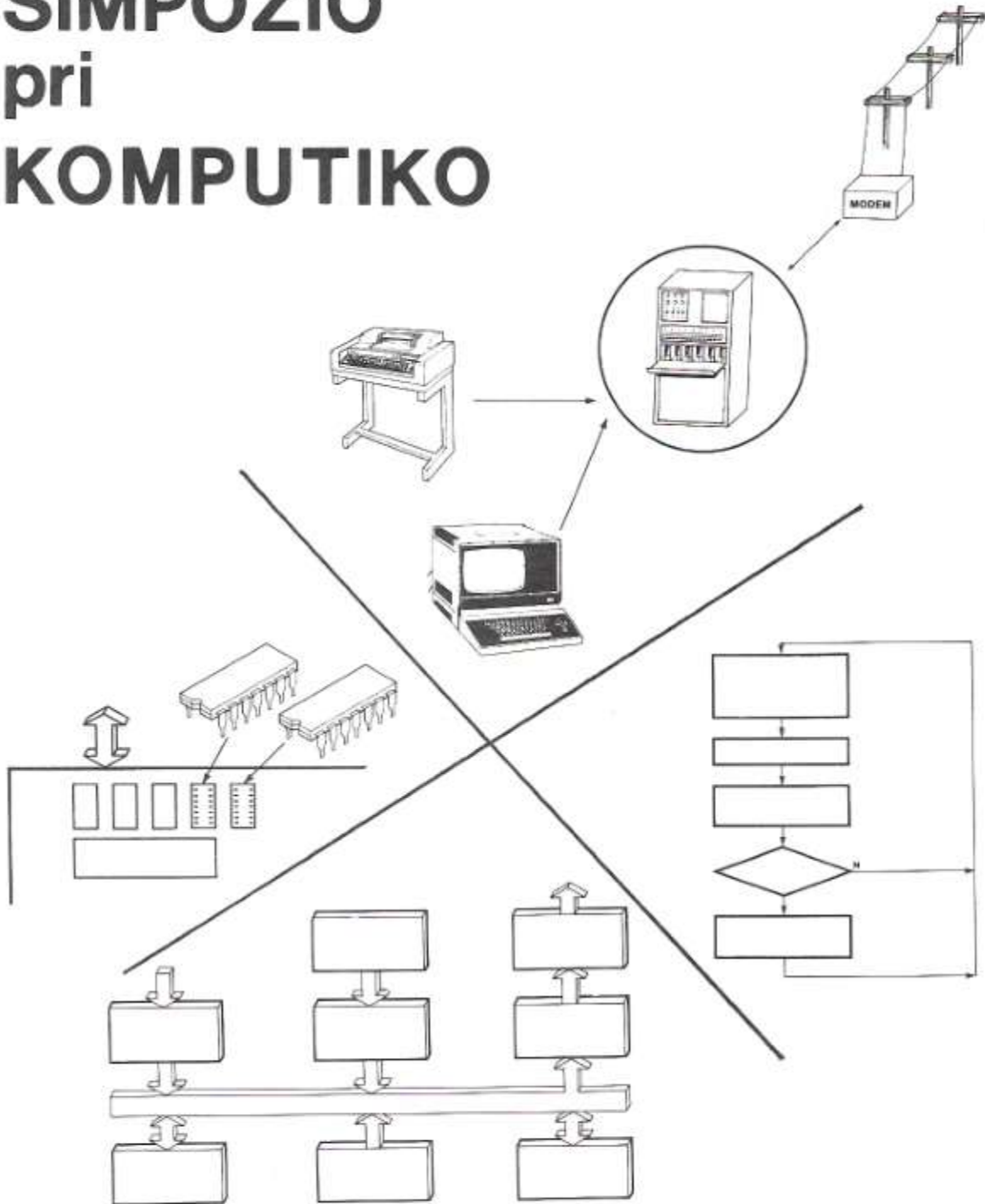


unua SIMPOZIO pri KOMPUTIKO



Rennes, Francujo, Julio 1981

Sub aŭspicio de UEA

**unua
SIMPOZIO
pri
KOMPUTIKO**

Rennes, Francujo, Julio 1981

S-ro C. BERTIN VRE/TDP
C.C.E.T.T.
B.P. : 1266
F - 35013 RENNES Cédex
FRANCUJO
Telekso : 740 284

1 Antaŭparolo

Ne malofte, ni legas pri manko kaj neceseco de scienca literaturo en Esperanto. Certe ĉiuj konsentas pri tio, sed ni devas studi unue kial ni troviĝas en tia situacio kaj poste kiel ni povas ŝanĝi ĝin.

Kial mankas scienca literaturo en Esperanto?

Tio estas kial ne ekzistas pli da fakaj eldonoj en Esperanto?

La ĉefa ĝisnuna formo de eldono estas libro aŭ revuo. Por eldoni libron, ne sufiĉas verkanto, necesas ankaŭ minimuma merkato, kiun ĝis nun ni ne havas en Esperantujo por la fakaj eldonoj. Por eldoni revuon, al la minimuma abonantaro aldoniĝas la neceso de grandega laboro (redaktado, presado, sendado, abonadministrado, informado, varbado...). Tio estas cikla problemo: la kresko de la nombro de esperantuzantoj kreskigos la sciencajn eldonojn en Esperanto, kaj la kresko de scienca literaturo en Esperanto kreskigos la nombron de la esperantuzantoj. Jam delonge multaj asocioj varbas al Esperanto, do ni prizorgu la alian flankon, tio estas: kreskigi la sciencan literaturon en Esperanto.

Kiel kreskigi la sciencan literaturon en Esperanto?

Ni jam konstatis ke eldoni fakajn librojn kaj revuojn ne estas facile. Ni devas trovi aliajn eldonmanierojn; pro tio mi proponis tiun simpozion (erare mi parolis pri kolokvo, kolokvo estas kunveno kun prezento kaj diskuto de diversaj temoj).

Eldoni kajeron de simpozio aŭ de kolokvo estas multe pli facile por iu ajn persono aŭ asocio: unue, la redaktado estas disigita inter ĉiuj verkantoj, due ne necesas varbi abonantojn kaj administri ilin, trie la redaktoro ne devigas sin labori pri tio dum pluraj jaroj kiel por revuo.

Alia avantaĝo estas la rapida eldono de la artikoloj post ilia verkado. Plie ĉar la simpozio estas tre faka, eldono de la kajero povas esti farata sen problemo, kadre de samfaka entrepreno aŭ firmao, sufiĉas ke ekzistu esperantista fakulo en tiu organizajo. Same la merkato vastiĝas de personoj al fakaj neesperantistaj organizaĵoj pro la fakeco de la kajero.

Poste, ni povos organizi kolokvon. Valoras organizi kolokvon nur se la ĉeestantaro anonciĝas sufiĉe multnombra (ĉirkaŭ 100 minimume). Eĉ se tio ankoraŭ ne eblas nun, tio ne devas malkuraĝigi nin organizi simpozion, tute male. Evidente kolokvo pli impresus la neesperantistan fakularon.

Terminologia flanko.

Alia afero estas la kreo, la adopto kaj la uzo de fakaj terminoj en la diversaj artikoloj. Pro la manko de kompetentaj komisionoj pri terminologio en multaj fakoj por proponi kaj rekomendi al ni terminojn, ni mem devas proponi (tio estas uzi) novajn terminojn; pro tio en la diversaj artikoloj estas uzataj malsamaj terminoj kun la sama signifo. Tio estas normala kaj ne ĉiam malbona ĉar tiel la legantoj konos la diversajn eblecojn kaj la uzo, se ne kompetenta komisiono, favoros unu kaj forpelos la aliajn. En ĉiuj kazoj, tio ne devas malkuraĝigi la verkantojn.

La organizanto: Christian BERTIN.

P. S: Estontece, ni devos esti pli severaj pri la tajpado de la kontribuajoj, ŝajnas ke multaj ne konsciis ke la eldono estos farita per nura fotokopiado de la sendita artikolo, konsekvence la kopio ne povas esti pli belaspekta ol la originalo. Same, multaj aŭtoroj ne petis la modelan kontribuajon antaŭ ol tajpi sian kontribuajon, tio estas bedaŭrinda. Aliflanke, ni devos poste rifuzi kontribuajojn jam prezentitajn al alia konferenco, tion ni ne faris ĉi-foje. La kajero estas jam sufiĉe ampleksa por antaŭvidi la organizadon de malpli ĝeneralaj simpozioj.

UNUA **KOLOKVO**PRI **KOMPUTORIKO**

(en esperanto)

Rennes, Francujo, Julio 1981

ALVOKO POR ARTIKOLOJ

Temoj

- Teoriaj fundamentoj pri informa prilaboro (matematikaj metodoj kaj teorioj ...)
- Komputara arkitekturo (progreso en teknologio...)
- Novaj hardvaraj produktoj (memoroj, mikroprocesoroj, icoj (Integraj Cirkvitoj), ...)
- Softvaro (operaciadaj sistemoj, programlingvoj, organizado...)
- Datenbazoj kaj informaj sistemoj (modeloj, teknikoj,...)
- Komputaraj retoj kaj komunikoj (radiaj, satelitaj, optikfibraj datenretoj, elektronika poŝto, interkonekto de retoj, privateco kaj sekureco de la komunikoj ...)
- Komputorado en scienco kaj industrio (parolo per pakoj, industria aŭtomatigo kaj robotiko,...)
- Sociaj kaj ekonomikaj efektoj
- Informa prilaboro kaj eduko (komputoruzaj edukaj sistemoj, komputoroj en lernejo...)
- Komputoroj en la ĉiutaga vivo (distro, klerigo...)
- Protokola reprezento kaj normigo
- Novaj servoj : videotekso, teletekso, faksimilo...

Partoprenaj kondiĉoj

La aŭtoroj devas verki artikolon kies amplekso devas esti inter 5 kaj 20 paĝoj (inkluzive resumon, enkondukon, tekston, konkludon (eventuale), bibliografion, glosaron kaj biografion kun nigra-blanka foto).

La artikoloj devas esti originale verkitaj en esperanto.

La artikoloj devas esti tajpitaj aŭ kompostitaj kaj prezentiĝi sur papero kies formato estas A4 (21 cm x 29,7 cm) kiel la kune sendita modelo (nur unuflanka tajpado).

La aŭtoroj devas sendi la artikolan originalon kun unu kopio kaj konservi bonan kopion (kaze de poŝta perdiĝo) antaŭ la fino de Junio 1981 al la sekvanta adreso :

S-ro C. BERTIN VRE/TDP

C.C.E.T.T.

B.P. : 1266

F - 35013 RENNES Cklex

FRANCUJO

Telekso :740 284

Ekzemplo de paĝo :

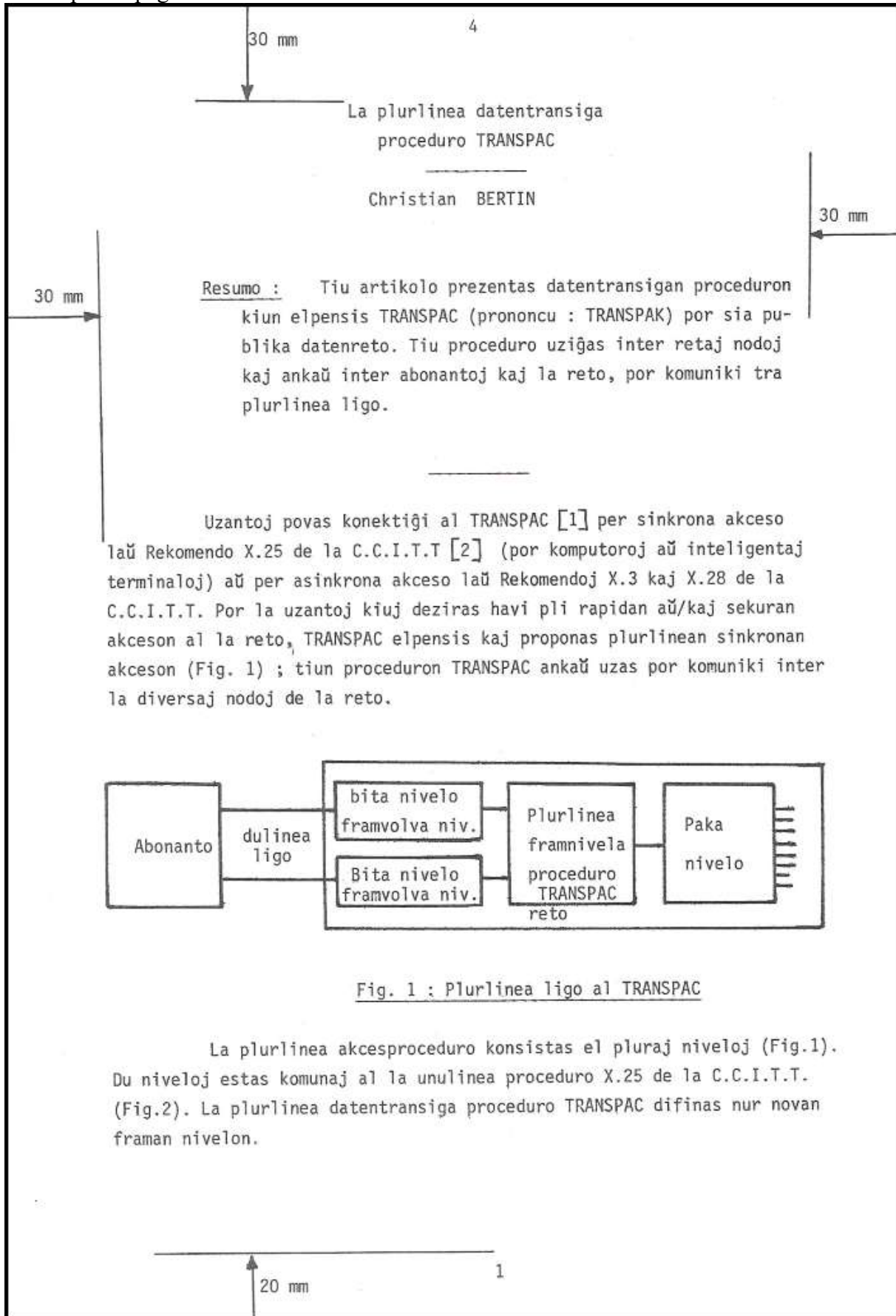


Fig. 1 : Plurlinea ligo al TRANSPAC

La plurlinea akcesproceduro konsistas el pluraj niveloj (Fig.1). Du niveloj estas komunaj al la unilinea proceduro X.25 de la C.C.I.T.T. (Fig.2). La plurlinea datentransiga proceduro TRANSPAC difinas nur novan framnivelon.

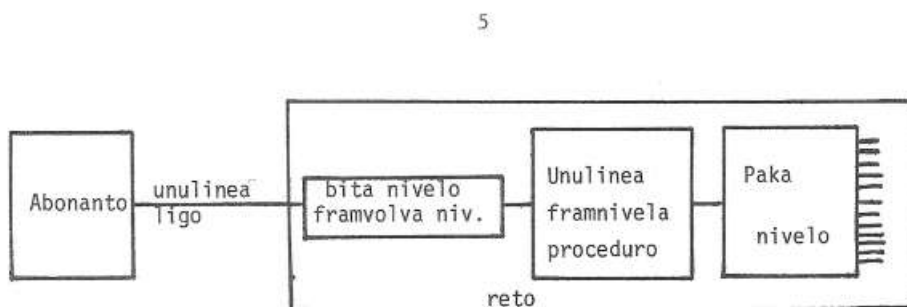


Fig. 2 : Unulinea ligo al TRANSPAC (X.25)

I. Strukturo de la diversaj framoj (datenblokoj) :

Pluraj tipoj de framoj ekzistas : la reinitaj kaj la informaj framoj.

I.1. La reinitaj framoj :

I.1.1. Formato de la reinitaj framoj :

.....

Bibliografio

- [1] : Spécifications techniques d'utilisation du réseau TRANSPAC.
A.D.E.T.T.I. 33,rue censier F-75005 PARIS
- [2] : Draft revised Recommendation X.25 (7-15 februaro 1980)
C.C.I.T.T. 2,rue Varembe CH-1211 GENEVE 20

Glosaro

C.C.I.T.T. : Comité Consultatif International Télégraphique et Téléphonique. (Internacia Konsultiĝa Komitato pri Telegrafio kaj Telefonio)

Cirkvito : Komunikvojo inter du korespondantoj.

Framo : Nomo de aparta datenbloko en X.25 kiu utilas aparte kiel ujo por datenpakö. Ekzistas ordonframoj kaj respondframoj. (Angle: frame, France: trame).



Biografio

Christian BERTIN, inĝeniero pri datenprilaboro (INSA, Rennes, 1972) laboris ĉe IRIA pri la reto CYCLADES ĝis 1976, ekde 1976 laboras ĉe la C.C.E.T.T. en Rennes (Centro Komuna Esplora pri Brodkasto kaj Telekomuniko), Francio.

LA CIBERNETIKO KAJ LA SOCIO
SZÉKELY Péter /Petro Bekej/

Resumo: Kiom da homoj interesiĝas pri la fortoj movantaj la socion? Multaj, ĉu ne? Tiu ĉi artikolo temas pri la ekkonaj eblecoj de la socio. Pro la mallongeco ĝi celas nur pensinstigon.

Juna kaj vaste ne tro konata scienco de la nuntempo estas la cibernetiko.

Cibernetiko estas la scienco de la regado, scienco, kiu okupiĝas pri la leĝoj de la regado de kompleksaj dinamikaj sistemoj.

Ankaŭ la viva mondo estas komplika cibernetika sistemo. La procezoj de la vivaj organismoj, kaj mem la vivaj organismoj rigardeblas preskaŭ kompletaj aŭtomatoj. Ili perceptas signojn, indikojn, interpretas ilin, kaj reagis laŭ siaj intelektoj. La plej evoluinta, plej komplika estaĵo de la viva mondo estas la homo.

La homo estas socia estaĵo, kaj ek de la pratempoj lin plej interesas la alia homo. Homoj formas grandajn komunumojn. Tiuj komunumoj formas la homan socion. Neeviteble necesa kondiĉo de la homa ekzisto estas, ke la homaj individuoj submetu sin sub la leĝojn de la homo socio.

Serioza celo de la nuntempaj pensistoj estas ekkoni la komunumon en kiu ni vivas. Tiu komunumo estas kompleksa dinamika sistemo, grandega, konstante movilanta sistemo de la homaj rilatoj, kies movigajn fortojn oni povas ekzameni per la helpo de la cibernetiko.

Por ekzameni la socion, kiel cibernetikan sistemon, oni devas klarigi, ke la socio ne rigardeblas aŭtomato en tia senco, kiel la viva organismo, aŭ la diversaj partoj de ĝi. Do en tiu senco nek la sociaj procezoj rigardeblas aŭtomatikoj. Ĉe la socio mem oni ne povas malkovri tian similecon al la maŝinoj kaj aŭtomatoj, kiel ĉe la viva mondo.

En multaj kazoj ŝajnas, ke la historiaj procezoj okazas laŭ certa programo, ja difinitan eksteran kaŭzon sekvas difinita "respondo". Tamen tre vasta estas la rondo de tiuj eksteraj influoj, de la eblaj respondoj, kaj de iliaj specifaj variantoj. Pro tio ili haveblas nur ege malproksiman parencecon al la maŝinoj. Ties ĉefa kaŭzo estas, ke la socio konsistas el homoj. t.e. el senchavaj estaĵoj,, La nombro de la konsciaj agoj, reagoj estas multoble pli granda ol tiu de la laŭleĝaj biologiaj reagoj.

La socio estas la plej komplika sistemo en la mondo ek-konebla far ni, La socio estas la plej alta nivelo je la evoluo de la materio en tiu senco, ke la nombro de siaj interrilatoj estas la plej granda, Tamen la socio ne ri-gardeblas "senespere" kompleksa sistemo. La socia sistemo estas tia aŭtomato, kiu relative pli facile ekkoneblas ol ekz. la biologia sistemo, do oni pli facile povos regi ĝin en tempo antaŭvidebla. Inter tiaj striktaj kondiĉoj cibernetiko speciale ekzamenas la socion kiel aŭtomaton respondanta la influojn de sia medio aŭtomate. La konsciaj agoj tute diferencas de la aŭtomataj /agoj/.

La socio kiel sistemo havas priskribeblajn, modeleblajn procezojn. La socia sistemo estas tia amaso de la elementoj de la socio, kiu havas organizitecon. Socia sistemo estas la kapitalista, aŭ la socialista reĝimo, kaj ankaŭ iu uzino, eventuale iu organizita unuiĝo. La modeloj, kiujn oni povas krei ĉiuj estas neperfektaj, multege simpligitaj. Tio estas bezonata, ja la trarigard-ebleco, modelebleco de la ekstreme multaj rilatoj ne sukcesas esti perfekta dum finita tempo. Oni devas neglekti grandan parton de la specifaj cirkonstancoj.

La eventoj de la socio neniam povas ripetiĝi, ilin tre multe influas la specifaj kondiĉoj. En certa senco tio pravus je ĉiu procezo, estu ĝi ĉu fizika, ĉu kemia, ĉu io ajn alia. Ĉe la eventoj okazintaj en sociaj sistemoj la graveco de la specifaj, kvazaŭ hazardaj variantoj povas esti elstara. Ekzemple du militoj neniam eblas simile unu

la alian tiom, kiom du kemiaj reakcioj, ktp.

Ĉe la reagoj de iu socia grupo oni devas atenti tiajn variantojn, kiel ekz. aĝo, edukiteco, nacieco, religio, sekso, familia medio, ktp. Tiajn multvariantajn analizojn nur tiel sukcesas fari, se oni la signifan parton de la variantoj rigardas parametro, eventuale neglektas ilin.

Plurajn komplikaĵojn kaŭzas, ke la socian fenomenon, kiun oni ekzamenas, modelas, ĝenerale estas malfacila, kaj multkaze eĉ neebla esprimi matematike, kiel matematika varinto, kaj pro tio ĉe la kalkuloj oni povas uzi nur grave neprecizajn proksimigojn. Eĉ se sukcesus trovi la matematikan varianton de iu socia procezo, evidentiĝus tuj, ke tiu varianto ŝanĝiĝas ankaŭ laŭ tempo. Ekzemple iu persono en diversaj tempoj kun diversaj konoj, scioj partoprenas la taskojn, procezojn de la socio.

La nuntempaj sociaj modeloj en la praktiko ne taŭgas. Ilin nur en laboratorio sukcesas efektiviĝi, kaj la rezultoj estas sufiĉe bagatelaj. Ju pli preciza estas la laboratoria modelo de iu socia procezo, des malpli ĝenerala estas lia valido. Pri la modeloj elprovitaj en la praktiko malkaŝiĝis, ke ili estas ĝustaj nur sur certaj terenoj, je certa tempo, ĉe certaj sociaj grupoj. Pri tio oni diras, ke la sociaj modeloj havas fenomenologian karakteron.

Hodiaŭ oni ankoraŭ ne kapablas fari tia specajn sociajn modelojn, ĉe kiuj modelinda procezo estas multvarianta "funkcion. Eĉ la lasta tempe prosperanta matematika evoluo nur etmezure helpas je la problemo de la multaj variantoj. Tamen oni devas emfazi, ke la modelado de sociaj procezoj estas solvebla. Pli kaj pli multaj kaj bonaj modeloj estas kreitan, kaj ĉiam pli sukcese eblas konsideri la ĝis nun ne menciitan, tamen tre gravan faktoron, ke la propra moviĝo de la socio mem, estas en dialektika interrilato kun la propra movilo de siaj elementoj. Dum ĉiu momento de tiu movilo efektiva estas la unueco de la neceso kaj hazardo, krome ankaŭ tio konsidereblas, ke la funkciado de la procezo neniam efikas nur en unu direkto, sed ĉiam ankaŭ

reagas.

Per ludo-teoriaj modeloj sufiĉe bone modeleblas multe de socia procezo /ekz. klasbataloj/.

Eĉ ĝis nun oni modelis multegan nombron da procezoj, ekzemple la efikon de la telekomunikiloj, nome la efikon de la propagando, aŭ la sintenon de la elektanataro je la elekta decido, kaj laŭ tiu modelo takseblas la rezulto de la elektoj kun 1-2 procenta eraro. Tiajn modelojn oni faris jam en 1966 dum la elektado en Norda-Rajna-Vestfalio. Iu socia sistemo el iu ajn vidpunkto rigardeblas sistemo, sed ĉiam laŭ ĝia informa reto iĝas organizita sistemo. Ĝuste pro tio estas necesa la disvastigado de la informaj kanaloj.

Se ni rigardas ian relative simplan socian subsistemon, iun fabrikon, videblas, se la instrukcioj /informoj/ de la direktoro devas trapasi multajn nivelojn ĝis la celita loko, tiam pli granda estas la probableco, ke la informo didifektiĝas, miksiĝas kun bruo. Tiukaze la sistemo povas iĝi tre malstrikta,

Pli kaj pli disvastiĝas la informo prilaborantaj garnituroj, pli kaj pli prosperas la "revolucio" de la informoj. La informo-prilaborantaj garnituroj ebligas la aŭtomatigon de la socia produktado super ĉiun ĝis nun akiritan nivelon.

La homo sin mem intencas anstataŭigi per maŝinoj, cibernetikaj iloj, respektive per maŝinoj, robotoj, regataj far komputiloj. Memkompreneble li volas ilin efektiviĝi en sia propra homo "strukturo". Ni sukcesas trovi la signojn de tiu laboro observante la lastajn jarcentojn, inkluzive ankaŭ la robotojn de la hodiaŭaj scienc-fikciaĵoj.

Jam en la mezo de la 60-aj jaroj oni faris la unuan cerbomodelon uzante ĉelsisteman konstruon. Kompreneble ĝi estis sufiĉe primitiva. Samtempe industriaj robotoj provas substitui la malfacilan, lacigajn partojn de la homa laboro. En la evoluantaj kapitalistaj landoj jam je la 70-aj jaroj funkciis plurcent industriaj robotoj. La evoluigo celas, ke la robotoj estu pli grandmezure programregataj,

ekzemple helpe de ciferegado.

La ideo de la ĉelbaza konstruo, de la ĉelaŭtomato pre-
prezentiĝis jam en 1950. Johano Neumann ellaboris la teorion de
la t.n. sinreprodukantaj aŭtomatoj. La esenco
de la ĉelaŭtomatoj estas tio, ke ili bezonas nur unuspecan
cirkvicon la tiel nomitan ĉelon. Per la apliko de multegaj
ĉeloj, sen io alia elemento elformiĝas la ĉelspaco, en kies
interno organizeblas aŭtomatoj. En 1973 ankaŭ en Hungario oni
kreis ĉel-cirkvicon. Nuntempe oni lacadas atingi pli bonajn
prepareblecojn.

Antaŭe oni sukcesis aŭtomatigi nur fizikajn procezojn, hodiaŭ
per la helpo de komputiloj oni jam kapablas farigi simplajn
intelektajn agojn, do tiel iom post iom disvastiĝos la kompleksa
mekanikizado. La homo pli kaj pli liberiĝas de la laboro kaj
reproduktiĝos la vivkondiĉoj. Tio permesos la abundecon de la
havaĵoj, kaj grandmezure helpos la progreson al la laŭnecesa
distribuo. Granda problemo estas, kion faros la homo de la
morgaŭo, se oni aŭtomatigas ĉiam pli grandan parton de la korpa
kaj intelekta laboro. Egalvaloran problemon signas la kerskanta
energiobezono. Laŭ Norberto Wiener la nova industria revolucio
estas kvazaŭ duakra glavo. Aplikebla kaj por helpi... kaj por
pereigi la homaron. Tiuj senlaborecoj, kiujn ni ekkonis ĝis nun
plejparte efikis ĉe la korpa laborado, kaj nur pere de ĝi
atingis la sferon de la intelekta laboro. Per la danĝero ne la
maŝinoj minacas, kiel en Anglujo nek ili minacis en la tempo de
la maŝindetruoj. Sole la homoj minacas unu la alian per la
maŝinoj. Pro tio la akiraĵoj de la sciencoj kaj tekniko egale
povas kunporti benon aŭ damnon, kaj kiun tio sole de ni, homoj
dependas. La maŝinoj de la 19-a jarcento mekanikizis la korpan
laboron, la cibernetikaj iloj kapablas aŭtomatigi la intelektan
laboron.

La kompuilo kapablas lerni tian serion de "operacioj, kiu
estas proksime al la arto. Oni ankoraŭ ne scias ki-
umaniere efektiviĝas la homa cerbo artan kreadon, kaj pro

tio oni sukcesas igi nek la maŝinojn kreadi artaĵojn. Nur la produktaĵo de la arta kreado estas analizebla per maŝinoj. Certaj fizionomioj esprimeblas matematike, kaj laŭ tio oni povas doni ordonon al la maŝino ke ĝi kreu rezultaton similan al iu artaĵo. Tion oni efektivigas tiel, ke la komputilo per hazarda cifergenerado pretigas ian produktaĵon laŭ la indikitaj reguloj, inter difinitaj limoj. Tia produktaĵo povas esti poemo, komponaĵo, eĉ desegnaĵo

A. Zichaël Noll farigis 90 paralelajn sinus-grafikaĵojn laŭ lineare kreskantaj periodoj, kaj li ricevis tre plaĉan, modernan "ŝtofo-modelon". La maŝinoj laŭ stokastikaj programoj faris eĉ pentraĵ-similajn kolorajn bildojn.

Iu juna hungara esploristo, Nikolao Havas kreis programon, kiu konfidas al la maŝino la ekkonon kaj aplikadon de la komponadaj reguloj. Li plenigis la memorion de la maŝino per melodioj; farigis statistikon pri la legaleco de la melodioj kaj konante ilin li ordonis la maŝinon komponi. En la Scienca Universitato "Kossuth Lajos" en Debrecen oni establis la rekurzivan vortaron de la hungara lingvo. Tiu tasko estas nesolvebla sen komputilo. Tiu vortaro prezentas grandan prespektion ĉe la ekkono de la sistemo kaj evoluo de la parolata lingvo.

Nuntempe naskiĝis nova koncepto ankaŭ en la estetiko. Tiu nocio estas la estetika informo. La mezuro de la Moles-speca estetika informo estas elkalkulebla, kaj ĝenerale estas granda. La estetika informo ne identiĝas kun la estetikaĵo. Sur la nivelo de niaj hodiaŭaj konoj ni ankoraŭ ne kapablas difini tian estetikan informon, kies kvanto rekte proporcius al la beleco.

Do la model-perspektiva evoluo de la sciencoj pli kaj pli proksimigas la homaron al la ekkono de sia socio, pere de komputiloj.



Faka aŭtobiografo

En 1973 mi finis la studadojn kiel komputista inĝeniero. Poste mi laboris ĉe maŝinindustria projekta instituto kiel komputil-programisto kaj funkciigisto apud CII 10010 kaj IBM-360.

Ekde 1980 mi estas la estro de komputil-centralo de konstru-industria entrepreno, kaj la projektanto de unu entrepren-direkta perkomputila sistemo.

urbo EGER en HUNGARIO, jun.19. en 1981.

La ĉapelitaj literoj en informatiko

Rudolf-Josef Fischer

Institut für Medizinische Informatik und Biomathematik der
Westfdlischen Wilhelms-Universitdt Münster Htifferstr. 75, D-4400
Münster

(Direktor: Prof. Dr. med. et Dipl.-Math. F. Wingert)

1 Enkondukaj rimarkoj

Preskaŭ ĉiuj Eŭropaj lingvoj, escepte de la Nederlanda kaj la Angla, uzas literojn kun diakritaj signoj. Tamen la internacia norma klavaro nur konas la Latinan alfabeton. Tio kaŭzas multajn problemojn, kiam oni ekzemple per komputoro manipulas tekstojn. Germana leĝo ordonas, ke oni en dokumento ne rajtas ŝanĝi la formon de familia nomo. Sed ambaŭ oftajn versiojn "Jäger" kaj "Jaeger" oni nur povas enkomputorigi kiel "JAEGER".

Pliaj problemoj kaj iliaj solvoj estas diskutendaj, elektante la internacian lingvon Esperanto kiel ekzemplon, kiu spegulas la samajn bazajn ecojn tiurilate kiel la naciaj lingvoj, aldonante apartajn punktojn.

Svarmas nuntempe proponoj por nova skribmaniero en Esperanto. Ne mirigas, ke kelkaj proponoj estas tute diversaj, depende de la necesoj kaj la celoj, el kiuj ili kreiĝis. Dum la diskuto en Esperantistaj ĵurnaloj kaj seminarioj montriĝis, ke la plej multaj Esperanto-parolantoj ne scias kelkajn bazajn faktojn pri la graveco de la Esperanta alfabeto rilate la informatikon, kiu ludas nuntempe eble la decidan rolon en la debato pri solvo de la lingva problemo en la Eŭropa Komunumo.

2 La ĝenantaj literoj

Ekde la unuaj jaroj de Esperanto ne ĉesis la kritiko, ke la Esperanta alfabeto enhavas la literojn kun supersignoj, t.n. ĉapelitajn literojn, nome ĉ, ĝ, ĥ, ĵ ŝ kaj ŭ. Certe ili ofte kaŭzis kaj kaŭzas problemojn dum presado kaj tajpado. Eĉ nun ankoraŭ ekzistas tute freŝdataj tajpiloj, kiuj ne ebligas, skribi akcentojn por formi la ĉapelon "^"; la litero ŭ eĉ nur esceptkaze estas adekvate tajpebla. La Fundamento, baza deviga regularo de la lingvo Esperanto, jam agnoskis ĉi tiun problemon kaj permesis alian skribmanieron per posta h al ch, gh, hh, jh, sh kaj anstataŭon de ŭ far u.

(Ĉi tiu eblo forigas la problemojn dum presado kaj tajpado.

3 Ne-Fundamentaj aliaj skribmanieroj

Des pli rimarkindas, ke Esperantistoj ege malofte uzas la postan h. Tiel oni foje povis legi en Germanlingva gazeto, ke ĉiu litero en Esperanto havas nur unu prononcon; sed du liniojn poste oni perturbis la leganton, ŝajne informante, ke "sapo" prononciĝas "ŝapo", tamen "safo" (ŝafo) "schafo".

Krome oni rimarkas, ke la Esperantistoj en praktiko ofte uzas aliajn ne-Fundamentajn skribmanierojn, ekzemple ŭ anstataŭ ŭ, kaj ĉ aŭ ĉ anstataŭ ĉ ktp.. Kiam la lingva kontrol-instanco de Esperanto, la Akademio, aperigis en aŭtuno de la 1980-a verdikton kontraŭ ĉiu skribreformo asertante, ke ĉiuj presejoj povas laŭ-Fundamente presi la Esperantajn literojn, ĝuste la fama Esperant-lingva ĵurnalo HEROLDO devis prezenti la tekston de la verdikto uzinte ŭ por ŭ kaj do pruvante la kontraŭon.

MAAS [1] uzas postan x anstataŭ h por laborado per komputoro, la Inversa Vortaro [4] uzas ".".

Tio ĉio indikas, ke ekzistas ankoraŭ gravaj problemoj pro la ĉapelitaj literoj kaj ke la Fundamenta posta h ne estas kontentiga.

4 Malavantaĝoj por literumado kaj legado

La malŝaton, uzi la Fundamentan postan h, verŝajne kaŭzas parte la timo, aspektigi nacilingvajn vortojn tro "fremdaj" (ekz. "shtato", "poshto"), parte la konscia aŭ malkonscia oboeo al la Fundamenta leĝo, ke oni skribu ĉion kiel oni prononcas kaj inverse. Fakte ĉi tiu lasta eco de Esperanto estas unu el la plej elstaraj avantaĝoj kompare kun aliaj, naciaj lingvoj. Allasi esceptojn al tiu regulo, signifas forĵeti la konvinkigan argumenton: "Kompreneble Esperanto kiel planlingvo havas senesceptan prononc- kaj skrib-regularon, tiel ke oni lernas malriproĉeble literumi kaj prononci dum kvarona horo."

Enkaze de uzo de posta h komencanto ne plu scias prononci legante vortojn kiel "longhara", "senchava" kaj multajn aliajn; same, maldistingante ŭ de u, malklaras la prononco de "praulo" kompare kun "fraŭlo" ktp.. Tio rimarkeble malplifaciligas la lernadon kaj forprenas multe de la avantaĝoj, kiuj estas necesaj por la interkomunikilo Esperanto, se oni volas racie kaj sukcese pruvi ĝian superecon super naciaj lingvoj.

5 Gravaj nuntempaj taskoj en informatiko

Nuntempa lingvo ne nur estas uzata kiel interkomunikilo inter homoj, sed devas taŭgi ankaŭ kiel interkomunikilo inter maŝino (komputoro) kaj homo.

Dum prilaborado de tekstoj estas tre avantaĝe, ke oni povas uzi simplajn algoritmojn, kiuj funkcias unuop-litere. Tio signifas, ke algoritmo traktante certan pozicion en vorto ne devas konsideri alian, ekzemple la postan, literon. Kompreneble tio ne estas deviga postulo; sed diferenco inter la fizika longo (numero de la signoj) kaj la logika longo (numero de la fonemoj) de vorto ege komplikigas algoritmojn kaj foruzas labortempon de komputoro.

Plie, por solvi la Eŭropan lingvo-problemon, oni provas, per nacilingvoj solvi la sekvantajn bazajn problemojn:

- a) aŭtomata parol-sintezo:
komputoro per parolilo voĉlegu tekston
- b) aŭtomata parol-analizo:
komputoro skribu parolitan tekston
- c) aŭtomata tekst-analizo: komputoro analizu vortojn kaj frazojn de enigita teksto kaj manipulinde parkerigu ilin por posta eldono de dezirataj informoj.

6 Problemoj de nacilingvoj kompare kun Esperanto

Dum la supraj taskoj nacilingvoj kaŭzas nevenkeblajn problemojn pro la malobeo al la regulo, ke oni uzu por ĉiu fonemo precize unu literon kaj inverse. Dum aŭtomata parol-sintezo dubaj vortpartoj kiel "sch" en la Germana vorto "Hduschen" ktp. devigas krei komplikajn listojn de esceptoj, kiuj neniam estas kompletaj kaj ebligas ankaŭ post granda elspezo nur, prononci ĉ. 90 % de Germanaj vortoj ĝuste. La uzon de posta h en Esperanto kondukas al la samaj problemoj.

Dum aŭtomata parol-analizo ekestas dubaj kazoj prononcinte ĉ, ĝ kaj c, ĉar ili estas prononce anstataŭigeblaj per tŝ, dĝ kaj ts respektive. Do, la vortkombino "gratsono" estas komputore skribata "gracono", se la parolanto ne klarige enmetas paŭzeton inter t kaj s.

Dum aŭtomata tekst-analizo ekiĝas la samaj problemoj kiel por aŭtomata parol-sintezo, ĉar parol-sintezo jam aplikas specialan

tekst-analizon. Krome unu baza tasko de tekst-manipulado ne plu eblas, nome laŭ-alfabeta ordigo. Uzante ĉi anstatal ĉ, "cheno" en tia vortaro estas antaŭ "cico" anstataŭ malantaŭe laŭ la Fundamenta alfabeto.

Do, oni povas resumi, ke uzante la Fundamentan skribmanieron kun posta h Esperanto kaŭzas la samajn problemojn kiel naciaj lingvoj, kiuj nur tute malkontentige estas uzeblaj per komputoroj.

Ĉar Esperanto tiurilate tiam ne superas naciajn lingvojn, forfalas gravega racia argumento por ĝi. Aliflanke estas aparta kondiĉo por Esperanto, ke ĝia alfabeto restu tiu de la Fundamento, ĉefe pro sociaj kaŭzoj, ĉar la komuna obero de la Fundamento ĝis nun kuntenis la dise en la mondo loĝantajn Esperanto-parolantojn.

7 Proponoj kompromisaj

Ekzistas diversaj proponoj, kiel resti laŭ-Fundamenta kaj plenumi la postulojn de informatiko.

- a) Oni proponas atendi, ĝis kiam la komputoroj estas adaptitaj al la lingvoj aŭ almenaŭ ĝis kiam la internacia norma klavaro kaj do ĉiuj tajpiloj havas la Esperantan alfabeton kun ĉapelitaj literoj.

Ĉar ĉiuj lingvoj ne povas postuli, ke ĉiu klavaro de la mondo havu la nacilingvajn diakritajn signojn, ankaŭ Esperanto ne povas. Prave la Angla-lingvanoj povus argumenti, ke ilia lingvo ja ne postulas iun ŝanĝon, kaj tion kaŭzas la fakto, ke por la Angla sufiĉas la Latina alfabeto, do ne iu kontraŭstarenda politika aŭ ekonomia premo. Tial kiel la aliaj lingvoj ankaŭ Esperanto devas ebligi la nuran uzon de Latinaj literoj.

- b) Oni uzas anstataŭ posta h alian, ne-Esperantan literon. Al tiu kategorio apartenas la jam menciitaj praktikoj, uzi x aŭ ".". Tiam oni evitas dubajn prononcon kaj tekst-analizon. Tamen la problemoj de parol-analizo restas, laŭ-alfabeta vort-ordigo ne eblas, kaj oni ne povas uzi simplajn algoritmojn, kiuj laboras unuop-litere.

- c) Oni uzas alian specialan signon anstataŭ h.

Internacia Organizo por Normigado en la 1975-a prezentis proponon kun speciala signo, kiu en la interna ordo de la komputoraj signoj

estas post la alfabetaj signoj. Tiam laŭ-alfabeta ordigo eblas; restas la problemoj de parol-analizo, kaj oni ne povas uzi simplajn algoritmojn. Krome, kiel ankaŭ dum propono b, ekrano kaj presilo normalkaze prezentas ne-Fundamentan skribmanieron. Plie, propono c preteratentas, ke ĝi ne estas realigebla per ĉiu maŝinaro.

En la universitato de Münster ekzistas ekzemple ekranoj, kiuj produktas ASCII-kodon, kiu aŭtomate interne estas transformata al EBCDIC-kodo, tiel ke ne eblas, tajpi specialan signon, kiu interne ricevas pozicion post ĉiuj alfabetaj literoj. Do, la propono de tiu Organizo ĝenerale ne estas solvo.

d) Oni uzas specialan klavaron kaj maŝinaron.

SHERWOOD [5] konatigis la sistemon PLATO, en kiu ekrano neriproĉeble prezentas la ĉapelitajn literojn. Tamen interne la kodo enhavas post la koncernaj literoj du specialajn signojn, kiuj ebligas jes laŭ-alfabetan ordigon, sed ne simplajn algoritmojn. Krome kompreneble ne ĉiu disponas pri la sistemo PLATO aŭ pri alia tia.

e) Oni uzas kelkajn klavojn kun specialaj funkcioj.

SCHNELL [3] proponas, uzi la ASCII-kodon kaj krome kelkajn klavojn por la literoj ĉ, ĝ, ĵ, ŝ kaj ŭ, tiel ke ili interne estas post la alfabetaj literoj. Li evitas tiam malsimplajn algoritmojn, sed ne povas laŭ-alfabete ordigi. Krome oni devas forigi la literon ĥ entute, kaj plie, kiel antaŭe menciite, ne ĉiu povas uzi originan ASCII-kodon. La kodo EBCDIC ne ebligas, tajpi rekte signojn, kiuj estas interne post la alfabetaj (escepte de la ciferoj, kiuj ĉi tie tute ne taŭgas). Fine riproĉeblas, ke oni devas rezigni pri la normalaj funkcioj, kiujn la klavoj uzataj nun por alia celo havis.

8 Metodo por forigi kelkajn malavantaĝojn

Ekzistas certa metodo por eviti malavantaĝojn de iu kodo, nome, ke ne eblas laŭ-alfabeta ordigo kaj/aŭ ke ne sufiĉas simplaj algoritmoj, kiuj funkcias unuop-litere. Oni devas transformi la kodon al iu alia interna kodo kun 28 signoj, kiuj respektive rilatas al la literoj de la Esperanta alfabeto. Tiam restas du problemoj:

- kvankam tute simplaj estas la necesaj algoritmoj, kiuj trans-

formas post enigo kaj antaŭ eligo la kodon, ili foruzas tempon kaj do monon. (Same oni povas solvi la problemojn de nacilingvoj kun ĉapelitaj literoj; do, Esperanto ĉi-kaze ne superas la nacilingvojn tiurilate.)

- pro la transformo la internaj tekstoj ne estas legeblaj, kio normalkaze estas ege ĝena. Ne eblas ekzemple, tajpi vortaron rekte; ĉiam oni devas krei vortaron per programo, uzante la simplan transforman algoritmon. Inverse ne eblas, legi la enhavon de datumaro sen programo.

9 Alia skribmaniero

Restas propono malobei la Fundamenton kaj uzi alian skribmanieron, nome skribi anstataŭ ĉ, ĝ, ĥ, ĵ, ŝ kaj ŭ: tx, dy, q, y, x kaj w respektive. Oni devas klare vidi, ke tiu propono signifas ŝanĝon de la Esperanta alfabeto, ĉar kompreneble oni nur tre malfacile povus konservi la saman vortordon kiel en Plena Ilustrita Vortaro [2]. Krome oni jes povas aŭtomate produkti novkodajn tekstojn el malnov-alfabetaj, sed inverse ne ĝenerale ĝuste: dum la konsonant-kombinoj "tx" kaj "dy" en certaj kazoj ne eblas, aŭtomate decidi, ĉu transformi al ĉ aŭ tŝ kaj al ĝ aŭ dĵ respektive. Pri tio ne kulpas la nova alfabeto, sed tio estas nur alia aspekto de la problemo dum aŭtomata parol-analizo, kaŭzita far la malbeo al la regulo "unu fonemo - unu litero" en la Fundamenta alfabeto (vidu 6).

Aliflanke ĉi tiu propono forigas ĉiujn antaŭe menciitajn problemojn dum la bazaj taskoj de informatiko (vidu 5). Tia Esperanto superus la nacilingvojn rimarkeble tiurilate: Eblus laŭ- (nov)alfabeta ordigo kaj apliko de simplaj algoritmoj sen transformpaŝoj. La regulo "unu fonemo - unu litero" estus pli strikte obeata (kaj strikte, se oni permesas skribi "ts" anstatal "c"). La ĝisnunaj tekstoj restus legeblaj, ĉar la ŝanĝo de la alfabeto afliktas nur malgrandan parton de la vortaro. Farhoma intermikso inter la malnova kaj nova skribmanieroj ne eblas.

10 Resumo

El la multaj proponoj rezultas du tute diversaj tipoj:

Unua, kiu estas realigebla por ĉiuj lingvoj: oni lasas la metodon, kiel oni enigu Esperantajn tekstojn, tute libera, depende de la maŝinaro aŭ bontrovo. Sed oni postulas, ke la rezulta presita

	6	7b	7c	7d	7e	9
	aldona h	aldona x ktp.	postalfa- beta aldona signo laŭ normigo	pluraj fiksaĵ aldonaĵ sig- noj (PLATO)	pluraj di- versaĵ aldonaĵ signoj (SCHNELL)	modifita alfabeto
Fundamenta ⁺⁺)	jes	ne	ne	jes	ne	ne
ĉie realigebla	jes	jes	ne	ne	ne	jes
ebligas aŭto- matan trans- formon	ne	jes	jes	jes	jes (sen Ĥ)	jes ĉar su- perflua!
laŭ-alfabete ordigebla	ne	ne ⁺)	jes	jes	ne ⁺)	jes (laŭ la nova al- fabeto!)
simplaj algoritmoj	ne	ne ⁺)	ne ⁺)	ne ⁺)	jes	jes
supereco de Esperanto sen transformo	ne	ne	ne	ne	ne	jes

tabelo 1: superrigardo pri avantaĝoj (jes) kaj malavantaĝoj (ne) de diversaj skribmanieroj

+) jesigebla per aŭtomata transformo (vidu 8)

++) "Fundamenta" signifas: Fundamenta skribmaniero sur ekrano kaj sur presita eligaĵo

teksto ĉiukaze estu Fundamenta (au konforma al la nacia lingvo), do, kun ĉapelitaj literoj aŭ kun aldona h. Krome oni fiksas la internan 28-signan kodon, kiu reprezentas la 28 literojn de la Fundamenta alfabeto. Ĉiuj informatikistoj devas uzi tiun norman kodon, por ke intern-kodaj tekstoj kaj koncernaj programoj (kiuj povas dependi de la kodo) estu ĝenerale uzeblaj kaj interŝanĝeblaj. En la kazo de Esperanto pro obeo de la Fundamento oni tiam rezignas pri la grava ŝanco, ke Esperanto povas superi la nacilingvojn rilate kelkajn bazajn taskojn de la informatiko. Oni konscie elspezas la necesan tempon kaj monon por la ĉiam uzendaj algoritmoj por post-eniga kaj por antaŭ-eliga transformo.

Dua, kiu nur eblas por planlingvo kiel Esperanto: oni anstataŭigas la ĝisnunan eblon de posta h per nova skribmaniero, konsciante, ke ĝi signifas samtempe alternativan alfabeton, kio rezultas ankaŭ al alternativa vortordo. Per tio oni atingas superecon de Esperanto super naciaj lingvoj rilate bazajn taskojn de informatiko (vidu 5), neglektante laŭ-literan obeon de la Fundamento.

Se iam Esperanto estos oficiale agnoskata, certe la koncernaj aŭtoritatuloj postulas ŝanĝon de la lingvo. Estus bone, se la Esperantistoj almenaŭ por certe kritikota punkto jam nun havus pli bonan alternativon. Tio povus ankaŭ malgrandigi la riskon, ke post la mondvasta akcepto de ia Nov-Esperanto la Fundamento kaŭzos novan Malnov-Esperanto-movadon.

11 Literaturo

- [1] MAAS, Heinz Dieter:
Aŭtomata tradukado en kaj el Esperanto.
9-a Konferenco de la Internacia Kibernetika Asocio, Namur 1980
- [2] Plena Ilustrita Vortaro de Esperanto, S.A.T., Paris 1978
- [3] SCHNELL, Roland:
Propono por normo de la Esperanto-alfabeto ĉe komputoroj. kontakto 66 (1980) 8-9
- [4] SCHLUTER, Klaus:
Inversa Vortaro, Nürnberg 1972
- [5] SHERWOOD, Judith:
PLATO Esperanto materials.
Studies in Language Learning 3, 1981

Komputila Traktado de Esperanta Teksto

Bruce Arne Sherwood

Esplorlaboratorio pri Perkomputila Instruado kaj Fako de Fiziko
Universitato de Illinois le Urbana-Champaign

252 Eng. Res. Lab.

103 S. Mathews

Urbana, Illinois 61801, Usono

La instruado de Esperanto per la PLATOsistemo por perkomputila instruado kreis bezonon por diversaj tipoj de komputila traktado de Esperanta teksto. Ĉi tio ebligis detalan trarigardon de multaj aspektoj de la prilaboro de Esperanta teksto. Temoj priskribataj inkluzivas ortografion, fonetikon, morfologion, leksikon, semantikon, kaj sintakson. Pluraj aplikoj estas priskribataj.

The teaching of Esperanto using the PLATO computer-based education system has led to the need for various kinds of computer processing of Esperanto text. This has made it possible to review in detail many aspects of Esperanto text processing. Topics covered include orthography, phonetics, morphology, lexicon, semantics, and syntax. A number of applications is described.

Enkonduko

En lingvistika kurso pri internacia komunikado kaj planlingvoj (Sherwood & Cheng 1980), la PLATO-sistemo por perkomputila instruado estis uzata kiel unu el la instruiloj en la studo de Esperanto, kun interagaj lecionoj verkitaj plejparte de Judith Sherwood (J. Sherwood 1981, B. Sherwood 1980b, 1981). Multaj diversaj tipoj de perkomputilaj prezent-manieroj estis esplorataj, inkluzive de diversaj tipoj de komputila traktado de Esperanta teksto, kiaj sintezo de parolado, preparo kaj uzo de leksikonoj por ĉeterminalaj lego-ekzercoj, kaj simpla permaŝina tradukado. Sperto kun ĉi tiuj esploroj ebligas iom kompletan resumon de la bazaj aspektoj de komputila prilaboro de Esperanto.

Ortografio kaj komputila reprezento

Estas dudek ok fonemoj en Esperanto, ĉiu reprezentita en ortografio per unusola litero. La alfabeto estas abcĉdefgĝĥĥijĵklmnoprsĝtuŭvz, kie la literoj signas proksimume la samajn fonetikajn valorojn kiel en iuj formoj de la Internacia Fonetika Alfabeto, escepte de ĝ = [dʒ], ĥ = [x] (kiu jam preskaŭ malaperis de la lingvo, plejparte anstataŭigita per k), ĵ = [ʒ], kaj ŭ = [w]. Tradicie la ŭ skribatas kiel ŭ, sed oni ofte vidas aliajn formojn en multaj eldonoj, ekz. ŭ, ŭ, ktp. La supersignoj kaŭzas problemojn en multaj komputilaj sistemoj kiuj desegnatitis nur por la angla (ofte eĉ nur por majuskloj). Ĉi tio igis iujn uzantojn de komputiloj proponi diversajn konvenciojn por reprezenti supersignojn, ekz. ĵ kiel "j." (Schlüter 1972) aŭ "j>" (Frank 1979). Mi mem opinias ke oni jam atendis tro longe por ke komputiloj respektu lingvajn konvenciojn, kaj estas la komputiloj kiuj devus ŝanĝiĝi, ne la lingvoj. Escepte de la angla kaj la nederlanda, preskaŭ ĉiuj eŭropaj lingvoj havas supersignojn, kaj la Internacia Organizaĵo por Normigado (1975) starigis konvenciojn por trakti ilin. En la estonto oni postulos ke komputilaj sistemoj povu trakti diversajn ortografiojn.

Tamen, la nuntempa socia pozicio de Esperanto estas tia, ke ni ne povas esti tiel postulemaj kiel la parolantoj de naciaj lingvoj. Krome, estus teknike tre avantaĝe se la Esperanta ortografio estus subaro de la naciaj ortografioj, kio ĝi estus sen la supersignoj, ĉar solvoj kiaj tiuj de la Internacia Organizaĵo por Normigado kutime konsistas el tio, ke al baza literaro komuna al ĉiuj latinliteraj lingvoj oni aldonas iujn signojn propraj al la koncerna lingvo-grupo.

Do por la franca oni aldonas é, è ktp, kaj por la germana

oni aldonas ŭ ktp. Postuli ke klavaro de franca komputilo aŭ skribmaŝino havu krom la apartajn francaĵojn ankaŭ Esperantaĵojn estas iom tro. Oni prave rimarkas, ke modernaj pres-teknikoj plejparte eksolvas la malnovan problemon de manko de supersignoj ĉe presistoj. Tamen, kiel Eichholz (1980) trafe argumentas, la situacio ĉe komputiloj ankoraŭ bezonas raciigon, ankaŭ pro interligoj inter diversaj komputiloj. Ekzemple li rimarkas, ke franca komputilo (kun é kaj germana komputilo (kun ü) bone povas interŝanĝi informojn en la angla, ĉar la angla literaro estas subaro de la storaj kaj eliraj kapabloj de ambaŭ sistemoj. Male, la Esperantaj supersignoj kaŭzos ĝenojn. Kiel Eichholz diras, ne sufiĉas ke la ortografio de Esperanto estu tiel same oportuna kiel la ortografio de aliaj lingvoj: ĝi devas esti sambona kiel la angla por interŝanĝo de komputilaj informoj inter diversnaciaj komputilaj sistemoj.

Kompreneble ekzistas la Zamenhofa konvencio ke ŝ estu reprezentata per sh, kaj ŭ skribatu simple kiel u. Tamen ĉi tio estas tre malkonvena en multaj komputilaj situacioj, ĉar ĝi enkondukas gravajn problemojn por alfabetigo, sintezo de parolado, aŭtomata tradukado, ktp, pro tio, ke oni ne povas per programo distingi inter ekzo longhara kaj longĝara, inter ĉashundo kaj ĉaŝundo, ktp. Homo povas facile distingi per kunteksto, sed komputilo ne. Kiel ni rekonu per programo, ke "ankau" (ankaŭ) ne estas imperativa verbo, aŭ ke "audi" (aŭdi) ricevu emfazon ĉe la a, ne la u?

Do la Zamenhofa konvencio tute ne estas uzebla en komputilaj traktadoj. Estas ankaŭ notinde, ke Esperantistoj neniam multe uzis la formojn kun -h, plejeble pro sento de malbela precipe ĉe gh kaj jh. Fischer (1980) rimarkigas, ke oni ofte en naci-lingvaj informoj forlasas la supersignojn aŭ uzas aliajn konvenciojn, do verŝajne -h ne kontentigas eĉ ekster komputiloj.

Eichholz proponas uzon de -x: ĉ = cx, ĝ = gx, ĥ = hx, ĵ = jx, ŝ = sx, kaj ŭ = ux. Kiel faris li, mi dum la resto de cxi tiu alineo uzos tiun sistemon por montri, ke gxi estas ne tro malbela, eble pro ia rilato inter x kaj ^ kaj La avantaĝo de -x anstataŭ -h devenas de la neuzo de x en Esperanto kaj gxia preskaŭ lasta loko en la latina alfabeto, tiel ke "masxo" nepre ordigxos post "masko", dum "masho" ordigxus antaŭ "masko", kontraŭ la tradicia ordo. Poste Eichholz trovis en "Esperanto", 1962 septembro, konsilojn el la Unua Internacia Konferenco pri la Lingva Problemo en la Scienco kiuj preskaŭ tute kongruas (escepte de sugesto uzi w por ux). Eichholz emfazas, ke oni uzu -x ne gxenerale sed nur en komputilaj kaj aliaj situacioj kie

necesas. Cxiam kiam eblas, oni uzu la tradician ortografion. Li ankaux notas, ke komputiloj iam povos auxtomate elpresi la tradiciajn supersignojn kiam estas storita -x.

La sugesto kaj argumentado de Eichholz estas tre bonaj. Tamen Fischer (1980 kaj persona letero) montras problemojn ankaŭ en ĉi tiu solvo. Eĉ se x estas proksime de la fino de la alfabeto, ĝi tamen ne estas plejlasta. Do "neuzo" (ne-uzo) devus antaŭi "neŭtrala", sed "neuzo" sekvus la vorton "neuxtrala". Dum Eichholz diras ke en neniu radiko la litero z sekvas c, g, h, j, aŭ s, Fischer notas ke jes ekzistas "zigzago", kaj en vortkunmetoj iu ajn litero povas stari antaŭ z (kaj ofte kunmetitaj vortoj estas listigitaj en vortaroj). Verŝajne estus treege malmultaj esceptoj al la norma ordo se ni uzus -x, sed ekzistus kelkaj. Pli grave, Fischer notas ke uzo de -x komplikigas multajn komputilajn procedojn. Ekzemple, aŭtomata dispartigo de vorto ĉe fino de linio devus kontroli, ĉu estas -x, kiun oni certe ne movu al la komenco de nova linio. Multaj algoritmoj dependas de la longo de vorto. Kun enmetitaj literoj x, la longo de la vorto ne kongruas kun la nombro de sonoj, nek kun la longo de la vorto sur ekrano se la terminalo ŝanĝas x en supersignon, kio ĝenas ĝustan elmontron de teksto.

Fischer vidas nur du verajn solvojn: aŭ uzi internan kodon 1-28 por reprezenti nian alfabeton (kun la neceso konverti al ĉi tiu reprezento ĉe klavara eniro kaj el ĝi por ekrana aŭ presa eliro, ĉar tiuj kodoj ne korespondos ekstere al la ĝustaj klavoj kaj signoj), aŭ ŝanĝi la alfabeton radikale. Li proponas skribi ĉ = tx, ĝ = dy, ĥ = q, ĵ = y, ŝ = x, kaj ŭ = w. Notu ke niaj fonemoj ĉ kaj ĝ estas dusonaj (tŝ kaj dĵ), kaj en la propono de Fischer oni skribus ilin per la korespondaj literoj, do tx kaj dy. Li ankaŭ sugestas skribi la dufonan fonemon c kiel ts. Tiel perkomputila konverto de parolado al skribo (se tio iam ĝenerale eblos!) estus unuforma. Notu ke en ĉi tiu nova skribsistemo estus 26 literoj, kaj la ordo de vortoj en vortaro estus malsama kompare kun la nuna ordo. Ekzemple, ŝipo (= xipo) sekvus vinon en la vortaro. Fonetike ĉio ĉi estas ĝusta, sed ĝi multe ŝanĝas la aperon de multaj vortoj al malpli internacia formo (Interŝanĝo de y kaj j iomete plibonigus la aferon). Eble malsimileco kun jam internaciaj formoj ne devus esti grava baro, sed ĝi sendube vekus akrajn kontraŭstarojn, ankaŭ pro la nefundamenteco de la propono. Tamen Fischer pravas pri la logika strukturo de la problemo kaj la proponitaj solvoj. Kiam la Akademio proklamis ke ne ekzistas ortografiaj problemoj, ili certe ne estis pensantaj pri la intimaj rilatoj inter signo kaj kodo ĉe

komputiloj sed nur pri la aspekto de presita informo. Kaj ili plejeble ne rimarkis tion, ke la Esperanta alfabeto ne estas tute unuforma rilate al transskribo de parolado (c-ts, ĉ-tŝ, ĝ-dĵ).

La PLATO-sistemo ĝuste traktas supersignojn por larĝa gamo da lingvoj, inkluzive de Esperanto, per uzo de kaŝita rega kodo en la interna reprezento de litervicoj. Dum elmontraĵoj operacioj ĉi tiu kodo interpretiĝas kiel komando por retroiri kaj surmeti supersignon laŭ la sekvanta kodo. Se la subkuŝanta litero estas majuskla, la supersigno estas levita supreten por pli bona apero. Pro tio ke la rega kodo havas nombran valoran pli altan ol z, rezultas ĝusta alfabeto ordigo: ekzemple, ŝafo sekvas supon. Notu ke operacioj pri litervicoj devas perlabori ŝ kiel tri-kodan unuon (s, rega kodo, kodo kiu signas cirkumflekson anstataŭ alian supersignon). Oni povus uzi kaj ja uzas alternativan metodon por la rusa kaj aliaj nelatina alfabetoj en la PLATO-sistemo. Eblas krei ĝis 126 specialajn literformojn kaj uzi ilin kiel duan alfabeton. Oni do povus krei 28 Esperantajn literojn kunligitajn al kodoj 1-28, tiel evitante plurkodajn komplikaĵojn, sed ni ne faris tiel, parte ĉar ekzistas aliaj bonaj uzoj por tiuj kreeblaj signoj (ekzemple, por fari moveblan bildeton de aŭtomobilo el pluraj desegneblaj "literoj"). La jena solvo estas plejparte kontentiga, eĉ se la plurkodeco de la ĉapelitaj literoj ĝenetas en iuj algoritmoj. Estas ankaŭ vere ke la tajpado de Esperantaj vortoj kaj frazoj estas iom ĝena por lernantoj, ĉar ili devas frapi ekstran klavon por surmeti supersignon. Ankaŭ por mi estas notinde, ke mi ne povas tajpi tiel rapide en Esperanto kiel en la angla, pro la ekstraj klavofrapoj bezonataj.

La PLATO-sistemo permesas ĝustan trakton de lingvoj pro tio ke ĝi desegnatis por instruo, inkluzive de instruo de lingvoj por kiu la angla alfabeto tute ne sufiĉas, kaj pro tio ke estis ĉe la desegnistaro idealo, ke komputiloj adaptiĝu al homoj kaj ne inverse. Tamen, eĉ se pro multa laboro la PLATO-terminaloj permesas ĝustan eniron kaj elmontron de multaj lingvoj, restas aliaj komercaj iloj kiuj ĉiam bezonas apartan traktadon. Ekzemple, mi tajpis ĉi tiun artikolon ĉe terminalo, kaj nur post multa prilaboro mi surpaperigis ĝin per kunligata tajpilo. Tiu moderna tajpilo permesas programan regadon de pozicio, laŭ tre fajna lokigo. Speciala programo retroiras kiam ĝi renkontas supersignan kodon, sed dum multaj monatoj de pledado oni ne aranĝis la programadon tiel, ke la supersigno aperu pli alte ĉe majuskloj, do mi ofte devis post eltajpo manskribi pli altan cirkumflekson sur C. Jen la tipo de ĝeno kiu venas eĉ

en relative bona situacio kia ĉe PLATO. Estas vere ke ankaŭ multaj naciaj lingvoj havas ĉi tiajn problemojn, sed estas nekontentige ke Esperanto ortografie ne estu pli bona ol naciaj lingvoj, kaj ke ĝi estu malpli bona ol la angla.

Fonetiko

La Esperanta skriba kaj sona sistemoj treege faciligas perkomputilan konverton de teksto en sintezan paroladon. En la prononcado de tiuj, kiuj estas konsiderataj "bonaj" parolantoj, estas nemulta alofona variado de la Esperantaj fonemoj. (Por bonega diskuto de la soci-lingvistika signifo de "bona" Esperanta prononcado, vidu Wells 1979:22-26.) Ĉar la prononco de litero ne multe varias laŭ kunteksto de apudaj literoj, oni povas simple eligi ĉiam la saman sonon por ĉiu aparta litero. Kontraste, la hispana skribsistemo, kiu fidele kaj senescepte reprezentas la sonojn, tamen postulas ortografiajn kaj fonologiajn regulojn por preskaŭ ĉiu litero (la litero *c* havas tri prononcojn -- *k*, *s*, kaj *ĉ*, "un beso" prononcatas "umbeso", ktp). Por ne paroli pri la teruraj malfacilaĵoj de la angla skribsistemo! Ne tute validas en Esperanto la principo "unu litero, unu sono", almenaŭ se oni konsideras ke por eligi *c*, *ĉ*, kaj *ĝ* oni devas komandi sintezilon produkti dusonajn kombinojn *ts*, *tŝ*, kaj *dĵ*, sed "unu litero, ĉiam la samaj sonoj" estas adekvate valida (sed en la kontraŭa direkto de parolado al teksto estas dusencaĵoj). Por detaloj pri sintezo de Esperanto vidu Sherwood 1978, 1979a, 1979b, 1980b, 1981.

La regulo por emfazo estas tre simpla: dum algoritma marŝo tra vorto, elektante kaj listigante por ĉiu litero korespondajn sonajn kodojn por sintezilo, notu la poziciojn de la vokaloj, kaj je la fino de la vorto ŝanĝu la kodon por la antaŭlasta vokalo tiel ke ĝi estu pli longdaŭra. (Je la fino de tuta frazo tiuj emfazitaj vokaloj ricevos pli altan voĉfrekvencon, kiel ni vidos poste.) Mi trovis, ke bona Esperanta ritmo rezultas se ĉiuj ne-emfazitaj vokaloj havas egalan daŭron kaj emfazitaj vokaloj havas daŭron 50% pli longan. Se la vorto enhavas nur unu vokalon, ne emfazu ĝin. Jen sufiĉe grava punkto: kiel en la latinidaj lingvoj, en Esperanto unu-vokalaj vortoj kutime estas negravaj kaj tial ne-emfazendaj. En la angla, germana, kaj rusa, unu-vokalaj vortoj povas esti gravaj kaj emfazendaj, tiel postulante komplikan sintaksan aŭ leksikan traktadon por distingi, kiujn vortojn oni emfazu kaj kiujn ne. Se la unuvokala vorto estas ĉe la fino de frazo, mia algoritmo tamen emfazas ĝin, kiel en "Mi volas pli". Se la lasta vorto en frazo estas plurvokala, mi ne nur emfazas la antaŭ-

lastan vokalon per plilongigo (kaj intonacia plialtigo), sed mi ankaŭ plilongigas la lastan vokalon, pro konata fenomeno, ke plejlastaj sonoj en frazo prononcatas malrapide.

Demandoj kun respondo jes aŭ ne en Esperanto komenciĝas per la vorteto ĉu kies ĉeesto aŭ foresto montras, ĉu la intonacio de la demando altiĝu aŭ falu. (Komparu: "ĉu vi iros?" "Kien vi iros?") Jen situacio signife pli simpla ol ĉe la plejmulto de eŭropaj lingvoj, en kiuj estas la aliaj demandoj (ne jes/ne) kiuj estas markitaj per "demando-vortoj", tiel postulante leksikonon de tiaj vortoj. Krome, tiuj demando-vortoj (kiu, kiam, kial, ktp) ne ĉiam estas je la komenco de la frazo: "Kun kiu vi iros?"

Je la fino de frazo, la listo de emfazitaj vokaloj permesas konstruon de intonacio. En normala deklara frazo la intonacio falas malrapide de la komenco ĝis la fino, kun lokaj altiĝoj ĉe la emfazitaj vokaloj. En ĉu-demando, anstataŭ falo, post la lasta emfazita vokalo estas leviĝo. Mi uzas sintezilon "Votrax", kiu akceptas kodovicon de 8-bitaj kodoj, el kiuj 6 bitoj regas sonon (estas 40 sonoj, kun 22 alidaŭraj vokaloj kaj 2 paŭzoj) kaj 2 bitoj regas kvar nivelojn de voĉfrekvenco. Formigo de intonacio konsistas el aldono de taŭgaj 2-bitajoj al la listo de 6-bitaj sonoj. Oni preferus havi pli larĝan gamon da frekvencoj. Kun nur kvar, mi konstruas sinsekvon 3-4-2-3-2-3-1, tiel ke la unua emfazo estas pli alta kaj la fino pli malalta ol la resto. Tio donas impreson de malrapide falanta intonacio, kun lokaj suprenirantaj perturboj ĉe la emfazitaj silaboj. Jen ekzemplo de frazo ("La rapida kato nun dormas"), kun plilongigitaj kaj altigitaj emfazitaj silaboj, kaj kun la tutlasta vokalo ankaŭ plilongigita:



Eĉ sen ia ajn sintaksa aŭ leksika pritrakto, la rezulta sinteza Esperanto estas surprize bona kaj komprenebla. Ĝi povus esti plibonigita per uzo de la gramatikaj finaĵoj ĉe Esperantaj vortoj por rekoni supraĵe la sintakson de la frazo, tiel ebligante pli naturecan ritmon kaj intonacion, sed mi ankoraŭ ne provis tion. Notu ke la simpleco de la sinteza procedo, kaj ĝia relative bona kvalito, dependas ne nur de la ortografio sed ankaŭ de la sonsistemo mem. Estas granda avantaĝo havi nur kvin facile distingeblajn vokalojn

(la angla havas dek du) kaj tre simplan ritmon (en la angla estas tre komplika vario de vokala daŭro depende de emfazo kaj apudaj sonoj). Tamen spertoj kun la uzo de sinteza Esperanto en instruado (Sherwood 1980b, 1981), kun sintezilo kiu ne tre bone produktas iujn konsonantojn, klare montras la fakton, ke la Esperanta konsonanta sistemo ne estas tiel ideala kiel la vokala sistemo. Estas sufiĉe subtilaj distingetoj inter iuj konsonantoj, kio ĝenas multajn lingvanojn. Ekzemple, la franca ne havas ĝo-sonon, do por francoj estas malfacile paroli kaj aŭdi distingeton inter ĝo kaj ĵo. La sonfamilio so, ŝo, co, ĉo kreas analogiajn problemojn por iuj lingvanoj, ĉar multaj lingvoj ne havas ĉiujn tiujn distingetojn. Estas konate, ke japanojn ĝenas la distingo inter lo kaj ro. Ĝuste kiel la anglaj vokaloj tro abundas kaj do tro similas unu la alian, estas tiel ĉe la Esperantaj konsonantoj, kaj eraretoj fare de sintezilo facile konfuzas unu konsonanton kun alia. (La angla havas kaj multajn vokalojn kaj multajn konsonantojn.)

Povas esti, ke ni bezonas grandetan konsonantaron por kapabli pruntepreni novajn vortojn el multaj lingvoj. Krome, kun iom da boneta instruado oni povas lerni kiel distingi inter la konsonantoj, kaj ĉio ĉi ne estas granda parto de la lernado de la lingvo. Sed oni ne fanfaronu, ke la sonoj kaj skriboj de Esperanto estas ĉiel tute idealaj. Oni ankaŭ vidu, ke la problemo de la supersignoj pli fundamente estas problemo de troriĉeco de la konsonanta sistemo. Laŭ analizo de konversacio (Sherwood 1980a), la sonoj ĉ, ĝ, ĥ, j, ŝ kaj ŭ estas maloftaj kaj sume reprezentas nur 2.1% de la sonoj (vidu Apendicon). Elimino de ĉi tiuj maloftaj sonoj jes simpligus la ortografion sed, multpli grave, forigus gravajn prononcajn problemojn por multaj lingvanoj. Dum la tuta vivo de Esperanto oni senĉese proponis ortografiajn reformojn. Verŝajne ĉi tiuj reformoj estis plejofte prezentitaj nur cele al forigo de la supersignoj en la skribosistemo, kun konservo de ĉiuj sonoj, sen zorgo pri la troabundo de la konsonantaro. Laŭ mia vidpunkto, pure ortografiaj ŝanĝoj estas negravaj kaj apenaŭ tuŝas la Fundamenton. Forigo de la sonoj mem postulus ĝisfundan, radikalan, kaj socie danĝeran reformon de la vortostoko, kion sole la Akademio rajtus fari.

Alia ekzemplo de komputila trakto de Esperanta fonetiko estas la esploroj kaj aplikoj de Giorgetti-Mas (1980) ĉe Nancy, Francio, pri perkomputila rekono de parolataj Esperantaj vortoj. Jam ekzistas por kelkaj lingvoj komercaj sistemoj kiuj povas rekoni unuopajn vortojn kun precizeco de pli ol 95%, se la parolata vorto estas unu el cent aŭ kelkcent registritaj vortoj kaj se la sistemo jam konas la

voĉajn proprajojn de la individua parolanto. Giorgetti-Mas kreis sistemon kiu registras Esperantajn vortojn parolitajn de instruisto. Sur ekrano lernanto vidas vorton kaj provas prononci ĝin, kaj ŝilia vorto estas komparata kun la ĝusta instruista vorto. Ĉar multaj eraroj de komencantoj estas antaŭvideblaj, la instruisto registras ne nur la ĝustan sed ankaŭ komunajn neĝustajn prononcojn. Se la vorto de la lernanto ne kongruas kun la ĝusta vorto, la sistemo komparas ankaŭ kun la aliaj por povi respondi ekrane ke "Ne, vi diris aĵo: diru aĵo." Kompreneble se la vorto de la lernanto ne kongruas kun iu ajn instruista vorto, la sistemo povas nur peti ripeton: ĝi ne kapablas konverti la parolatan vorton en literojn. Tia konverto, tiel multe sopirata, ankoraŭ ne estas realaĵo ie ajn en la mondo. Notu ke por ĉiu situacio la sistemo devas kompari la vorton de la lernanto kun nur kelkaj (4-5) instruistaj vortoj. Do kontraste kun aliaj aplikoj kie kelkcent-vorta komparado necesigas detalan konon de la karakterizoj de la parolanta voĉo, jen ĉi tiu instrusistemo povas funkcii kun la voĉoj de diversaj lernantoj.

Ĉu estus pli facile konverti Esperantan paroladon al teksto ol ĉe aliaj lingvoj? Ekzistas iuj faciligoj ĉe Esperanto. Ekzemple, en la rekono de flua parolado (kio estas multpli malfacila ol rekono de izolaj vortoj), granda problemo estas divido en vortojn, ĉar kutime neniu paŭzo okazas inter vortoj (la spaco inter vortoj en la skriba lingvo estas nur konvencio). En Esperanto tiu vortodivido estas helpata per tio, ke la plejmulto de vortoj finiĝas per vokalo aŭ diftongo (proksimume 70%, kaj pliaj 25% aldonas aŭ -n aŭ -s) kaj ke plursilabaj vortoj havas emfazon ĉiam ĉe la antaŭlasta silabo. Tamen la ĉefa problemo estas la dubsenceco de la akustika signalo. Ĉiu sono havas plurajn eblajn identigojn, kiujn homaj aŭskultantoj precizigas baze de lingvistika kaj kunteksta kompreno, kiuj estas tre malfacile programeblaj en komputilo.

Morfologio

Esperantaj morfemoj, ĉiuj nevariaj, dividiĝas en du kategoriojn: tiuj kiuj bezonas "gramatikajn finaĵojn", kiujn mi nomos "enhavaj" morfemoj, kaj tiuj kiuj ne bezonas tiajn finaĵojn, kiujn mi nomos "funkciaj" morfemoj. En ĉi lasta kategorio estas la difina artikolo la, prepozicioj, pronomoj, la "primitivaj" adverboj ne derivitaj de enhavaj morfemoj, nombroj, konjunkcioj, kaj interjekcioj. La gramatikaj finaĵoj estas la jenaj:

- a adjektivo
- o substantivo -j pluralo -n akuzativo
- e adverbo

	verbo	pasiva participo	aktiva participo
pasinta	-is	-it-	-int-
nuna	-as	-at-	-ant-
estonta	-os	-ot-	-ont-
infinitiva	-i	imperativa -u	kondiĉa -us

Krom signi rektajn objektojn, la akuzativo signas "direkton al" ĉe adverboj (dekstre, dekstren) kaj en prepoziciaj frazoj (en la domo, en la domon). La akuzativo ankaŭ anstataŭas prepoziciajn frazojn en esprimoj de kvanto kaj tempo (Mi atendos dum semajno = Mi atendos semajnon). Pro tio ĉi ĝuste diras Maas (1980) ke la MANKO de -n signas la SUBJEKTON, ne ke -n signas la akuzativon, ĉar -n povas signi diversajn aferojn. Funkciaj morfemoj povas memstari, sed ili ankaŭ povas nedevige havi gramatikajn finaĵojn: ena, ene.

Tiuj morfemoj tradicie nomataj prefiksoj kaj sufiksoj fakte estas enhavaj morfemoj, kaj tio kion oni ofte nomas derivadon en lernolibroj verkitaj por parolantoj de eŭropaj lingvoj fakte estas kunmetado. Ekzemple, kato katido, ido; bona, bonulo, ulo; skribi, reskribi, ree. Jen oni vidas ke, escepte de la gramatikaj finaĵoj, apenaŭ ekzistas "ligitaj" (neliberaj, nesendependaj) morfemoj en Esperanto (sed la enhavaj morfemoj kompreneble bezonas gramatikajn finaĵojn por formi vortojn). Ekzistas vira (ĉj) kaj ina (nj) kares-sufiksoj kiuj ligiĝas al stumpigitaj radikoj: patro, paĉjo; onklino, onjo; Petro, Peĉjo. Notu ke ĉi tiuj nekutimaj sufiksoj ne enhavas vokalojn, kaj ke la stumpigitaj radikoj estas la solaj esceptoj al la regulo, ke ĉiuj morfemoj estas nevariaj. Ekzistas teknikaj sufiksoj, precipe en kemio kaj medicino, kiuj ne uzeblas sendepende, kiel estas ankaŭ pri la sufikso -i- por nomigo de landoj (Francio, Japanio). Gramatikaj finaĵoj kaj karesaj, teknikaj, kaj landaj sufiksoj estas la solaj vere ligitaj morfemoj. La rigora aglutineco de nevariaj morfemoj faras la strukturon de Esperanto plej proksima al la turka kiam oni aplikas universalajn mezurojn kia la meznombro de morfemoj en vorto (Brozovil 1976), sed kontraste kun la turka la produktemaj "sufiksoj" estas fakte radikoj, donante al Esperanto iujn el la proprecoj de izola lingvo kia la ĉina.

Unu grava kategorio de funkciaj morfemoj meritas apartan diskuton pro dubeco, ĉu iuj el ili estu konsiderataj enhavaj morfemoj, kaj pro tio, ke ili bezonas specialan komputilan trakton en la prilaboro de teksto. La "korelativoj" inkluzivas 45 erojn kiuj almenaŭ por memor-

helpo povas estis konsiderataj kiel konstruitaj el kvin komencoj kaj naŭ finoj:

		-o	aĵo
		-a	tipo de
nedefinitiva	i-	-e	loko
demanda	ki-	-u	persono, specifa aĵo
elmontra	ti-	-al	motivo
inkluziva	ĉi-	-am	tempo
negativa	neni-	-el	maniero
		-es	posedo
		-om	kvanto

Tiuj korelativoj kiuj finiĝas per -o (kio, tio, ĉio) funkcias simile al substantivoj, kaj tiuj kiuj finiĝas per -a havas proprajojn similaj al adjektivoj. Sed notu ke adverbaj funkcioj de loko kaj maniero estas reprezentitaj per du malsamaj finaĵoj (-e kaj -el), dum adverboj formitaj el enhavaj morfemoj ĉiam finiĝas per -e: bone, oriente. La komencaj elementoj de la korelativoj certe estas ligitaj morfemoj. Ŝajnas plej simple, kaj eble necesas, trakti korelativojn kiel nedivideblajn funkciajn morfemojn, almenaŭ por komputilaj celoj. Ankaŭ ekzistas pseŭdo-korelativoj konstruitaj per analogio kun la enhava morfemo ali, kiu formas la normalajn vortojn alia kaj alie. Plejeble pro ĝia utileco kaj la hazardo, ke la radika morfemo finiĝas per la formo alies ekuziĝis komune. Mi havas la impreson ke aliaj korelativaj finaĵoj malofte uzatas kun ĉi tiu radikoj.

Ĉar enhava morfemo devas enhavi almenaŭ unu vokalon kaj devas kombini almenaŭ kun gramatika finaĵo, kiu ĉiam enhavas vokalon, vortoj kiuj enhavas nur unu vokalon nepre estas funkciaj morfemoj. Tial estas utile dividi vortojn en unu-vokalajn kaj plur-vokalajn kategoriojn kaj trakti ilin malsame. Krom diversaj kutimaj unulingvaj kaj dulingvaj vortaroj, la retroira vortaro de Schlater (1972) estis tre helpa por malkovri la specialajn kazojn diskutotajn.

Unu-silabaj vortoj (vortoj kun nur unu vokalo)

Inter la unu-vokalaj vortoj, se la vorto finiĝas per -i ĝi estas pronomo, escepte de ĉi, pli, pri, kaj tri. Se la vorto finiĝas per -in, ĝi estas akuzativa pronomo, escepte de kvin. Por komputilaj celoj ĉi tiuj eroj estas esceptoj al utila regulo, sed en tradiciaj priskriboj de Esperanto ne ekzistas regulo kiu deklaras, ke "ĉiuj unu-silabaj vortoj finiĝante per -i estas pronomoj, do ĝi ne estas "escepto" en la lingvo. Tamen Esperantaj parolantoj surpriziĝas kaj

amuziĝas kiam ili eknotas ĉi tiujn rilatojn, ĉar unuavide oni emas supozi, ke inter la unusilabaj vortoj estas nur la pronomoj kiuj finiĝas per -i.

Du tre oftaj vortoj estas nekutimaj pro tio ke ili havas pli da plursenceco ol havas la plejmulto de Esperantaj vortoj. Temas pri de (libro de montras posedon, venas de montras originon, kaj verkita de montras aganton) kaj dum (en dum mi legas ĝi estas adverbo sed en dum la vintro ĝi estas prepozicio). La rolo de ĉi tiuj vortoj devas esti elfosita de la kunteksto.

Plur-silabaj vortoj (vortoj kun pli ol unu vokalo)

Nombroj inter dek kaj dudek ofte skribatas kiel unu vorto (dekunu, dekdu, dektri, dekkvar, dekkvin, dekses, deksep, dekok, deknaŭ), sed ĉe pli altaj nombroj la dekoj estas apartigitaj (tridek kvin). Notu ke iuj el la nombroj inter dek kaj dudek false ŝajnas havi gramatikajn finaĵojn: dekunu kaj dekdu havas "imperativajn" finaĵojn, dektri havas "infinitivan" finaĵon, kaj dekkvin havas "akuzativan" finaĵon. Tial konsilindas kontroli por nombroj inter dek kaj dudek antaŭ ol procedi pluen en la analizo de plur-vokalaj vortoj. Oni rajtas diri ke ĉi tiuj unu-vortaj formoj perfortas norman Esperantan ortografion, ĉar estas ne la antaŭlasta silabo "dek" sed la fina nombro kiu estas emfazenda. Ĉar unu havas du silabojn mem, dekunu ne estas problemo tiurilate, sed la aliaj estas esceptoj en sintezo de parolado per konverto de teksto.

Escepte de amen kaj tamen, fina -n indikas akuzativon kaj povas esti fortranĉita post tia identigo. Post la fortranĉo de fina -n, fina -j post "a" (adjektivoj), "o" (substantivoj), aŭ "u" (korelativoj kiaj kiu) indikas pluralecon kaj povas esti fortranĉita post identigo. Notu ke la ofta kunmeto malplej ne estas pluralo, tial neprigante kontrolon ke la lasta vokalo ne estas "e" (-ij ne okazas en Esperanto). Korelativoj plej bone traktatas kiel specialaj kazoj, identigitaj aparte, interalie por certigi ke la vortoj iu/kiu/tiu/ĉiu/neniu ne ŝajnu imperativaj verboj. Post kontrolo de akuzativo (-n), pluralo (-j), kaj korelativoj, la gramatikaj finaĵoj -o (substantivo), -a (adjektivo), -e (adverbo), -is (pasinto), -as (nuntempo), -os (estonto), -i (infinitiva), -u (imperativa), kaj -us (kondiĉa) nedubece identigas vortajn kategoriojn, kaj la finaĵoj povas esti fortranĉitaj, lasante radikojn. La solaj esceptoj verŝajne estas minus (subtraho, ne kondiĉa verbo min-us), unu (ne imperativa verbo un-u), oni kaj ili (kutime pronomoj, ne infinitivoj on-i kaj il-i!), kaj, inter la oftaj kunmetoj, malpli kaj multpli (kiuj ne estas infinitivaj verboj).

Se la vorto ne komenciĝas per gramatika finaĵo, en ĝi ne povas esti enhavaj sufiksoj, sed povas esti prefikso (malantaŭ). Oni devus kontroli por la nombraj finaĵoj -dek, -cent, -mil: dumil tricent kvindek unu = 2351. Se estis gramatika finaĵo, povas esti participaj finaĵoj (pasivaj kaj aktivaj pasinto, nuno, kaj estonto -int-, -it-, -ant-, -at-, -ont-, -ot-) aŭ sufiksoj (il, id, ar, ist, ktp). Ĉi tiuj finaĵoj ĉiuj komenciĝas per vokalo, kio faciligas trovon kaj identigon. Tamen, dum tre malmultaj vortoj imitas gramatikajn finaĵojn, la participaj kaj sufiksaj finaĵoj estas ofte imitataj. Ekzemple, lekanto povas esti analizita kiel lek-ant-o (kiu lekas) aŭ kiel lekant-o (floro), kaj radaro povas esti rad-ar-o (aro da radoj) aŭ radar-o (elektronika sistemo). En multaj kazoj falsaj finaĵoj povas esti identigitaj kiel tiaj: facila ŝajne havas sufikson il (fac-il-a), sed ne ekzistas morfemo "fac". Simile, mastodonto (bestego) finfine ne povas analiziĝi kiel participo kun -ont-, ĉar "mastod" estas sensenca. Por povi malakcepti "fac" kaj "mastod" oni bezonas kompletan morfemaron disponeblan al la komputila programo.

Estas similaj konsideroj por prefiksoj. La du vortoj malgraŭ kaj malica ŝajne komencas kun la tre produktema prefikso mal, sed ĉi tio ne estas la normala analizo, eĉ se ludema retroformado liveras graŭ (pro) kaj ica (la malo de malica). Depende de la strukturo de la analizo, la komunaj kunmetoj malpli kaj multpli povus ŝajni infinitivoj. Ekzistas manpleno da aliaj ŝajnaj malvortoj, kiaj malario kaj maleolo. Veraj dusencaĵoj okazas en kazoj kiaj renomo (famo, ne re-nomo) kaj persone (person-e, ne per-son-e). Utila proksimuma regulo verŝajne estas: prenu la plej longan morfemon kiu trovatas en la leksikono, prefere al sinsekvo de pli mallongaj morfemoj. Ĉi tiu regulo probable funkcias bonete, eble pro la emo de parolantoj eviti dusencajn kunmetojn, escepte en intenca vortoludo (kia ĉe Raymond Schwartz).

Resumo de morfologia prilaboro

La sekvantaj reguloj resumas la antaŭan diskuton pri morfologia analizo.

- 1) Unu-silabaj vortoj: vortoj kiuj finiĝas per -i estas pronomoj kaj tiuj kun -in estas akuzativaj pronomoj. Esceptoj: ĉi, pli, pri, tri, kaj kvin. La komunaj vortoj de kaj dum bezonas kuntekstan traktadon por identigi la ĝustan signifon.

- 2) Plur-vokalaj vortoj: Kontrolu por dek ĝis deknaŭ. Forprenu -n por akuzativo, kun esceptoj amen kaj tamen. Forprenu -j por pluralo se antaŭigita de -a, -o, aŭ -u. Kontrolu por korelativoj. Forprenu gramatikajn finaĵojn, escepte en ili, oni, minus, malpli, multpli, kaj unu. Kontrolu eblajn nombrajn finaĵojn -dek, -cent, -mil. La resto devas analiziĝi por ebla kunmetado.

Propraĵoj utilaj en analizo de kunmetoj inkluzivas la fakton ke multaj produktemaj "sufiksoj" komenciĝas per vokalo kaj enhavas nur ĉi tiun solan vokalon, ke iuj "prefiksoj" estas ege produktemaj, kaj ke la plej longa morfemo ofte estas pli bona elekto ol alia analizo. En la traktado de du fabeloj kaj unu eseo, rezultis ke la sekvantaj morfemoj, inkluzivante kaj tradiciajn "prefiksojn" kaj prepoziciojn, estis produkteme uzataj kiel prefiksoj: al, bo, de, el, ek, en, ge, ne, re, dis, eks, for, kun, mal, mis, pra, sen, kaj tra.

Leksikono

En la strukturigo de komputile bazata leksikono, eble utilas konscii pri du sufiĉe diversaj vidpunktoj pri la gramatikaj kategorioj de Esperantaj enhavaj morfemoj. La tradicia vidpunkto, bone resumita de Wells (1969:7-9), estas ke enhavaj morfemoj dividiĝas en la kategoriojn substantivo, adjektivo, kaj verbo: martel apartenas al la substantivoj, bel al la adjektivoj, kaj kur al la verboj, tiel ke la bazaj vortoformoj estas martelo, bela, kaj kuri. Ŝanĝoj en la gramatikaj finaĵoj estas permesataj: marteli, belo, kuro. Sed por fari pluajn derivaĵojn oni devas uzi sufikson, kiel en martelado, ĉar ŝanĝi marteli al martelo liverus objekton, ne agon. En sia vortaro Wells estas fidela al ĉi tiu priskribo, donante gramatikan kategorion por ĉiu enhava morfemo.

Alternativan priskribon urĝas Szerdahelyi (1976, 1978), kiu argumentas ke la enhavaj morfemoj mem ne havas esencan gramatikan karakteron sed estas tute abstraktaj ĝis gramatika finaĵoj aldoniĝas. Laŭ lia vidpunkto, vortoj eniras la lingvon en siaj plenaj formoj (tio estas, martelo, ne la nuda radiko martel), ĉar prunteprenado el aliaj lingvoj devas eki de vorto, ne morfemo. Poste, procedoj kiuj estas aŭtonomaj en Esperanto liveras nudan morfemon per eltiro el la plena vorto. La vojo laŭ kiu la vorto eniris la lingvon do determinas la formon de derivado, ne iu esenca kategorio de la nuda morfemo. Mi konfesas, ke de praktika vidpunkto

mi ne vidas, kiel observebla diferenco povus rezulti el la du teorioj, eĉ se tiu de Szerdahelyi povus utili en la priskribo de la prunta procedo mem. La tradicia priskribo provizas utilan memorhelpan aparaton, kio plejeble ankaŭ utilos en komputila leksikono. Tial plejblas ke por multaj celoj estos avantaĝe asigni kategorion al ĉiu enhava morfemo.

Semantiko

La plejmulto de Esperantaj vortoj havas pli mallarĝan gamon da signifoj ol okazas en naciaj lingvoj, kaj samsoneco estas evitata kiel eble plej multe (sed ne tute, kiel oni vidas en radar-o/rad-ar-o kaj en la pluraj signifoj de de kaj dum). La principo de neceso kaj sufiĉo, proponita de René de Saussure (frato de la fama Ferdinand) kaj akceptita en 1913 de la antaŭulo de la Akademio de Esperanto, permesas kaj kuraĝigas la forlason de kuntekste nenecesaj sufiksoj. Ĉi tio povas kaŭzi problemojn por la komputila prilaboro de Esperantaj frazoj. Konsideru ĉi tiun paron da frazoj (Cherpillod 1979), en kiuj nedevigaj sufiksoj estas montrataj inter parentezoj:

Tiu ĉi desegn(aĵ)o estas tre plaĉa.

La desegn(ad)o de tiu figuro okupis min kvaronhoron.

Ĉar la kunteksto klarigas la signifon, estas bona stilo ellasi ĉi tiujn sufiksojn, almenaŭ kiam oni parolas aŭ skribas al homoj! Sed estas klare ke komputila programo havos malfacilaĵojn en la rekono de sufiĉa kunteksto por ĝuste identigi la signifon de desegno.

Problemoj de ĉi tipo igas min skeptika ke normala Esperanto havus signifan valoron kiel inter-lingvo en perkompuita tradukado, kiel proponita de Kelly (1978) kaj aliaj. La lingvo devus esti strikte limigita en siaj vortoformoj kaj sintakso, kaj se oni devas fari tiel, oni povus principe eki de iu ajn baza lingvo. Por multaj celoj, komputila traktado de Esperanta teksto estas pli simpla ol la prilaboro de aliaj lingvoj, kaj perkompuita tradukado el Esperanto en alian lingvon certe estas multe helpata de la gramatikaj finaĵoj. Sed finfine ĝi estas lingvo uzata de homoj, kaj kiam temas pri semantiko, la komputilo kiel kutime estas je granda malavantaĝo. Iasence inversa estas la laboro de Munnich (1979), kiu kreis komputilan lingvaĵon "Progreso", lerte kaj invente ekspluatante subaron de Esperanta sintakso tiel, ke programo verkita en Progreso estas legebla kiel normala (sed limigita) Esperanta teksto.

Sintakso

Mi povas malmulton diri pri sintakso, krom emfazi ke la gramatikaj finaĵoj ege simpligas la taskon determini supraĵan strukturon de frazo, eĉ sen leksikono. Kvankam tia analizo ankoraŭ ne estis aplikata al sintezo de parolado, evidentiĝas ke eĉ kruda analizo de vortofrazoj povus esti uzata por fari signifajn plibonigojn en la ritmo kaj intonacio de sinteza parolado. Distinga trajto de Esperanta sintakso estas relative libera ordo de subfrazoj, danke al eksplicita elmonro (-n) de la akuzativo. Mi iom parolos pri komputila tradukado poste.

Aplikoj

Komputila traktado de Esperanta teksto estas aplikata laŭ diversaj manieroj en perkomputila instruado de Esperanto pere de la PLATO-sistemo. Mi jam citis artikolojn pri interagaj instrumaterialoj kaj pri sintezo de parolado. Chin-Chuan Cheng kaj mi verkis programeton por traduki tre simplajn Esperantajn frazojn al la angla, parte por instrui al ni mem pri la procedo kaj parte kun la espero, ke post plua evoluigo lernantoj trovus utila kaj interesa la eblecon eksperimenti, konstruante Esperantajn frazojn, kun ĝustiga respondo reprezentita per la angla traduko. La baza skemo konsistas en la identigo de la gramatikaj kategorioj de la vortoj kaj grupigo de la vortoj en subfrazojn. Malgranda leksikono enhavas anglajn ekvivalentojn de unuopaj radikoj, kun specialaj notoj por malregulaj anglaj pluraloj kaj verboj. Ni ne provis dispecigi kunmetitajn vortojn, do vortoj devas enhavi sole unu morfemon (krom gramatikaj finaĵoj. Maas (1980 kaj personaj komunikoj) de la Universitato de Saarlando jam iris multpli profunden per ekspluato de potencaj iloj por manipulado de frazaj strukturoj. Li rimarkigas ke la malpli granda meznombro da signifoj de Esperantaj vortoj estas granda helpo, sed ankaŭ li ankoraŭ ne alfrontis la defian problemon de morfologia kaj semantika analizo de kunmetitaj vortoj. Maas emfazas, ke la karakterizaj mal-vortoj en Esperanto ofte prezentas eksterordinarajn malfacilaĵojn por reprezento en aliaj lingvoj.

Michael Pogue, talenta postdiploma studento de elektronika inĝenierado (nun laborante en Bostono), kreis legajn ekzercojn kiuj faciligas eltrovon de nekonataj signifoj far de la studantoj sed samtempe donas al ili utilan ekzercadon pri disdivido de vortoj en la konsistajn morfemojn. Teksto aperas ĉe la ekrano de PLATO-terminalo.

Se estas vorto kiun la studanto ne konas, ŝili simple tajpas la vorton, dividante ĝin en unuopajn morfemojn. Ekzemple, la vorto malbona devas esti tajpita "mal-bon-a", en kiu kazo la programo respondas kun la signifoj de ĉiu elemento. Se la studanto tajpas nedividitan vorton "malbona", la respondo estas "radiko ne en vortaro". Kun iom pli da programado, kompreneble estus eble provi analizi la nedividitan vorton por la studanto, sed en ĉi tiu legata kunteksto ŝajnas preferinde doni al la leganto ekzercadon pri analizo de vortoj en morfemojn, ĉar ĉi tio estas tre grava elemento en lerta legado de Esperanto.

Dum Pogue entajpis pliajn legajn ekzercojn, ŝajnis utile uzi morfologian analizon por helpi lin enigi la difinojn. Pro teknikaj kialoj, kaj pro tio ke ni ankoraŭ ne kreis plenan enkomputilan leksikonon, ĉiu legaĵo havas sian propran leksikonon. Li kaj mi evoluigis programon kiu prezentas la vortojn de la nova teksto sur la ekrano, unu post unu, montrante plurajn elektojn de eblaj aŭ probablaj morfologiaj analizoj. Ekzemple, la vorto malfacila estas montrata en la formoj "malfacil-a", "malfac-il-a", "malf-ac-il-a", "mal-facil-a" kaj "mal-fac-il-a". Oni simple indikas la kvaran formon, kaj la programo tiam petas difinojn de tiuj elementoj ne jam troveblaj en la kreskiĝanta leksikono.

Jam studeblas du fabeloj de Andersen, kaj la letero de Zamenhof al Borovko. Poste mi uzis multon el ĉi tiu programo por statistike analizi duonhoran sonbendan konversacion inter William Auld kaj Peter de Smet (Sherwood 1980a). En apendico mi ripetas el tiu studo la frekvencojn de fonemoj kaj gramatikaj finaĵoj.

Kompreneble estas aliaj eblaj aplikoj de la teknikoj priskribataj ĉi tie. Mi esperas, ke ĉi tiu resumo utilos kaj al tiuj interesataj pri instruaj aplikadoj kaj al tiuj kiuj ŝatus statistike studi Esperantajn tekstojn.

Dankoj

Mi dankas Chin-Chuan Cheng, Michael Pogue, kaj Judith Sherwood por kunlaboro kaj helpo. Mi ankaŭ dankas miajn hejmajn fakojn pro malavara ebligo de studado de lingvistiko dum la jaro 1979-1980, kaj la Fakon de Lingvistiko pro gastamo dum tiu periodo.

Referaĵoj

- Brozovi, Dalibor (1976). Pri pozicio de Esperanto en lingva tipologio. En Zlatko Tišljarić, red.: Internacia Lingvistika Simpozio. Zagreb: Internacia Kultura Servo. 33-84,
- Cherpillod, André (1979). Necesoj kaj sufiĉo. Heroldo de Esperanto, aprilo 3.
- Eichholz, Rüdiger (1980). Lingva konsilejo. Lumo (organo de la Kanada Esperanto-Asocio), Vintro, 13-17.
- Fischer, Rudolf-Josef (1981). La ĉapelitaj literoj en informatiko. Manuskripto.
- Frank, Helmar (1979). Planlingva dokumentado. En T. Carlevaro kaj G. Lobin, red.: Einführung in die Interlinguistik. Alsbach: Leuchtturm-Verlag. 199-209.
- Giorgetti-Mas, Marie-Thérèse (1980). Instruo de prononcado per komputilo. Konferenco de la Internacia Kibernetika Asocio, Namur, Belgio.
- International Organization for Standardization (1975). Keyboard for international information processing interchange using the ISO 7-bit coded character set -- Alphanumeric area (ISO 2530-1975 E). Keyboards for countries whose languages have alphabetic extenders -- Guidelines for harmonization (ISO 3243-1975 E). [Klavaro por internacia informatika interŝanĝo uzante 7-bitan kodaronalfanombra kampo (ISO 2530-1975 E).
Klavaroj por landoj kies lingvoj havas ekster-alfabetajn erojn
Gvidlinioj por harmoniigo (ISO 3243-1975 E).]
- Kelly, Ian (1978). How Esperanto can aid machine translation. [Kiel Esperanto povas helpi perkomputilan tradukadon.] Computer Weekly, januaro 19, 14. Estas aldona diskuto en postaj numeroj (februaro 9 kaj 16, marto 23 and 30, aprilo 6, 13, 20, kaj 27, kaj junio 15).
- Maas, Heinz Dieter (1980). Aŭtomata tradukado en kaj el Esperanto. Konferenco de la Internacia Kibernetika Asocio, Namur, Belgio.
- Münnich, Antal (1979). Enkonduko en la program-lingvon PROGRESO/1. En Carlevaro, T., & G. Lobin, red.: Einführung in die Interlinguistik [Enkonduko en la Interlingvistikon]. Alsbach, Germany: Leuchtturm-Verlag. 211-225.

- Schlüter, Klaus (1972). *Inversa Vortaro*. Nurnberg: Ludwig Pickel.
- Sherwood, Bruce (1978). Fast text-to-speech algorithms for Esperanto, Spanish, Italian, Russian, and English. [Rapidaj algoritmoj por sintezi paroladon el teksto en Esperanto, la hispana, la itala, la rusa, kaj la angla.] *International Journal of Man-Machine Studies* 10, 669-692.
- Sherwood, Bruce (1979a). Schnelle textgesteuerte Sprachsynthese- Algorithmen für Esperanto, Spanisch, Italienisch, Russisch und Englisch. [Rapidaj algoritmoj por sintezi paroladon el teksto en Esperanto, la hispana, la itala, la rusa, kaj la angla.] *Grundlagen-studien aus Kybernetik und Geistes-wissenschaft* 20, 109-121. (Mallongigo kaj traduko de Sherwood 1978.)
- Sherwood, Bruce (1979b). The computer speaks. [La komputilo parolas.] *IEEE Spectrum* 16, 18-25.
- Sherwood, Bruce (1980a). Statistical analysis of conversational Esperanto, with discussion of the accusative [Statistika analizo de konversacia Esperanto, kun diskuto pri la akuzativo]. Prelego prezentita ĉe Simpozio pri Gepatra-lingva Influo ĉe Esperanto, Urbana, Illinois, Usono, 1980 majo 10.
- Sherwood, Bruce (1980b). Spertoj pri sintezo de Esperanta parolado. Konferenco de la Internacia Kibernetika Asocio, Namur, Belgio.
- Sherwood, Bruce (1981). Speech synthesis applied to language teaching. [Sintezo de parolado, aplikata al lingvo-instruo.] *Studies in Language Learning* 3.
- Sherwood, Judith (1981). PLATO Esperanto materials. [Esperantaj materialoj ĉe PLATO-sistemo.] *Studies in Language Learning* 3.
- Sherwood, B. A., & C. C. Cheng (1980). A linguistics course on international communication and constructed languages. [Lingvistika kurso pri internacia komunikado kaj planlingvoj.] *Studies in the Linguistic Sciences* 10, 189-201. Ankaŭ havebla de ERIC Clearinghouse, dokumento ED 182 995.
- Szerdahelyi, István (1976). La semantika modelo de Esperanto. En Zlatko Tišljarić, red.: *Internacia Lingvistika Simpozio*. Zagreb: Internacia Kultura Servo. 85-152.

Szerdahelyi, Istvki (1978). Vorto kaj vortelemento en Esperanto.
Literatura Foiro no. 50, 8-10.

Wells, John (1969). Esperanto Dictionary (en la serio "Teach Yourself
Books"). London: Hodder and Stoughton. New York: David McKay.

Wells, John (1979). Lingvistikaj aspektoj de Esperanto. Rotterdam: Centro
por Esploro kaj Dokumentado de la Monda Lingvo-Problemo.

Apendico: Frekvencoj en duonhora konversacio (3730 vortoj)

Fonemoj

e	11.4%	j	2.8%
a	11.3	v	2.3
i	9.9	g	1.0
o	7.8	f	0.9
n	7.4	c	0.9
s	6.8	ĉ	0.9
t	5.7	b	0.8
r	5.6	h	0.7
l	5.1	ĝ	0.5
k	4.5	ŭ	0.5
u	3.5	z	0.3
m	3.2	ŝ	0.15
d	3.0	ĵ	0.07
p	2.8	ĥ	0.005

Gramatikaj kategorioj

adjektivoj	7.4%	(13.8% kun -n)
substantivoj	14.0	(24.1% kun -n)
pronomoj	11.1	(8.2% kun -n)
korelativoj	7.6	(22.1% kun -n)
adverboj (-e)	6.5	(1.3% kun -n)

verboj

pasintaj	3.3
nunaj	9.3
estontaj	0.7
infinitivaj	2.3
imperativaj	0.5
kondiĉaj	0.6
aliaj	36.8

ESPERANTO KIEL PONTLINGVO EN KOMPUTILRETOJ -
MANKOJ KAJ RIMEDOJ

A.P.M. Witkam

BSO/Automation Technology b.v.
P.O. Box 8348 3503 RH Utrecht
The Netherlands

RESUMO

Oni povas uzi Esperanton kiel pontlingvon en plurlingvaj komputilretoj ekipitaj kun facilecoj por maŝin-tradukado. En tia rolo ĝi bonege plenumas la necesojn de normeco, reguleco, kompakteco kaj tenado, sed ĝia nivelo de maldubasenceco estas iom dubinda. La jena artikolo koncentriĝas sur la neperfektaĵoj de Esperanto rigardante leksikan kaj strukturan maldubasencecon. Sekvas prezento de strategio por solvi tiu problemon en la komputilĉirkaŭaĵo. Montriĝos ke la lingvo mem havas neuzitajn potencon por solvoj, kiuj estas uzeblaj kaj etendeblaj por produkti maldubasencajn kaj tamen kompaktajn frazojn.

1. ENKONDUKO

La koncepto mem de pontlingvo (PL) kiel pivoto en sistemo de maŝintradukado (MT) inter diversaj fonto- kaj cel-lingvoj estas jam vaste konata kaj akceptita, kaj ankaŭ uzata en kelkaj instalaĵoj. Rigardante la planadon aŭ uzadon de iu PL, kelkaj aŭtoroj (ref. 1 - 4) iam proponis kaj rekomendis la uzadon de Esperanto. Konsiderante la aspektojn de evoluigo kaj tenado, Esperanto estas aloga kiel PL pro sia laŭregula strukturo kiu ebligas procezi ĝin en komputilo, kaj pro ĝia facila lernebleco.

Antaŭ mallonge (5,6) oni proponis Esperanton kiel PL-on por grand-skalan uzadon en internacia komputilreto, kie ĝi alportos gravajn ekonomiajn avantaĝojn de normigo kaj kompakteco.

Sed estas ankaŭ alia kriterio por juĝi la funkciadon de iu specifa PL : ĝia grado de maldubasenceco. Kompare al aliaj homaj komunikilaj lingvoj Esperanto estas bonega en tio rilato. Ĝi enhavas nenian aŭ ege malmultajn el la malfacilaĵoj kiujn oni trovas ĉe aliaj lingvoj; ekzemple :

- parolhomonimeco (konfuzo de vorto kategorioj);
- maldubasenceco de kazoj aŭ nombroj (konfuzo de la nominativo/akuzativo, singularo/pluralo);
- problemoj de apudmetado (malklareso de la sfero de la adjektivoj);
- leksika homonimeco (nerilataj sencoj de la sama vorto).

Aliflanke, komparo kun aliaj porokazaj PL-oj (speciale verkitaj por MT-sistemoj) ŝajnas montri mankojn de Esperanto ĝuste kie temas pri maldubasenceco; dum rilate al la aliajn kriteriojn (normeco, kompakteco) ĝia supereco estas jam konvinka. Indas rimarki ke kompreneble la du celoj de komptakto kaj maldubasenceco konfliktas inter si.

La jena artikolo enfokusigas la atenton sur dubasenceco kiel ŝajna 'malfortaĵo' de Esperanto. Tial, enfokusitaj estos problemoj, anstataŭ meritoj. Parto 2 rezumas la diversajn tipojn de dubasenceco kiujn oni povas rekonti en Esperanto, kaj donas impreson pri la eksistantaj solvoj por ili. En parto 3 troviĝas skizon de strategio por ricevi maldubasencan Esperanto PL-on en komputilsistemoj.

2. DUBASENCECOJ EN ESPERANTO KAJ MANIEROJ POR EVITI ILIN

Kvankam malpli oftaj kaj malpli gravaj ol en multaj etnaj lingvoj, Esperanto tamen enhavas dubasencecojn de diversaj tipoj, kiuj estas en tiu ĉi parto mallonge esploritaj. Tiu ĉi parto, kies celo estas pli ilustrati ol doni kompletan liston, pritraktos fenomenojn semantikajn aŭ leksikajn (2.1 - 2.3), kaj fenomenojn sintaksajn aŭ strukturajn (2.4 - 2.7).

Oni indikos la diversajn ilustraĵojn, kaj samtempe la diversajn jam ekzistantajn manierojn por ilin eviti. Tiuj manieroj ĉefe elvenas el fontoj rekonitaj kaj aŭtoritataj, kiel la plej gravaj Esperanto-lernolibroj kaj vortaroj (8 - 11), kaj kelkaj monografioj (12 - 13).

Ne ĉiuj el la jenaj solvoj estas ordinare uzataj de parolantoj kaj skribantoj de Esperanto, parte ĉar ili estas nekonataj, sed ankaŭ ĉar oni simple ne bezonas ilin. Dum komunikado inter homoj, la ricevanto povas fidi al sia instinkta scio de la situaĉia ĉirkaŭaĵo por korekte interpreti la alie dubasencan frazon. Malsimile al homoj, komputiloj ne aparte bone interpretas la dubasencan lingvon, kaj tiu ĉi neceso revigligos kaj eĉ plifortigos la jam ekzistantajn aron de solvoj por dubasenceco.

2.1 Polisemio kaj homonimeco (dubasenceco de vortaj radikoj)

Polisemio (kunligitaj sencoj de la sama vorto) kaj homonimeco (ne kunligitaj sencoj de la sama vorto) oni ne ĉiam povas precize distingi unu de la alia. Sed dum homonimeco okazas nur ĉe limigita nombro (7) de vortradikoj, polisemio havas malfermitan limon al nebuleco kaj malprecizeco, kaj tial estas pli malfacile ĝin ekzameni. La sekvonta divido estas laŭ la tipo de solvo.

- a. Sinonimeco. Oni rezervas anstataŭan vortradikon por unu el la sencoj : *

<i>ero</i>	{ <i>ero</i> <i>erao</i>	<i>element</i> (An) <i>era</i> (An)
<i>talio</i>	{ <i>talio</i> <i>taliumo</i>	<i>waist</i> (An) <i>thallium</i> (An)

* La tradukado estos al la angla (An) aŭ al la franca (Fr).

- b. Precizigo. Unu aŭ ambaŭ de la signifoj de dubasenca vortradiko estas precizigitaj per aldono de alia radiko :

kubo	{ kubo luĉ'kubo	cube (An) die (An)
pendoli	{ pendoli pendol'veturi	swing (An) commute (An)
trepano	{ krani'trepano min'trepano	trepan (An) drill (An)
marko	{ fabrik'marko poŝt'marko	brand (An) stamp (An)
rivero	{ ĉef'rivero flank'rivero	fleuve (Fr) rivière (Fr)

Ofte tio neceso por precizigo rezultas el la principoj de vorto-formiĝo en Esperanto, kio estas bazita sur limigita grupo da radikoj. Do, vortoj kunmetitaj el du radikoj anstataŭas sendependajn radikojn de aliaj lingvoj. Sed kelkaj distingo (kiel *fleuve/rivière*) estas apartaj nur por kelkaj lingvoj, kaj do la pli ĝenerala vorto (*rivero*) ne estas vere dubasenca en la aliaj lingvoj.

- c. Apartaj fakoj. La dua signifo de vorto estas foje limigita al speciala fako aŭ kategorio, kaj tion oni trovas plej ofte indikita en la vortaroj (astronomio, juro, zoologio, muziko, ktp.). Kategoriaj vinjetoj aŭ mallongigoj nin helpas distingi en la jenaj ekzemploj :

baro	{ (fiziko)	barrier (An) bar
lukso	{ (fiziko)	luxury (An) lux
parabolo	{ (literaturo) (matematiko)	parable (An) parabola
genio	{ (mitologio)	genius (An) guardian spirit

- d. Apostrofo. Ĉe malgranda grupo de homonimoj oni uzas apostrofon por la pli malofta signifo :

koro	{ korp kor'o	heart (An) (biblia mezur-unuo por enhaveco)
------	--------------------	--

- e. Ekskluzivo. La kialo de kelkaj dubasencasajoj estas erara aŭ konversacia uzado, kaj ili ne estas subtenitaj sed eĉ klare malkonsilitaj aŭ notitaj kiel eraraj en la aŭtoritataj vortaroj :

absolvi	{ (erara)	exonerate (An) finish (studies) (An)
---------	--------------	---

<i>ĉelo</i>		<i>cell</i>	(An)
	(konversacia)	<i>cello</i>	(An)

2.2 Dubasenseco de vort-strukturoj

La vortoj en Esperanto estas formitaj el radikoj, senŝanĝajn afiksoj kaj finaĵoj, pere de aglutina sistemo. Tio signifas ke la kunmetitaj elementoj, la morfemoj, restas klare videblaj kaj identigeblaj. Tio faciligas la interpretadon kaj la analizon de la vorto. Tamen, estas malmultaj okazoj kiam la divido de iu vorto en morfemoj kaŭzas konfuzon. Tie ĉi ni distingas tri tipojn de solvo :

a. Indiki la morfemo-disigon. Uzado de la apostrofo por distingi morfemojn malebligas eraran interpreton : (pliaj troviĝas en (12)) :

<i>nom'ado</i>	<i>nomination</i> (An)
<i>nomad'o</i>	<i>nomad</i> (An)

<i>mark'oto</i>	(persono kiu estos markata)
<i>mar'koto</i>	(koto kiu troviĝas en la maro)

b. Evito de elizioj. Tio estas foje unu el la anstataŭaj eblecoj, kaj foje la sola solvo :

<i>kromproduktoj</i>	<i>pli da produktoj</i>
<i>kromoproduktoj</i>	<i>(aĵoj produktitaj el kromo)</i>

c. Enmetado de iu gramatika finaĵo. Por la tiel nomitaj Schnitt-homonimoj, Wüster (13 : 9.2) rekomendis enmeti la adjektivan 'a', aŭ la adverban 'e' je la ne-ĉefa morfem-limo.

2.3 Tempa/loka dubasenseco de prepozicioj.

La uzado de prepozicioj kiel *antaŭ* *before* (An) estas dubasenca ĉar ĝi povas aludi al tempa sekvo kaj ankaŭ al spaca rilato. Kvankam tiu duobla rolo en lingvoj estas preskaŭ universala, ja ekzistas esceptoj, kiel en la franca : *avant/devant*.

Por eviti problemon en tiutipa celo-lingvo, oni povas sekvi la proponon en (8), kaj uzi la precizan formon :

<i>iam antaŭ</i>	<i>avant</i> (Fr)
<i>ie antaŭ</i>	<i>devant</i> (Fr)

Samkiale oni konsilas la diversajn funkciojn de la prepozicio 'de' : *ekde, de post, detempe* ktp.

2.4 Konfuzo pri la gramatika kategorio de vorto

Ĉiu ajn el la sekvantaj naŭ unumorfemaj vortoj povas troviĝi en du el la tri kategorioj : prepozicio, konjunkcio, adverbo :

dum
ĝis
krom
por
anstataŭ
apenaŭ
kvazaŭ
antaŭ
ĉirkaŭ

kelkaj el tiuj vortoj oni povas uzi nur kune kun preciza partikulo, kiel ekzemple por ke, antaŭ ol. Estas ankaŭ proponoj etendi tiun uzadon (dum ke, dum kiam, ĝis kiam) (13), sed ŝajne mankas ĝenerala konsento pri ili (8).

2.5 Sintaksa dubasenceco kaŭzita de 'de'

La prepozicio ⁱde¹ estas en Esperanto grava, sed ankaŭ tre problemiga elemento. Krom la semantika dubasenceco de tempo/loko (vidu 2.3) oni renkontas ankaŭ la jenajn malsimilecojn en signifo :

de	1.	FROM	(ekiri de)
	2.	OF	(posedo)
	3.	BY	(la aganto en la pasiva formo)

Por eviti dubasencecon en frazoj kiel

la administradon de la filioj de la asekuritoj;
la manĝado de la rato de la kato;
la nutrado de la infano;
la infano forprenita de la patrino;

(8) rekomendas laŭnecesan uzon de tiuj variantoj :

<i>disde</i>	=	(AWAY) FORM	(An)
<i>al</i>	=	OF	(An)
<i>far (fare de)</i>	=	BY	(An)

kaj (9, 135) proponas uzi la akuzativon :

la manĝo de la raton de la kato.

La PIV (8) ankaŭ konsilas, kiam eblas anstataŭigi 'de' per iu pli preciza prepozicio, kiel : *en, el, pri, pro, super, ktp.*

2.6 Konfuzo de aktiva/pasiva objekto ĉe verboj kun IG

Krom problemoj kaŭze de la prepozicio *de*, dubasencecon inter pasiva aŭ aktiva objekto kaŭzas ankaŭ transitivaj verboj kun la sufikso IG. Ekzemple en,

La farmisto manĝigas siajn kokinojn.

ne estas klare ĉu la kokinoj estas nutritaj aŭ manĝitaj. La PIV (8) konsilas solvi tiu dilemon per enmetado de la aktiva (*ant*) participon aŭ la pasiva (*at-*) participon antaŭ la sufikso IG.

2.7 Struktura dubasenceco kun Prepozicia Frazo (FP)

En pluraj lingvoj foje ne estas klare ĉu la PF estas subordigita al la verbo, aŭ al la antaŭa Substantiva Frazo (SF).

Ekzemple

He saw the girl with the binoculars (An) :

'Li vidis la knabinon per/kun la bonokloj'.

En Esperanto, ĉar oni havas pli precizajn prepoziciojn (per = *by means of*, kun = *along with*'), la nombro de tiutipaj dubasencaĵoj estas limigita, sed ili tamen okazas :

Oni forprenis la infanon de la virino.

Kiel la PAG (9, 272) indikas, la sekvonta ŝanĝo en vort-ordo rezultas en la maldubasenca frazo

Oni forprenis de la virino la infanon.

Ĝenerale Esperanto permesas flekseblecon en la aranĝo de la ĉefaj vortgrupoj (subjekto, objekto, modifoj ktp), rilate unu al la alia. Aliflanke, oni havas fiksitajn regulojn por aranĝo de la vortoj kaj elementoj (SF, PF) en la vort-grupo mem. Tio ebligas kunmeti frazojn tiamaniere ke la limoj inter gramatikaj vort-grupoj estas facile korekte identigeblaj. Oni povas decidi pri tiuj limoj ankaŭ per la ĉeesto de la distinga akuzativa finaĵo kaj la akordo inter substantivo kaj ĝiaj determinaj adjektivoj.

3. CELE AL STRATEGIO POR MALDUBASENCA ESPERANTO-PL

En tiu Ei parto ni turnas nin al la specifaj konsideroj rilate al la uzado de Esperanto inter komputiloj. Tie ĉi ni alparolas ne nur spertulojn kaj laborantojn en la tereno de komputil-lingvistiko kaj arkitekturo de informad-sistemoj, sed ankaŭ la komunumo de Esperanto-spertuloj, kies helpo povus esti nemalhavebla.

3.1 La ĉefa rolo de PL

Konsiderante iun jam ekzistantan homan lingvon kiel PL-on, kreas la problemon, ke oni emas forgesi, ke tiu jam ekzistanta lingvo estos uzata laŭ tute alia maniero : ĝiaj frazoj ne plu celos homajn uzantojn, nek estos konstruitaj de ili.

Ĉe unu flanko de la transmisilo linio, komputil-programo nomita PL-KODANTO konstruos frazojn en la PL. Post konservado kaj transsendado tra la tuta reto, tiujn frazojn legos alia komputil-programo, nomita PL-MALKODANTO, ĉe la alia (malproksima) flanko. Tiuj du programoj kunlaboros kun pliaj elementoj en la mult-lingva komputil-sistemo (fig. 1) tradukante natur-lingvajn tekstojn el fonto-(FL) al cel-lingvo (CL), ekzemple el la angla al la franca.

La PL estas unuavice ponto inter komputil-programoj. Krome, ĝin kontrolos homo por evoluigo kaj tenado, sed ne tio estos ĝia ĉefa rolo. Tial, en la konstruado de PL-frazoj oni povas cedi faktorojn kiel akcenton, ritmon, eviton de monotoneco kaj ripetado ktp, favorante tipajn komputil-faktorojn, kiel maldubasencecon kaj standardeccon. Tiu ŝanĝo povos okazi en la gramatika kadro de la lingvo adoptita kiel PL, kaj ĝi estas grava kriterio por la planado de la PL-KODANTO kaj PL-MALKODANTO.

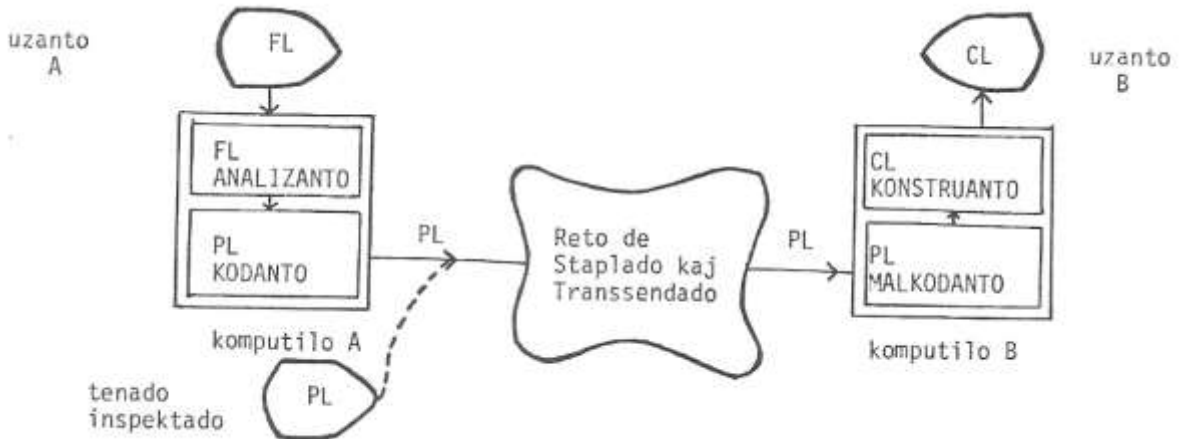


Fig. 1. Sur-PL-bazita tradukad-sistemo. La uzantoj normale funkcias nur je la nivelo de FL/CL. Tenado-laboristaro povas rekte interveni je la nivelo de PL.

3.2 La interna kaj ekstera PL-formatoj

La interna PL-formato estas tia formo en kia la PL estas staplita en la donitaj-bankoj kaj vojaĝas tra la reto de transdonado. La formato estas fakte kompaktita serio de bitoj, kaj ĝi celas ŝpari maŝinajn rimedojn kaj telekomunikadan band-larĝecon. Ĝi ĝenerale konsistas el miksaĵo de kod-elementoj diverslongaj kaj fikslongaj, kaj por homo ĝin analizi estas tede.

Lingvistoj kaj aliaj spertuloj implikitaj en la evoluigo kaj tenado de la PL-bazita sistemo devus povi inspekti kaj manipuli la PL-on plimalpli oportune. Tiacele, la interna PL-formato (la maŝin-lingvo) estas projektita laŭ ekstera formato, kaj inverse.

Ĉe speciale planitaj porokazaj PL-oj, la ekstera formato de la PL frazo estas agregato de donitaĵoj, kiun oni plej bone povas kompari kun asemblea lingvo (fig. 2.) Oni povas ĝin legi se oni alkitimiĝas al la apartaĵoj kaj detaloj de numeraj kaj mnemonikaj kodoj, kolum-aranĝo, ktp.

Sed kiam oni uzas Esperanton kiel PL-on, la PL-frazo estas nenio alia ol ordinare videbla (kaj, libervole, aŭdebla) Esperanto-frazo. La respondeca spertulo povas sin rekte kaj tute koncentriĝi sur lingvistikaj aspektoj, kaj la eksteran formaton oni plej bone povas kompari al altnivela program-lingvo. Tio estas unika avantaĝo de Esperanto kiel PL.

37.	49.	02.	SATZ	INSATZ	SATZ ETIMESIN-NO. 32									
32.	61.	01.	WORT	INSATZ	SECONDLY	34	0	0	0	6	6	0	ZWEITENS	
32.	62.	02.	WORT	INSATZ	.	9	0	0	0	0	0	0	.	
32.	63.	03.	WORT	INSATZ	WE	3	12	29	0	0	0	0	WIR	
32.	64.	04.	WORT	INSATZ	CAN	6	17	0	0	0	0	0	KOENNEN, KANN	
32.	65.	05.	WORT	INSATZ	MEAT	3	6	27	0	0	0	0	MITZE, MEIZEN	
32.	66.	06.	WORT	INSATZ	TIME	1	0	0	0	0	0	0	.	
32.	67.	07.	WORT	INSATZ	STEEL	3	26	0	0	0	0	0	STAHL	
32.	68.	08.	WORT	INSATZ	PROVE	4	23	0	0	0	0	0	UEBER	
32.	69.	09.	WORT	INSATZ	A	2	8	0	0	0	0	0	ZIH	
32.	70.	10.	WORT	INSATZ	CERTAIN	5	0	0	0	0	0	0	GEWISS	
32.	71.	11.	WORT	INSATZ	CRITICAL	5	6	0	0	0	0	0	KRITISCH	
32.	72.	12.	WORT	INSATZ	TEMPERATURE	3	27	0	0	0	19	21	22	TEMPERATUR
32.	73.	13.	WORT	INSATZ	.	9	0	0	0	0	0	0	.	
32.	74.	14.	WORT	INSATZ	AND	11	6	0	0	0	0	0	UND	
32.	75.	15.	WORT	INSATZ	THEN	14	6	0	0	0	0	0	DANN	
32.	76.	16.	WORT	INSATZ	ALLOW	6	23	0	0	0	16	0	0	GESTATTEN
32.	77.	17.	WORT	INSATZ	IT	3	12	29	0	0	0	0	ES	
32.	78.	18.	WORT	INSATZ	TO	4	23	0	0	0	12	6	0	ZU
32.	79.	19.	WORT	INSATZ	COOL	5	6	0	0	0	0	0	0	KUEHL, KUEHLEN
32.	80.	20.	WORT	INSATZ	AT	4	23	0	0	0	0	0	0	BEI
32.	81.	21.	WORT	INSATZ	DIFFERENT	5	0	0	0	0	0	0	0	VERSCHIEDEN
32.	82.	22.	WORT	INSATZ	RATES	3	27	29	0	0	21	22	0	GESCHWINDIGKEITEN
32.	83.	23.	WORT	INSATZ	.	0	0	0	0	0	0	0	0	.
32.	86.	01.	SATZ	INSATZ	WORTZAML = 23									
32.	81.	01.	WORT	ADVERB	ADVERB ZWEITENS									
32.	81.	01.	WORT	ADVERB	WORT DANACH								DECO 14 0 0 0 0 0 SECO 0	
32.	91.	01.	WORT	ADVERB	ENDE ZWEITENS								DI-BITS 9	
32.	82.	02.	WORT	BISES	SONDERZEICHEN 1								DECO	
32.	83.	03.	WORT	BISES	PRONOMENI WIR									
32.	84.	04.	WORT	VERB	VERB KOENNEN, KANN								DECO 6 17 0 0 0 0 SECO 0	
32.	84.	04.	WORT	VERB	MILFS/MODALVERB KOENNEN, KANN									
32.	84.	04.	WORT	VERB	SUBJEKT ZU MILFSVERBPOS. 3) WIR									
32.	84.	04.	WORT	AMBIG	I KOENNEN								DECO 6 17 29 0 0 0 SECO 0	
32.	84.	04.	WORT	VERB	VERB NACH MILFSVERB 5 MITZE, MEIZ									
32.	84.	04.	WORT	VERB	ENDE IKA 5 IKA 2									
32.	84.	04.	WORT	VERB	ENDE KOENNEN								DECO 6 17 29 0 0 0 SECO 0	
32.	85.	05.	WORT	BISES	PRONOMENI MITZE, MEIZEN									
32.	85.	05.	WORT	SURST	VERB MITZE, MEIZEN								DECO 3 6 27 0 0 0 SECO 2 7	
32.	85.	05.	WORT	VERB	VERB OR SUBST/WORT DAVOR KOENNEN								DMIM1 6 17 29 0 0 0 SMIM1 0	
32.	85.	05.	WORT	AMBIG	Z MEIZEN								DECO 3 6 27 0 0 0 SECO 2 7	
32.	85.	05.	WORT	SYMBI	ENDE MEIZEN								DECO 6 27 0 0 0 0 SECO 2 7	
32.	85.	05.	WORT	VERB	VERB MEIZEN								DECO 6 27 0 0 0 0 SECO 2 7	
32.	85.	05.	WORT	VERB	OBJEKT(ACT.), POS. 7 STAHL									
32.	85.	05.	WORT	VERB	ENDE IKA 8 IKA 12									
32.	85.	05.	WORT	VERB	ENDE MEIZEN								DECO 6 27 0 0 0 0 SECO 2 7	
32.	86.	06.	WORT	BISES	PRONOMENI STAHL									
32.	88.	08.	WORT	PRAP	PRAPPOSITION UEBER								DECO 4 24 0 0 0 0 SECO 0	
32.	88.	08.	WORT	PRAP	ENDE UEBER								DECO 4 24 0 0 0 0 SECO 0 0	
32.	10.	10.	WORT	BISES	ADJEKTIVI GEWISS									
32.	11.	11.	WORT	BISES	ADJEKTIVI KRITISCH									
32.	12.	12.	WORT	BISES	PRONOMENI TEMPERATUR									
32.	13.	13.	WORT	BISES	SONDERZEICHEN 1									
32.	15.	15.	WORT	ADVERB	ADVERB DANN								DECO 14 0 0 0 0 0 SECO 0	
32.	15.	15.	WORT	ADVERB	WORT DANACH GESTATTEN								DI-BITS 6 23	
32.	15.	15.	WORT	ADVERB	ENDE DANN								DECO	

Fig. 2. Impreso de la ekstera formato de porokaza PL.

Por certigi ke la ekstera PL-formato provizas kompletan kaj fidindan reprezenton de interna duplikato, la konvertado inter la du formatoj devas esti kiel eble plej rekta. Unu tia ekzemplo estas la konvencia kodado de signoj, la fiksita korespondado inter la almontrita signo kaj la binara kod-elemento. Sed por Esperanto, rekonita pro la konstanteco de ĝiaj morfemoj, kodi morfemojn ŝajnas pli taŭge. Tio implicas korespondadon de 1 al 1 inter morfemo (serio de signoj) kaj binaro koda elemento (numero de morfemo), kaj tio karakterizas Esperantan PL-on.

3.3 La tento aldoni al la interna kodo

Asigninte binarajn numerojn al Esperanto-morfemoj, unuavice ŝajnas pli interese jene daŭrigi :

- asigni 2 aŭ pli internajn kod-numerojn al la sama ekstera morfemo, por prizorgi polisemion kaj homonimecon, dubasencecon de prepozicioj ktp. (vidu 2.1, 2.3 - 2.5);
- aldoni internajn kod-elementojn por solvi kelkajn okazojn de struktura dubasenceco, por funkcii kiel gramatikaj etikedoj, aŭ kiel vort-grupaj apartigiloj, ktp. (vidu 2.6, 2.7).

Sed el tiu metodo sekvos kelkaj gravaj malavantaĝoj :

- I. Se la diferencigo kaj aldinaĵoj en la interna kodo ne estas spegulitaj en la ekstera PL-formato, la lasta donos malkompletan kaj do malfidindan bildon de la informo kiu estas vere transsendita. Tiu apartigo inter la interna kaj estera formatoj igos homan inspektiston sin turni al la interna formato.
- II La internajn kod-plilongigojn oni povus mapi sur eksterajn simbolo (numeroj, mnemonikoj) por ilin inkluzivi en la ekstera PL-formato. Malgraŭ sia kompleteco kaj fidindeco, la ekstera formato ne plu konsistos el puraj Esperantaj frazoj. Ĝi estos ŝargita de etikedoj kaj kodoj kiuj necesigos konvencion kaj alkutimigon, verŝajne ne subtenitajn per lingvistoj kaj Esperanto-spertuloj ekster la proksima komputilsistema ĉirkaŭaĵo. Rezultos PL kiu pli kaj pli devios el Esperanto, kun malpli oportuna homa kontakto kaj malkreskanta perspektivo de normeco.

3.4 La avantaĝoj de eksteraj solvoj por dubasenceco

Serĉante strategion por produkti maldubasencajn PL-frazojn, ni devas serĉi rimedojn kiuj obeas la karakteron de la PL-on elektitan. La eksterordinara trajto de Esperanto kompare al aliaj PL-oj, estas ĝia "malfermecon en la jenaj tri sencoj :

1. normeco : tutmonda kaj kun ĝenerala celo (ne limigita al ia MT-instalaĵo, aŭ al la limigita aplikebleco de la produktoj de iu fabrikanto);
2. alirebleco : la PL estas alirebla por ĉiu ajn kiu pretas lerni Esperanton (la plej facile lernebla lingvo);
3. evolukapabo : post jarcento da sperto, Esperanto ankoraŭ havas sufiĉan potencialon kreski (kvante kaj kvalite).

Kiel oni montris en parto 2, atentema esplorado de Esperanto montras ke ekzistas solvoj por ĉiuj tipoj de dubasenceco. Tiujn solvojn enhavas la lingvo mem, ili estas integraj al ĝi. Ilia uzado estas malofta, nur pro la malofteco de neceso solvi dubasencecojn en homa parolado.

Sed ĉe PL komunikado inter komputiloj, kie la maldubsenceco devenas esenca, oni povas komplete uzi la jam haveblan stokon de solvoj, tute en la kadro de la ekzistanta lingvo.

La signifa avantaĝo de tio ĉi strategio sur tio konsiderita en 3.3, estas la kohereco kio rezultas inter la interna kaj eksterna PL-formato. Oni strikte konservas korespondecon de 1 al 1 inter la binaraj kod-elementoj kaj la percepteblaj Esperanto-morfemoj; tiu igas la eksteran formaton kompleta kaj fidinda.

Plue, oni ne difektigos per entrudigi ekster-lingvajn elementojn la unikan homan alireblecon kiun Esperanto donas. La PL restos pura kaj klara Esperanto.

Laŭ tio strategio oni evitas dubasencecon per malfermitaj iloj : videblaj kaj komunikeblaj lingvaj elementoj; tio ebligas implikado kaj sin-devoto de larĝa grupo de lingvistoj kaj Esperanto-spertuloj.

Al la solvoj listigita en parto 2 oni povas aldoni la sekvantajn ĝeneralajn rimarkojn .

I. Vort-orĉo. La libera vort-orĉo en Esperanto ĝenerale servas por tri celoj :

- klareco (malĉasenceco);
- belsonco kaj akcento;
- adaptado al la gepatra lingvo de la uzanto.

Kiam homoj konstruas frazojn, oni kutime sekvas tiujn tri celojn farante iutipan kompromison. Sed kiam komputiloj konstruas tekston por ĝin sendi al aliaj komputiloj, grave estas nur la unua punkto el la tri, kaj do oni povas programi la PL-KODANTON konforme al tio. Alia helpilo por solvi ĉasencecon inter komputiloj sen afekti homan alirecon, estas pli multa uzado de la komo por marki la limon de vort-grupo en malfacilaj okazoj (en la PL-kodado oni traktas interpunkcion kaj interspacojn kiel morfemoj).

II. Vort ripetado. Anstataŭ ol anaforaj elementoj (personaj pronomoj, montraĵ pronomoj) aŭ en apozicio al ili, la PL povas simple ripeti la substantivojn kaj la proprajn nomojn kiam temas pri malĉasenceco. Tio validas en longaj kaj komplikaj frazoj, kaj ankaŭ por aludoj el unu frazo al la antaŭa.

Homaj uzantoj, kiuj tipe ne bezonas malĉasencitajn frazojn, kutimas eviti ripetadon kaj preferas pli varian stilon.

Kvankam PL bazita sur tio strategio estos tute Esperanto, ĝi estos karakterizita per tre rimarkinde multaj ĉasenceco-rimedoĵ : alternativaj vortoj kaj partikoloĵ, specifa vort-orĉo kaj monotona stilo.

3.5 Kiel konservi kompaktecon

Oni notis jam en la enkonduko ke sekvado de malĉasenco konfliktas kun kompakteco. Malĉasencen oni ofte asocias kun balasto, kiu ŝajnas esti la kontraŭa de kompakteco.

Bonŝance la situacio ne estas tiom grava ĉe Esperanto-PL :

- kelkaj solvoj de ĉasenceco konsistas el antaŭi unu partikolon (prepozicion, ktp.) per du, aŭ aldoni afikson (2.3, 2.6); tiuj solvoj klare ŝanĝas la kompaktecon;
- aliaj solvoj de ĉasenceco tute ne influas la kompaktecon : antaŭi morfemon per alian alternativan morfemon, aŭ ŝanĝi la vort-ordon (2.5, 2.7);
- fine oni havas kategorion kiu unue ŝajnas malutila : la vort-ripetado sugestita en la antaŭa parto (3.4); sed por la tiel nomita pliiga kodado ĝi preskaŭ nenian tuŝas en tiu ĉi okazo.

Pliiga kodado, *incremental coding (An)*, estas tekniko laŭ kiu oni trasendas la kompletan kod-elementon kiu korespondas al leksika morfemo nur kiam la morfemo okazas por la unua foĵo. Ĉiu plua aperigo de la sama morfemo estos pritraktita per la interna interŝanĝo de stako-montriloĵ. La ekstera rezulto estas ke kompleta ripetado de substantivo ne kostos pli ol aludo per anafora elemento. La teknikon de pliiga kodado oni povas apliki kune kun aranĝo de diverslongaj kaj fikslongaj kodado.

Kvankam estas klare ke oni devas cedi iom-da de la kompakteco de Esperanto-PL favorante malĉasencecon, ŝajnas ke el la originale (5,6)

gajnita 54 % de la kompakteco, oni povas reteni la pluparton, kaj rezultos neta gajno de pli-malpli 40 %. Oni bezonas pluan elserĉadon por pruvi tion, inkluzive novajn statistikojn de ofteco.

3.6 La limoj de PL-maldubasenceco

Oni devas elĵeti iluzion ke iam, ie, PL povos esti tute maldubasenca; plue, ĉar la koncepto de dubasenceco havas malfermitan limon en la direkto de polisemio, malklareco kaj malspecifeco. Ankaŭ en tradukado de lingvoj oni devas trakti idiomojn kaj metaforojn kiuj ne estas simple transfereblaj.

En la ricevanto komputilo PL=MALKODANTO funkcias ankaŭ tute alia mekanismo, nome tio de maldubasencigantan vortaro.

Oni ricevas bonpromesajn rezultojn kun tiutipa mekanismo por la angla (15). Tio igas bonajn esperojn ricevi la bezonantan funkciadon por Esperanto PL.

La plibonigo bezonata por tiuj pli subtilaj dubasencaĵoj multe dependas ol la celo lingvo. Kune kun kompakteco, tio estas la kialo por kio restanta kaj fina maldubasenciigo devas okazi en la ricevanta komputilo, post la trasendado de PL.

Referencoj

1. Kelly, I.D.K., 1977, "Esperanto - the Logical Intermediary", working paper;
2. Falla, M.E., "Comments on 'The Case for Machine Translation by Number Language'", informal working paper;
3. Luxemburg, 1977, "Overcoming the Language Barrier", 3rd European Congress on Information Systems and Networks;
4. String, P., "Esperanto kiel objekto kaj instrumento de aŭtomata komputado", in : "La Internacia Lingvo; Sciencaj Aspektoj", D. Blanke (ed.), Kulturligo de GDR, Berlin, 1979;
5. Witkam, A.P.M. and J.J. Hillan, "Resolving language barriers in international videotex communication", International Conference on New Systems and Services in Telecommunications, Liège, 1980;
6. Witkam, A.P.M., "Natural Language Translation in Information Distribution Networks", IFIP/UNESCO computer network conference COMNET '81, Budapest;
7. Vaitilavičius, A., "Esperanto (UEA)", June 1972;
8. Waringhien, G. (ed.), "Plena ilustrita vortaro de Esperanto", SAT, Paris, 2nd ed., 1977;
9. Waringhien, G. and K. Kalocsay, "Plena analiza gramatiko de Esperanto", UEA, Rotterdam, 4th ed., 1980;

10. Butler, M.C., "Esperanto-English Dictionary", 1967;
11. Waringhien, G., "Grand Dictionnaire Esperanto Français", SAT, 1976;
12. Givoje, M., "Esperantonimoj", Pisa, 1979;
13. Laster, E., "Die Verhältniswörter des Esperanto", Saarbrücken, 1968;
14. Wells, J., "Lingvistikaj aspektoj de Esperanto", Rotterdam, 1978;
15. Kelly, D. and Ph. Stone, "Computer recognition of English word senses", North-Holland, Amsterdam, 1975.

Andrjuŝĉenko V.M.

La studo de la komputil-devenaj lingvoj el la
lingvistika vidpunkto.

La ade disvastiĝanta uzado de la komputiloj por la inform-prilaboraj soci-bezonoj manifestas elformiĝon de la nova elemento de nia kulturo, konsiderata kiel la tuto de "la instrumentaro, teknika ekipo, sociaj institucioj, religio, moroj, kutimoj kaj lingvo" [1,5]. La universaleco de la komputiloj, kiel inform-prilaboriloj, estigas antaŭkondiĉojn por la komputiligo de la tuta moderna kulturo, rigardata kiel "la tuto de la ne biologie heredata informaro de la homaro aŭ de iuj pli limigitaj kolektivoj" [1,12]. La kulturo estas ne la inform-deponejo, sed komplika signo- kaj komunik-sistemo, funkcianta kiel uzilo kaj samtempe kreilo de la sociaj informoj. Do, tiurilate ĝi konsistas el la lingvaro konstruata super la origina "natura" lingvo de la kolektivo. Tiuj superkonstruitaj lingvoj, nomataj duaecaj (sekundaraj) imit-sistemoj, krom baziĝi sur la natura lingvo, reproduktas ties ĉefajn strukturajn trajtojn [1, 6-7]. Tio validas pri ia "specialaj" [2,231] religiaj, slangaj, kultaj, profesiaj kaj alaj lingvoj, ankaŭ pri la lingvoj de literaturaj ĝenroj kaj skoloj. La samo validas pri la "sciencaj lingvoj". U.Weinreich [3,171] rimarkigas: "...artefaritaj lingvoj, kreitaj de filozofoj, en multaj esencaj trajtoj de sia logika strukturo koincidas kun la natura lingvo", kaj

poste li citas la eldiron de Morris: "... formalaj lingvoj, esplorataj de la modernaj logiko kaj matematiko, enkorpigas la formalan strukturon de la realaj kaj ankaŭ eblaj lingvoj, ordinare uzataj por paroli pri ĉiutagaĵoj; ĉiudetales ili spegulas la esencajn karakterizojn de la realaj naturaj lingvoj".

Laŭ ke la komputil-uzo iĝas fakto de la kulturo, la komunikado kun komputilo ekzistigas la novan tipon de la komunik-situacio, ĉe kiu la homaj-komputilaj lingvoj ekludas rolon de sociaj dialektoj. Sekve tiuj lingvoj, estante elementoj de la kulturo, heredas ecojn de la naturaj lingvoj. Do ili devas soci-dialekte evolui en intima interefiko kun la natura lingvo, direkte al ĉiam plia proksimo al ĝi [4].

Aliflanke la atento de programistoj ĉiam pli altir-iĝas al la natura lingvo. Ekzistas du ekstreme kontraŭaj opinioj pri la uzado de la natura lingvo por la homa-komputila komunikado. Iuj opinias nereala fari komputilon vere amasa instrumento, sen povigi la ordinaran uzanton komuniki pere de la natura lingvo [5,8]. La aliaj opinias la uzadon de la natura lingvo nereala kaj eĉ malutila [6,29].

La aŭtoro de la artikolo pli solidaras kun la unua el la opinioj, tamen proponas distingi la du esence diversajn situaciojn: la aŭtomatigitan datum-prilaboron en la natura lingvo kaj la veran dialogon kun la komputilo. Koncerne la unuan situacion ni tute konsentas kun la starpunkto de A.P.Jeršov, kiu skribis rimarkindajn vortojn: "... Verŝajne paroli pri la plena (eĉ el la vid-

punkto de la estiĝintaj realaj bezonoj, sed ne la mak-simumismaj deziroj) komputiligo de la produktad-rilatoj estas nereala, sen instrui al la komputilo akcepti kaj kompreni la ofican prozon. Ajna provo laŭordone formaligi la formojn de la oficaĵaj dokumentoj pere de la speciala kaj apriore strikta simboleca lingvo ŝajnas absolute nereala tasko, kaj laŭesence, kaj laŭ la metodo de la fakta plenumado... Iufoje aperas volismaj provoj trudi al la komputil-uzantoj iun kvazaŭnaturan lingvon, laŭigitan al la limigitaj kapabloj de iu fuŝe-rapide konstruita lingvo-prilaborilo. Sed ĝenerale la problemo restas praktike ne tuŝita, kaj ni ekzemple eĉ ne scias: ĉu la ofica prozo estas vere formala aŭ, pli ĝuste, limigita lingvo, al la homo ruze sukcesas enigi eĉ en la reglamentitan kadron de la striktaĵaj oficaĵaj rilatoj la neelĉerpeblon de sia naturo".

Koncerne la duan situacion, nome la dialogon inter homo kaj komputilo, mia opinio uzindas specialaj lingvaj rimedoj (diferencaĵoj de la natura lingvo), kiuj formas socian dialekton por la komputiligita interkomunikado. Tiu dialekto "nature" kreskas el la modernaj homaj-komputilaj komunikiloj: komandaj, direktivaj, labor-regaj kaj programaj lingvoj surbaze de la struktura, sed ne laŭaspekta proksimigo al la natura lingvo (COBOL prezentas ekzemplon de la supraĵa kopiado de la ekstera formo de la natura lingvo, kiu tute ne proksimigas lin al la natura lingvo).

Por firmigi tiun starpunkton, ni devas konsideri la ecojn de la natura lingvo, kiuj ĝin distingas el la aliaj signaroj.

Aliflanke ni devas evidentigi la ecojn de la

homaj-komputilaj komunikiloj, kiuj propras ankaŭ al la naturaj lingvoj. Krom tio necesas trovi la lokon de la "alkomputilaj" lingvoj inter la aliaj semiotikaj sistemoj.

Trajtoj kaj ecoj de la naturaj lingvoj.

Malgraŭ ke ekzistas pli ol 5000 diversaj "naturaj" lingvoj, "estus ne erara la aserto, ke ekzistas nur unusola homa lingvo, en ĉiuj latitudoj, kiu estas laŭ-esence unueca" [2,217]. Tiu vidpunkto de J.Vandries formas bazon de la ĝenerala lingvistiko, de la lingvistika lingvotipologio, kaj de ties direkto, kiu provas serĉi kaj formuli la lingvajn universalajojn, nome la leĝkonformecojn karakterizajn por la lingvo ĝenerale aŭ almenaŭ prezentitajn en la plimulto de la lingvoj [7].

Inter la plej ĝeneralaj universalajoj ni trovas la ecaron, klu difinas la objekton, kutime nomatan "natura lingvo". Verŝajne ne eblas difini tiun objekton alimaniere, ĉar la lingvo tiom trapenetas nian tutan ekzistadon, nian tutan kulturon, ke nenia pervorta difinaro takseblas kiel tute kompleta (vd. [8,10-33]). Ni rimarku, ke la termino "natura lingvo" estos uzata en la senco de "ordinara" lingvo, "lingvo proprasence", "io, ordinare nomata lingvo". Do ni kontraŭigos lin ne al la termino "artefarita lingvo", sed al la nocio "speciala, neordinara" lingvo, same kiel oni kontraŭigas la "komputilan grafikojn" kaj la "komputilan muzikon" al la "ordinaraj" grafiko kaj muziko, sed ne al la "naturaj".

(Komparu la vidpunkton de V.I.Abaev en la artikolo "Pri la termino "natura lingvo" [9], kie li traktas la terminon "natura" kiel "kreita de la naturo" kaj tiamaniere rezervas ĝin por la lingvoj de bestoj.)

Ni opinias, ke ĉiu artefarita lingvo, se ĝi akiris (programist-dire) sufiĉe multe da uzantoj kaj estas intense uzata, povas fariĝi same "natura" kiel ordinara lingvo, kvankam ĝi konservas specialan uzo-sferon, do restas speciala, faka lingvo. Ne necesas ŝvit-pene serĉi ekzemplojn: multaj nunaj naciaj lingvoj estas artekreitaj kaj neordinaraj en certaj komunik-situacioj, tamen oni taksas ilin naturaj, ekzemple la germanan lingvon (Bühnendeutsch). Matureco" de lingvo dependas de ties laŭo al la ĝeneralaj homaj lingvoj.

La plej kompleta resumo de la homlingvaj karakterizoj estas prezentita en la artikolo de Ĉ. Ĥokkett "La problemo de la lingvaj universalajtoj" [10], kiun ni ĉefe sekvas kompletigante per aliaj, speciale menciataj fontoj, ĉar tiu artikolo precipe celas distingi inter la hom- kaj best-lingvoj, do ĝi ne konsideras la por ni interesajn aspektojn de la komparo de la natura lingvo kun la alkomputilaj lingvoj.

Origine kaj prae la natura lingvo havas parolan karakteron kaj uziĝas en la voĉ-aŭda kanalo. Tio determinas la nedaŭrecon de la lingvosignala ekzistotempo, la trean plion (redundancon) kaj fidindecon de la parolo. Multaj komunik-sistemoj ne uzas voĉ-aŭdan kanalon, aŭ ilia "parolo" estas senartikulacia. La artikulacieco, la diserigeblo de la parolo en la diskretajn figurojn

konserviĝas ankaŭ en skribaj lingvoj; normale skribitaj komunikaĵoj en la naturaj lingvoj destiniĝas por prononcado kaj perceptado en la voĉ-aŭda kanalo, do per tio la naturlingvaj skribaj tekstoj diferencas de la skribaj komunikaĵoj en specialaj lingvoj (inform-lingvoj, program-lingvoj, formal-logikaj lingvoj ktp).

Por la natura komunikado origine kaj principe necesas ekzisto de la adresantoj ("parolantoj") kaj adresatoj ("aŭskultantoj"), kiuj efektivigas ĉiudirektan "dissendon" kaj unudirektan ricevadon. La parolantoj kaj la aŭskultantoj alternigas siajn rolojn, sed ilia rilato al la parol-ago estas nesimetria, ĉar la parolanto estas ankaŭ aŭskultanto, sed la malo ne veras. La komunikaĵ-strukturo, same kiel la struktura ecaro de la lingvosistemo, rezultas el la kompromiso inter interesoj de la parolantoj kaj aŭskultantoj [1]. En la animala regno okazblas kaj la unudirekta elsendado, kaj la ricevado sen distingo de la fonto. Dum la homa-komputila komunikado, la informo-fluo kutime estas unudirekta, kaj la programoj ordinare estas "monologoj", kio kaŭzas la dialogeman pensmanieron de la programisto (konsideru la teston de la ligikaj kondiĉoj kiel analogion al la demando).

La naturlingva sistemo ne herediĝas biologie, sed ekposedo pri unu el la sistemoj kapabligas lerni ankaŭ la aliajn, kaj ekposedo pri ĉiu nova sistemo pliigas la lernokapablon pri la sekvaj [13].

La bestlingvoj verŝajne ĉiam estas biologie heredataj. La homo havas ne pli ol ekposedemon por la lingvoj, sed nepre ekposedas iun lingvon kaj nur en certa aĝo. La plen-

valora ekposedo de lingvo en malpli frua a²ĝo jam ne eblas. Krome la manko de la ĝustatempa ekposedo pri iu lingvo malhelpas la mensan kaj socian evoluojn kaj eĉ rezultigas la antaŭtempan morton.

Inter la enhav-kontinuo kaj la parol-kontinuo lokiĝas la sistemo de la diskretaj kaj senchavaj lingvaj signifantaĵoj, konsistanta almenaŭ el tri signo-klasoj: signiloj, formiloj kaj anstataŭaĵoj [3,166].

La diskreteco kaj senchaveco (simbola signeco) de la objektoj estas precipa karakterizo de la homlingvoj kontraŭe al la bestlingvoj. La signiloj simbolas eksterlingvaĵojn (signataĵojn), la anstataŭaĵoj simbolas lingvajn objektojn (referencaĵojn), la formiloj servas por la konstruado de novaj signoj kaj por la kombinado de signoj en esprimojn. La tri samajn tipojn ni trovas en la program-lingvoj kaj en la formal-logikaj lingvoj. Nome la identigiloj (datum-nomoj) estas signiloj kaj anstataŭaĵoj (parametroj), la kleoj ("ŝlosilvortoj") kaj la indicoj estas formiloj.

Rilate al la eksterlingva signataĵo la lingva signo estas arbitra, tamen en la sistemo de la lingvo ĝi estas grandparte motivita. Tiukaze la vorto "arbitra" signifas "ne bilda". En la bestlingvoj la kvanto de la bildaj (ikonecaj) elementoj estas tre granda. En la danco de la abeloj la direkto de la danco kaj ties rapido montras la alcelan direkton kaj la kvanton de la mangaĵo. En la natura lingvo ikonaĵoj havas la funkcion de la bildaj (poeziaj) rimedoj. Tiaj elementoj preskaŭ tute mankas en la formal-logikaj lingvoj. En la program-lingvoj PL/I

kaj COBOL ikonaĵoj prezentiĝas kiel "bildoj" en la atributo PICTURE, kaj en la lingvoj PL/I kaj FORTRAN - kiel elementoj de la format-listo. Sed eĉ ikonaĵoj de la naturaj lingvoj kaj tiuj de la program-lingvoj estae motivitaj ene de la lingvo, nome ili estas sisteme rilatigeblaj al aliaj signoj. Kiel rimarkigas R. Jacobson, "parono-mazio aŭ laŭsenca proksimigo de la fonologie similaj vortoj sendepende de ties etimologia interligo ludas konsiderindan rolon en la vivo de la lingvo" [14,11-12]".

Tiu aŭtoro citas kiel ekzemplon la sekvajn anglajn vortojn: bash frapi, mash - premmiks, smash - frakasi, crash bruege katastrofi, dash - ĵeti, lash - vipi, hash - hak-dispecigi, rash - impeti, bresh - rompi, clash - interpuŝi, trash - rubo, plash - plaŭdi, splash - ŝpruci, flash - trembrili. Ĉi-ekzemple ni vidas, ke la sonkunigo [æf] asociiĝas al la ideo de la "detrua movo", sed ja ne pro ke la movo mem iel koncernas la sonkunigon [æf] sed pro la efiko de la konsiderata signifosistemo. Analogie en la program-lingvoj la "bildoj" (ŝablonoj) ne per si mem bildigas la valorojn de variabloj, sed nur rilate al la konstantaro de la lingvo.

La lingvaj signoj havas semantikan tropecon, ĉar eblas aŭ eventuale eĉ necesas uzi ilin en figura senco. La kapablo de la natura lingvo al la figura (metafora, metonimia) vortuzo estas ĝenerale konata. En la jargono de la programistoj terminiĝis la anglaj metaforoj "dump" rubejo, rubamaso aŭ forfaligi, kaj "debug" - sencimigi, seninsektigi. En la alkomputilaj lingvoj tiuj vortoj gramatikiĝas kaj perdas sian sencan tropecon. Verŝajne en la formal-

logikaj, klasig-indeksaj same kiel en la algoritmaj lingvoj la figura vortuzo estas nebla escepte de la variablo-nomoj, ĉar la ceteraj vortoj estas esence gramatikaj elementoj (formiloj).

La energi-elspezoj por la parol-estigo ne dependas de la graveco de la komunikaĵoj kaj estas malkoncernaj al la komunikeblo. La signataĵoj de la eksterlingva mondo povas kaj enesti la komunik-situacion, kaj foresti en ĝi. La sencekzisto por la komunikaĵoj en la natura lingvo estas sendependa de la formala logikeco. La intenso de la best-komuniko rilatigas sin al la fiziologiaj statoj (ju pli egiĝas timo, malsato, seksa instinkto (libido), des pli egiĝas "bleko", "voko" ktp). Malamiko, mangaĵo, ino devas ekzisti en la komunik-situacio por iĝi ties objekto. La bestosignaloj ne povas esti "erarigaj". Evidente, ke en la nebiologiaj komunik-sistemoj (la natura lingvo, ĉiuj specialaj lingvoj) tute ne necesas kriegi por esprimi eĉ la plej intensajn dezirojn (se tiuj estas iel esprimeblaj !), krome la objekto de la komunikaĵo povas foresti en la komunik-situacio aŭ tute malekzisti, kaj la komunikaĵo povas esti kaj vera, kaj malvera, se nur ĝi estas gramatike korekta. Nun oni ja scias, ke la komputiloj tute ne ĉiam "diras veron", kaj ne ĉiam pri tio kulpas la programisto.

La sistemo de la natura lingvo estas malfermita, ĉar en ĝi senfine povas aperi kaj novaj signiloj kaj novaj signifoj, kiuj estas uzataj kaj esprimataj enkadre de la gramatika sistemo. La gramatikaj sistemoj dum paso de tempo modifiĝas. Novaj komunikaĵoj libere kreigas analogie

al la malnovaj aŭ per la transformado de tiuj. La sistemoj de la best-komunikado, verŝajne estas ne tute fermitaj, sed pri ilia variemo ni scias tro malmulte. Laŭ disvastigita opinio la formal-logikaj sistemoj kaj la programlingvoj estas tute fermitaj. Sed tio ne estas vera, ĉar la sistemo de variabloj estas malfermita kaj signantaĵe kaj signataĵe. La difinoj de novaj "terminoj" eblas en multaj lingvoj (ekzemple en ALGOL-68 kaj PASCAL). Ankaŭ la gramatikaj sistemoj de la program-lingvoj iom post iom varias, kion pruvas multenombraj "dialektoj" de FORTRAN, ALGOL-60, PL/I kaj la familioj de la "al-iu-similaj" lingvoj.

La strukturo de la natura lingvo semiotike estas (almenaŭ) dueca, ĉar en ĝi interefikas du subsistemoj: la kinematika subsistemo de la parol-movaj strukturaj unuoj (parol-sonoj, silaboj) kaj la plerematika subsistemo de la strukturaj unuoj de la enlingva sencoplano (de la signiferoj semoj). La interago de la subsistemoj konsistas el tio, ke la infinita nombro de la signifero-kombinoj estas kodata nur per la limigita nombro de la fonologiaj senco-distingopovaj unuoj. Ĉ.Ĥokkett skribas ĉi-rilate: "Nek unu besta komunik-sistemo, el la konataj al la aŭtoro, montras ioman duecon de la struktura organizeco" [10,57]. La program-lingvoj havas duecan organizon, ĉar en iliaj priskriboj la partoj difinantaj la alfabeton, apartigilojn, datum-nomojn, konstantojn kaj aliajn "pleremojn", estas sendependaj de la partoj, priskribantaj la konstruojn (esprimojn, frazojn ktp).

Ĉi-kaze la "dueco" signifas ekziston de "almenaŭ du subsistemoj". Kompreneble, povas ekzisti pli ol du subsistemoj.

Ĉiu el la interefikaj semiotikaj subsistemoj estas el la gramatika vidpunkto paradigme kaj sintakse organizita. La paradigma plano formiĝas el la sistemo de la gramatikaj kategorioj nepre esprimendaj en la parolo (ekzemple, en la rusa lingvo tio estas la genro, nombro, kazo de la substantivo, la tempo, aspekto, modo, voĉo de la verbo, la akordigo de la substantivoj kaj adjektivoj, la akordigo de la verbo kaj subjekto en la preterito laŭ genro ktp). Sintagmoplane funkcias du ĉefaj signo-kombin-mechanismoj, nome kunigo kaj enigo [3,194]. La nombro de kunigoj kaj enigoj estas limigita statistike, dum ke la kunig- kaj enig-reguloj estas limigitaj strukture (la proekcieblo de la strukturo sur la vektoron de la signantaĵoj). La longecoj de la signovicoj, formitaj de la projekcio de la strukturo sur la vektoron de la signantaĵoj, estas statistike akordigitaj laŭ la niveloj de la enmeto.

Pri la paradigmaro de la bestlingvoj oni scias nenion. En la formal-logikaj lingvoj la paradigma plano estas maltre evoluigita. Distingeblas nur nemultaj elemente konstruitaj paradigmoj (ekzemple, interrilatoj kaj opozicioj de la nocioj "pli ol", "ne malpli ol", "malpli ol", "ne pli ol", kvantoroj V, E evidentigataj en la rilatigo de la konfirmaj kaj malkonfirmaj propozicioj ktp). En la alkomputilaj lingvoj la paradigmaro estas nekompareble pli riĉa. Unue, en la evoluintaj lingvoj

kiel PL/I oni havas riĉan kaj sufiĉe komplikan paradigmon de la datum-tipoj, kiu realigas la tutan sistemon de opozicioj:

```
((DEC:BIT) : (FIXED:FLOAT)) : COMPLEX:PICTURE) :
(CHAR:BIT) : (LABEL:EVtaT:TASK: (POINTER:OFFSET) :
AREA) )
```

La kompletan paradigmon de la duumaj opozicioj formas la kompar-operacioj:

```
(( (LE:LT) : (NL:EQ) ) : (GE:GT) : (NG:NE) ) )
```

kio kompareblas, ekzemple, al la sistemo de la rusaj kazoj [15,182]:

```
(( (nominativo:akuzativo) : (genitivo-1:genitivo-2) ) :
((instrumentalo:dativo) : (prepozitivo-1: prepozitivo-
2) ) ) .
```

Ordinare, ju pli evoluigita estas la paradigmaro, des pli limigaj estas la signo-kombin-reguloj (sintagmaro).

En la naturaj lingvoj krom la natura senca kombineblo (komparu "rondforma ortangula tablo") observendas ankaŭ la gramatika kombineblo (komparu la faman ruslingvan frazon de L.V.ŜĈerba: "Glokaja kuzdra ŝteko budlanula bokra i kurdjaĉit bokrjonka" kiu estas gramatike "komprenebla" por la ruslingvanoj kaj estas "tradukebla" Esperanten proksimume kiel "Gloka kuzdrino ŝteke ekbudlis la virbokron kaj kurdadas la bokridon"). La kombin-reguloj de la natura lingvo havas la strukturon, kiu ebligas al la komunikanto la komprenon eĉ malgraŭ la sufiĉe granda rompo de la reguloj.

"... La kunigo de la nekombineblaj signaĵoj estas kutima artifiko uzebla fare de la skribanto/parolanto por rimarkigi al la leganto/aŭskultanto novajn, neantaŭviditajn

de la komunik-kodo rilatojn inter la signaĵoj. Laŭŝajne la improvizitaj kombinaĵoj de la nekunigeblaj signaĵoj eblas en ajna kulturo" [3,195]. La rompo de la gramatiko povas esti akceptita de la komunikato kiel la gramatike korekta sed sensencigita eldiro. La formal-logikaj kaj alkomputilaj lingvoj heredas ĉi tiujn ecojn de la naturaj lingvoj, sed tamen ili metas tro severajn, el la vidpunkto de la natura lingvo, limojn por la gramatika ĝusteco de la frazoj, samtempe ebligante konstruon de la arbitre longaj esprimoj, same kiel de la arbitre profundaj enmetoj. "Por ĉiuj (naturaj) lingvoj la signo-kombino fariĝas aŭ kunmete aŭ enmete kaj ambaŭ manieroj uziĝas en la kernaj propozicioj. La transformoj ne rezultigas iujn novajn tipojn de la signo-kombinado. La nombro de la kombin-niveleoj teorie estas senlima, tamen tre malofte oni trovas pli ol tri-nivelan kunigon kaj kvar-nivelan enigon" [3,194]. La fenomeno estas konata sub la nomo "hipotezo de Ingve" [16]. Ni formulas ĝin per la terminoj de la analizo pere de stako (magazeno). Ni havu ekzemple la frazon: "Hieraŭ estis ege varma vespero". Ni imagu jenan analiz-algoritmon. La vortoj de la frazo enstakiĝas demaldekstre-aldekstre. Se la vorto enstakigata formas sintaksan unuon (sintagme aŭ senpere kombineblas) kun la tuje antaŭa vorto aŭ sintaksunuo, tiam elformiĝas la nova, pli granda unuo, kiu ekanstataŭas la ĵusan, kaj la "rigardo-truo" de la magazeno relokiĝas malsupren. Sed se la nova unuo ne formeblas, la truo relokiĝas supren, kaj en ĝin metiĝas la nova vorto. Do unuapaŝe ni enstakigas la vorton "hieraŭ" kaj movas la truon supren, poste ni enstakigas la vorton "estis". Ĉar "estis" ne estas senc-

kompleta, ni ne povas kombini ĝin en pli grandan unuon kun la vorto "hieraŭ", do ni suprenmovas la truon kaj ekkonsideras la vorton "ege". Ĝi ne formas unuon kun la vorto "estis", do ni ree suprenmovas la truon kaj ricevas en ĝi la vorton "varma". La vorto "varma" unuigeblas kun la vorto "ege", pro tio ni formas novan unuon "ege+varma" kaj anstataŭigas per ĝi la vorton "ege". Poste ni malsuprenmovas la truon kaj metas en ĝin la vorton "vespero", kiu kombineblas kun la unuo "ege+varma". Ni formas la novan unuon "ege+varma+vespero", per kiu ni anstataŭigas la unuon "ege+varma", kaj malsuprenmovas la truon. Nun en la truo troviĝas la unuo "ege+varma+vespero", kiu formas unuon kun la vorto "estis". Do ni konstruas la novan unuon "estis+ege+varma+vespero", per kiu ni anstataŭigas la vorton "estis", kaj malsuprenmovas la truon. Finfine ni ricevas la kompletan frazon kiel la unuon. Ĝi havas jenan strukturon:

(Hieraŭ (estis ((ege varma) vespero)))

1 2 3 4,3 4,3,2,1

La ciferoj montras la nombron de la okupitaj magazenĉeloj post la apero de la respektiva vorto kaj la plenumo de la reduktoperacio. La "hipotezo de Ingve" asertas, ke plej ofte la kiomo de la samtempe okupitaj magazenĉeloj dum la sintaksa analizo ne superas 7 ± 2 (kio akordiĝas kun la eksperimente trovita amplekso de la operacia memoro de la homo [17]) kaj ke la lingvoj havas specialajn rimedojn por retenigi la ampleksojn kaj la profundecon de la frazo en inter la indikitaj limoj. (Necesas rimarkigi, ke nia priskrib-maniero diferencas de tiu de V. Ingve. Interalie, laŭ nia formulo la malprogresaj konstruoj postulas plej malmulte da memoro,

ekzemple: "Tre kontraste projekciitaj sur la ekranon bildoj estas malbonaj", sed laŭ V. Ingve tiaj estas ĝuste la pro-gresaj, ekzemple "Hieraŭ estis vespero, varma ege". Ver-ŝajne la diferencaj postuloj pri la direkto de la disdubiĝo prezentas la parolantoj kaj la aŭskultantoj: el la vidpunkto de aŭskultanto (analizo) pli avantaĝas la malprogresaj strukturoj, sed el la vidpunkto de parolanto (sintezo) pli avantaĝas la progresaj. En la naturaj lingvoj maloftas kaj pure malprogresaj kaj pure progresaj strukturoj, ĉar la kompromiso inter parolantoj kaj aŭskultantoj ekvilibrigas la direktojn de la disdubiĝo.

La profundec-hipotezo klarigas la du disvastigitajn fenomenojn de la naturaj lingvoj, nome la projekciecon de la fraz-strukturoj kaj la tre oftajn okazojn de la rompo de tiu projekcieco. Ja por la sukcesa funkciado de la stak-imitata analiz-meĥanismo laŭ la senperaj unuoj necesas ke la unuo-strukturo formu la regulan krampo-strukturon. Aliflanke, se la stako superpleniĝas, necesas rompi la projekciecon, kio eble klarigas la multajn (sed regulajn) rompojn de la kadra konstruo en la germana lingvo. Komparu:

1. (Ich (bin (ihm (gestern ((ganz unerwartet) ((in (der Leipziger Strasse)) begegnet))))))
2. ((Ich (bin (ihm (gestern ((ganz unerwartet) begegnet)))) (in (der Leipziger Strasse))).

"Mi renkontis lin hieraŭ tute neatendite sur la Lajpciga strato".

Unu el la rimedoj reteni la frazon en inter limoj de la permesebla profundeco estas la apoziciigo (apartigado) de la disvastigitaj frazeroj. Se la komplementoj aŭ adjek-

toj (adverbialoj) "eksterkrampiĝas" la entuta profundeco de la frazo malpliĝas je la konto de la "plilarĝigo de la arbo", kiel en la lasta ekzemplo. Se apartiĝas la subjekto (ekzemple "Mia malnova kaj bona amiko, li ĉiam apogis min en malfeliĉo kaj ĝojis pri miaj sukcesoj"), tiam rompiĝas la projekcieblo de la arbo de la senperaj unuoj. Tiu fenomeno analogias al la uzado de la senkondiĉaj saltinstrukcioj en la program-lingvoj (en la ĵusa ekzemplo "li" ekvivalentas al "GO TO" al la teksto "mia malnova kaj bona amiko").

Al la metodo de la reduktado de la signantaĵoj en la pli altgradajn unuojn senpere rilatas la sekva fenomeno malkovrita de ni [18] surbaze de la materialo de la 21 lingvoj: nome, la mezlongecoj de la lingvaj unuoj apartenaj al la sinsekvaj niveloj (la mezlongeco de la morfemo, vorto, ritma grupo kaj frazo) formas geometrian vicon kun la denominatoro $a_0 = 2.718\dots$ kaj

$$\frac{|\bar{K}_i - \bar{a}_0^i|}{s/\sqrt{n}} \leq t_{0.05}$$

Kie \bar{K}_i estas la mezlongeco de la lingva unuo de la i -a nivelo, s estas la korektita meza kvadrata devio, $t_{0.05}$ estas la devio de la mezvaloro de la samplo ĉe la 5%-a nivelo de la signifeco, a_0 estas mezo en la ĝenerala tuto.

Surbaze de tiu reguleco estis proponita hipotezo, ke la redukto-procedo por la lingvaj komunikaĵoj konsistas el la sinsekva anstataŭigo de la pluraj kod-unuoj de la malalta nivelo per nur unu unuo de la senpere pli alta nivelo.

Verŝajne la projekcieco de la esprim-strukturoj kaj la retenado de la profundeco kaj de la nombro de la konsistigaj unuoj en ĉiuj niveloj en inter certaj limoj karakterizas ĝeneralan econ de la natur-lingvaj tekstoj, kiuj respondas al la psikofiziologiaj kaj mnestikaj karakterizoj de la homa organismo.

La formal-gramatikaj rimedoj de la naturaj lingvoj, reguligantaj la redukt-procedon de la signantaĵoj en la pli altnivelajn unuojn, povas tre varii. Depende de la karaktero de la redukt-reguloj kaj redukt-operacioj la lingvoj klasigeblas je lingvistikajn (morfolgiajn) tipojn. Ekzistas tia sinsekvo de la tipoj, ke ĉiu tipo rigardeblas gramatika metalingvo por ĉiuj sekvantaj. La amorfa lingvo [19] estas gramatika metalingvo por ĉiuj lingvoj de ĉiuj tipoj.

Por vidigi tion, ni uzos la notadon, analogian al tiu en la Normala Formo de Backus.

La lingvoj havas la kvar ĉefajn gramatikajn klasojn de elementoj, kiuj estas

Radikaj: R - puraj radikoj (ekzemple ruse "kino")

B - bazoj (ekzemple "stoja-" en la rusa vorto "stojatj")

Helpaj: A - afiksoj (prefiksoj, sufiksoj, konfiksoj, infiksoj, ekzemple en la rusa vorto

"pre-kras-n-ejŝ-ego" la afiksoj estas

elementoj "pre-", "-n-", "-ejŝ-")

A*- fleksioj ("-ego" en la vorto "prekrasnejŝego")

P - partikuloj (ekzemple la rusaj "ne", "li", prepozicioj k.a.)

La puraj radikoj kutime ne kombiniĝas kun la afiksoj, sed nur kun la bazoj. La radikoj kombiniĝas kun la radikoj, bazoj kaj partikuloj. Ni rimarku, ke la radikoj kaj bazoj analogias al la identigiloj (datum-nomoj) de la program-lingvoj, la afiksoj al la indicoj de la tabeloj, al la priskribiloj de la datum-precizo kaj datum-longo (ekzemple en PL/I); la partikuloj - al la "ŝlosil-vortoj" (kleoj, hieroglifoj) de la program-lingvoj.

La esprimo, kiu enhavas sinsekvon de la radikoj kaj helpaj elementoj, povas esti prezentita kiel "tipa vorto", ripetita tiomfoje kiom da signifaj elementoj (signiloj, identigiloj) troviĝas en la esprimo, se konsideri ties eblajn kombinojn kun la helpaj elementoj.

Ekzemple la ruslingva vort-kombinaĵo "prekrasnejŝij sad bolsogo starinnogo doma" (la belega ĝardeno de la granda antikva domo) prezentebblas jene

$$\{ [A] R \{A\} [A^*] \}_5$$

kie la rektaj krampoj entenas la elementojn aperigeblajn nul- aŭ unu-foje, la figuraj krampoj entenas la elementojn aperigeblajn kiom-ajn-foje, t.e. nul-, unu-, aŭ plur-foje (la indico 5 ĉe la plej ekstera ferma figura krampo indikas la nombron da vortoj en la vort-kombinaĵo). Ĉi tie estas prezentita en la ĝenerala formo tipa plej eta ero de tiu rusa vort-kombinaĵo. Efektive, la element-konsisto en la vortoj de la vort-kombinaĵo estas jena:

pre	-	kras	-	n	-	ejŝ	-	ij
A		R		A		A		A*
sad								
R								
bols - ogo								
R A*								

$$\begin{array}{cccc}
 \text{star} & - & \text{in} & - & \text{n} & - & \text{ogo} \\
 \text{R} & & \text{A} & & \text{A} & & \text{A}^* \\
 \\
 \text{dom} & - & \text{a} \\
 \\
 \text{R} & & \text{A}^* \\
 \hline
 \{[\text{A}] \text{R} \{\text{A}\} [\text{A}^*]\}_5
 \end{array}$$

Ni traktu la fleksiojn "kunfanditaj" afiksoj esprimantaj per unu formo kelkajn gramatikajn signifojn, kaj ni traktu afiksoj la formojn esprimantajn unusolan signifon. Plue, la bazokreaĵajn afiksojn ("-n-" en la vorto "prekrasnejŝij", "-in-" en la vorto "starinnij") ni ne traktu afiksoj, sed ties kombinaĵon kun la radikoj ni rigardu bazo. Tiam ĝeneral-forme la esprimo en la fleksia lingvo (la rusa, la latina ktp) prezenteblos jene:

$$\{[\text{P}] \{\text{A}\} \text{B} \{\text{A}\} [\text{A}^*] [\text{P}]\}.$$

La aglutina lingvo estas pli simpla ol la fleksia, ĉar ĝi ne havas fleksiojn, tial la aglutin-lingva esprimo prezenteblos jene:

$$\{[\text{P}] \{\text{A}\} \text{B} \{\text{A}\} [\text{P}]\} .$$

La enkorpiga lingvo estas pli simpla ol la aglutina, ĉar ĝi permesas "enmeton" de la radikoj antaŭ la bazo, tial ni povas malaperigi la plej eksterajn krampojn kaj prezenti la enkorpig-lingvan esprimon jene:

$$[[\text{P}] \{\text{R}\} \text{B} \{\text{A}\} [\text{P}]]$$

Kaj fine, la amorfa lingvo estas la plej simpla, ĉar en ĝi mankas kaj la bazoj kaj la afiksoj. La amorf-lingva esprimo prezenteblos jene:

$$[\text{P}] \text{R} \{[\text{P}] \text{R}\}$$

Ĉar la bazo prezenteblos kiel radikoj+afikso, la fleksio - kiel partikulo (la diferenco inter ili estas tia, ke

la afikso garnas la vorton, sed la partikulo garnas kaj la vorton, kaj la vort-kombinon). Ni povas rigardi la fleksilingvan gramatikan strukturon prezentebla enkadre de la aglutinlingva gramatika strukturo (alivorte, la aglutina lingvo estas gramatika metalingvo por la fleksia), analogie la gramatika strukturo de la fleksia kaj aglutina lingvoj prezentebblas enkadre de la gramatika strukturo de la enkorpigaj lingvoj kaj, fine, la gramatikaj strukturoj de la tri supre-menciitaj lingvotipoj estas prezentebblaj enkadre de la gramatika strukturo de la amorfa lingvo. Tial la amorfa lingvo rigardeblas gramatika metalingvo por ĉiuj tri lingvotipoj.

La naturaj lingvoj evoluas tiel, ke la amorfaj, gramatikiginte kelkajn partikulojn en afiksojn, kunfandinte tiujn kun radikoj kaj fiksinte kelkajn partikulojn kiel vorto-finformilojn, igas la enkorpigaj. La enkorpigaj, finforminte aparte ĉiun radikon kaj iginte ĝin bazo, iĝas la aglutin-tipaj. La aglutinaj, kunfandinte la afiksojn en fleksiojn, iĝas la fleksiaj. Eblas ankaŭ la inversa transformilo de la fleksiaj al la amorfaj. Kompreneble ĉi tie estas priskribata ideala lingvo-tipa skemo, sed puraj tipoj ne troveblas en la lingvo, ĉar en ĉiu lingvo oni povas trovi divers-tipajn konstruojn. La lingvoj estas klasigataj laŭ la ĉefa super-reganta tendenco. Tiel la fleksiaj lingvoj estas klasigataj al la fleksiaj malgraŭ ke en tiuj povas aperi kaj la aglutinaj kaj la enkorpigaj konstruoj. La lingvoj, havantaj la regulan finformadon de la interrilatantaj vortoj per la interakordiĝantaj afiksoj, estas taksataj kiel la aglutinaj (kondiĉe ke en tiuj mankas fleksioj, sed malgraŭ

ke en tiuj povas aperi la enkorpigaj konstruoj). La lingvoj "preferantaj" kunigi la radikojn en la tutaĵon kaj finformi tiun afikse aŭ partikule estas taksataj la enkorpigaj, kvankam en tiuj renkonteblas ankaŭ aglutinaj konstruoj. La senafiksaj lingvoj kiuj havas vortojn, funkci-kapablajn jen kiel la plensignifaj jen kiel la helpaj (partikuloj), estas taksataj la amorfaj. Ofte la tipigo de iu lingvo estas diskutinda. La lingvotipologio fakte utilas ne por la propra-senca lingvo-klasigo, sed por la reprezento de la eblaj lingvostrukturoj.

En la alkomputilaj lingvoj ni trovas analogiajn inter-rilatojn. La lingvo SIMPOLIZ-64 [20] tipigeblas amorfa, ĉar ĝi ne havas afiksojn, kaj la partikuloj finformas (determinas la sintaksan signifon) de la unu- aŭ plur-radika sinsekvo. Ekzemple:

```
)STRING( ARRAY 10 REAL SREAD
)X(
21
RREAD
)Y( ARRAY 21 REAL
21
RREAD
)71(
1
=
...
```

En tiu programo la puraj radikoj (variablo-identigiloj) aperas markite per inversaj parentezoj, la propraj nomoj (konstantoj) estas prezentitaj per entjeroj, kaj la partikuloj (deklaroj kaj instrukcioj) estas substrekitaj. Ankaŭ la operacisignoj (en la ekzemplo tio estas "=") klasiĝas

partikuloj.

Ĉiuj komputilaj kaj asemblilaj lingvoj strukture analogias al la enkorpigaj lingvoj, ĉar la partikuloj havas fiksitajn lokojn kaj rilatas al unu, du aŭ tri radikoj aŭ bazoj. Krom tio ekzistas afiksoj (delokigoj).

La program-lingvoj de la tipo de ALGOL-60, FORTRAN, PL/I, COBOL ktp lokiĝas inter la enkorpigaj kaj aglutinaj lingvoj.

La SIMPOLIZ-64-tipa lingvo evidente rigardeblas gramatika metalingvo por ĉiuj algoritmaj lingvoj, ĉar enkadre de lia strukturo prezentblas ajna-lingvaj programoj. La lingvo SIMPOLIZ-64 estis intence ellaborita kiel pont-lingvo (aŭ lingvistik-dire "interlingvo") por la tradukilo el ALGOL-60. Notindas, ke la pont-lingvon (entradukilan reprezenton) oni proponis ellabori ankaŭ amorf-struktura [19,132]. La lingvo ALMO [21] ellaborite kaj kiel programad-lingvo kaj kiel pont-lingvo disponigas pli ampleksan strukturaron: ek de la amorfaj (en la aritmetikaj esprimoj) ĝis la aglutinaj (operaci-specifoj, tablo-indicoj) kaj tiusence ĝi estas pli komplika ol SIMPOLIZ-64 do malpli facile realigebla.

Aŭstataŭiginte en la natur-lingva teksto la signilojn per la identigiloj kaj rigardante la afiksojn kaj partikulojn program-lingvaj operacioj, instrukcioj, funkcioj ni povas vidi en la teksto la "koditan" "programon" de ĝia kompreno. Do, la lingvistikaj strukturoj konsidereblas komputaj strukturoj nur etparte realigitaj program-lingve.

La ĵusa analizo montras, ke la program-lingvoj imitas multajn naturlingvajn ecojn kaj proksimas al la naturaj lingvoj laŭ la realigitaj strukturoj. Sed tiu proksimeco estas ne nur struktura, ĉar la program-lingvoj plenumas

preskaŭ ĉiujn semiotikajn funkciojn karakterizajn por la naturaj lingvoj.

La semiotikaj funkcioj de la signaroj

La ĵus menciitaj ecoj de la naturaj lingvoj rekte koncernas al la ĉefobjektoj de la komunika situacio: parolanto, aŭskultanto, teksto (komunikaĵo), enhavo (de la komunikaĵo), lingvosistemo (kodo) kaj komunik-kanalo [22]. Oni opinias, ke la parolanto estas aktiva elemento, estiganto de la komunik-situacio, kies sinteno pri la esprimata en la komunikaĵo rilato al la aliaj objektoj de la komunik-situacio nomiĝas semiotika (lingvistika) funkcio. Estas distingataj jenaj semiotikaj funkcioj:

La metalingvistika, kiu estas sinteno al la esprimado de sia rilato al la sistemo de la uzata lingvo.

La kognitiva, kiu estas sinteno al la esprimado de la pensa enhavo.

La simbola, kiu estas sinteno por la reprezento en la parolo de la realaĵoj.

La reprezento-noma, kiu estas sinteno por la nomado de la realaĵoj.

La bilda (poezia), kiu estas sinteno por la senpera, enkomunikaĵa respegulado de la signifaj ecoj de la objektoj de la enhavo.

La ĝeneral-komunika, kiu estas sinteno por la ligeco (kohero) de la komunikaĵo.

La specialaj komunikaj funkcioj (fatika, konatiga, emotiva ks), kiuj estas sinteno por la estigo kaj

pluigo de la kontakto, agordado de la adresato, esprimado de la emocia stato de la parolanto ka.

La sistemo kapablas plenumi iujn funkciajn depende de la posedo pri la respektivaj rimedoj, kiuj estas signoj specialigitaj por la plenumo de iuj el la ĵus menciitaj funkcioj. Ekzemple, la metalingvistika funkcio efektivigeblas fare de la sistemo nur se ĝi posedas rimedojn por la sinpriskribo. Ĉar la natura lingvo kapablas je la priskribo de ĉiaj signaroj, ĝi laŭdifine posedas rimedojn por la sinpriskribo. En ajna program-lingvo eblas priskribi (per verki analizan programon) tiun program-lingvon.

Ankaŭ la gesto-lingvoj kaj la skrib-sistemoj disponas pri la por-sinpriskribaj rimedoj. La formal-logikaj lingvoj ordinare ne havas tiajn rimedojn, sed por kelkaj el ili verŝajne eblas ĝeneraligo ebliganta la sinpriskribon. Por tio necesas fiksi la predikaton "SIGNIFAS". La aliaj troveblaj en la tabelo signaroj ne disponas pri la metalingvistikaj rimedoj.

La kognitiva funkto plenumblas fare de la signaro se tiu havas rimedojn por esprimi predikatojn (ecojn, kvalitojn), logikajn rilatojn kaj kvantorojn. Evidente, ke la kodoj (ekzemple la Morse-aboco) kaj eble la best-lingvoj ne disponas pri tiaj rimedoj. La signalaroj (ekzemple la voj-signoj) kaj la klasig-indeksaj inform-retrovaj lingvoj nur limigite disponas pri tiaĵoj.

La simbola funkcio laŭdifine propras al ĉiuj signaroj, ĝi estas la sola plenumata de la kodoj.

La reprezenta-noma funkcio plenumblas de la signaro, se en tiu distingeblas signiloj-nomoj kiel io diferenca de la formiloj kaj anstataŭaĵoj. Tiu funkcio estas karakteriza por ĉiuj signaroj escepte de la kodoj kaj versajne la best-lingvoj.

La bilda funkcio propras al la signaro, se tiu havas bildigilojn, kiuj estas gestaj, sonaj, grafikaj rimedoj por bildigi signataĵon (notaĵon). La klasig-indeksaj lingvoj ne havas tiajn rimedojn, kaj estas malklare, ĉu ili ekzistas en la best-lingvoj. La program-lingvoj kaj la formal-logikaj lingvoj disponas pri kelka kiomo de tiaj rimedoj (en la program-lingvoj tio estas ŝablonoj, bildoj, elementoj de la format-listo; en la formal-logikaj lingvoj tio estas divers-tipaj sagetoj, anguloj ktp).

La plej grava funkcio estas la ĝeneral-komunika. Ties esprimiloj en la naturaj lingvoj estas la gramatikaj kategorioj de persono, tempo, modo, la personaj montraĵaj pronomoj kaj pronomaj adverboj, unuvorte la rimedoj servantaj por fiksi la en la konkreta diro esprimatajn rilatojn inter la parolanto kaj aŭskultanto kaj ties spac-tempan lokilon [23,139]. Pri tiaj rimedoj disponas nek la formal-logikaj, nek la klasig-indeksaj lingvoj, kaj nur etgrade la signalaroj kaj alkomputilaj lingvoj (parametroj, indikiloj, eventoj ka).

Kaj fine, la specialaj komunik-funkcioj esprimiĝas per la apelativoj (vokiloj), interjekcioj, kaj la esprimoj kiel "Ĉu vi aŭdas?", "Ĉu klare?" ktp, kiuj destiniĝas por la estigo kaj pluigo de la kontakto, kaj por "agordado" de la

kunparolanto. Simile al la ĝeneral-komunika funkcio, tiuj funkcioj ne propras al la klasig-indeksaj kaj formal-logikaj lingvoj kaj nur ete propras al la signalaroj kaj la alkomputilaj lingvoj. (Ekzemple, la fiksado de parametroj por la tradukilo analogias al la konatiga funkcio, la enkonstruitaj instrukcioj por ĝustigo ("seninsektigon, senmisigo) de la programo plenumas la konatigan kaj fatikan funkcion ktp.).

La interkomparo de diversaj signaroj konkludigas, ke la alkomputilaj lingvoj pli intime ol la aliaj signaroj ligiĝas kun la natura lingvo kaj aliaj sistemoj proksimaj al ĝi, nome tiuj de gestoj, belarto (ekzemple danc-arto), skrib-sistemoj.

La analogio al la natur-lingvaj rimedoj en la program-lingvoj.

En la program-lingvoj ni renkontas multajn konstruojn analogiajn al la natur-lingvaj. Tiel la konstantoj analogias al la personaj nomoj, la enkonstruitaj funkcioj - al la vortoderivaj kategorioj kaj nesintaksaj kazoj de la naturaj lingvoj, la tabel-dimensioj - al la kategorio de nombro, la parametroj, variabloj de la etikeda tipo, indikiloj, evento-nomoj al la pronomoj kaj nekvalitaj adverboj, la apartigiloj al la partikuloj, la operacisignoj - al la sintagmaj rilatoj, la bildoj kaj formatoj - al la ikonecaj signoj.

La natura lingvo estas tiel aranĝita, ke inter interesoj de parolantoj kaj aŭskultantoj estiĝas kompromiso [12], kiu rezultigas ellaboron de specifaj konstruoj, kies analogioj renkonteblas ankaŭ en la program-lingvoj. Entute la program-

lingvoj evoluas direkte de la "poraŭskultaj" lingvoj al la "porparolaj". Tamen, la evoluo de la specifaj ecoj de la "porparolaj" lingvoj necesigas evoluigon de la korelativaj ecoj de la "poraŭskultaj" lingvoj. Tiel la plimultigo de la datum-tipoj korelativas kun la plimultiĝo de la enkonstruitaj funkcioj (analogio al la vortoderivaj funkcioj kaj kazoj), la plimultiĝo de la elementaj operacioj (analogio al la sintagmaj rilatoj) korelativas kun la plikomplikilo de la rango- kaj uzad-reguloj de la datum-tipoj (analogio al la sintagma akordigo). Por la program-lingvoj validas kelkaj universalaj formolitaj analogie al la natur-lingvaj [19,182-222]. Ekzemple por la naturaj lingvoj validas jena regulo (universalajo): la nombro de la laŭgenraj diferencoj en la singularo ne superas la nombron de tiuj en la pluralo. Por la program-lingvoj eblas formuli analogian universalajon: la nombro de la datum-tipaj diferencoj por la skalaraj variabloj ne malsuperas tiun por la indic-havaj. El tiu vidpunkto la program-lingvoj estas ankoraŭ nesufiĉe esploritaj. Supozeble la evoluigado de la naturlingvaj ecoj en la alkomputilaj lingvoj kaj laŭigo de tiuj al la ĝeneralaj leĝoj propraj al la naturaj lingvoj, faros el la alkomputilaj lingvoj iom pli oportunan kaj naturan komunikilon inter homo kaj komputilo, ol ili estas nun. Proponeblas kelkaj, ŝajne jam realaj, vojoj por proksimigo de la alkomputilaj lingvoj al la naturaj. Nome

- ebligi programadon sen videblaj (eksplicitaj) egaligoj (asignoj) [4]
- malnecesigi la eksplicitajn rego-saltojn [4]
- ebligi vastigon de lingvo per novaj difinoj kaj

enkonduko de la metalingva nivelo [4]

- permesi la polisemion (plursencecon), sinonimion (samsencecon), homonimion (samnomecon) kaj la semantikan tropecon (sengramatikigo de la partikuloj)
- evoluigi la aparaton de la prisilento (analogio al la elipso)
- morfologiigi la strukturon de la identigiloj (per monstro de la datum-tipo en la strukturo de la identigilo ktp)
- enkonduki uzon de la nedifina datum-tipo kaj de la aparato de ĝia dinamika difinado
- enigi en la lingvojn la aparaton de la propozicia kalkulo kaj predikatoj (evoluiĝo de la kognitivaj rimedoj)
- evoluigi la rimedojn por la estigo kaj pluigo de la kontakto (analogio al la konatiga kaj fatika special-komunikaj funkcioj)
- pliriĉigo de la neindicaĵaj rimedoj por la elektado el la datum-agregatoj (analogio al la natur-lingvaj difinoj)
- evoluigi la kaŭzajn-sekvajn konstruojn
- evoluigi la rimedojn de la modaleco de la plenum-tempo (analogie al la rimedoj de la kompilad-tempo)
- evoluigi la rimedojn por ŝanĝo de la dialektoj (interdialekta komuto)

(La lastaj tri ebloj vastigas la ĝeneral-komunikan rimedaron.)

Laŭfunkcia klasigo de la signaroj.

sistemoj	kodoj	klasig- -indek- saj kaj inform- retrovaj lingvoj	signal- aroj	best- lingvoj	gesto- kaj belart- lingvoj	skriboj	natura lingvo	alkompu- tilaj lingvoj	formal- logikaj lingvoj
funkcioj									
la meta- lingvist- ika	-	-	-	-	±	±	+	+	±
la kogni- tiva	-	±	±	?	+	+	+	+	+
la simbola	+	+	+	+	+	+	+	+	+
la prezen- ta-noma	-	+	±	?	+	+	+	+	+
la bilda (poezia)	-	-	±	+	+	+	+	±	±
la ĝeneral- -komunika	-	-	±	+	+	+	+	±	-
la esprim- konatiga kaj fatika	-	-	±	+	+	+	+	±	-

Литература:

1. Лотман Ю.М. "Статьи по типологии культуры", ТГУ, Тарту, 1970.
 2. Вандриес Ж. "Язык. Лингвистическое введение в историю", Гос. соц. -экон. из-во, М. 1937
 3. Вейнрейх У. "О семантической структуре языка", в сб. "Новое в лингвистике", вып. У, Из-во "Прогресс", М. 1970.
 4. Цейтин Г.С. "Мертвы естественных языков в языках программирования", в сб. "Машинный перевод и прикладная лингвистика", МПИИЯ им. М.Тореза, М. 1974
- ~~Бухарин~~
5. Ершов А.П. Предисловие к сб. "Взаимодействие с ЭВМ на естественном языке", ВЦ СО АН СССР, Новосибирск 1978
 6. Ердоногов Г.Г., Новоселов А.Г. "Автоматизация процессов накопления, поиска и обобщения информации", Из-во "Наука", М. 1979.
 7. Успенский Б.А. "Проблема универсалий в языкознании", в сб. "Новое в лингвистике", вып. 5, Из-во "Прогресс", М. 1970
 8. Налимов В.В. "Вероятностная модель языка", Из-во "Наука", М. 1979
 9. Абаев В.И. "О термине "естественный язык", ВЯ, 4, 1976
 10. Хоккет Ч.Ф. "Проблема языковых универсалий", в сб. "Новое в лингвистике" вып. У, Из-во "Прогрессе", М. 1970
 11. Ельмслев Л. "Пролегомены к теории языка", в сб. "Новое в лингвистике", вып. I, Из-во иностр. литературы, М. 1960, стр. 305 и далее.
 12. Успенский Б.А. "Проблемы лингвистической типологии в аспекте различия "говорящего" /адресанта/ и "слушающего" /адресата/, в сб. "To Honor Roman Jakobson", Mouton, The Hague, 1967.
 13. Иванов В.В. "Некоторые проблемы современной лингвистики", "Народы Азии и Африки", 1963, 4.
 14. Якобсон Р. "В поисках сущности языка", в сб. "Сборник переводов по вопросам информационной теории и практики", ВПИИЯ, № 16, ВПИИЯ, М. 1970.

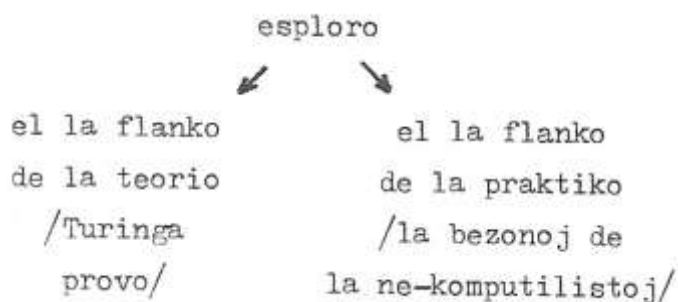
15. Якобсон Р. "К вопросу об общей теории падежей. Общие значения русских падежей", в сб. "Материалы к спецкурсу по структурной типологии языков", ИВЯ при МГУ, М. 1965; оригинал статьи: R. Jakobson "Beitrag zur allgemeinen Kasuslehre. Gesamtbedeutungen der russischen Kasus", "Travaux du Cercle Linguistique du Prague", 6, 1936.
16. Ингве В. "Гипотеза глубины", в сб. "Новое в лингвистике", вып. IV, Из-во "Прогресс", М. 1965.
17. Миллер Дж. "Магическое число семь плюс или минус два", в сб. "Инженерная психология", М., 1964.
18. Андрущенко В. М. "Зависимость между средней длиной лингвистической единицы и номером уровня, к которому она принадлежит", в сб. "Уровни языка и их взаимодействие. Тезисы научной конференции", 9-й МГПИИЯ им. М. Горького, М. 1967
19. Успенский Б. А. "Структурная типология языков", Из-во "Наука", М. 1965
20. Жоголев Е. А., Лебедева Н. Б. "СИМПОЛИЗ-64 - язык для программирования в символических обозначениях", Из-во МГУ, М. 1965
21. Богданов В. В., Ермаков Е. А., Маклаков А. В. "Программирование на языке АЛМО", Из-во "Статистика", М. 1976
22. Jakobson R. "Closing Statement: Linguistics and Poetics", in: "Style in Language", ed. by A. Sebeok, Cambridge, Mass., 1960
23. Ревзин И. И. "Структура языка как моделирующей системы", из-во "Наука", М. 1978

Per niaj homaj vortoj ni parolu al la maŝino!

GERÓ Peter

Resumo: Skribante pri la eblaj rilatoj inter komputiloj kaj homaj lingvoj temebblas aŭ pri la homa lingvo kiel objekto, aŭ pri la homa lingvo kiel ilo de maŝina prilaborado. Ĉi tiu artikulo montras kelkajn karakterizajn ekzemplojn por la ila rolo de homa lingvo. En la kadroj de la mallonga supervidado oni trovas ankaŭ la unuan publikatan program-plenumadon de la program-sistemo Esploranto.

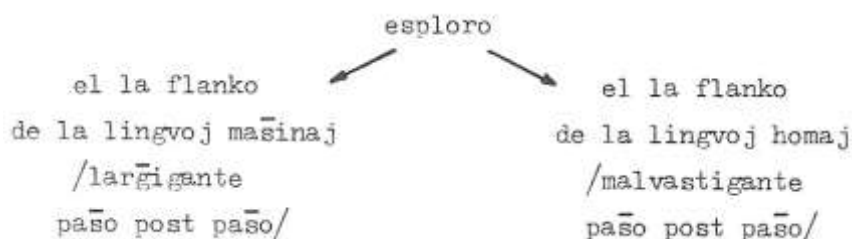
Programado, regado de maŝinoj per vivanta homa lingvo /tio estas, sen aparta programlingvo/ estas grava problemo de la teorio kaj de la praktiko tiel same. Estus sennecese en ĉi tiu eldonaĵo, se mi volus konigi la teorian problem-lanĉon, la Turingan provon. Inter la multaj eksperimentoj por parta kontentigo de la Turinga, provo oni povas vidi ankaŭ la eksperimentojn de hom-lingva programado. Aliparte, el la flanko de la praktiko, pli kaj pli forte aperas la necesaj pretendoj de tiuj, kiuj volas uzi la komputil-teĥnikon sen estiĝo komputil-teĥnikaj fakuloj.



Malklara estas la supra formulado de la klopodoj; plie nur sento, ol preciza starigado de iu tasko. Sed ĉi tiu malprecizeco originas ne nur el mi. Ankaŭ scienculoj esplorantaj la demandon eĉ dum jaroj malkaŝas ofte sendecidecon, kiam ili devus precize konturigi eĉ siajn proprajn temojn.

La problemo tuta, la senliga uzado de la vivanta lingvo por regi maŝinojn, ŝajnas malfarebla. La denombrado de Mealy pri la kvar tipoj de la gramatikoj per maŝino prilaboriĝeblaj tute ne eblas impliki iun ajn hom-lingvan gramatikon.

Se ĉi tiu malfarebleco veras, tiam restas nur du vojoj: aŭ postuli en la homa lingvo diversajn ligadojn, aŭ forigi kelkajn el la ligadoj de la programlingvoj. Tiel kaj ĉi tiel ni venas al sufiĉe artifika part-aro de la homa lingvo - ja kiu part-aro eblas esti tre taŭga por siaj taskoj.



Ni superrigardu kelkajn el la diversaj rezultoj kaj eksperimentoj!

Estas karakteriza produkto de la praktikaj pretendoj la program-lingvo COBOL. Ĝin estigis interfaka grupo por ne-komputilista uzo /1/.

Granda parto de COBOLa programo estas tia, kvazaŭ oni devus plenigi blanketon. Eta ekzemplo el Report Section /tabelo-desegna ĉapitro/:

Ø1 FIRST-LINE

TYPE PAGE HEADING.

Ø2 LINE 1.

Ø3 COLUMN Ø5 PICTURE XXXXXXXX
SOURCE CURRENT-DATE.

Ø3 COLUMN 4Ø PICTURE 99
SOURCE PAGE-COUNTER.

Ø2 LINE PLUS 1.

Ø3 COLUMN 25 PICTURE XXXXXXXXXXXX
VALUE 'TOP OF PAGE'.

La aliaj partoj de la programo estas anglaj frazoj gramatike preskaŭ korektaj, kiel ekzemple:

```
PERFORM FILLING -THE -MATRIX
  VARYING INDEX-OF-COLUMNS
    FROM 1 BY 1 UNTIL END-OF-LINE,
  AFTER INDEX-OF-LINES
    FROM 1 BY 1 UNTIL END-OF-PAGE.
```

```
aŭ:      READ CARD-FILE;
          AT END DISPLAY 'FINISHED',
          STOP RUN.
```

kaj tiel plu.

Ekzistas ĉirkaŭ 300 tielnomataj okupitaj vortoj en la program-lingvo COBOL. La uzo de ĉi tiuj vortoj estas permesata nur en la rolo por kiu ili, okupitiĝis. Sekve ne okazebblas en la programo, ke ekzemple LABEL estu la nomo de iu datumo aŭ EXIT de iu procedo.

COBOL estas vera part-aro de la angla lingvo. Ĝia gramatiko estas treege simpligita; ĝi estas preskaŭ medio-sendependa. Verboj /la ŝlosil-vortoj de la instrukcioj/ unusence determinas tiujn 3-4 eblecojn, kiel la konkreta instrukcio konstruiĝi povas.

En Hungario COBOL ne estas tro populara. Ĝia alsimiliĝado al la angla lingvo ne avantaĝas por ni. Ne-programistoj sentas COBOL-on tro ligita, programistoj sentas ĝin amasi vortojn superflue.

Ankaŭ Progreso/1 estas produkto de pretendo praktika. Ĝia celo estas: krei universalan malgrand-komputilan dialogan eblecon de tekst-prilaborado, unuavice por instruada uzo. La fundamentaj vortoj de la lingvo estas vortoj Esperantaj, el kiuj oni povas kombini frazojn. Ekzemple:

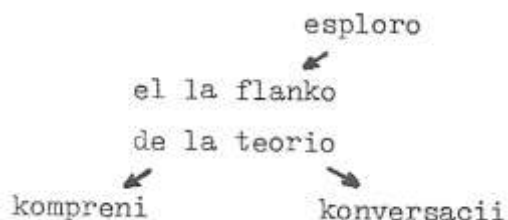
```
LA MAGIA VORTO NE EGALAS JE IO NOMOTA LA KOMENCAĴO + "UR" AŬ "BR" AŬ "MR"
NOMOTA LA GRAKADO + IO 5 KARAKTRA + IO AJN.
```

El la vidpunkto de aliĝo al la homaj lingvoj estas speciala virto de Progreso/1 la uzo de prefiksoj. Estas la prefiksoj, kiuj signas la kunapartenantajn datumojn dum la datum-cirkulado inter la programsegmentoj.

Multaj da partoj de Progreso/1 estas ankoraŭ ne optimaj. Ellaborado de Progreso/2 jam progresas.

Per komputilo uzebla fazo de Progreso/2 bedaŭrinde ankoraŭ ne ekzistas.

La eksperimentoj startintaj el la teoria flanko montras bildon ne malpli varian, ankaŭ pro tio, ke oni povas provi la respondadon laŭ du manieroj. Kiel /3/ skribas: -- Inter tiu, ke "kompreni" kaj "doni akcepteblan respondon" tute ne estas limo akra. Samtempe ĉiuj legantoj konas tiujn du anim-statojn, kiujn ni provas nun elkapti per helpo de analogio. Unu estas, kiam ni faras intensan forto-streĉon por kompreni la esencon de iu argumentado, la alia estas la stato, kiam la diskuto nin ne interesas, sed, volante ŝajni kelkfoje ni balancas la kapon, aŭ ni interkrias per tiaj senenhavaj frazismoj, kiel "vere interese!"



La sistemo STUDENT, menciita en /3/, estas bona ekzemplo por la unua varianto. La STUDENT povas solvi tiajn taskojn, kiuj eblas modeliĝi per unuagrada ekvacio-sistemo. Ekzemple oni povas entajpi la sekvan tekston: La sumo de tri nombroj estas 9. La dua nombro estas per 3 pli granda ol la duoblo de la unua nombro. La tria nombro egalas je la sumo de la unuaj du nombroj. Kiom estas tiuj tri nombroj? Kaj la STUDENT respondas:

LA UNUA NOMBRO ESTAS 0.5

LA DUA NOMBRO ESTAS 4

LA TRIA NOMBRO ESTAS 4.5

Eĉ la plej simplaj rezultoj de la alia deiro estas jam eksterordinare vidindaj. /4/ desegnas ekzemple tiun 35-linian PROLOG-lingvan programeton, kies plenum-liston mi ĉi tie publikas Esperantigite. La program-plenumo startigeblas per la vorto PAROLADU.

OS/360 PROLOG VERSION 2 (79APR20)

(* CPU TIME: 0 016 SEC 2 CALLS 0 FAILURES *)

	39*	- PAROLADU.
MI JAM SCIAS	40*	MI ESTAS HUNGARULO.
HUNGARULO	41*	KIU ESTAS MI?
MI NE SCIAS	42*	KIL ESTAS VI?
MI JAM SCIAS	43*	VI ESTAS PROGRAMO
	44*	SKRIBITA EN PROLOG.
MI JAM SCIAS	45*	LA PLEJ BELA PROGRAMO
	46*	ESTAS VI.
MI JAM SCIAS	47*	KIL ESTAS VI?
PROGRAMO SKRIBITA EN PROLOG		
LA PLEJ BELA PROGRAMO		
MI NE KOMPRENAS	48*	VERE?
MI ENUAS VIN SAME	49*	MI VIN ENUAS.

(* CPU TIME: 3 593 SEC 1165 CALLS 299 FAILURES *)

50* - STOPU

(* CPU TIME: 0 010 SEC 2 CALLS 0 FAILURES *)

Ĉi tiu programo povas reakcii kelkajn fiksjajn eblecojn.

/3/ kaj /6/ skribas pri la sistemo ELIZA, kiu povas pridemandi pacientojn psiĥologist- aŭ kuracist-maniere. La esenco de la algoritmo estas, ke la programo serĉas ŝlosil-vorton en la teksto skribita per la paciento, kaj per la ŝlosilvorto ĝi formas respond-frazon, eble demandon. Se ĝi ŝlosil-vorton ne trovas, tiam aŭ ĝi revenas al ŝlosil-vorto pli frua, aŭ kreas frazon indiferentan.

/6/ skribas pri Parry, kiu kreiĝis pro tio, ke psiĥologist-studentoj per ĝi ekzercu la pridemandadon. Estas temoj, al kiuj Parry revenas multfoje, ofte neatendite. Ĝi reakcias ekstre akrasente la plej malgrandan insultadon, suspekton tio estas, la ŝlosil-vortojn esprimantajn ĉi tiujn.

La perlo de la temo estas, laŭ mia opinio, la konversacio, en kiu ELIZA kaj Parry estis lasita unu kontraŭ la alia. Lasu min nun montri kelkajn komencajn frazojn de ĉi tiu konversacio /laŭ /6//:

- Bonan vesperon. Sciigu vian problemon, mi petas. La frazon finu per punkto aŭ per demandosigno.
- Homoj kelkfoje taŭzas miajn nervojn.
- Mi ne estas certe, ke mi komprenis vin perfekte.
- Tiam atentu pli bone.
- Mi Densas, ke estas vi, kiu devus pli bone atenti.
- Vi havas la rajton ekspliki vian propran opinion.

Kaj tiel plu. Scianta la algoritmon /ELIZA provas perforti la pacieron diri ŝlosil-vorton, Parry kaptas ĉiun okazon por ofendiĝi/ ni povas imagi la procezon de la program-plenumado. Sed estus mensogo, se mi dirus, ke mi ne kredus, ke ĉi tiu konversacio estas homa! Mi pensas, ke nuntempe ĉi tie estas la pinto de la esploroj startigitaj el la teoria flanko.

Ankaŭ mi mem havas kelkajn rezultojn en ĉi tiu tereno. Mi komencis mian esploradon el la flanko de la praktiko kaj el la flanko de la vivantaj homaj lingvoj. Mi malkovris, ke la teoremoj tezintaj la maleblecon de la prilaborado de tutaj homaj lingvoj estas malprecizaj, pro erara miksado en sia argumentaro. Ilia argumentaro, origininta el la problemo de maŝina tradukado el homa lingvo en homan lingvon, ne donas konvenan respondon al la problemo de la plenumado de komputilaj programoj skribitaj homlingve.

Mi serĉis la minimon de la ligaro malforlasebla por la maŝina prilaborado. Mi vidis, ke estas nur malmultaj ĉi tiuj ligoj ekster la ligoj sentataj per la sobra cerbo apriore naturaj.

La praktika rezulto de mia laboro estas la prilabor-programo Esploranto /7/. Kiel unuan lingvon, per kiu mi ĝi efektivigis, mi elektis la indoeŭropan minimum-lingvon Esperanton. /Ĝia esperanteco ne estas ĝia unuebleca stato, ĉar ĝi, konstruiĝinte el komponantoj, ŝanĝeblas facile./ La tuta lingvo-priskribo por Esploranto konsistas el tri paĝoj, inkluzive la necesajn konojn pri Esperanto. La pluj lingvaj reguloj estas, ke ekzemple oni ne skribu la vortojn m a l d e n s i g e kaj tiel plu.

Mi pensas, ke ekzemploj estas pli vidigaj ol ĉiu priskribo:

```

--> Deka fazo dialosa; 1981.05.29. <--
-> Programeto por montri kelkajn el la nunaj eblecoj
-> de Esploranto:
-> Unue legu la stop-signon! La komenco de la ciklo:
-> legu ankau\ la datumon. Se la datumo egalas je la stop-
-> signo: tiam stopu.
-> Alifoje skribu la datumon nomate. Daŭrigu ekde la ko-
-> menco de la ciklo.

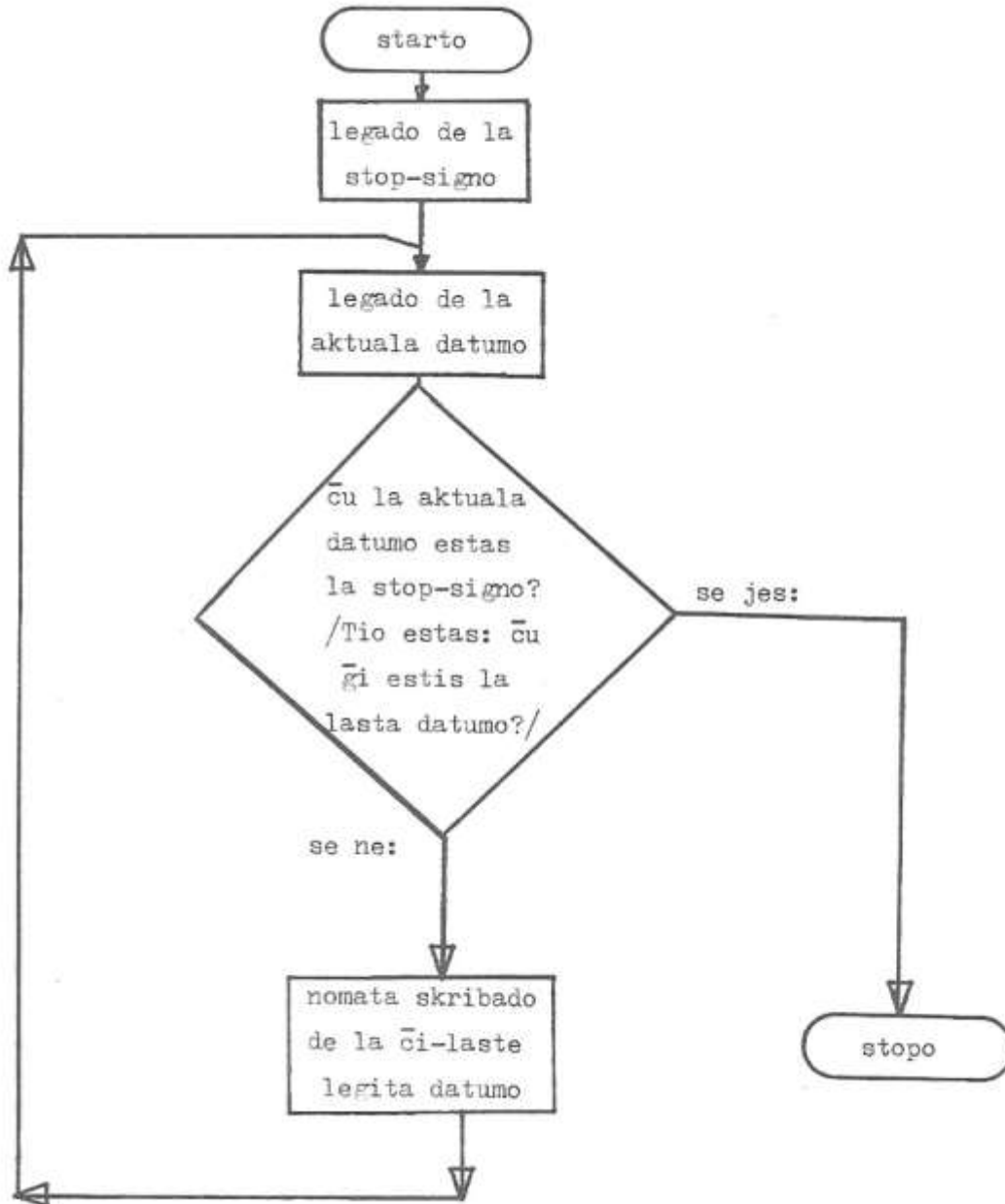
PROGRAMETO POR MONTRI KELKAJN EL LA NUNAJ EBLECOJ
DE ESPORANTO:
UNUE LEGU LA STOP-SIGNON! LA KOMENCO DE LA CIKLO:
LEGU ANKAŬ LA DATUMON. SE LA DATUMO EGALAS JE LA STOP-
SIGNO: TIAM STOPU,
ALIFOJE SKRIBU LA DATUMON NOMATE. DAŬRIGU EKDE LA KO-
MENC0 DE LA CIKLO.

-> *f*
-> 1981
  DATUMo
    =
  1981
-> ??????????
  DATUMo
    =
  ??????????
-> *TEKSTO*
  DATUMo
    =
  *TEKSTO*
-> *f*
      S t o p o .
-> Skribu Y-on!
-> SKRIBU Y-ON!
  !!!> La datumo ne havas valoron.
-> Esrlisu Y-on per la stop-signo. Nomate skribu
-> von.
-> EGALIGU Y-ON PER LA STOP-SIGNO. NOMATE SKRIBU
  YON.
  Yo
    =
  *f*
-> leŝu von!!!
-> LEGU YON!!!
-> 3.14159
-> Skribu la stop-signon. Ankau\ Yon
-> SKRIBU LA STOP-SIGNON. ANKAŬ YON
  !!!> La frazo-fino-signo mankis post la lasta frazo.
  !!!> La frazo ne enhavis instrukcion.
      (Mi pensas, ke la ĉi-lasta instrukcio daŭriĝas.)
  *f*
  3.14159
-> Kiel vi fartas?
-> KIEL VI FARTAS?
  !!!> La frazo ne enhavis instrukcion.
  !!!> La tuta program-sesmento ne enhavis instrukcion.
      Mi dankas al vi por la interesa konversacio.
-> --> La program-plenumo finiĝis. <--
  JOB74 -- STOP 1

```

Sur la figuro mi signis la hom-tajpitajn liniojn per sageto. La pluaj linioj estas la respondoj de la sistemo Esploranto. Videblas, ke la entajpitan programon la maŝino reskribas; per ĉi tiu reskribo ĝi signas, ke la legado de la programo estas plenumita sen morfologia eraro.

La unua montrata programeto plenumas ĉi tiun simplan procezon:



En la aliaj ekzemploj oni vidas pli-malpli korektajn instrukciojn kaj la reakciojn de la Esploranto: la plenumadon aŭ la erar-mesaĝojn.

Kiel videblas,

- la programoj novas, sed ne devas havi nomojn;
- la program-nomo povas konsisti el multaj vortoj;
- oni novas uzi grandajn kaj malgrandajn leterojn tiel same;
- ankaŭ la datum-nomoj estas skribeblaj per grandaj aŭ malgrandaj leteroj,
- la datum-nomoj povas sed ne devas enhavi ligilon: Y-o kaj Yo kaj y-o kaj yo kaj Y-O kaj YO signifas la saman datumon;
- oni povas komenci novan alineon ie ajn;
- se la aktuala linio estas plena, tiam la vortoj povas esti apartigate ĉe ligilo aŭ alie ajn;
- la vort-ordo estas senliga,
- senlige uziĝeblas artikoloj /la/ kaj aliaj help-vortoj /unue, tiam, alifoje, ankaŭ ktp./;
- - la datumojn povas havi numerajn kaj tekstajn valorojn tiel same;
- la program-plenumo estas daŭrigebla ekde laŭplaĉa etikedo;
- la etikedoj povas konsisti el multaj vortoj;
- la program-plenumo estas stopigebla per la instrukcio "stopu";
- la fino de la programo /segmento/ estas signita per malplena linio;
- la fino de la tuta kontakto per la Esploranto estas signita per malplena linio denove;

kaj kiel ne videblas sur la, unupaĝa listo:

- la program-plenumo estas stopigebla ankaŭ per la instrukcio "haltu" aŭ "finu";
- la Esploranto povas doni pli ol dudek diversajn erar-mesaĝojn;

- ĉiuj novaj vortoj uzataj en iu programo tuj enmiksiĝas en la vortaron;
- la nuntempa vortaro enhavas 100-150 vortojn.

Mi akcentas, ke la Esploranto estas provizore ne sistemo uzina, sed sistemo eksperimenta. En somero de 1981 ĝi ankoraŭ ne konis pli ol 10-11 tipojn de instrukcioj; ĝi ne povis plenumi operaciojn, nek aritmetikajn, /escepte egaligadon,/ nek karaktrajn, /escepte egaligadon,/ nek logikajn /escepte ke "X-o ne malegalas je Y-o" estas la sama, kiel "X-o egalas je Y-o/".

La sistemo Esploranto startis sur nova vojo, optimiante por la homa sed ne por la maŝina efikeco. Sed ĉi tiu estas ankoraŭ nur la teĥnika parto de la afero.

Nenia ilo povas solvi la problemojn, nur helpi la homan problem-solvan adon. La programsistemo Esploranto - kiel ankaŭ la aliaj montritaj lingvoj kaj lingvo-eksperimentoj - estas nur ponto por ju pli ne-komputilistaj fakuloj al la uzado de unu el la plej grandefikaj kaj grandefektaj iloj de nia epoko: la komputilo.

Bibliografio kaj aliaj fontoj

- /1/ COBOL, American National Standard /Amerika Nacia Normo/
- /2/ MÜNNICH Antal: Progreso/1 szövegfeldolgozó programnyelv specifikációja /Difino de la tekst-prilaborada programlingvo Progreso/1/, INFELOR, Budapest, 1972.
- /3/ J. NIEVERGIW - J. C. FARRAR - E. M. REINGOLD: Matematikai problémák megoldásának számítógépes módszerei /Komputilaj metodoj por solvi matematikajn problemojn/, Műszaki Könyvkiadó, Budapest, 1977. /Originale: Computer Approaches to Mathematical Problems, Prentice-Hall, Inc./
- /4/ SZEPEDI Peter - FUTÓ Iván: PROLOG kézikönyv /Manlibro de PROLOG/, NIMIGÜSZI, 1977.
- /5/ La programo plenumiĝis per komputilo IBM 370/145 en SZÁMOK
- /6/ Számítógépes párbeszédék /Komputilaj konversacioj/ en: Korszerű vezetés Moderna administrado/, 1979. 7-8.

/7/ GERÓ Peter: La programsistemo Esploranto eksperimento por kompreni la hom-lingvajn programojn; prelego en Žilina, konferenco 1981.

/8/ La programo plenumiĝis per komputilo PDP 11/40 en SZÁMOK.

Glosaro

SZÁMOK : Uemzetközi Számítástechnikai Oktató es Tájékoztató Központ
/Internacia Komputad-teĥnika Instruada kaj Informada Centro/, Budapest.

Biografio



GERÓ Peter, originale agregaciulo de matematiko kaj fiziko, laboris dum kelkaj jaroj kiel programplananta matematikisto. Ekde 1978 li instruas altnivelajn programlingvojn, programplanadajn kaj program-skribadajn metodojn en SZÁMOK. Lia speciala esplor-temo estas /ekster Esploranto/: objektiva mezuro de la kvalito de la programoj

La subrutino de eligiloj
 en la operaciad-sistemo por regado
 de teknologiaj procezoj

 Vladimír NĚMEC

Resumo; La artikolo resumas bazajn principojn por kreado de priserva subrutino de eligiloj en la operaciad-sistemo por regado de teknologiaj procezoj kaj starigas cefajn postulojn, kiujn ĝi devas plenumi.

La operaciad-sistemoj por regado de teknologiaj procezoj havas aŭ apartajn subrutinojn por unuopaj specoj de eligiloj aŭ universalan subrutinon por ĉiuj eligiloj ebligantaj pokodan aŭ povortan eligon kaj duan subrutinon por eligiloj ebligantaj poblokan eligon.

La plej temporaba estas priservo de eligiloj ebligantaj povortan al pokodan eligon, do ni pli detale pritraktos ilin.

Ankaŭ el la vidpunkto de la uzanto-proceso /u-proceso/ oni povas paroli pri povorta aŭ pobloka komunikado.

Ĉe pokoda aŭ povorta eligo la kodo aŭ koda duopo estas transdonata al la eliga subrutino kutime en registrumo. Ĉe pobloka komunikado la u-proceso transdonas al la eliga subrutino adreson de la storo kie troviĝas la raporto. La raporto estas sinsekvo de informoj, kiuj estas eligotaj kaj havas sian kapon, korpon kaj eventuale finkodon.

Se la raporto ne havas sian finkodon, signifas tio, ke la raporto ne estas finita. Tiu ĉi solvo ebligas eligadi longajn protokolojn per sinsekvo de nefinitaj ra-

portoj kaj la eligilo estas rezervita por la koncerna u-proceso ĝis la eligo de lasta raporto kun la finkodo.

En la operaciad-sistemo povas esti difinitaj kelkaj finkodoj. La finkodo laŭ sia speco povas esti por diversspecaj eligiloj malsame interpretita: por skribmaŝino kiel liniado, por presilo kiel paĝado, k.t.p.

La ebleco elsendi nefinitan raporton ŝparas la storon, kio povas esti ofte postulata.

La kapo de la raporto enhavas jenajn parametrojn:

- a/ Formon de la raporto. La raporto povas enhavi unu, du aŭ tri kodojn en unu vorto. En lasta kazo la raporto enhavas nur majusklojn kaj ciferojn.
- b/ Nomon de la eligilo /Povas tio esti ekzemple du- aŭ tricifera numero/
- c/ Longecon de finita al nefinita raporto. La longeco de la raporto estas libervola
- d/ Postulon de la reaktivigo post la eligo de la raporto, aŭ post la transskribo de la raporto el la storo de u-proceso en la bufran storon /bufron/ de operaciad-sistemo. Tiu ĉi postulo havas signifon nur tiam, kiam la operaciad-sistemo ebligas al la u-proceso daŭrigi en kia laboro pli frue ol estas tuta raporto transskribita en la bufron aŭ eligita. Tiam la u-proceso rajtas sinsekve postuli eligon de kelkaj raportoj el diversaj lokoj de sia rezervita storo. La operaciad-sistemo informas pri la eligo aŭ transskribo de la raporto en ĝia kapo. Se la u-proceso deziras eligi pluan raporton, sed ĝi jam ne havas plu spacon por ĝi, ĝi povas malfunkciigi postulinte sian reaktivigon.
- e/ Indikon pri la urĝeco liberigi la spacon por plua raporto.

La operaciad-sistemo povas eligadi la raporton rekte el la storo de u-proceso, aŭ transskribi ĝin en propran bufron. En dua kazo la storo por la raporto de

u-proceso estas pli rapide liberigita. En la operaciad-sistemo povas havi ĉiu eligilo propran bufron, kiu en la kazo de povorta eligado flanke de u-proceso povas esti cikla aŭ la operaciad-sistemo havas "liberajn bufrojn", kiuj plenigitaj per raportoj estas envicigataj en prioritatan vicon por koncerna eligilo. Tamen la nombro de tiaj bufroj estas limigita kaj ilia plenigo povas kaŭzi haltigon de u-proceso. Cikla bufro estas tiu, kiu povas esti samtempe malplenigata kaj plenigata, kio okazas en ciklo.

Pli konvena solvo estas krei ciklajn bufrojn sur ekstera storo, en kiujn estas transskribataj raportoj se la bufroj en la operaciad-sistemo estas plenaj. Estas konvene krei sur ekstera storo apartajn bufrojn por ĉiu u-proceso kaj ĉiu eligilo. Tiuj ĉi ciklaj bufroj estas difinitaj per sia radiuso:

$$L_B = k \cdot r$$

L_B : kapacito de la bufro

k : longeco de la bloko

r : bufra radiuso

La raporto estas transskribata en la ciklajn bufrojn sur ekstera storo pobleke. La longeco de raportaj blokoj estas konstanta. Kiam la bufro en la operaciad-sistemo estas malplena, la plej malnovaj raportoj estas transskribataj el la cikla bufro sur ekstera storo en la liberigitan bufron en la operaciad-sistemo.

La cikla bufro sur ekstera storo ebligas konservon de la plej freŝa historio de raportoj. Se tiu ĉi bufro estas plena, la plej malnovaj raportoj estas anstataŭigataj per la plej freŝaj. Pro tio neniam povas okazi, ke iu u-proceso estus haltigita pro plenigitaj bufroj. Tiu ĉi solvo precipe estas konvena por eligado de raportoj pri difektoj kaj eraroj el la regata aŭ kontrolata teknologia procezo, kiam interesas nin precipe lastaj informoj. Por eligado de diversaj protokoloj ni ciklajn bufrojn sur eksteraj storoj ne uzas, car ni ne povas allasi perdon de iaj informoj, aŭ uzante ilin ni post ilia plenigo haltigas

u-proceson ĝis parta liberigo de tiuj ĉi bufroj. En ĝuste projektita sistemo oni povas certigi tion, ke la ciklaj bufroj sur ekstera storo neniam pleniĝos, ĉar la bufra radiuso povas esti operatore redifinebla.

Por ebligi en daŭre laboranta komputil-sistemo riparojn de difektitaj periferieroj, eventuale ilian profilaktikon estas konvene solvi ilian aŭtomatan anstataŭigon per alia periferiero. Aŭtomata anstataŭigo estas necesa ankaŭ tiam, kiam koncerna periferiero estas en la stato "daŭre okupita" kvankam ĝi nenion eligas.

La anstataŭigo povas esti:

a/ ĉena

b/ ronda

Ĉe ĉena anstataŭigo la taskon de difektita aŭ elsaltita periferiero transprenas alia periferiero, speciale destinita por tiu ĉi tasko.

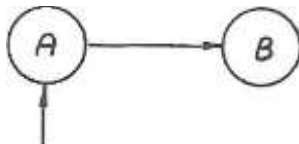


Fig. 1 : ĉena anstataŭigo

La taskon de la eligilo A transprenas la eligilo B, al kiu rekte ne estas sendataj la raportoj de u-procesoj.

Tiu ĉi anstataŭigo postulas apartajn rezervajn eligilojn, kiuj en la kazo de funkciantaj bazaj eligiloj ne estas eluzataj.

Pli konvena estas ronda /cirkla/ anstataŭigo:

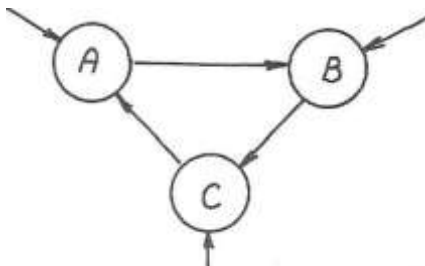


Fig. 2 : Cirkla anstataŭigo

Oni povas difini anstataŭigon de tri aŭ pluraj eligiloj. Ĉiu el eligiloj eligas raportojn al ĝi sendataj de u-proceso

kaj en la kazo de unu elŝaltita eligilo dua transprenas ĝian taskon apud sia. La sistemo ne averias eĉ tiam se almenaŭ unu el eligiloj en la cirklo funkcias.

La eligiloj povas esti aligitaj al u-proceso aŭ operatore aŭ programe. Kelkaj aperaciad-sistemoj limigas la nombron de operatore aŭ programe aligitaj eligiloj. La operatore aligitaj eligiloj principe estas daŭre aligitaj al koncerna u-proceso kaj alia proceso ne rajtas komunikadi kun ili. Programe aligitaj eligiloj estas portempe aligitaj al la u-proceso, kiu fininte sian komunikadon liberigas ilin por aliaj u-procesoj postulantaj komunikadon kun ili.

Tiu ĉi solvo de la problemo ne ĉiam estas akceptebla, ĉar u-proceso ne povas eligadi siajn raportojn per kia ajn eligilo, eventuale devas atendi liberigon de portempe programe aligita eligilo. La plej konvena estas tia subrutino, kiu ebligas eligadon per kiu ajn eligilo al kiu ajn u-proceso ne haltigante ceterajn.

Konkludo: Subrutinoj de eligiloj devas plenumi jenajn postulojn:

- a/ Aŭtonome eligadi al ĝi senditajn raportojn
- b/ Ebligi komunikadon al kiu ajn u-proceso kun kiu ajn eligilo
- c/ Ne haltigi aŭ ne atendigi u-proceson pro komunikado kun eligilo
- d/ Ebligi poblin komunikaĵon
- e/ Ebligi sendon de nefinitaj raportoj
- f/ Aŭtomate anstataŭigadi difektitajn, elŝaltitajn aŭ "daŭre okupitajn" eligilojn



Biografio :

Vladimir NĚMEC, inĝeniero, laboris en la j. 1968-74 ĉe Trafika Altlernejo en Žilina. Ekde la j. 1974 laboras ĉe URAP /Instituto por racionaligo de industrio/ en Žilina, en la fako Aŭtomatigo de la sistemo de la regado. CSSR

HSZR-1

Retokonstrua kaj -desegna programo

ANGYAL József

Resumo: La programo HSZR-1 helpas multe en la konstruo de logika reto en retoplanada laboro. El la rilatoj antaŭaj kaj postaj al la agadoj ĝi kunmetas la logikan reton kaj desegnas ĝin en formo CPM aŭ MPM sur linipresilon. Ĝi analizas la tempon en la reto kaj difinas la kritikan vojon.

Unu el la plej efikaj kaj plej ofte uzataj procedoj inter la praktikaj aplikadoj de la operaciesploro estas la retoplanado. La komencan rapidan disvastiĝon de la retoplanado malhelpis la permana konstruado kaj desegnado de logikaj retoj. La plej malfacila kaj la plej multan tempon pretenda agado dum retoplanado estas la konstruado kaj prizorgado de la reto. Tiel la reton preparantoj ne volonte kontrolas kaj redesegnas la jam pretigitan reton. Pro tio la logika reto relative frue disiĝas de la realeco. Ankaŭ tio estas unu el la kaŭzoj pro kiu post la komenca disvolviĝo multaj trompiĝis pri la retoplanado.

Dum la evoluo de la retoplanado la apero de grandplenumaj komputiloj signifis turnopunkton. Komence nur la tempoanalizo pretendanta multe da tempo estis prilaborita per komputilo, sed la pretendoj de la uzantoj kaj la eblecoj de komputiloj instigis la komputistojn

solvi novajn taskojn. Tiel la plejmulto de komputilproduktaj firmaoj kaj programprilaboraj institutoj pretigis programaron pri retoplanado. Tiuj programaroj solvis jam automate la taskojn de la retoplanado. ili estis la sekvantaj:

- tempoanalizo,
- koncentro de energifontoj,
- analizo kaj planado de kostoj,
- aktualigo.

La aplikado de programaroj pri retoplanado denove prosperigis la retoplanadon. Per tiuj programaroj ni povas aktualigi de tempo al tempo la retoplanojn, kaj tiel la tabeloj eligitaj per ili povas doni datumojn proksimajn al la realeco. Sed samtempe la aktualigo de logika reto en desegnaĵo eklipsis pli kaj pli. Fariĝis praktiko, ke la pretigitajn logikajn retojn oni aktualigis poste nur por eligaj tabeloj de programaroj pri retoplanado. La eventualaj ŝanĝoj de logikaj rilatoj en la reto ne estis transprenitaj por la desegnaĵo de la reto. Tiel la investado antaŭeniris kaj jam ne estis disponebla aktuala foje replanita kaj redesegnita-logika reto.

Pro tio kreskis la pretendo por plani kaj desegni logikajn retojn perhelpe de komputilo. Tiun ĉi pretendon kontentigas la PROJACS /MPM desegno/, MANDAS /MPM desegno/ kaj la tie prezentota HSZR-1 /CPM kaj MPM desegno/.

La programaro HSZR-1

Disvastiĝis du procedoj pri la retoplanado: CPM kaj MPM. Ambaŭ havas malsamajn avantaĝojn kaj malavantaĝojn

La avantaĝo de procedo CPM estas, ke eblas pli facile videbligi ĝin kaj pli rapide trarigardi ĝin. Ĝia malavantaĝo estas, ke ĝi nur komplike povas ŝanĝi la agadojn, trakti atendojn.

La procedo MPM facile traktas la agadŝanĝojn kaj atendojn perhelpe de la distancoj inter la rilatoj. Ĝia malavantaĝo estas, ke la reto MPM ne estas sufiĉe travidebla, kaj ĝia tempoanalizo pretendas ekstreme multe da kalkulo.

Inter la adeptoj de MPM kaj CPM estas nepacigebla kontraŭeco, ĉar ne konsiderante la konkretan taskon ne eblas decidi unusence kiu procedo estas pli bona. Sed solvo ekzistas: apliko de kombinita procedo pri retoplanado, kiu kunigas la avantaĝojn de la du procedoj. Tio estas "tempoanalizo de la reto" per MPM, "prezento kaj desegno" per CPM. Sed tio sen komputilo estas malfacila tasko.

La programaro HSZR-1 faras la planadon de la logika reto, tempoanalizon laŭ MPM aŭ CPM. Tiel la apliko de HSZR-1 donas kvar eblecojn:

- tempoanalizo kaj desegno per CPM;
- tempoanalizo per CPM, desegno per MPM;
- tempoanalizo per MPM, desegno per CPM;
- tempoanalizo kaj desegno per CPM.

Inter ili la tria kombinita procedo MPM/CPM estas la plej grava. La dua kombinita procedo CPM/MPM ne havas praktikan gravecon.

Ecoj de la programaro HSZR-1

- ĝia uzo ne pretendas konojn pri komputotekniko,

- bazaj konoj pri retoplanado sufiĉas por ĝia uzo,
- desegnas la logikan reton en bone travidebla formo, tiel konzervas unu el la pli grandaj avantaĝoj de la retoplanado: la travideblecon,
- ĝi desegnas la logikan reton sur linipresilo.

Servoj de HSZR-1

- konstruo de logika reta,
- desegnado de logika reto en formo CPM aŭ MPM,
- enmeto de nefaktaj agadoj /ĉe la reto CPM/,
- indiko de logika erara, ĝia ĉesigo,
- modifo kaj aktualigo de logika reto,
- tempoanalizo laŭ CPM aŭ MPM,
- desegno de laŭtempa reto /ĉe CPM reto/,
- kunkonstruo de partaj retoj,
- genero de enigaj datumoj sur magnetaj diskoj por la programaroj /PERT, PCS, PMS, PROJACS,..../

Konstruo de logika reto: Unue ni listigas la agadojn por la laboro de HSZR-1. La vicordo de la agado estas hazarda, laŭplaĉa. /Ni povas skribi ilin laŭ ilia apero./ Poste ni numerigas ilin ekde 1. Tiel ni citos ilin laŭ ilia numero. Sekvas la kolekto de logikaj rilatoj: por ĉiu agado ni devas doni la numeron de la ĵus antaŭa kaj posta agadoj. Tiel ni donas ĉiujn logikajn rilatojn dufoje, sed tiel ni povas korekti la erarojn faritajn far la uzantoj.

Ekz.: oni malbone donis la logikan rilaton aŭ tute forgesis ion.

El tiuj logikaj rilatoj HSZR-1 konstruas la logikan reton, en kazo de reto CPM, ĝi eĉ numerigas la agadojn.

Ĉe logikaj retoj ofte okazas, ke necesas enmeti nefaktajn agadojn kiel sekvenco de logikaj rilatoj. Tion HSZR-1 mem faras. Ĉe tasko pri retoplanado unu el la plej malfacilaj taskoj estas la pretigo de la komenca reto. Ĝuste pro tio eraroj oftas en logikaj retoj. La programo provas helpi en la ĉesigo de tiuj logikaj eraroj, skribante mesaĝon pri la eraro. Se estiĝas buklo en la reto pro la malbone donitaj datumoj, HSZR-1 atentigas la uzanton pri la buklo. Por plifaciligi la serĉadon de la eraro ĝi ĉesigas la buklon. Ĝi disŝiras la buklon ĉe ĝia komenco, kaj tiel reprezentas ĝin.

Modifo, aktualigo de logika reto: La datumojn de la unuopaj subretoj storas HSZR-1 sur magnetbendo. Por modifi la logikan reton necesas doni nur la modifantajn datumojn. Por modifi la bazajn datumojn ekzistas du eblecoj:

- modifi la logikajn rilatojn,
- modifi la nomojn de agadoj, resp, la datumojn pri tempo.

Se ni modifas logikan rilaton, ni donas la "novan" ĵus antaŭan kaj postan agadojn. La programo forigas la ĝisnunajn logikajn rilatojn apartenantajn al la agado, kaj konsideras la nove donitajn. Per tiuj modifoj ni povas;

- forigi
- enmeti,
- ligi antaŭen aŭ malantaŭen.

Se ni modifas la datumojn rilatantajn al agado, tiam ni lasas malplena tiujn partojn de la karto, kiujn ni ne

volas modifi. HSZR-1 reskribas la modifendan parton per la nova enhavo. La modifantaj datumoj ĉiam troviĝas sur karto.

Desegnado de logika reto: Ni povas desegni la konstruitan logikan reton laŭ CPM aŭ laŭ MPM, eĉ dum la pretigo iam ajn ni povas transformi la reton el CPM en MPM aŭ male. Ni havas eblecon desegnigi la subretojn en partoj por pli bone travidi ilin, kaj eblas malpligrandigi la subretojn per HSZR-1.

Sur la reto CPM ĉiu agad-linio, entenas la numeron de la agado, super kaj sub ĝi la 24 karakteran mallongigon de la nomo de la agado. Sur MPM desegnaĵo de reto la supera flanko de la agad-kvadrato enhavas la numeron, la suba flanko la labortempon. En la agad-kvadrato troviĝas la 24 karaktera mallongigo, sube la komenco de la agado kaj lia tuta tempo /se ni ne malgrandigas la reton/.

Se ni malgrandigas la reton horizontale, la mallongigo estas nur 16 karaktera.,

Sur ambaŭ retodesegnaĵo /se ni aldonis la tempodatumojn/ indikiĝas ankaŭ la kritika vojo.

HSZR-1 tiel disigas la reton, ke disŝirante la presil-paperojn kaj metante ilin unu apud la alia principe en ambaŭ direkto la reto povas esti senfina.

La unua kaj 2a bildo montras la desegnaĵon CPM kaj MPM de la sama reto./

Tempoanalizo laŭ CPM aŭ MPM: Samtempe kun la konstruo HSZR-1 analizas ankaŭ la tempon en la unuopaj subretoj.

/se ni havas tempodatumojn./ Ni povas peti tempoanalizon laŭ CPM aŭ MPM.

Kaze de tempoanalizo laŭ MPM ni povas indiki la distancon de la rilatoj. Por du sinsekvaj agadoj la programo supozas la rilaton fino komenco kun tempo 0 /normala sekvo/. Nur tiuj rilatoj estas donendaj, kiuj diferenciĝas de tiu. Ni povas uzi kvar rilatan distancon:

komenco-komenco, komenco-fino, fino-komenco, fina fino.

Desegnado de CPM reto laŭtema

Se oni indikis pritempajn datumojn, la programo aŭtomate desegnas la laŭtempan reton kun la tempoakso. La rezulton de la tempoanalizo ni povas peti en formo relativa aŭ kalendara. En la lasta kazo ankaŭ la akso estas laŭkalendara. /6-a bildo: detalo de laŭtempa reto./

La programaro HSZR-1 estas taŭga unuavice por prilabori retojn kun subretoj. Ni desegnas la unuopajn subretojn / maksimume 300 agadoj/ unuope. La interligon inter la unuopaj subretoj certigas la interligaj rilatoj.

Genero de enigaj kartoj sur magnetan diskon

Per HSZR-1 eblas generigi la enigajn kartojn por la necesaj agadoj. Kun la desegnado de subretoj ĝi iĝas konvena por trakti retoplanojn konsistantajn el plurmilojn da agadoj reprezentantaj subretoj.

Enigaĵo de HSZR-1

Unue oni devas fari la liston de la numerigitajn agadoj. Poste por ĉiuj agadoj ni indikas la vicnombron de la ĵus sekvantaj Ekz.: Post la kvara agado tuj sekvas la 7a, 8a,

13a, 18a kaj la 23a agadoj. Ni devas ilin kodigi

4	7	8	13
4	18	23	

Sur unu karto estas tri logikaj rilatoj. Estas pli bone indiki ankaŭ la ĵus antaŭan ankaŭ la ĵus postan agadojn. Ekz.: la 34-an agadon ĵus antaŭas la 24a kaj la 31a agadoj. Kodigo: -34 24 31

La antaŭadon ni signas per "-". Post la kodigo ni povas ankaŭ nomi la agadojn kaj se ni volas tempoanalizon, ni devas doni ankaŭ la tempajn datumojn. HSZR-1 skribas en la reto ankaŭ la 24 karakteran mallongigon de la agado. Post la agadoj ni povas doni la unuojn de la tempo. /Ekz.: TAGO, SEMAJNO, .../

Eligaĵo de HSZR-1

La eligaĵo aperas sur linipresilo.

Sur la unua paĝo estas skribita la titolo de la tasko, la nombro de agadoj kaj de nefaktaj agadoj, nombro de logikaj rilatoj kaj eventoj.

Ekde la dua paĝo skribiĝas la detala nomo de la agadoj, kaj la respondeca organizaĵo / kvar steloj inter ili/ partigite 25-ope.

Poste sekvas resuma tabelo. En unu linio estas: la 24 karakterna mallongigo de nomo, la vicnombro de la ĵus antaŭa kaj la posta agadoj, daŭro de la agado la plej frua fino kaj la tuta rezerva tempo.

En kazo de tempoanalizo post la supraj tabeloj la programo skribas la tempon de la laboro. / la longon de la kritika vojo/

Kaze de tempoanalizo MPM poste sekvas la resuma tabelo pri la rilataj distancoj.

Desegnado de logikaj retoj: La programo HSZR-1 presas la unuan 160 /aŭ 120/ karaktrajn longan vertikalan zonon de la reto. Poste ĝi iras al la komenco de la sekvonta folio. El la presitaj folioj estas facile kunmeti la reton. Sur ĉiuj paĝoj estas skribitaj la numero, la titolo de la tasko kaj la dato de la prilaboro. /1a kaj 2a bildoj/

La desegnado de la laŭtempa reto okazas simile. /3a bildo/

Avantaĝoj kaj malavantaĝoj de perkomputila desegnado de reto

La plej granda avantaĝo estas la redukto de la permana laboro kaj la rapideco. La tempo por pretigi la reton malpliiĝas je 40 %-j. Teĥnika desegnisto ne estas bezonata. Eblas rapide, facile modifi. Tio estas tre grava ĉe aktualigo, ĉar tio necesas ĉirkaŭe kvaronjare. Se ŝanĝiĝas la logiko de la reto, necesas rekonstrui kaj redesegni la reton. Sen komputilo tio estus tro multe da tempo.

La malavantaĝo de la perkomputila retodesegnado estas la pli granda reto. Tion ni povas faciligi per dispartigo en subretojn. Malfacile solvebla tasko estas la vertikala dispartigo de la reto. La desegnaĵo sur la presita folio ne estas samnivela kun iu farita far desegnisto. Tiun ĉi problemon eblas faciligi per desegno-maŝino, aŭ per la redesegnado de la presita folio far teĥnika desegnisto.

RESUMO

La programo HSZR-1 kapablas desegni en unu paŝo subreton konsistantan maksimume el 300 agadoj en la formoj CPM aŭ MPM. El tiuj subretoj ni povas konstrui eĉ reton konsistantan el pluraj miloj da agadoj per la sistemo de interligaj agadoj.

La programaro HSZR-1 tute eligas la permanan konstruon kaj desegnadon de reto el la tasko pri reto planado. Per ĝia uzo neniiĝas la laciĝa laboro de la enigo de datumoj. Tiel neniiĝas ankaŭ la enigaj eraroj.

La labortempo de HSZR-1 dependas de la komplikeco de la reto kaj de la nombro de la agadoj. La tempo lineare ŝanĝiĝas kun la grandeco de la reto.

Meze komplikan subreton konsistantan el 1200 agadoj la HSZR-1 konstruas, analizas la tempon kaj desegnas dum 7 minutoj.

Ĉe aktualigo de logika reto, la modifo, rekonstruo bezonas nur kelkajn minutojn, tiel per la uzo de HSZR-1 oni povas atingi, ke la logika reto ĉiam respegulu la aktualan situacion. La reto planada laboro iĝas kolektiva laboro, tiel la unuopaj ecoj de la personoj jam ne influas la rezulton.

La programo HSZR-1 estas skribita en la lingvo FORTRAN kaj povas esti uzata en ĉiuj mezaj komputiloj.

Nun ĝi laboras sur la komputiloj 1903/A, IBM360, R22, R40.

Literaturo

Kaufmann Desbazeille: La matematika bazo de kritika vojo.

Mk. 1972.

La ununura rajtanta por vendi HSZR-1:

NOVEX Találmányfejlesztő es Ertékesítő Külkereskedelmi RT

Budapest V., Március 15 tér 1 Telefono: 184-022

Telekso: 226054

Kodo de NOVEX: SW 9/39

BIOGRAFIO



Mi naskiĝis en Budapeŝto 1952. Mi finis la unuversitatajn studojn en 1977 kiel diplomita matematikisto.

Jam en la universitato ekinteresis min la komputistiko. Depost la ekkono de retoplanado estiĝis mia pli preciza intereskampoj: perkomputilaj problemoj de la retoplanado.

Ankaŭ en mia laborejo - NIM Instituto pri Industriekonomio kaj organizo - mi okupiĝas pri tio. La retoplanadan kaj desegnan programon HSZR mi faris dum du jaroj en 1978-79. Poste ni jam utiligas ĝn en la praktiko sukcese. Nuntempe mi laboras pri pri la prilaboro kompleta programaro pri retoplanado.

DISDONO DE LA ŜARĜO INTER TELEFON -SUBCENTRALAJ MANIPULANTOJ

GYULA CSÁSZÁR

BHG Telekomunika Entrepreno, Budapeŝto, Hungario

Resumo

La alvenantajn vokojn en la telefon-subcentralojn peras la manipulantoj al la flankaj stacioj. Dum la dimensiono oni devas konsideri krom la teknikaj vidpunktoj ankaŭ la interesojn de vokantoj kaj manipulantoj. Tiu artikolo solvas la problemon fare de modelo je starado en vico /vicada/, kaj demonstras la avantaĝojn de la tiel nomata "glatiga" vokdisdono, helpe de kalkulado kaj simulado. La matematikajn detalojn enhavas apendico.

1. Enkonduko

Dum la evoluigo de telefoncentraloj necesas taksi la kapaciton de la pretiĝanta sistemo. Per la ĝusta dimensiono de certaj partoj de centraloj pligrandigeblas la trafik-plenuma kapablo. La celo estas estu tia, ke la teknika realigo de la centralo estu tia, ke ju pli ĝi prezentu bonan liveradon. En tiu laboratorio ni okupiĝas pri la de elformado de la manipula sistemo, prezentita en la BHG Telekomunika Entrepreno dum la projektado de storita program-regadaj /SPR/ telefoncentraloj.

La SPR signifas la regadon de la telefoncentralo fare de ia komputileca aparato. La konstruon kaj la malkonstruon de la parol-interrilatoj regas la programaro de la komputilo. En la ceteraj la manipulanta sistemo signifas la kolektivon de aparatoj kaj proceduroj /aparataro kaj programaro/ necesaj por servi la alvenantajn vokojn, kune kun la manipulanta personaro.

Dum la planado de la manipulanta sistemo oni devas konsideri kaj la telefonabonantojn kaj la manipulantojn.

Gravas, ke la vokantoj ne devu atendi multon por la manipulado, aliflanke la ŝarĝo de la manipulantoj malpermesatas esti pli ol ia fiziologie ebla valoro. /Laŭ la sveda poŝtofiko maksimume 0.7 erlango po manipulanto [1] /

En la fakliteraturo estas nombraj ĝeneralaj metodoj kiuj okupiĝas pri ĉi-tipaj servadaj sistemoj [2]. Ankaŭ en la BHG oni okupiĝis pri kelkaj manipulaĵaj sistemoj, sed la esploroj por la elformigo de optimuma manipula sistemo el trafika vidpunkto komenciĝis nur samtempe kun la evoluigo de SPR-subcentraloj [3-4]

En la subcentraloj pretitaj ĝis nun oni adaptis diversajn manipulantajn sistemojn. Se ĉiuj manipulantoj estas okupataj, tiam la alvenantaj vokoj devigis atendi kaj oni devas informi la manipulantojn pri tio ke estas atendantaj vokoj. En okazo de subcentralo de reduktita kapacito kie estas nur unu manipulanto, oni povas signali la statojn de la alvenantaj linioj sur la sola manipula aparataro - ekzemple forme de lampo-tabulo. En tiu okazo ekzistas unusenca konformigo inter la alvenantaj linioj kaj la manipulaĵaj cirkvitoj. En grandkapacita subcentralo, en kazo de pluraj manipulantoj, ankaŭ realigeblas unusenca konformigo, sed tio ne estas laŭcela el pluraj vidpunktoj /ekzemple, la manipulantoj ne povas helpi unu la alian kaj ankaŭ dum malgranda trafika periodo necesas tiuj manipulantoj./ En tiu kazo oni aplikas tiel nomatan "koncentratan manipulado", kiu signifas, ke la manipulantoj konas la statojn de nur kelkaj alvenantaj linioj konektataj al ili - sed ne de ĉuj.

Dum la trafik-pintoj estas multaj servendaj vokoj, la atendada tempo estas longa, do la celo estas malŝarĝi la centralon per atendada trafiko, pro tio konekti ju pli malmultajn vokojn al manipulanto samtempe. Sed el alia vidpunkto estas pli laŭcela tio, se la manipulanto konas la statojn de pluraj linioj ĉar en ĉi tiu kazo li estas pli bone informata, kaj tiel li pli bone aktiveblas.

Tiuj estas kontraŭaj kondiĉoj, por retrovi la optimumon oni devas konsideri vidpunktojn trafikajn kaj psikologiajn.

Oni povas apliki diversajn strategiojn por disdoni la atendantajn vokojn al la manipulantoj. Sube ni detale konigas la manipulantan sistemon en kazo de duspeca disdona strategio. Pere de taksado do la rezultoj de la esploroj ni konstato la avantaĝojn de la tiel nomata "glatiga" vokdisdono.

2. Priskribo de la manipula sistemo

La ekzamenata manipula sistemo estas speciala sistemo de amasa servado /vicado/ enhavanta plurajn atendigajn vicojn. La servado ĝenerale ne okazas en ordo de alveno, kaj la distribuo de la periodo de servado proksimigeblas plej bone per logaritme normala distribuo.

La manipulantoj havas apartajn atendigajn vicojn, kaj ekzistas unu komuna vico en kiu la atendantaj vokoj estas disdonitaj ankoraŭ al neniu manipulanto /v. la unuan figuron/. En la komunan vicon alvenas la vokoj nur tiam, kiam pleniĝis ĉiu manipulanta vico. Se dumtempe malpleniĝas individua atendiga loko, tuj "trairas" en ĝin la unua voko el la komuna vico.

Kiam ekzistas neplenaj individuaj vicoj, tiam ni esploris duspecajn strategiojn de la envicigo:

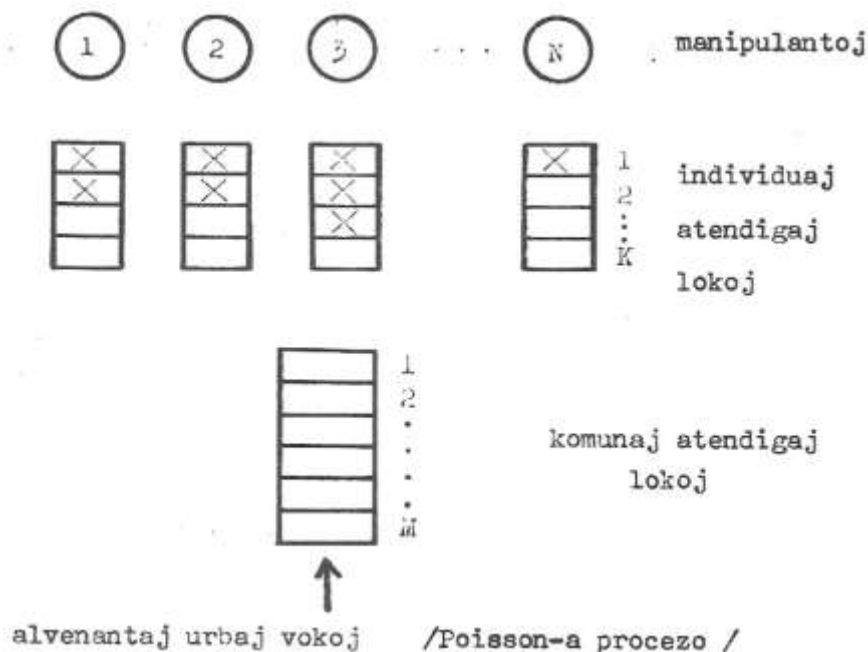
- HAZARDA strategio, kiam la rega programo elektas vicon hazarde /laŭ egalmezura distribuo/ el inter neplenaj individuaj, sendepende de la nombro de starantoj en tiu vico;
- GLATIGA strategio, kiam la alvenantaj vokoj "iras" al unu el la neokupitaj manipulantoj aŭ al unu el la plej mallongaj vicoj /se ekzistas pluraj tiaj vicoj, tiam ili elektas ankaŭ hazarde/.

/En ambaŭ kazoj tiuj vokoj perdiĝas, kiuj alvenas tiam, kiam la sistemo estas plena. En la centralo nur tiom da vokoj povas esti, kiom da konektanta linio - tiel nomata alvenanta linio - estas. Ĝis kiam ĉiuj estas okupataj, ne alvenas novaj vokoj./

La longecon de komuna vico ni determinos tiel, ke la probablo de la perdedo estu malpli granda ol fiksita $\epsilon > 0$.

En nia modelo ni supozas, ke la centralo estas senemberasa, nome, ke la regado kaj la konektilaro ne malhelpas la vokojn. Tio kondiĉeblas en kazo de la ĝenerale laŭleĝa unu procenta perdedo de la konektilaro.

Inter la malokupiĝo de manipulanto kaj la komenco de servado de la unua voko staranta en lia individua vico, forpasas tiel - nomata tempo. Ĝi dependas de homaj faktoroj, ekzemple de tio ke la manipulanto kiel rapide rimarkas ke voko alvenis, aŭ kiel rapide li komencas la servadon de la voko jam ĉeestanta,



1. figuro La manipula sistemo

La hazarda strategio realigeblas teknike pli facile, sed la glatiga disdono atendeble donas pli bonan liveradon.

Nome, en la ĉi-lasta kazo la vokoĵ ĉiam alvenas al la malplej okupita manipulanto, kaj tiel la larĝiteco de manipulantoj estos pli egalmezura. Intiucie estas kompreneble, ke la glatiga strategio estos pli bona konsidere ankaŭ la atendadajn karakterojn. Tion demonstras kaj la simuladaj esploroj, kaj la kalkuladoj.

3. Esploraĵ metodoj

3.1. La ĝeneralan sistemon oni povas esplori nur per komputila simulado, ĉar konvenaj kalkuladaj metodoj estas nekonataj. Ni faris simuladan programon SIMULA-67-lingve, kiu plenumblas en komputilo CDC 3300 de la Hungara Scienca Akademio.

Per la programo ni povas esplori la priskribitan sistemon. La rezultoj de simulado estas statistikecaj /konfidindecaj

intervaloj/. Por atingi statistikan ekvilibron, necesas observi multajn vokojn, pro tio la tempo de la program-plenumo estas relative longa.

Enigaj datumoj de la programo:

a/ sistempaj parametroj:

- nombro de la manipulantoj
- nombro de la individuaj kaj komunaj atendigaj lokoj
- maniero de la envicigo de voko /HAZARDA al GLATIGA/

b/ manipuladoj parametroj:

- tipoj de la distribuo de reaga kaj servada tempo /konstanta, eksponenciala, logaritme normala/ kaj ties parametroj /ekspekto, varianca devio/

c/ voka parametro estas la alvena intenseco /ĉiam supozante Poissonan procezon/

d/ plenumaj parametroj estas la nombroj de voko dum la prepariĝo kaj en la fazoj de observo

La rezultoj de la simulado:

Por ĉiu datumtipo ni gajnas

- averaĝojn po fazoj kaj por la tuta plenumo
- 95-procentajn konfidindecajn intervalojn

a/ ĝeneralaj simuladaj rezultoj:

- nombro de la servitaj pretendoj
- nombro de la perditaj pretendoj pro la plenigitaj vicoj
- la longeco de la periodo de la observo

b/ la finita trafiko po manipulanto

c/ karakteroj de la atendigaj vicoj /manipulantaj, komuna kaj tuta sistema/:

- distribuo de la nombro de atendintoj
- averaĝa atendada tempo
- trafiko atendada probableco /frekvencio/ de atendado, distribuo de la atendada tempo.

3.2. Ĉar la perkomputila simulado deziras multan tempon, supozante Markofan-modelon ni okupiĝis pri la solvado de sistemo de stat-ekvacioj bazitaj sur statistika ekvilibro.

Unue ni kalkulis per fermita formulo la stat-probablojn por okazoj de 2,3,4 manipulantoj. Ni supozis, ke en la individuaj vicoj povas atendi maksimume nur unu voko, ĉar la ĝis nun plan-

itaj sistemoj estis tiaj. La kalkulaj rezultoj montris, ke inter la atendataj tempoj gajnitaj per la dupecaj strategioj precipe en kazo de pluraj manipulantoj estas konsiderinda diferenco en avantaĝo de la glatiga disdono.

3.3. La solvado de stat-ekvacioj per fermita formlulo estas tre pezmovebla jam en kazo de 4 manipulantoj, pro tio ni prilaboris algoritmon por iom ajn la manipulantoj kaj individuaj atendigaj lokoj, sed nur por glatiga disdono. Kun la algoritmo pretiĝis ankaŭ programo. Pri la detaloj vidu la apendicon.

4. Rezultoj de la esploroj

Sube en tabeloj ni montras procente, per kiom kreskas la atendataj tempoj kontraŭ la glatiga okazo, se ni aplikas la hazardan metodon. En la tabeloj, R signifas la trafikon po manipulantoj kaj K signifas la nombron de la individuaj atendigaj lokoj. La unua tabelo koncernas la eksponenciale distribuitan manipulanadan tempon, sed la dua koncernas la pli realan logaritme normalan distribuon. La datumojn de la unua ni gajnis kalkule kaj simule, sed en la dua estas nur similitaj rezultoj.

R \ K	1	3	5
0.5	168	432	---
0.6	92	244	---
0.7	49	168	202

1. tabelo Procenta kresko de la averaĝa atendada tempo en kazo de la hazarda strategio relative la glatigan. La sistemo havas 4 manipulantojn, kaj la servada tempo estas eksponenciale distribuita /ties parametro estas la unuo/.

R \ K	1	3	5
0.7	63	212	263

2. tabelo Procenta kresko de la averaĝa atendada tempo en kazo de la hazarda strategio relative la gla-

tigan. La sistemo havas ankau 4 manipulantojn, sed la distribuo de la servada tempo estas logaritme normala. /Ties karakteroj estis $M=15.9$ -ekspekto- kaj $\sigma = 0.1$ -varianca devio- kalkulitaj el datumoj de mezuradoj okazintaj en BHG en 1968./

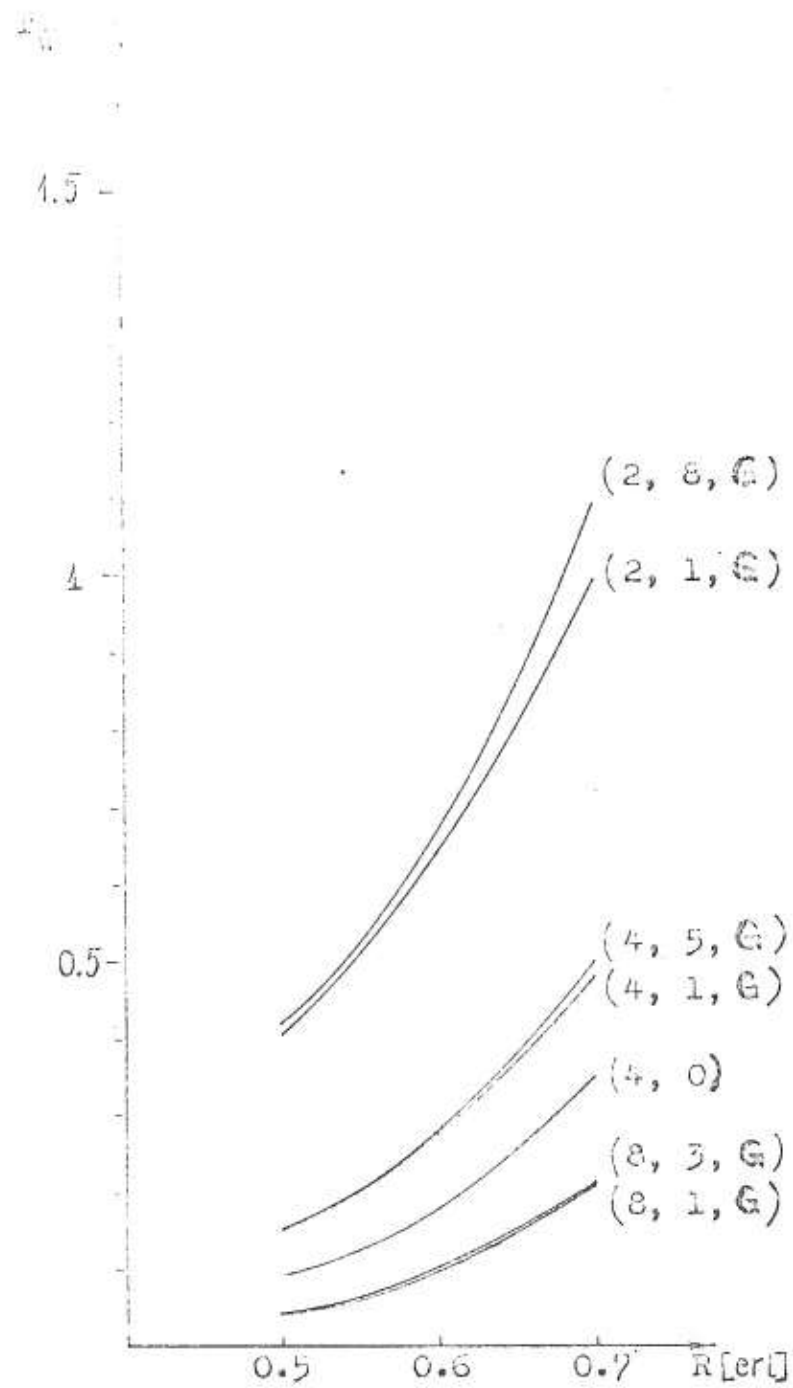
El la tabeloj videblas, ke la glatiga disdono estas konsiderinde pli favora. Krome estas ankaŭ favora kvalito, ke la probablo de la atendado estas malpli olanda ol eĉ tiu de la tute atingebla fasko $/Z=0/$, en la kazoj de ambaŭspeca distribuo. Eĉ tiu probablo malgrandigas, se la longeco de la individuaj vicoj kreskas plu.

Sur la dua figuro videblas, ke fiksinte la nombron de la manipulantoj kaj aplikinte la glatigan strategion dum la kreskigo de la longeco de vicoj, la atendada tempo pligrandiĝas nur nuance. Do ne esencas, kiom da atendiga loko estas ĉe la manipulantoj. La noto $/2,8,G/$ signifas tiun sistemon kun glatiga disdona strategio, en kiu estas 2 manipulantoj kaj 3 individuaj atendigaj lokoj. Ni notis per $/4,0/$ la tute atingeblan sistemon, kiu havas nur komunan vicon /kaj pro tio ekzistas nur unu strategio/.

Krome la sistemo kun glatiga stratezio havas tiun favoran kvaliton, ke ĝi estas pli bona ol ĉiu ajn hazarda sistemo havanta samnombrajn manipulantojn. Tio estas, se $/N,K_G,G/$ notas glatigan sistemon kun N manipulantoj kaj K_G individuajn lokojn, tiam validas, ke ĝi estos pli bona ol ĉiu sistemo $/N,K_h,H/$ por laŭplaca $K_G / K_h = 1,2,\dots$

Laŭ la simulado tiu aserLo ankaŭ pravas por logaritme normala distribuo.

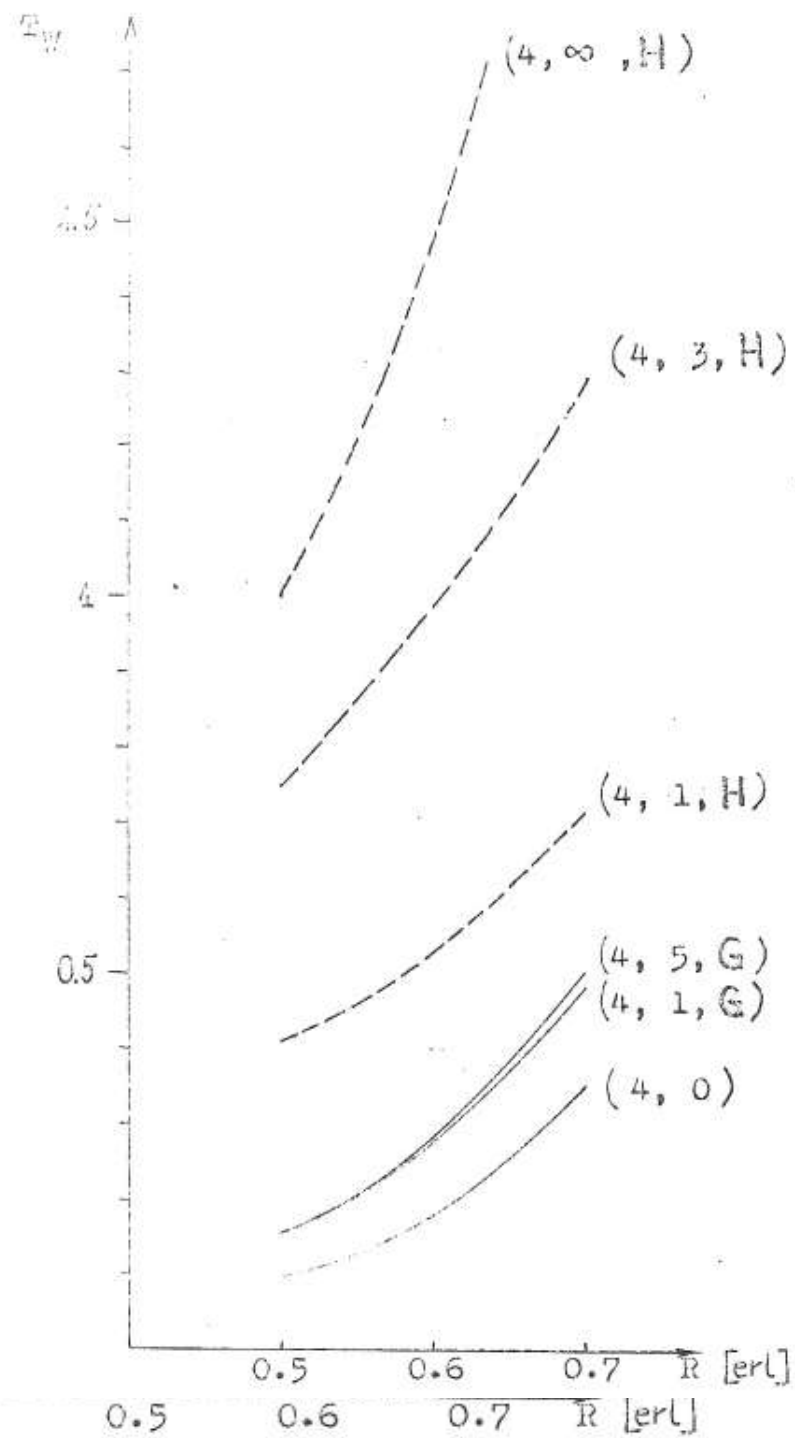
Por komparo, sur la tria figuro videblas la efiko de la parametro K je ambaŭ strategioj, kiam estas 4 manipulantoj en la sistemo. La datumoj de la sistemo $/4,\infty,H/$ dedukteblas el la modelo signita per $H/M/1$ en la fakliteraturo. En ĉi-lastata sistemo estas nur unu servisto, kaj senfine multaj atendigaj lokoj.



2 figuro Formiĝo de averaĝa atendada tempo en kazoj de diversnombraj manipulantoj kaj individuaj atendigaj lokoj, aplikante la glatigan disdonon

T_w : Ekspekto de la atendata tempo

R : Trafika ŝarĝo po manipulanto



3. figuro Formiĝo de averaĝa atendada tempo je ambaŭ stratcgioj en kazoj de 4 manipulantoj kaj diversnombraj atendigaj lokoj.

T_w : Ekspekto de la atendada tempo

R : Trafika ŝarĝo po manipulanto

5. Konkludoj

La rezultoj de la esploroj subtenas aplikon de la glatiga vok-disdono, ĉar

- 1./ Se oni aplikas tiun strategion, la karakteroj de la atendado konsiderinde pliboniĝas.
- 2./ Fiksita sistemo kun glatiga metodo estas pli bona ol ĉiu sistemo kun hazarda metodo en la kazo de samnombraj manipulantoj.
- 3./ La nombro de individuaj atendigaj lokoj praktike ne influas la sistemkarakterojn. La atendadaj karakteroj praktike ne ŝanĝiĝas, pro tio pli elaste konsidereblas kromaj vidpunktoj dum la dimensiumado.
- 4./ La probablo de la atendado estas malpli granda ol eĉ tiu de la tute atingebla fasko eĉ kun la kreskado de longo de individuaj vicoj iomete plu malkreskiĝas.

Konsidere ĉi tiujn avantaĝojn, estas motivita la ĝenerala enkonduko de la glatiga vok-disdona strategio.

APENDICO

Sube ni supozas Markofan modelon: kaj la alvenantaj intervaloj kaj la servadaj tempoj estas eksponenciale distribuitaj.

1. Priskribo de la statoj

La sistemon en statistika ekvilibro oni karakterizas per la probabloj de la statoj. Dun la priskribo de la eventualaj statoj ni senkonsideras la komunan vicon, ĉar el karakteroj de sistemo, sen ĝi, oni povas elkalkuli facile la probablojn en kazo de laŭplaĉe longa komuna vico.

En la ceteraj pro simpleco ni inkluzivas la servatajn vokojn en la nombron de tiuj, kiuj atendas en la individuaj vicoj. Laŭ tio estu $K:=1$ la nombro de la individuaj lokoj, kaj se ni parolas pri j -longa vico, tiam ĝustadire tio signifas nur $j-1$ atendantojn, okaze de $j > 0$.

Ni karakterizas la statojn per tio, ke kiom da peco estas el diversaj vicoj:

Notu i_j la nombron de j -longaj vicoj / $j=0,1,\dots,K$ /, tiel ni gajnas / $K+1$ -elementajn vektorojn:

$$I = (i_0, i_1, \dots, i_K) \quad \text{Kompreneble}$$

$$0 \leq i_j \leq N \quad \text{kaj} \quad i_0 + i_1 + \dots + i_K = N \quad (1)$$

La nombro de tiuj vektoroj estas $\binom{N+K}{N}$, sur la 4-a figuro ni montras ĉiujn 20, okaze de $N = K = 3$.

2. Karakterizo de la transiroj inter statoj

Sur la kvara figuro ni montris per kontinuaj sagoj la eventualajn transirojn je alveno de novaj vicoj, kaj per mal-kontinuaj sagoj la transirojn je finiĝoj de servadoj.

Notu λ la alvenan intensecon, kaj μ la finiĝantan, tiam λ/μ estas la oferita trafiko al la sistemo. Por simpligi, ni estu $\mu=1$.

Nun ni esploras, kiel oni povas karakterizi la transirojn helpe de la koordinatoj.

- Transiroj je finiĝoj

Tiujn ni povas klasifiki laŭ la longo de vico, en kiu finiĝis la servado. Se tio estis j -longa, tiam estos per unu pli multa / $j-1$ -longa, kaj per unu pli mallonga j -longa vico. La tiel rezultitan vektoron el vektoro I , notu $\hat{E}_{j-1/I}$.

La inverson de tiu transformo notu $f_{j-1/I}$:

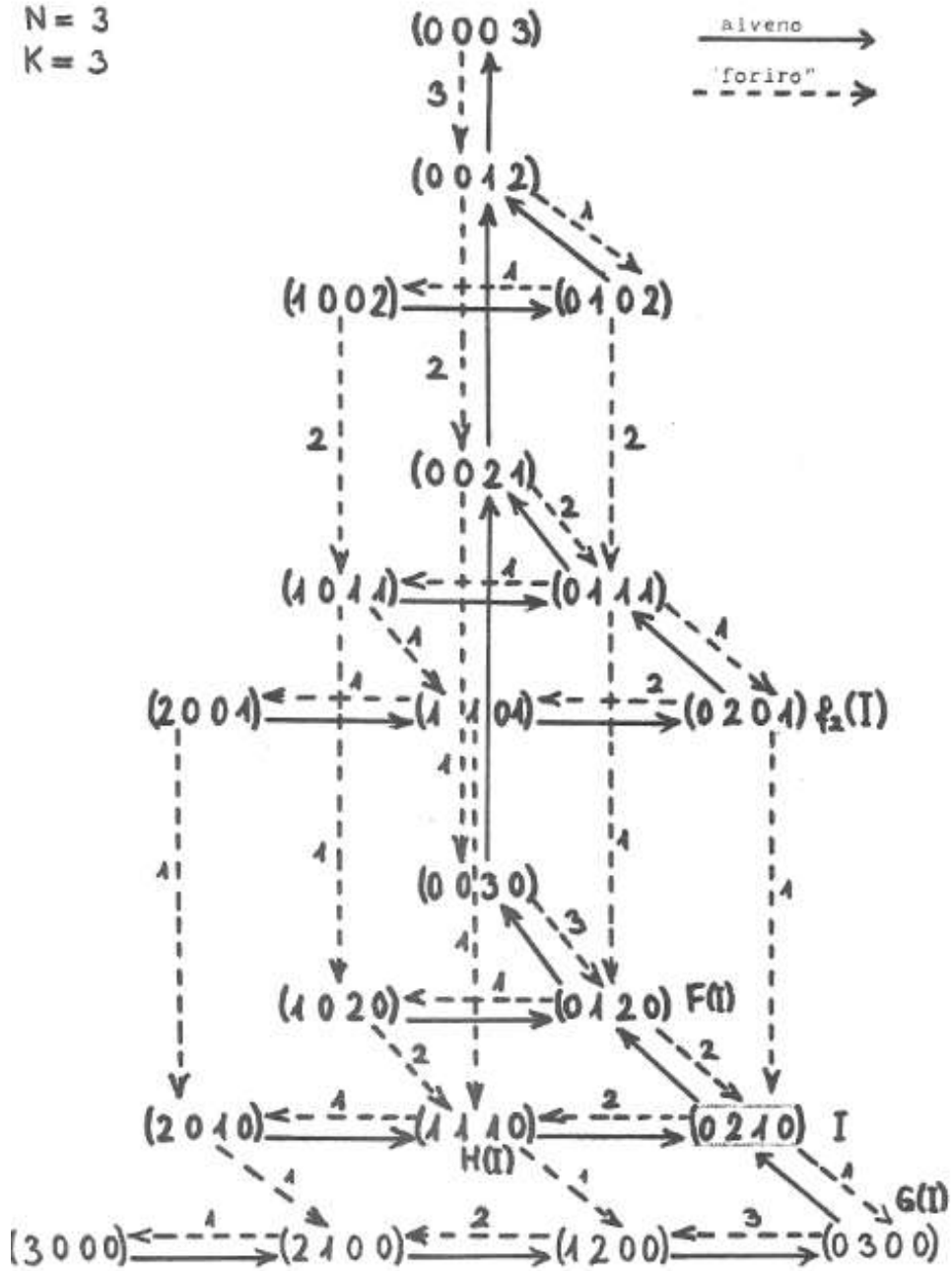
devas esti kreskata la j -a koordinato per unu, kaj mal-kreskata la $(j-1)$ -a.

- Transiroj je alvenoj

Estu $m = m/I$ la longo de la plej mallonga vico, la nova vico enviciĝas en ĉi tiun, pro tio la nova stato estos $F/I = f_{m/I/I}$: laŭire de maldekstro dekstren, la

vektoron I , la unuan pozitivan elementon oni devas mal-kreskigi, kaj la sekvan oni devas kreskigi per unu.

$N = 3$
 $K = 3$



4. figura La spaco de statoj

Necesas ankaŭ la inverso de transformo F , kaj kiel pri-
penseblas, ekzistas maksimume nur du vektoroj, por kiuj
 $F/J/ = I$, nome

$$J_1 = G/I/ = g_m/I/ \quad \text{kaj} \quad J_2 = H/I/ = g_{m-1}/I/$$

Poste oni povas skribi la ekvacion por stato I [5] :

la sumo de ĉiuj elirantaj intensecoj multiplike per pro-
bablo de la stato egalas la sumo de termoj, kiuj estas la
produktoj de eventualaj enirantaj intensecoj kaj λ -
konvenaj statojprobabloj:

$$\begin{aligned} (\lambda + N - i_j) \cdot p/I/ &= \lambda \cdot \left\{ p(G/I/) + p(H/I/) \right\} \\ &+ \sum_{j=1+m/I/}^K (1+i_j) \cdot p(f_{j-1}/I/) \end{aligned} \quad (2)$$

Oni povas noti tiujn ekvaciojn por ĉiu stato, tiel ni
gajnas sistemon de linearaj ekvacioj konsistantan el

$\begin{bmatrix} N+K \\ N \end{bmatrix}$ ekvacioj, kaj tiun sistemon ni devas solvi.

Por noti la matricon de la sistemo de ekvacioj, oni de-
vas konstati ian ordon de la statoj, kiu povas esti laŭplaĉa,
sed oni povas trovi tian, kiu havas certajn avantaĝojn.

3. Ordigo inter la statoj

Konsideru du statvektorojn I, J . Ni diru ke I estas pli
malgranda ol J ($I < J$), se $i_K < j_K$ aŭ $i_K = j_K$ sed $i_{K-1} < j_{K-1}$
aŭ ... sed

Ĝustadire realiĝas leksikografika ordigo: la koordinato
havanta pli grandan indicon, havas pli grandan "lokan valo-
ron".

En la tiel ordigita spaco, la transiro fare de alveno ĉi-
am direktiĝas al pli granda stato, la transiro fare de fin-
iĝo ĉiam direktiĝas al malpli granda stato /vidu la kvaran
figuron!/. El tiu sekvas, ke la matricoj estas "preskaŭ" sup-
ra triangula matricoj, ĉar

- sub la diagonalo ekzistas nur unu elemento diversa de 0

- ĝia valoro egalas λ
- plej ofte ĝi situas senpere sub la diagonalo.

4. La matrico de la ekvaciaro

Notu $M_N^{(K)}$ la matricon, $P_N^{(K)}$ la vektoron de la stat-probabloj, kiu estas solvo de ekvaciaro $M_N^{(K)} \cdot P_N^{(K)} = 0$. Tiu sistemo determinas nur la kvocientojn de vektor-elementoj, ĉar la sumo en vertikaloj egalas nulo.

Pro tio ni elektu je unu la "probablon" de la plej granda stato, kaj adiciu $-\lambda$ -al ambaŭ flankoj de la lasta ekvacio. Estu $M_N^{(K)}$ la tiel modifita matrico! La nova ekvaciaro estas:

$$M_N^{(K)} \cdot P_N^{(K)} = \left[\begin{array}{c} 0 \\ \vdots \\ 0 \\ -\lambda \end{array} \right] \left\{ \begin{array}{c} N+K \\ N \end{array} \right\} \text{-elementa vektoro} \quad (3)$$

En la lasta vertikalo de $M_N^{(K)}$ la sumo estas $-\lambda$, kaj pravas la suba aserto:

Se $I_N^{(K)}$ notas la $\begin{pmatrix} N+K \\ N \end{pmatrix}$ -ordan matricuniton, kaj

$$M_0^{(K)} = M_N^{(0)} = (-\lambda) \quad (4)$$

tiam validas la sekva rikura konstruado /v. la 5-an figuron/

- unue oni metas blokojn sur la diagonalon. La m -a bloko egalas $M_{N-m}^{(K-1)} = m \cdot I_{N-m}^{(K-1)}$ / $m = 0, 1, \dots, N$ /
- poste super la m -an blokon oni metas samordan diagonalan matricon : $m \cdot I_{N-m}^{(K-1)}$ -n./sed ĝi estas malpli granda ol la $(m-1)$ -a bloko estanta maldekstre de ĝi/
- finfine apud la maldekstran angulpinton de m -a bloko ni metu $\lambda - n$.

La ekvaciaron (3) ni solvas tiel, ke fariate eliminadon unue iĝas $M_N^{(K)}$ vera supra triangula matrico, poste reprogresante de la "plej granda" stato, substituas la gajnatajn valorojn, kaj tiel kvazaŭ rulvolvas la sistemon de ekvacioj.

$$\begin{aligned}
 & \left[\begin{array}{c} M_N^{(k-1)} \\ \lambda \left| \begin{array}{c} I_{N-1}^{(k-1)} \\ M_{N-1}^{(k-1)} - I_{N-1}^{(k-1)} \end{array} \right. \\ \vdots \\ \lambda \left| \begin{array}{c} m \cdot I_{N-m}^{(k-1)} \\ M_{N-m}^{(k-1)} - m \cdot I_{N-m}^{(k-1)} \end{array} \right. \\ \vdots \\ \lambda \left| \begin{array}{c} (m+1) \cdot I_{N-m-1}^{(k-1)} \\ M_{N-m-1}^{(k-1)} \end{array} \right. \\ \vdots \\ \lambda \left| \begin{array}{c} N \cdot I_0^{(k-1)} \\ M_0^{(k-1)} - N \cdot I_0^{(k-1)} \end{array} \right. \end{array} \right] \\
 & M_N^{(k)} =
 \end{aligned}$$

15

5. Kulvolvado de la ekvaciaro

Sur ambaŭ flankoj de (3) ni plenumas la saman transformigojn. Ĉar la elementoj de dekstra vektoro egalas nulon, escepte nur unu, pro tio la ŝanĝoj aperas nur en la maldekstra matrico.

Adiciu la horizontalojn ĝis tiu, kiu formas la lastan de la m -a bloko. Tiam en ĉi tiu horizontalo

- la elementoj antaŭ la diagonalo egalas nulon
- en la diagonalo estas $(-\lambda)$
- post la diagonalo, super la $(m+1)$ -a bloko la elementoj egalas $m+1$, kaj post ili denove estas nuloj.

Per tiuj transformigoj /por $m=0,1,\dots,N/$ ni modifis la ekvaciaron tiel, ke en ĉiu ekvacio estas nur nekonatoj apartenantaj al maksimume du blokoj. Helpe de tio oni povas ŝpari per storo en programo /ne necesas stori ĉiujn nekonatojn samtempe/.

Per la ĵus menciita transformado ni atingis ĝustadire, ke la eliminado de subdiagonalaj elementoj estas simpla en kazo de tiuj statoj, por kiuj per iu " m "

$$i_m + i_{m+1} = N \quad (5)$$

Ĉar ni gajnis, ke por $m=N-1$

$$p/I/ = \frac{1 + i_{m+1}}{\lambda} \cdot P \left\{ \text{numero de } (K-1)\text{-longaj vicoj} = 1 + i_m \right\} \quad (6)$$

kaj mem la maldekstra supera bloko de $M_N^{(K)}$ ankaŭ estas samstruktura matrico, laŭ la antaŭa metodo oni povas demonstri, ke

$$p/I/ = \frac{1 + i_{m+1}}{\lambda} \cdot P(I, F/I/) + \frac{1}{\lambda} \cdot P(F^i_m/I/, f_{m+1}/I/) \quad (7)$$

/(,] estas maldekstre malfermita intervalo laŭ la enkondukita ordigo en la tria alineo/.

Fluaj esencaj rimarkoj prezentiĝas el (2), se mankas termino koneksa kun $C/I/$ aŭ $H/I/$.

a/En kazo de tiuj statoj, en kiuj estas liberaj manipulantoj

/ $i_0 > 0$ / mankas $H/I/$ kaj tiam $p/I/$ esprimeblas tiel:

$$p/I = \alpha/I \cdot p(f_0/I) + \sum_{j=2}^K \sum_{r=0}^{i_1} \beta_r^{(j)}/I \cdot p(g_0^r f_{j-1}/I) \cdot (1+i_j) \quad (8)$$

/Rimarko: dekstre troviĝas nur probabloj de tiuj statoj kiuj estas pli grandaj ol I, do (8) estas konvena por la menciita celo - por transformigo je supra triangula matrico./

La koeficientoj α, β kalkuleblas el rikura formulo gajnebla el (2), kaj esprimeblas helpe de polinomoj Brockmeyer-aj

$$w_r^{(v)} = \sum_{s=0}^r (v-1+s) \cdot \frac{\lambda^{r-s}}{(r-s)!} \quad \text{tiel:}$$

per $v = v/I = i_2 + \dots + i_K$

$$\alpha/I = \frac{w_{i_1}^{(v)}}{w_{1+i_1}^{(v)}} \quad , \quad \beta_r/I = \frac{w_r^{(v)} \cdot w_{i_1-r}^{(v)}}{w_{1+i_1}^{(v)} \cdot \binom{i_1}{r} (1+i_1)} \quad (9)$$

/ β_r/I ne dependas de j /

b/ La kalkuledon de okazo $K=2$ oni povas fari laŭ a/ punkto, ĉar tiam la statoj estas tri-elementaj vektoroj, kaj kie $i_0=0$ tie $i_1+i_2=N$, do (5) efektiviĝas.

c/ Se estas nur du manipulantoj, $N=2$, $M_N^{(K)}$ konsistas nur el tri blokoj:

- La dekstra suba "bloko" konektas kun la "plej granda" stato, kies probablon ni egaligis 1;
- La meza bloko, por kiu pravas, ke en (2) mankas G/I /tio validas ĉiam por la $(N-1)$ -a bloko /;
- Finfine la maldekstra supra bloko, kiu estas pli malgranda, sed samstruktura matrico, kaj pro tio ankaŭ por ĝi facile plenumblas la transformigo je supra triangula.

In la ĝenerala kazo $(N > 2, K > 2)$, la interrilatoj (5) - (9) okazigas esencajn simpligojn, sed oni devas daŭrigi la eliminadon.

Referencoj

- [1] Anderberg, M. - Arestig, M.:
Studies on Human Factors in a PABX
Proceedings of 7th International Symposium on
Human Factors in Telecommunications, 1974.
Montreal, Canada pp. 1-16.
- [2] Störmer, H.:
Verkehrstheoretische Grundlagen für die Planung von
handbedienten Vermittlungsanlagen
A.N.C. 12 [1958], 119-126.
- [3] Ágostházi, M. - Szádeczky-Kardoss, T. - Konkoly, L. -
Császár, Gy.:
Internaj raportoj en EHG 1979/80.
- [4] Konkoly, L. - Császár, Gy.:
Prelego en la X-a Hungara Operacio-serĉada Konferenco,
samtitola kiel tiu ĉi artikolo
Debrecen, 1980.
- [5] Kleinrock, L.:
Queueing Systems I.
John Wiley & Sons, 1975.

Uzitaj fakvortaroj:

- Plurlingva fakvortaro de konceptoj pri entrepren-
direktada komputil-aplikado
- SZÁMOK 1972. Budapeŝto
- Matematika terminaro Dean, C. UEA 1954.
- Kelkaj konsideroj pri matematikaj terminoj en Esperanto
Olav Reiersol Universitet ,Blindern, Oslo 1963.
- - Hungara kompletigo al EK-vortaro de matematikaj
terminoj Budapeŝto, 1980.

Glosaro

erlango - unuo de la trafiko, egalas la produto de alvena intenseco kaj averaĝa servada tempo

leksikosrafika ordigo

- kutima ordiso de termoj en polinomo de pluraj nedeterminatoj

loka valoro

- alivorte signifanco. En nombrosistemoj la signifo de cifero dependas de loko kie ĝi staras

sistemo de amasa servado, servada sistemo, starado en vico, vicado

- La modeloj de \sim konsistas el alvena stokastika procezo de deziroj /vokoj, aĉetantoj, ktp./ kaj unu aŭ pluraj servistoj /manipulantoj, linioj, vendistoj, muntistoj, ktp./. La servada tempo de dezirantoj estas ĝenerale ankaŭ stokastika variablo. La matematikaj esploroj okupiĝas pri la kvantecaj probablecaj karakteroj de la sistemo / atendado, perdecado, ktp./.

Dankado

La aŭtoro esprimas dankon al sia estro , sen kies instigeto ne naskiĝintus tiu ĉi laboraĵo.

Pri la aŭtoro



Császár, Cy. estas 27 jaraĝa matematikisto. Li fariĝis diplomita en 1978, de tiam li laboras en la Instituto pri Evoluigo de la BHG Telekomunika Entrepreno, kie li okupiĝas pri la trafika dimensiumado de telefoncentraloj. Nun la unuan fojon li publikas fremdlingve. Ĉi-jare li plenumos ŝtatekzamenon en la Universtato Eötvös Loránd, en interlingvistika fako. Li kolektas poŝkalendarojn, ŝatas popoldanci kaj precipe ŝakludi.

Aŭtomata telekomunika sistemo regata de komputilo IBM S/7,, instalita ĉe Hungara Meteologia Organizaĵo

Csanád ENDREDI

Resumo: La konigota telekomunika sistemo estas aŭtomata direkta sistemo por transigi meteologiajn mesaĝojn fare de la regado de komputilo IBM S/7. Mi skizas la aŭtomatigan taskon, mi konigos la specifaĵojn de la aparataro, kaj flue mi prezentos la telekomunan programaron, kiun hungaraj fakuloj evoluigis apogintaj sin per baza programaro de IBM.

Enkonduko

La meteologiaj instancoj faras la prognozojn, la esplorojn surbaze de la observaj datumoj, kiujn oni mezuras en ĉiuj landoj kaj kiujn oni cirkuligas en telekomunika sistemo.

Por fidinde kolekti, interŝanĝi kaj distribui la observajn datumojn kaj por transigi la materialojn prilaboritajn de la meteologiaj mondaj kaj regionaj centraloj la Tutmonda Meteologia Orgsrizajo kreis la Tutgloban Telekomunan Sistemon.

La Novaĵcentralo ene de la Centra Prognoza Instituto de la Hungara Meteologia Organizaĵo estas ero de la Tutgloba Telekomunika Sistemo, tial ĝi havas la sekvajn farendaĵojn:

- funkciigi la mezrapidecajn datumtransigajn liniojn de Praha-Budapest, de Budapest-Bucuresti
- priservado de la reto ebliganta la trafikon de la ĝeneralaj meteologiaj datumoj kaj de la cirkvitoj liverantaj la aviadmeteologiajn datumojn
- regado de la enlanda teleksa reto

Por la aŭtomatigo de la telekomunikado Hungara Meteologia Organizaĵo aĉetis komputilon, tipo S/7. Ĝi estas por la aŭtomatigo de farendaĵoj, krome por krei datumaron je 24 horoj. Ĉar mi ne havis la eblecon de instalado de dua

komputilo, ni trovis ekipaĵojn enlande fabrikitaĵn, per kiuj oni povas vivteni la kontinuan linian konekson.

La fakuloj de la Hunjara Meteologia Organizaĵo zorgis pri la aparatara kaj programara kunmetoj inter tiuj ekipaĵoj kaj la komputilo IBM S/7.

La aparataro

Mi skize konigas la aparataron de IBM S/7, per kiuj ni povis atingi la aŭtomatigon de la telekomunika sistemo /figuro I./.

- I ekz. CFU /centra prilabora unito/, tipo 5010 E32
- 64 kilookbitoj da interna storo
- 2 ekz. da diskoj kun po 5 Megaokbitoj da kapacito, tipo 5022 M 02. La operaciada sistemo de IBM S/7, la programoj de la telekomunikada programaro kaj la meteologiaj mesaĝoj /bultenoj/ estas storataj sur ĝi.
- I ekz. digitala eniga/eliga modulo, tipo: 5013 A 0I, kiu manipulas la mezrapidajn datumtransigajn liniojn.
- 2 ekz. da telekomunikadaj linikroĉaj/multipleksaj moduloj, kiuj povas manipuli po 16 duonduoblajn liniojn funkciantaj en ek-halt reĝimo /start-stop/ tipo: 5098 N 0I. La duoblajn liniojn bezonas 2 poziciojn.
- 1 ekz. manipulista aparato /konzola tajpmaŝino/, tipo: 5028, ĝi ebligas la komunikadon por la manipulisto kun la komputilo en la IBM S/7 operaciada sistemo kaj en la telekomunika sistemo.
- I ekz. matrica presilo, tipo 7431, kiu aperigas la listaĵojn de la translatado, la rezultaĵon de la programa efektivilo aŭ la alvenintajn bultenojn.
- 1 ekz. karto-truigilo/legilo, kiu kapablas funkcii en la reĝimoj ĉu kunkroĉa /on-line/ ĉu senkroĉa /off-line/.
- 1 ekz. konveniga unito kapablanta akcepti 32 teleksajn liniojn, kiu povas labori en 4 reĝimoj, depende de la permana kroĉado inter la enigaj-eligaj punktoj:
 - aŭtomata, senhoma reĝimo; la IBM S/7 akceptas respektive direktas la trafikon de la telekomunika kanalo

- permana reĝimo; la teleksoj akceptas respektive direktas la trafikon de la telekomunika kanalo /la komputilo ne funkcias/
- aŭtomata reĝimo kun manipulista atento; la IBM S/7 direktas respektive akceptas la trafikon de la telekomunika kanalo. Oni povas atenti la trafikon per la alligata pasiva teleksa aparato.
- loka reĝimo; ĝi estas sendependa de la telekomunika kanalo, inter la IBM S/7 kaj lokaj teleksaj aparatoj
- modema /modulilo-malmodulilo/ konvenigilo, kiu konvenigas la mezrapidajn - 1200 bit/sekundo liniojn al la digitala eniga/eliga modulo. Ĝi havas duoblan taskon:

-permana reĝimo: la kontrolo de signaro, kiun oni prenas sur la kvardrata telefontipa cirkvito kaj malmodulas, okazas en la erardefenda unito laŭ la preskriboj de la Manlibro de Tutgloba Telekomunika Sistemo. Ĝi produktas "puran informon" por la truigilo kaj ĝi same atentigas "puran informon" de la trubendlegilo, kiun ĝi transigas kompletigante per la erardefendaj preskriboj tra la modulilo al la adekvata cirkvito. En tiu ĉi reĝimo IBM S/7 ne havas rolon en la direktado de la trafiko.

-aŭtomata reĝimo: la datummanipulado estas simila al la antaŭe skribita. Tio estas la diferenco, ke la konveniga unito kroĉas la enigajn/eligajn punktojn de la erardefenda unito ne al la truigilo kaj la trubendlegilo, sed al la adekvataj digitalaj enigaj/eligaj punktoj de la IBM S/7.

La supre konigita aparataro ebligas krom la aŭtomata reĝimo antaŭ la tradician permanan regimon, kio estas bezonata pro la manko de la dua komputilo.

Ĝenerala konigado de la telekomunika programaro

El la eco de la aŭtomatigata sistemo doniĝas, ke la programaro estas samtempa /real-time/, ĉar la bultenoj devas esti prilaborataj tuj ĉe ilia alveno. Ni organizis

tiam dispartigitan sistemon pro programadaj kialoj, kie troviĝas ses partoj de diversaj grandoj en la interna storo krom la rezida parto. La dimensioj de la partoj estas fiksj, ni dimensiumis ilin laŭ la programoj en ili efektiviĝantaj.

La bultenoj, kiuj partoprenas en la trafiko regata de IBM S/7, devenas el la orientaj partoj de Nord-Ameriko, el Gronlando, el la teritorio de Atlantiko kaj Norda Maro, el Nord-Afriko, el Proksima-Oriento kaj el Eŭropo. La bultenoj alvenas en norma formo preskribita en la manlibro de la Tutgloba Telekomunika Sistemo sur la mezrapidaj linioj en la kodsistemo CCITT No 2 kaj sur la teleksaj linioj en la kodsistemo CCITT No 5. La programaro konvertas la bultenojn en la ASCII kodon de senpara parito por la unueca traktado. La bultenoj alvenintaj en norma formo eniras en la datumbazon kaj ili povas esti reserĉataj dum 24 horoj. Bultenoj de ne norma formo aperas sur la manipulista stacio por esti korektataj, ili reiros en la trafikon post la korekto. Samtempe kiam oni enigis bultenojn en la datumbazon, oni devas post la selektado la bultenojn en norma formo transigi, tra la diversaj datumtransigaj linioj aŭ al la prilaborantaj programoj /figuro 2./.

Ne nur bultenoj alvenintaj el la enigaj linioj iros en la trafikon, sed ankaŭ tempfiksitaj programoj aŭ la rezultoj de la programoj efektiviĝantaj per la manipulista ordono.

La datumbazo

El la menciitaj bultenoj la programaro vivtenas tian datumbazon, en kiu ĝi ebligas por la uzanto la alpremeblecon de la storitaj datumoj pere de teleksaj, mezrapidaj datumtransigaj linioj aŭ programoj efektivegeblaj en la sistemo. Ni solvis la storadon kaj la retrovodon per du rikordaroj:

-identigila rikordaro /katalogo de bultenoj/, el tiuj bultenoj, kiujn la Hungara Meteologia Organizaĵo bezonas, aŭ ĝi devas transigi tra diversaj cirkvitoj.

En tiu ĉi rikordaro troviĝas ĉiuj tiaj informoj, kiujn la sistemo bezonas por regi la linian trafikon aŭ plenumi la datumbaz-funkciojn. Laŭ tio la rikordaro aŭ rikordo enhavas:

-datumojn necesajn por identigi bultenojn

- alvenan kaj eksendan tempojn de la bulteno
- la prioritaton de la dissendo
- la stor-adreson de la bulteno
- la dissendajn listojn, kiuj dependas de la alvena tempo
- la informojn bezonatajn por fari statistikojn /vidu poste/

Nature, la rikordaro bezonas konstantan aktualigon, depende de la dissendataj bultenoj kaj de la bezonoj por la meteologiaj servoj. La modifo de la jam ekzistantaj rikordoj fariĝas per kunkroĉa /on-line/ programo, la enmeto de novaj rikordoj aŭ forigado de malnovaj rikordoj fariĝas per senkroĉa /off-line/ programo.

- la dua rikordaro enhavas la bultenojn alvenintaj de la enigaj linioj aŭ bultenojn pretigaj de la programoj de telekomunika programaro. La stora dimensio estas proporcia al la kvanto de la datumoj dissendataj dum 24 horoj. La stora podimensio dependas nek de la storenda materialo, sed ĝi egalas al la sektoro, kiu estas la atingebleca kvanto de la disko. La alvenintaj bultenoj estas storataj vickontinue, en sektoroj ligitaj unu al la aliaj. En ĉiu sektoro unu montra nombro indikas la kvanton de la storata informo. La adreso de la sektoro de la bulteno estas storata en la unua rikordaro.

Statistikoj

La Tutmonda Meteologia Organizaĵo preskribas al ĉiuj meteologiaj centraloj la produktadon de statistikoj, iliajn formon kaj tempon en la manlibro de la Tutgloba Telekomunika Sistemo. Nia telekomunika sistemo informas laŭ preskribo pri la bultenoj, kiuj ne alvenis en la ĝusta tempo kaj ĝi faras statistikon pri tio, ke la alvenintaj bultenoj kiom longe malfruis, kalkulante de la eksenda tempo. Por plifaciligi la vivtenajn laborojn ni faras tabelon je tiuj bultenoj, kiuj alvenis, sed kiuj ne enestas en la datumbazo de la Hungara Meteologia Organizaĵo.

La elskribado de la rezultoj de la statistikoj okazas je la manipulista ordono, ĝenerale en la periodoj preskribi-

taj de la Tutmonda Meteologia Organizaĵo.

La kontakto inter la manipulisto kaj la komputilo

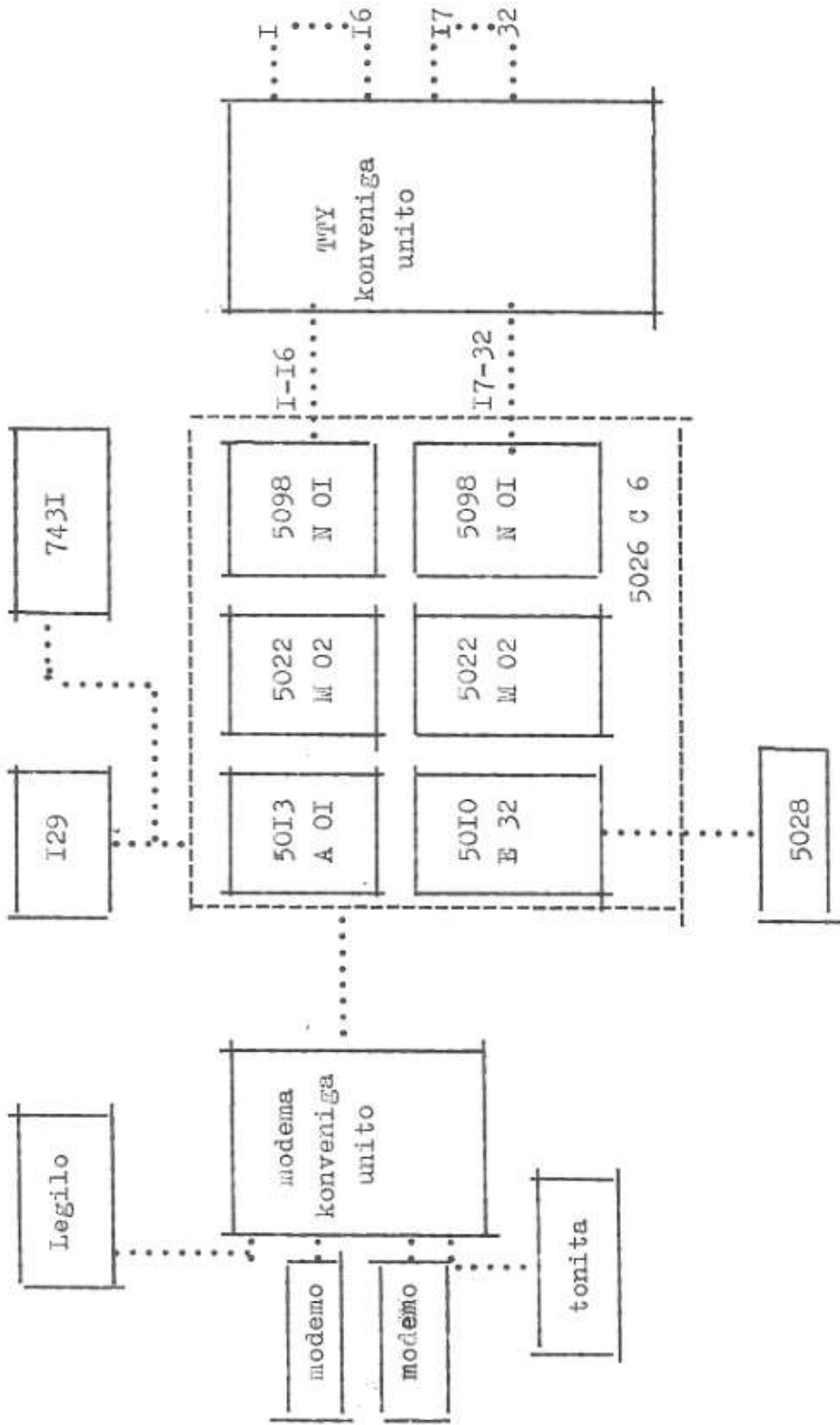
La manipulisto devas scii konstante pri la okazintaj eraroj en la trafiko direktata de la komputilo, pri la stato de la linioj, pri la mankaĵoj de la datuminterŝanĝo. La manipulisto povas informiĝi simple kaj rapide pri la ecoj de la vicantaj materialoj kaj li povas modifi la vicordon de la diversaj linioj, li povas ripari la materialojn, kiujn la sistemo trovis malbona, ktp.

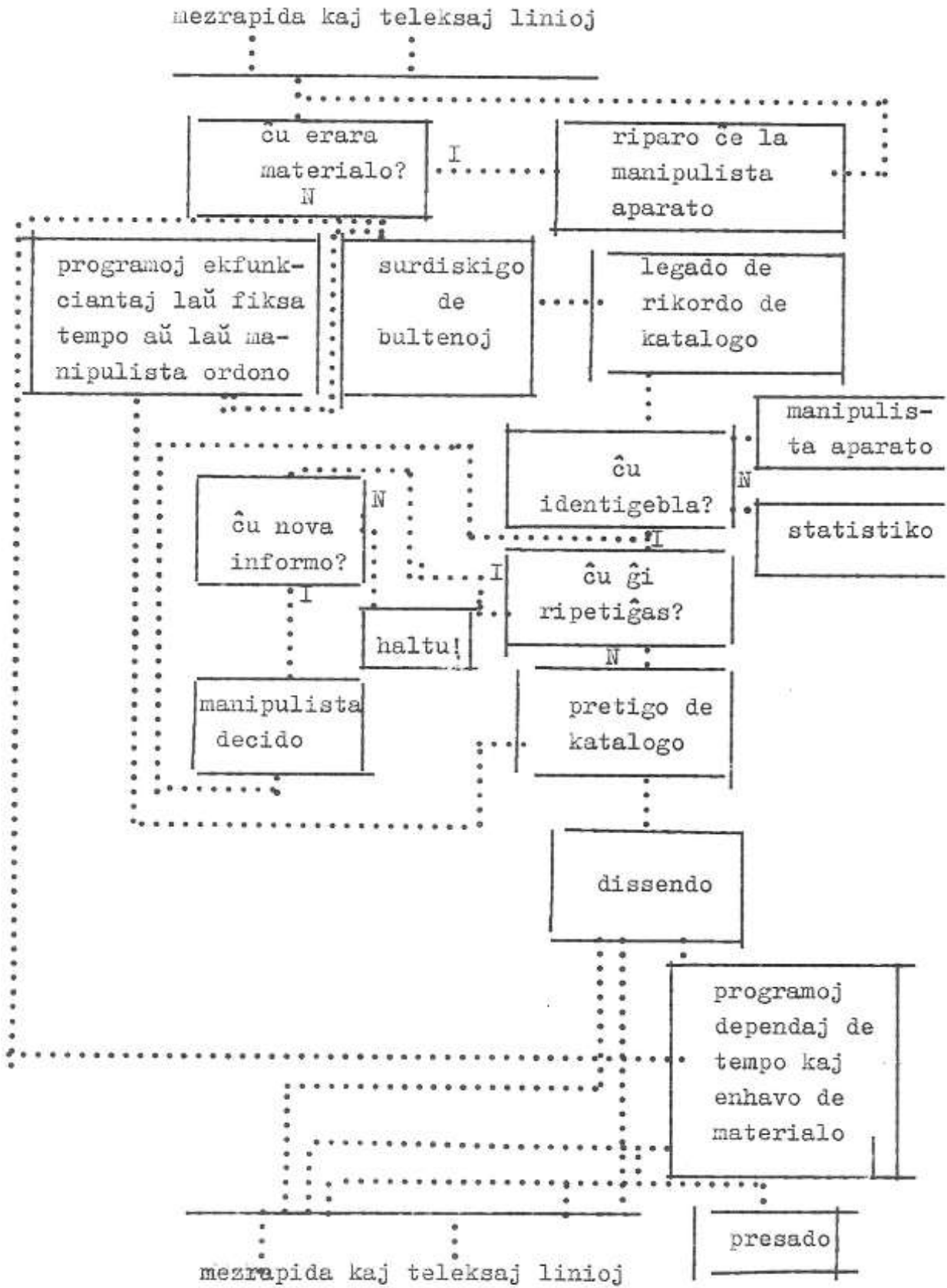
La unusola manipulista aparato traktata de la IBM bazprogramaro montriĝis malmulta por la farendaĵoj. Tial la sistemo uzas la du liniojn de la telekomunika multkroĉa modulo je manipulista rimedo. Oni povas efektiviĝi el tiuj du teleksaj linioj tiajn specialajn helpprogramojn, per kiuj oni povas modifi la plenumiĝantan funkciadon, kaj la telekomunika sistemo je unu el tiuj du linioj sciigas la manipuliston pri la malfacilaĵoj okazintaj en la trafiko.

Oni povas direkti per la manipulista aparato traktata de la bazprogramaro pakan /batch/ prilaboradon kaj ekfunkciigi tiajn programojn, kiuj ne ligiĝas strikte al la direktado de la telekomunikado.

Ni ŝatus esti solvintaj la konigitan telekomunikadan sistemon per la programpakedo IBM CCAP/7, sed la programaro fariĝis embargita, ne estis do ebleco por la aplikado.

La rezigno pri la luado de CCAP/7 ŝparas al la Hungara Meteologia Organizaĵo 12000 Dolarojn da instruada kosto kaj 6000 Dolarojn da lua kosto po jaro dum 5 jaroj.





Figuro 2.

Bibliografio

I./Manual on the Global Telecommunication System, Volume I.

Aŭtobiografio

Mi naskiĝis en 1955-01-13 en Budapeŝto. La komputadon mi ekŝatis dum la gimnaziaj jaroj, kiam mi okupiĝis unu jaron pri programado de komputiloj TPA. Post abiturienta diplomo mi frekventis programadan matematikistan fakon de Sciencuniversitato Lóránd Eötvös, kie mi diplomiĝis en 1976. Post la ŝtatekzameno mi eklaboris en la Novaĵcentro de la Hungara Meteologia Organizaĵo, kie mi partoprenis la evoluon de la telekomunikada komputado.



Csanád ENDRÉDI

BUDAPEST, Akácfa u. 13.

1204

MODULA PROGRAMADO

 Marc VANDEN BEMPT

Resumo: La artikolo prezentas diversajn flankojn de la programado laŭ moduloj, unu el la eblecoj por igi programojn pli facile legeblaj kaj modifeblaj.

1. Difino.

Pliaj difinoj de moduloj estas troveblaj en la literaturo. Plej ofte ili originas el iu metodo por disigi programojn en modulojn, kiel ekzemple en la jena difino:

"Modulo estas nombro da instrukcioj kiuj plenumas specifan funkcion kaj kiuj povas esti skribitaj preskaŭ sendepende de aliaj programpaŝoj."

Kvankam tiu ĉi difino estas bone uzebla, ĝi limiĝas je nur du vidpunktoj:

- 1 modulo = 1 funkcio,
- 1 modulo = kiom eble sendependa de la resto de la programo.

En la dua paragrafo ni koncize priparolos tiujn ĉi, sed ankaŭ aliajn vidpunktojn.

Por indiki la komunan econ de moduloj ni volas limiĝi je la jena priskribo:

"Modulo estas grupigo de nombro da instrukcioj en unu tutaĵon. Tiu ĉi grupo povas (sed ne devas) uzi datumojn kiuj ne estas atingeblaj el aliaj programpartoj."

Post mallonga priparolado de la diversaj eblaj vidpunktoj kaj de la avantaĝoj kaj malavantaĝoj, ni klopodos priskribi kelkajn praktikajn gvidliniojn por la konstruado de mo-

duloj; finfine ni indikos la diferencon inter tri specoj da moduloj: paragrafoj en programo, ĝeneralaj moduloj kaj kvazaŭ-ĝeneralaj moduloj.

2. Metodoj, postuloj.

Dum la disigado de programo en modulojn eblas diversaj kriterioj. Kelkaj eblecoj estas la jenaj:

2.1. Laŭ abstraktnivelo (Dijkstra).

Moduloj je la plej alta nivelo en programo estas abstraktaj. Ju pli malalta la nivelo, des pli konkretaj iĝas la moduloj. Je pli alta nivelo precipe la plenumu- kaj vok-instrukcioj estos uzataj; je pli malalta nivelo troviĝas "se-tiam-maltiam"-instrukcioj; tute sube troviĝas la plenumaj instrukcioj (adiciu, movigu, ...).

Ekz.: modulo je alta nivelo: plenumu komenco.

plenumu agado.

plenumu fino.

modulo je suba nivelo:

se ago (indeks-ago) = serĉita-ago,

tiam movigu serĉita-ago al elektita-ago,

maltiam adiciu 1 al indeks-ago.

2.2. Laŭ funkcia dekomponado (Constantine/Myers).

Modulo tie ĉi plenumas 1 funkcion de la tuto.

Ekz.: programo: plenumu legu-rikordon.

movigu spaco al kontrolo.

plenumu kontrolu-kaj-legu

ĝis (kontrolo ne = spaco).

...

modulo kontrolu-kaj-legu: plenumu kontrolu-rikordon.

se kontrolo = spaco,

tiam plenumu legu-rikordon.

...

2.3. Laŭ agadkolektol (Naur).

Laŭ la fundo-supren metodo oni konstruas el nombro da instrukcioj koheran tutaĵon (=modulo). Tiaj moduloj estas kunigitaj, ĉiam je pli altaj niveloj, ĝis la finprogramon.

2.4. Laŭ modulo-intervandoj (Parnas).

Tiu ĉi metodo diras ke la nombro da ligoj inter la diversaj moduloj devas esti minimumigitaj. Sed atentu: la "nombro" da ligoj inter moduloj ne estas priskribebla kiel funkcio de kontroltransiĝo, de la nombro da parametroj, nek de la nombro da komunaj datumoj. Per ligoj ni kontraŭe volas indiki la supozojn kiujn moduloj havas unu pri la alia: minimumigi tiujn aŭtomate igas la modulojn sendependaj de aliaj programpartoj.

Ekz.: modulo serĉu (nombro, vico, respondo):

```

movigu spaco al respondo.
se nombro < vico(1),
    tiam movigu 'tro malgranda' al respondo.
plenumu komparu, indeksvico de 1 per 1,
    ĝis (indeksvico > 10) aŭ (respondo ne = spaco).
se respondo = spaco,
    tiam movigu 'tro granda' al respondo.
kun modulo komparu:
    se nombro = vico (indeksvico),
        tiam movigu 'trovita' al respondo,
        maltiam se nombro < vico (indeksvico),
            tiam movigu 'netrovita' al respondo.
```

(Rimarko: la alvoko al modulo 'komparu' ne estas bone strukturita, ĉar testo kun AŬ kaj NE fakte estas netolerebla. Tion ni en ĉi tiu artikolo tamen ne konsideras).

Tiu ĉi modulo 'serĉu' supozas ke la vico estas ordigita pli-altiĝanta. La danĝero ekzistas nun ke alvokanta programo ne scias tion kaj do rericevos erarajn rezultojn de la modulo. Alia supozo ankaŭ estas ke la vico devas konsisti el precize 10 nombroj.

Tiaj supozoj estas evitendaj. Kiel tio eblas?

- aŭ la modulo je la komenco kontrolas ĉu la kondiĉoj estas plenumitaj.

Ekz.: kontroli ĉu la 10-nombra vico estas ordigita pli-altiĝanta.

- aŭ oni pliĝeneraligas la modulon tiel ke la supozo forfallas.

Ekz.: la testo "se nombro < vico (indeksvico), tiam movigu 'netrovita' al respondo" povas esti forlasita. Tiam la modulo jes estas malpli rapidfunkcianta por ordigitaj vicoj, sed ĝi estas ĝenerale uzebla por arbitraj vicoj.

aŭ oni klare indikas kiel komentario kiuj estas la supozoj.

Ekz.: "Tiu ĉi modulo atendas serĉendan nombron kaj vicon de 10 nombroj, laŭ plialtiĝanta ordo. Ĝi redonas kiel respondo 'trovita' se la nombro egalas unu de la dek; se ne ĝi donas 'tro malgranda', 'netrovita' aŭ 'tro granda', depende de ĉu la nombro estas pli malgranda ol ĉiuj nombroj de la vico, troviĝas inter la vicnombroj sen egali unu el ili, resp. estas pli granda ol la 10 nombroj."

2.5. Laŭ datumstrukturo.

Tiu ĉi metodo indikas ke la strukturo de la programo devas sekvi la strukturon de la datumoj uzitaj de la programo.

Tio tamen ne ĉiam eblas, certe ne uzante datumbenkaĵojn destinitajn al uzado de diversaj kaj diversfunkciaj programoj. Tamen en diversaj kazoj povas esti utila ke oni egaligu pli-malpli program- kaj datumstrukturojn. Tiel modulo nur uzos limigitan kaj klare indikitan parton de la tuta datumaro.

Anstataŭ ordigi do la instrukciojn laŭ la datumstrukturo, ni prefere klopodu formigi la datumojn laŭ la strukturo de

la programo uzanta ilin.

3. Avantaĝoj kaj malavantaĝoj de moduleco.

Avantaĝoj:

- pli facile kodigebla kaj korektebla, sendepende de aliaj programeroj.
- pli facile kompreneblaj programoj.
- pli facile ŝanĝebla: aldona funkcio povas esti aldonita je la dezirita loko en la programo.
- pli facile kontrolebla.
- programoj ne dependas de unu programisto.
- ebleco de korektecpruvoj.
- la programlogiko strikte sekvas la problemlogikon.
- estimado de la necesaj personoj kaj tempo por ellabori sistemon estas pli facila uzante modulajn sistemojn.
- diversaj personoj samtempe povas labori pri unu programo (tio jes demandas intereson de tiuj diversaj programistoj pri la tuto!). Komplikaj moduloj povas esti skribataj de spertaj programistoj.

Malavantaĝoj:

- "plej multaj programistoj ne komprenas modularecon": tio tamen estas nur komenca problemo.
 - postulas aldonan skribadon.
 - postulas pli da komputillaboro (maksimume 5 ĝis 10% tamen).
 - necesas kooperado inter la programistoj, same kiel informado de la projektinto de la programo al la programistoj.
 - postulas dokumentadon de la moduloj kaj de modul-alvokoj.
 - postulas iomete pli da stoko (maksimume 5 ĝis 10 %).
- Uzante optimumigantan kompilon tiu ĉi malavantaĝo tamen forfalas, eĉ povas iĝi avantaĝon. Tiaj kompileroj verŝajne pli bone funkcios sur modulaj programoj, tiel ke tiuj eĉ postulos malpli da stoko.

4. Teknikoj de modula programado.

Moduleco pli estas pensmaniero ol sekvado de reguloj. Farante oni lernas fari. Tamen ni povas indiki kelkajn punktojn pri kiuj oni povas atenti.

- Modula programado komencas je modula projekto.
- Moduloj povas esti:
 - paragrafoj aŭ kohera grupo da paragrafoj en programo,
 - laborunuo farenda de unu programisto,
 - sendepende kompilita unuo,
 - kodunuo komuna al diversaj programoj.
- Skribu simplajn modulojn. Modulo ne povas fari tro, tiel ke ĝia logiko ne estu malklarigita. Necesas kompromiso, rilate al la ĝeneraleco de modulo, inter:
 - kion la modulo povas: ju pli ĝenerala, je des pli da kazoj ĝi povas esti uzata,
 - kion la modulo faras: ju pli ĝenerala, des pli da laboro restos farenda de la alvokanta programo.

La limo inter la du alternativoj povas esti movigita favore al la unua alternativo, sen ke la dua estas forgesita: oni skribu modulojn kun diversaj eblecoj kaj lasu la elekton inter tiuj eblecoj al la vok-programo, kiu indiku per parametroj kiujn eblecojn ĝi deziras uzi.

En la modulo de 2.4 eblus ekzemple transdoni la nombron da elementoj de la vico al la modulo kiel parametro (kun maksimumo de ekz. 100) tiel ke tiu ĉi lasta estas uzebla por pli da kazoj.

- Ofte eblas skribi nombron da malgrandaj ĝeneralaj moduloj, kiuj je pli alta nivelo estas kunigitaj en kvazaŭ-ĝeneralaj moduloj (vidu par. 7): la ĝeneralaj moduloj tiam konsistigas la bazon el kiu la programoj estas kunmetitaj.
- Interne de la moduloj oni nepre ne forgesu atenti la jenajn punktojn:
 - strukturita programado

- pinto-suben projektado, programado kaj testado
- dokumentado
- normigado
- korektecpruvoj por ofte uzataj moduloj.

- Atentu minimumajn ligojn inter moduloj: kiom eble malmultaj supozoj de moduloj unu pri la alia aŭ pri vok-programoj, kaj inverse de tiuj programoj pri moduloj.

- Se ekzistas tiaj supozoj ili devas esti klare surpaperigitaj.

- Kiom eble plej malmultaj programoj povas esti influataj de modifo en unu de la moduloj.

- Moduleco devas esti uzata je ĉiu nivelo de la programo: ne taŭgas skribi programon kiel "plenumu komenco, plenumu agado, plenumu fino", se la modulo "agado" mem ne plu estas disigita en modulojn.

- Lokaj variabloj (indeksoj, helpvariabloj, ...) interne de modulo plej bone ne estu uzataj ekster la modulo.

- En/Eligo, kalkuloj kaj decidoj restu apartaj.
 Ekz.: programo = plenumu komenco. (= movigu ...)
 plenumu legi. (= enigo)
 plenumu kontrolo-legita. (= decidoj)
 plenumu prilaborado. (= decidoj, kaj je
 malpli alta nivelo: kalkuloj, movi-
 gu ...)
 plenumu skribi. (= eligo)
 plenumu fino. (= movigu ...)

5. Paragrafoj en programo.

Tia modulo estas tutaĵo da instrukcioj kiu troviĝas en programo en aparta paragrafo, ĉar ekzemple ĝi

- plenumas specifan funkcion,
- estas uzita plurfoje en la programo.

Pere de tiaj paragrafoj, programo iĝas pli bone aranĝita, ol se la instrukcioj simple troviĝas inter la aliaj. Samtempe ripetado de instrukcioj estas evitata. Se specifa paragrafo plurfoje estas alvokita, certe valoras la penon enkonduki specialajn kontrolojn en la paragrafon por sekurigi ĝin (en la senco de defensiva programado).

Ia sinsekvo de la moduloj en la programo ĝenerale estas la hierarkia sinsekvo. Unue troviĝas la modulo de la supra nivelo; ĝin sekvas la unua modulo de la dua nivelo, ktp. ĝis la suba nivelo, poste la dua modulo de la antaŭsuba nivelo, ktp.

Alivorte: la arbostrukturon de la moduloj en programo ni trairas de supro suben kaj de maldekstro dekstren.

6. Ĝeneralaj moduloj.

Tiaj moduloj estas skribataj kiel apartaj programunuoj, aparte prilaboritaj de kompilero. Ili povas esti alvokitaj sen iu modifo de diversaj aplikadprogramoj.

Necesas dokumentado pri tiaj moduloj: kiuj ekzistas kaj kiel uzi ilin?

Programo kiu alvokas ĝeneralan modulon ne nepre devas profunde koni tiun ĉi lastas. Ĝi konsideras la modulon kiel nigran skatolon, kun alirejo konsistanta el modulnomo kaj unu aŭ pluraj datumpasigzonoj: tiuj estas la elementoj specifendaj de la alvoko al la modulo.

El tio evidentiĝas la graveco de la dokumentado de ĝenerala modulo. Ĝi nome estos uzata de programistoj kiuj ne skribis ĝin, kiuj ne scias kiel ĝi funkcias, kiuj nur uzas priskribon de kion la modulo faras kaj de la necesaj parametroj.

Skribante modulon vi do nepre zorgu ke la dokumentado ne nur estu farita, sed ankaŭ ke ĝi klare kaj korekte priskribu la deziritan.

Kiel avantaĝoj de tiaj ĝeneralaj moduloj ni i.a. citas:

- minimumigo de programadkostoj,
- postulas kaj akcelas normigon inter la diversaj programistoj
- ŝparas stokon (se ne la modulo devas trograndiĝi por iĝi ĝenerala),
- evitas programaderarojn (almenaŭ ripetadon ties),
- lernigas pensi strukturite (kvankam ankaŭ nestrukturitaj moduloj povas esti skribataj ...).

Kiel eventualaj malavantaĝoj ni povas mencii:

- ne facilas skribi ĝeneralan modulon kiu devas uzi diversajn rikordarojn, depende de la alvokanta programo.
- skribi ĝeneralajn modulojn postulas krepovon: ĉiuj eblecoj devas esti antaŭviditaj. Per simpla modifo ofte eblas ege ĝeneraligi modulon.
- ĝeneralaj moduloj skribitaj ekster la propra entrepreno plej ofte ne sekvas la proprajn programadkonvenciojn.

7. Kvazaŭ-ĝeneralaj moduloj.

Certaj programpartoj ne estas skribeblaj kiel ĝeneralaj moduloj. Tiam ni povas klopodi skribi ilin kiel kvazaŭ-ĝeneralan modulon. Tiuj estas unuoj kiuj troviĝas en diversaj programoj kaj kiuj funkcias kun diversaj datumoj, sed kiuj estas sammaniere strukturitaj.

Unu el tiuj programunuoj do povas esti uzata kiel modelo por la alia, tiel ke la programadkostoj povas esti malpliigitaj. La modifokostoj ne multon malpliigatas ĉar modifo devas tamen okazi en ĉiuj kopioj (kvankam la loko kie ĝi devas okazi ne ĉiufoje devas esti reserĉita).

Parto de la instrukcioj de kvazaŭ-ĝeneralaj moduloj estas fiksa, alia parto estas modifenda laŭ la kazo. Anstataŭ ĉiam reskribi la fiksan parton ĝi povas esti skribita unufoje. Kiam oni volas apliki tiun modulon je specifa kazo, oni do nur devas kopii tiun parton kaj aldoni, je specifaj

lokoj en la programo, la nefiksajn instrukciojn.

La avantaĝo ne nur estas ke la fiksa parto devas nur unufoje esti skribita, sed samtempe ke ĝi devas esti testita nur unufojon.

8. Konkludo.

Por ke programoj estu pli bone legeblaj oni devas ordigi la instrukciojn en ili laŭ paragrafoj (same kiel libro aŭ artikolo). Ekzistas diversaj kriterioj laŭ kiuj tiu disdivido de programo en paragrafoj povas okazi. Ĝeneralaj programpartoj, plurfoje uzataj por solvi specifan problem(ar)on, povas esti skribitaj unufoje kiel ĝenerala modulo. Se tio ne eblas, oni povas tamen klopodi skribi kvazaŭ-ĝeneralan modulon por solvi la celitan problem(ar)on.

Bibliografio

La jenaj libroj kaj artikolo estas uzataj kiel bazo por la ideoj klarigitaj en ĉi tiu artikolo.

- Modular programming in Cobol, Russell M. Armstrong. John Wiley & Sons, New York k.a., 1973.
- Programma- en informatiestrukturen, Prof.Dr.Ir. Y.D.Willems. Kurso K.U.Leuven, 1977/78.
- Modulair en gestructureerd programmeren, red. van de EDP.- Analyzer. El Informatie 1976, nr.11, p. 625-632.

Biografio



Marc VANDEN BEMPT, licencposedanto pri Matematiko kaj pri Informadiko (K.U.Leuven, 1977 resp. '78) estas analizisto ĉe A.B.B. en Leuven (Asekurkompanio de la Belga "Boerenbond"), Belgio.

La mezurado de la programaro
d-rino Sarolta ZÁRDA

Resumo:

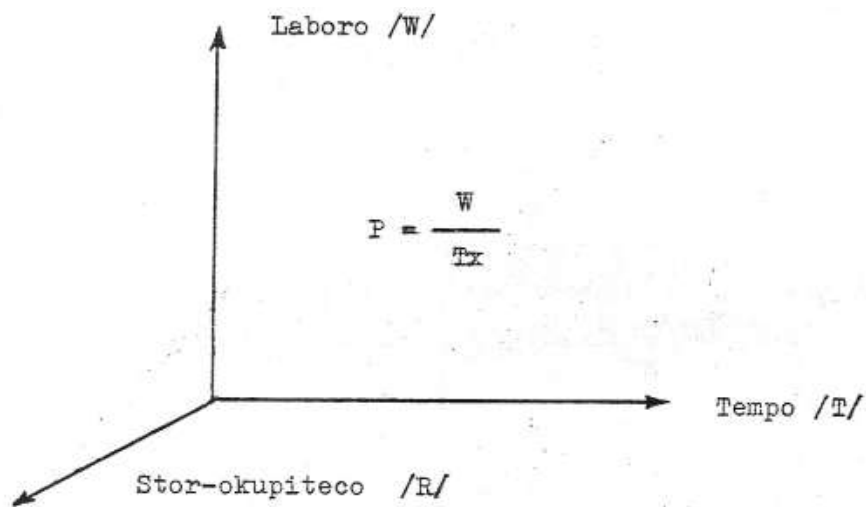
La mezurado de komput-teknikaj servoj pro la multspececo, diverseco en la prezo kaj potenco de la programaro kaj aparataro hodiaŭ jam estas necesa. Sur tiu grundo elformiĝis la fakscienca branĉo KOMPUTMEZURADO, kiu enkadrigas ege multajn metodojn de mezurado kaj komparado por analizi la programaron, aparataron, man-operaciaron plejparte sur la eksperimenta nivelo.

Unuecan sistemon donas por mezuri la programaron la nova teorio, ellaborita de Kenneth W. Kolence: fiziko de la programaro. La baza koncepto de la fiziko de la programaro estas elekto de 3 kvantaj karakterizoj, el kiuj ĉiuj aliaj devenas per matematikaj metodoj. Per la devenigitaj karakterizoj priskribebblas la plena strukturo de programaro. Sen-dube la teorio enhavas sufiĉe multe da abstrakcioj, tamen ĝi donas la unuecan sistemon de la mezurado de la programaro, kiu ebligas la konzistentan komparon sur la tereno de ekspluado kaj evoluo.

Hodiaŭ jam pli kaj pli antaŭeniĝas la direktado, mezurado kaj kontrolo de la komput-teknikaj potencoj. Ĝi estas necesa por ni dum ŝarĝad-karakterado, planado de konfiguracio, taktigo, potenco-plibonigo kaj kost-kalkulado. Nova eksperimenta metodo de la kapacito-direktado estas la fiziko de la programaro.

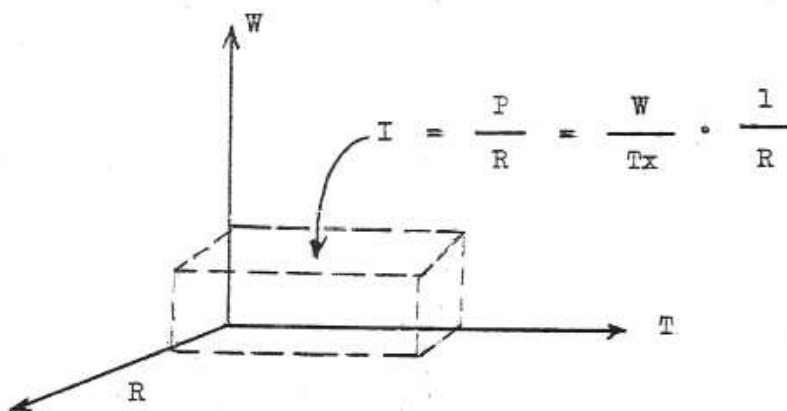
La bazaj nocioj de la fiziko de la programaro estas la programar-laboro /ĝian difinon vidu poste/, la tempo, necesa

por plenumado kaj la stor-okupiteco. La teorio dispartigas la laboron al labor-specoj, plenumantaj de diversaj rimedoj kaj elformas la nocion de unuo-vektoro, necesan por plenumo de iom de laboro. La plej gravaj devenigataj montriloj estas la potenco, la labor-denseco kaj la labor-intenso. La kunligo de la devenigataj montriloj kun la bazaj videblas laŭ sekvaĵaj figuroj:



Kie: T_x = tempo de la plenumado
 T_e = la suma forpasita tempo
 D = labor-denseco /saturiĝo/
 P = potenco

Labor-intenso: I



El la nocioj ni analizu pli detale du, la laboron kaj la potencon, ĉar pere de tio ni povas vidi ankaŭ la aplikon de la teorio.

La esprimo "programar-laboro" ĉirkaŭe egalas al per-komputila laborplenumo, sed pli precize ĝi esprimas, ke por la funkciigo de la programaro /programoj/ necesas difinit-kvanta potenco, labor-plenuma kapablo, laboro de la komputilo. Do ni povas rigardi kiel unuon de la laboro de la programaro ian agadon plenumitan far procesilo sur ian storilon /ekzemple transporto de unu oktumo sur konkretan storilon/. La laboron de la programaro ĉe IBM-komputiloj oni povas elkalkuli el ordinaraĵ mezuroj, kolektitaj de SMF.

La laboro de la periferio-unuoj:

$$\text{laboro} = \frac{\text{/nombro de la enig-/elig-agadoj/}}{\text{averaĝa blok-
longo/}}$$

La averaĝan laboron de la procesilo ni mezuras en tempo-unuoj. ni nomas, kiel potenco de la programaro /vidu poste pli detale/.

Tiel:

$$\text{Laboro de la centrala procesilo} = \frac{\text{/averaĝa
potenco/}}{\text{/centrala procesila tempo/}}$$

La programara laboro priskribebla per du bazaj metodoj. Laŭ nnu el ili ni priskribas la programaran laboron per konfiguracio-partoj /centrala procesilo, kanaloj, direktilo ktp./, en tiu kazo la karakterizo ne estas sendependa de la konfiguracio. Pli ĝenerala karakteriz-metodo estas, se la

laboron ni enklasigas en tiel nomatajn rimedarajn klasojn kunlige kun konkreta ŝarĝo, aŭ ĝia komponanto. Do la rime-dara klaso estas tia aro de rimedoj, kiu kapablas plenumi ian taskon. La ŝarĝon de la rimedara klaso ni povas karakterizi per la laboro de la centrala procesilo, magnetdiskoj, magnetbendoj, terminaloj ktp. La laborojn de opaj rimedaraj klasoj ni sumigas.

La laŭ rimedaraj klasoj dispartigitan formon de la laboroj ni nomas kiel labor-vektoro de la programaro. Tiu ĉi vektoro estas konstanto rilate de konkreta tasko, kiu prilaboras konkretan datumaron, sendepende de la konfiguracio kaj la tasko-konsisto. Nur la efikeco de la tasko-plenumado ŝanĝiĝas. La efikeco esprimas la efikon de la kunordigiteco de la sistemo kaj la rilaton de la konfiguracio al potenco-postuloj.

La labor-vektoron de la programaro ni povas ricevi, konsiderinda, ke la laboro de la komputilo sumiĝas el multaj komponantoj, el la laboro de multaj parto-unuoj. tiu konkreta tasko /programo kun datumoj/ el la laboro de iu parto-unuo postulas konkretan kvanton.

Tiel la programar-laboron ni povas esprimi per tia vektoro, kies komponantoj estas la apartaj parto-laboroj. /En la formulo ni signas ilin simile al kolumno-vektoro./ Ĉe komputiloj kun diversaj parametroj tiu ĉi vektoro estas diversa, sed la programaran laboron /tio estas la perkomputilan laboron, necesan por plenumo de la tasko/ ni povas esprimi ankaŭ en sendependaj de la komputilo unuoj. Ekzemple ĉe la centrala storo

anstataŭ la tempo ni uzas la nombron de la stor-ciklo. Tiel la vektoro de la programara laboro rilate de la tasko estas konstanta.

Se la elementojn de la labor-vektoro de la programaro po rimedaraj klasoj ni dividas per la tuta laboro, ni ricevas la unuo-vektoron de la programaro, kies element-sumo ĉiam estas 1. La plejparto de la unuo-vektoro de programaro estas konstanto. Tio signifas, ke rilate de konkreta kurado aŭ taskogrupa la sola ŝanĝiĝanta komponanto estas la plena laboro.

La distribuo de la laboro, uzata por la tasko depende de la rimedaraj klasoj ne ŝanĝiĝas. Per elformo de la unuo-vektoro de la programaro - se ni konas la plenan laboron - la distribuon de la ŝarĝado eblas bone pesi.

La laboron divide de la tempo ni ricevas la potencon de la programaro. La potenco de la programaro enkaze de la plena ŝarĝado sur la nivelo de la plena konfiguracio egalas al ĝenerala kompreno de prilabor-kapablo. La teorio de la fiziko de la programaro por ebligi la komparon de la prilabor-kapabloj inter konfiguracioj aŭ ene de konkreta instalacio laŭtempe, enkondukas la nocion de la plenumo-kapablo. La plenumo-kapablo estas la plena laboro, plenumita sur konkreta konfiguracio koncerne de parto-ŝarĝo dum tempo-unuo. Se ni la laboron disdividas je rimedaraj klasoj, ankaŭ la plenumo-kapablo povas esti esprimita en ilia rilato. Ekzemple se la tasko uzas 3 rimedaraj grupoj dum T_e tempo:

$$\text{Plenumo - kapablo} = \frac{1}{T_e} \begin{bmatrix} W \text{ centrala procesilo} \\ W \text{ disko} \\ W \text{ bendo} \end{bmatrix}$$

La formulo montras, ke se ni multiplikas la vektor-elementojn per Te^{-1} , tiam ni ricevas koncerne de opaj rimedaraj klasoj la dum Te tempa averaĝan potencon. Tiun ni nomas kiel relativa potenco de la rimedaraj klasoj. En tiu kazo la tempo koncernas la tutan konfiguracion kaj ne la opajn rimedarajn klasojn.

En kazo de elkalkulo de absoluta potenco la laboron de la rimedara klaso ni dividas per la propra tempo de plenumado. La plenumo-kapablon ni povas kunligi kun la absoluta potenco de rimedaraj klasoj. La kvantoj, per tipo

$$\frac{T_x \text{ /centralla procesilo/}}{T_e} \quad \text{/kompreneble simile difinite}$$

ankaŭ por la aliaj rimedaraj klasoj/ laŭesence difinas la procentan utiligecon. La sumo de utiligec-procentoj donas la faktoron de la simultana funkciigo /SF/ de periferioj.

$$SF = \frac{T_x \text{ / centralla procesilo /} + T_x \text{ /disko/} + T_x \text{ /bendo/}}{T_e}$$

La teorio difinas la teorie riceveblan potencon de la absoluta potenco de iu konfiguracio. La teoria potenco apartenas al iu nivelo de potenco-utiligo, kiu estis mezurita dum ideala ŝarĝo. A konkreta ŝarĝado ĉiam /plejparte/ eblas trovi idealan konfiguracion kaj inverse. La praktika problemo de la potenco - direktado estas la trovo de tiu ekvilibro. /Por ke la ekvilibro en kiel eble plej granda procento de la tempo ekzistu./

La kreantoj de la fiziko de la programaro ellaboris ankaŭ unuecan sistemon de la signoj. Ĝia baz-strukturo estas la sekva:

$V_q / V_{d1} V_{d2} /$, kie

V_q estas la dimenzio, ekz. laboro, tempo ktp.

V_{d1} nomo de la programar-unuo

V_{d2} konfiguracio au rimeara klaso.

Surbaze de tiu teorio en Hungario estis analizitaj komputiloj Honeywell /tipoj (66/60+ 66/20). Estas planata apliki la unuecan mezuradan sistemon, bazitan sur la fiziko de la programado ĉe komputil-serioj de Unueca Komputil-Sistemo kaj Unueca Minikomputil-Sistemo de socialismaj landoj.

Biografio

Mi naskiĝis en Budapeŝto. Mi diplomiĝis en 1974 sur la planad-
matematika fako de Budapeŝta Ekonomikista Universitato. Temo de
mia diplomlaboro estis la analizo
kaj apliko de la ebloj de
forrester- a sistem-dinmiko. El
tiu temo mi doktoriĝis en 1974. Mi
pli detale okupiĝis pri lingvoj
por simuli. Tiamaniere mi
kontaktiĝis pli proksime kun la
komputilo kaj programaro. La sekva
temo estis la simulo de la
programaro. En 1977 mi preparis la
simulan modelon de operacia
sistemo GCOS de Honeywell
komputiloj, poste mi okupiĝis pri
principaj ebloj de la datumbaz-
simulo. La simulo de la programaro kondukis min al efikec-analizo
de perkomputilaj servoj, al esploro de potenco-taksado. Nun mi
okupiĝas pri komparo de komputiloj, elekto de programaro el
komerca vidpunkto.



En la Budapeŝta Ekonomikista Universitato mi instruas
komputadon kaj la programlingvon FORTRAN, mi volonte okupiĝas
pri interesiĝantoj pri komputado studentoj kaj komencantoj.

PROPRECOJ DE PROGRAMARAJ PROCEZILOJ

Szóke László

Evoluiga Instituto de VIDEOTON

Hungario

Resumo: Perkomputila solvo de granda klaso de praktikaj problemoj bezonas samtempan uzon de kelkaj lingvaj niveloj, pro tio estas interese esplori programarajn procezilojn, kiuj ebligas aŭtomatan transiron de unu lingva nivelo al alia. En la artikolo estas ellaborita semiotika nociaro por difini kaj analizi abstraktajn programarajn procezilojn. Estas pruvita, ke ekzistas algoritmo por formala abstrakcio. Estas enkondukita la nocio de dinamika procezilo kaj kelkaj specoj de dinamikaj proceziloj estas priskribitaj.

0. Enkonduko

La arto de programado de komputiloj ekzistas kiel konsekvence de la jena baza pragmatika problemo : oni bezonas malsimplan konduton de la komputilo, sed havas por ĝia priskribo nur relative simplan lingvon.

La problemo estas inherenta al la programado. Ĝi ekzistas dum komputiloj mem ekzistas, ŝanĝiĝas nur la nivelo de la programaj lingvoj kaj tiu de la malsimpleco de la dezirarla konduto. Oni solvas la problemon, sed oni eĉ ne pensas likvidi ĝin, ja ĝuste tio estas la avantaĝo de la komputilo kompare al la aliaj gpecoj de maŝinoj, ke ĝi povas, dank' al la programaro, konduti multe pli komplika, ol tio eblus sen la programaro.

La problemo estas speciale streĉa, kiam oni projektas programan sistemonm tiom komplikan, kiel operaciaj sistemoj, aŭ se oni uzas lingvon tiom malriĉan, kiel maŝinkoda lingvo de ordinaraj mikroprocezoj.

Pri solvo de tiu ĉi problemo okupiĝas multaj esploristoj. Distingeblas du ĉefaj alproksimiĝoj:

- de supro malsupren, kiam estas esplorata la metodologio, kiel diserigi komplikan projektotaskon je pli simplaj taskoj;
- de malsupro supren, kiam oni esploras la problemojn de plialtigo de lingva nivelo.

La du alproksimiĝoj reciproke kompletigas unu la alian kaj la bazan problemon ili povas solvi nur kune. Tio memkompreneblas, ĉar ili ekiras de du kontraŭaj punktoj de la unueca solva vojo, de tiuj punktoj, kiuj estas donitaj. Tamen potencioj de la du direktoj estas diversaj.

La direkto "de supro malsupren" -- metodologia alproksimiĝo -- povas certigi porteblecon al la produkto, ĉar post iom da paŝoj de precizigo kaj erigo de la projekto oni povas ĝin konkretigi je diversaj lingvoj sen returno al analizitaj pli supraj niveloj. La direkto "de malsupro supren" lingvistikak alproksimiĝo -- ebligas pli bone eluzi povon de la donita lingvo, kaj rezulto de evoluigo de la lingvaj rimedoj ĝi helpas malstreĉigi la bazan problemon por estontaj projektoj.

Al la unua direkto apartenas esploroj pri la struktura disciplinigo de la pensado [Dahl, Dijkstra, Hoare 1972], [red. Kjørner 1980], kaj lingvoj por reprezenti projektojn dum ilia paŝo-post-paŝa precizigo, kiel VDL [Wegner 1972], META-IV [Bjørner, Jones 1978] kaj LDM [Balogh, Santa-Toth, Szeredi 1979]. Al la alia direkto apartenas esploroj pri makrorimedoj de la programado [Brown 1969], aŭtomatigo de produktado de tradukiloj [Brooker, Morris 1960], [Feldman 1966], [Ferguson 1966], [Feldman, Gries 1968], [Lecarme 1977].

Analizante la du menciitajn alproksimiĝojn, ni trovas, ke la unua iras de abstrakta problemo al konkretaj lingvaj rimedoj, kaj la dua iras inverse. Ĉi tiu dua direkto estas pli malfacila, ol la unua, ĉar tiu fakte havas naturan rimedon - la homan menson -- dum tiu ĉi nuntempe havas nur malmultajn taŭgajn rimedojn.

La homaj lingvoj (ĉu naturaja ĉu artefaritaj), kiel konate, estas produktoj de la homa socio kaj evoluiĝas kun la socio. La praktiko ellaboras novajn nociojn. La lingvo estas rimedo por ellabori ilin, kaj por enpreni kaj heredigi ilin al novaj generacioj. Estas lingvoj mortaj (ekzemple: latino) kiuj ne plu faras tion. La vivaj lingvoj riĉiĝas kaj evoluiĝas laŭ praktikaj bezonoj de la socio.

La prelego emfazas, ke la plejmulto de konataj programaj lingvoj estas "mortaj" en analogia senco, kaj por pli bone solvi nian bazan pragmatikan problemon ni bezonas "vivajn" programlingvojn. Vere, evoluo de programaj lingvoj fakto estas ilia ŝanĝo al aliaj, novaj lingvoj. Tiel aperas post COBOL kaj FORTRAN la lingvo PL1, post PL1, ALGOL, PASCAL kaj aliaj tiel aperas la lingvo ADA. La programaj lingvoj havas difinitan aron de nocioj, abstraktaĵoj, kaj la programistoj havas por uzi nur tiun, ĉiam difinitan, aron.

Por realigi komplikan projekton surbaze de relative simpla lingvo, irante "de malsupro supren", oni bezonas rimedojn por difini novajn nociojn, por larĝigi kaj evolui la bazan lingvon. Tiaj rimedoj estas makroj, uzeblaj en multaj lingvoj de asemblila nivelo. Certaj operaciaj sistemoj helpas konservi kaj heredigi bone funkciantajn makrojn. Estas necese analizi la problemon de ŝanĝiĝo kaj evoluiĝo de lingvoj, kaj la respektivajn trajtojn de la procezoj. La prelego resumas la ĉefajn rezultojn de tiu ĉi (ankoraŭ daŭrigota) laboro.

La plenumitan laboron determinis semiotika pozicio de la aŭtoro (grave influita de [Zemanek 1966]). Al la konataj (devenintaj de Chomsky [Chomsky 1959]) sintaktikaj nocioj estas aldonitaj nocioj de teksto kaj korpuso, kio rezultis pli riĉan prezentadon de la sintaktiko. Estas konstruita semantika modelo por la komputilaj lingvoj, kaj la nocio de algoritmo estas (post Markov [Markov 1954]) redifinita helpe de tiu modelo. Kiel baza pragmatikaĵo, estas enkondukita la nocio de abstrakta procezilo, kaj estas klasifikitaj la abstraktaj proceziloj.

Post semiotika esploro de la lingvaj proceziloj estas analizitaj eblecoj de algoritmaj transiroj en la konstruita semantika modelo, konataj (tradukado, interpretado, makroekstenso) kaj nekonata (formala abstrakcio). Estas difinita kaj analizita la klaso de dinamikaj proceziloj (proceziloj, kies lingvoj dumlabore ŝanĝiĝas). Fine, estas praktike realigitaj diverspecaj dinamikaj proceziloj.

La praktike realigitaj dinamikaj proceziloj estas aplikitaj Por programi procezdirektan realtempan operacian sistemon per mikroprocezila aparato. Ŝajnas, ke estis sukcese malstreĉigita la baza pragmatika problemo de la arto de programado.

1. Bazaj nocioj

Nia celo estas analizi la problemon de aŭtomata ŝanĝiĝo kaj evoluigo de programaj lingvoj, kaj la respektivajn trajtojn de proceziloj. Tiu ĉi problemo en la komputiko dume ne estas esplorata. Dume eĉ ne estas klara, ĉu ekzistas programaraj proceziloj kun tiaj trajtoj aŭ ne. Estas postulate, ke ili ekzistu. Por fari la unuajn paŝojn ni bezonas sistemon de nocioj, sufiĉe ĝeneralaj por priskribi ajnan programadan lingvon kaj ajnan programaran procezilon.

La programadajn lingvojn ni konsideras semiotikaj sistemoj (do programistoj kaj proceziloj estas konsiderataj subjektoj de tiaj sistemoj). La bezonata sistemo de nocioj deskribas sintaktikon, semantikon kaj pragmatikon de la programadaj lingvoj.

1.1. Sintaktikaj bazoj

Al la bazaj sintaktikaj nocioj apartenas kelkaj klasikaj nocioj [Chomsky 1959]:

alfabeto A -- finita subaro $A \subset \Sigma$ de ne pli ol numerebla aro $\Sigma = \{s_1, s_2, \dots\}$ de karaktroj s .

stringo σ -- vico el (aŭ vektoro sur) alfabeto A , havanta finitan longon $\rho\sigma$; $0 < \rho\sigma < \infty$ (aŭ rangon $\rho\sigma$). La i -an karaktron de la stringo σ ni signas per $\sigma[i]$.

Speciala stringo ϵ , konsistanta el nulo da karakteroj,

$$\rho\epsilon = \theta$$

estas enkondukita kaj nomita vakua stringo.

A^* signas la aron de ĉiuj nevakuaĵ stringoj kaj $A^* = A^+ \cup \{\epsilon\}$ signas la aron de ĉiuj stringoj sur la alfabeto A .

lingvo \mathcal{L} -- ajna subaro de A^* .

Lingvo, rekursive numerebla, povas esti reprezentita per

gramatiko G -- $G = (N, A, P, S)$, kie

A -- alfabeto (aŭ terminalaro)

N -- finita neterminalaro, $N \cap A = \emptyset$

S -- starta karaktero, $S \in V$; $V = N \cup A$

P -- finita produkciaro:

$$P = \{ \xi \rightarrow \eta \mid \xi \in V^* \times N \times V^* \ \& \ \eta \in V^+ ; V = N \cup A \}.$$

Krom tiuj ĉi konataj nocioj ni bezonas kaj

enkondukas kelkajn novajn:

teksto t -- finitlonga nevakua vico el la aro A^* .

korpuso T -- ajna aro de tekstoj sur la alfabeto A .

Estas facile pruvi la ekvivalentecon de tekstoj kaj stringoj. Se al la karakteraro Σ aldoni novan karakteron \cdot kaj uzi ĝin en tekstoj kiel separilon de stringoj, do ĉiuj tekstoj sur Σ povas esti konsiderataj stringoj sur $\Sigma^{\cdot} = \Sigma \cup \{\cdot\}$ kaj inverse.

La nocio de teksto, tamen, por ni estas utila. Se teksto t konsistas nur el stringoj de lingvo \mathcal{L} , ni diras ke la teksto t estas en la lingvo \mathcal{L} , kaj ni skribas $t \in \mathcal{L}$. (Do $t \in \mathcal{L}$ signifas, ke t ne havas sintaktikajn erarojn laŭ la lingvo \mathcal{L} .)

Havante iun lingvon \mathcal{L} , ni povas distingi korpusejn sur \hat{G} . La plej vasta korpusejo estas la aro de ĉiuj tekstoj $T = \{t | t \in \mathcal{L}\} = \mathcal{L}^+$. Malpli vasta korpusejo estas la aro de ĉiuj programaj tekstoj. Ankoraŭ malpli vasta -- la aro de ĉiuj strukturitaj programoj ktp.

Rimarkeblas analogieco de la nocioj:

stringo -- teksto,

lingvo -- korpusejo.

Stringo estas vico el alfabeto, dum teksto estas vico el stringaro. Lingvo estas stringaro, dum korpusejo estas tekstaro. Tiu ĉi "duetaĝa" sintaktiko estas utila por esploro de proceziloj.

Ni aldonu ankoraŭ paron de analogiaj nocioj:

gramatiko -- sintakso.

Gramatiko (en la klasika senco de Chomsky) limigas kaj difinas lingvon. La nocio "sintakso" ni uzas en la senco de "io, limiganta kaj difinanta korpusejo". Do estas "nula sintakso", difinanta \mathcal{L}^+ , "programa sintakso", difinanta korpusejo de ĉiuj programoj, "strukturiga sintakso", difinanta korpusejo de ĉiuj strukturitaj programoj, ktp.

Oni ne strebu difini pli precize la sintakson. Por certaj korpusejoj (ekzemple por \mathcal{L}^+) sintakson povas anstataŭi gramatiko, sed ĝenerale tio aŭ ne eblas, aŭ ne estas racie. Ĝenerale la sintakso devas esti intuitiva nocio. (Kiamaniere difini gramatikon por generi ĉiujn strukturitajn programojn, aŭ, en homa lingvo, ĉiujn sonetojn?)

1.2. Semantika modelo

Por evoluigi la nocian sistemon necesas scii, ke ĉiu lingvo havas semantikon. Ni ne endetaliĝu pri la semantiko de la semantiko. La semantiko estu reprezentita nur formale, pere de rilatoj inter certaj aroj.

Ni konsideru, ke al ĉiu stringo de ajna lingvo apartenas elemento de iu, sufiĉe granda, aro K :

$$\forall L \forall \sigma \in L \exists k \in K \{ \sigma v k \}.$$

La rilaton v ni nomas rilato de notado. La noton $\sigma v k$ legu jene: σ signifas k , aŭ k estas la koncepto de σ .

Krome ni konsideru, ke ekzistas aro R , ankaŭ sufiĉe granda, tia, ke ajna elemento $k \in K$ havas certan rilaton ω al iu elemento $r \in R$:

$$\forall k \in K \exists r \in R \{ k \omega r \}.$$

La rilaton ω ni nomas rilato de reflektado. La noton $k \omega r$ legu jene: k reflektas r , aŭ r estas la referento de k .

La sistemon de aroj L, K, R kaj rilatoj v, ω ni nomas semantika modelo:

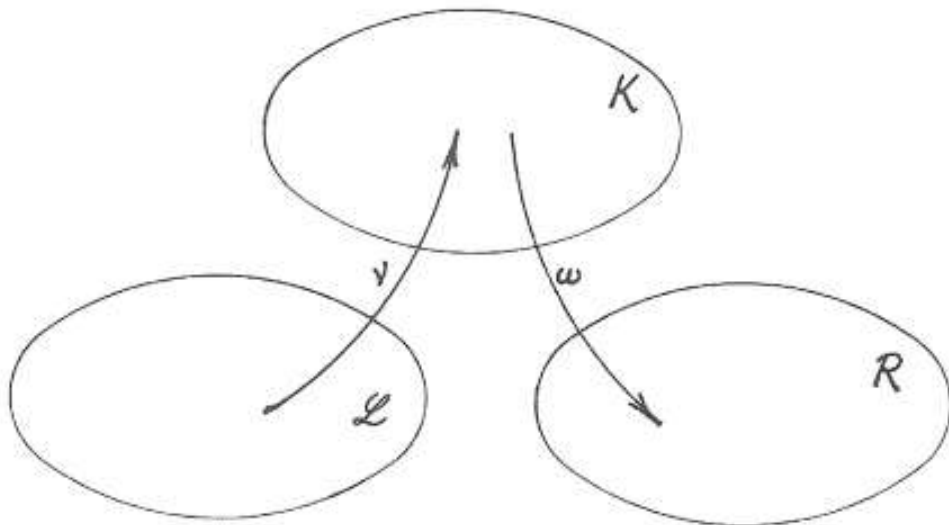
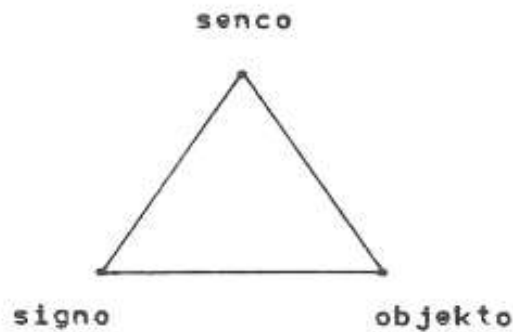


Fig. 1. Semantika modelo

Ĝin ni konstruis laŭ la konata skemo de Frege [Church 1956]:



Oni diras, ke la signo signifas la sencon, la senco reflektas la objekton, kaj la signo signas la objekton.

Ni ĝeneraligu la modelon. K kaj R estu komunaj por ajna lingvo $\mathcal{L}_1, \mathcal{L}_2, \dots$ de certa klaso. Pro tio signu $S(\mathcal{L}) = \{k \in K \mid \exists v k; \sigma \in \mathcal{L}\}$ la semantikan areon kaj $R(\mathcal{L}) = \{r \in R \mid \exists w r; k \in S(\mathcal{L})\}$ signu la objektan areon de la lingvo \mathcal{L} .

Estu teksto t skribita en iu lingvo \mathcal{L} : $t \in \mathcal{L}$. Ĝi konsistas el stringoj $\sigma_1 \sigma_2 \dots \sigma_n$. Ĉiu stringo σ_i signifas iun koncepton $\sigma_i \vee k_i$. Ni nomu la fono $\psi(t) \in K$ de la teksto t la vicon de tiuj ĉi konceptoj k_i . El la ĵusa ĝeneraligo de la modelo sekvas, ke $\psi(t) \in K$; la fono $\psi(t)$ de ajna teksto t mem estas koncepto.

1.3. Algoritmoj, proceziloj

Surbaze de ĉi tiuj sintaktikaj kaj semantikaj nocioj nun ni povas difini la nociojn de algoritmo kaj procezilo. La nocion "algoritmo" ni redifinas sekvante al Markov [Markov 1954]:

(1) Algoritmo, en nedefinita senco, estas ajna efektive kalkulebla funkcio sur la aro de ĉiuj tekstoj $T = \{\Sigma^*\}^+$.

(2) En ekzakta senco, algoritmon ni difinas jene:

Ni nomu operacio ajnan finite reprezenteblan subaron de $T \times T$; $T = \{\Sigma^*\}^+$. Do operacio estas aro de paroj $\{t_{x_1} \rightarrow t_{y_1}, t_{x_2} \rightarrow t_{y_2}, \dots\}$, havanta finitan reprezenton. Signu \mathcal{O} la klason de ĉiuj operacioj.

Lingvo \mathcal{L} estu nomita algoritma, se $S(\mathcal{L}) \subset \mathcal{O}$. Fono $\varphi(t)$ de ajna teksto $t \in \mathcal{L}$ estu nomita algoritmo, se \mathcal{L} estas algoritma lingvo.

Fakte tiu ĉi difino de la algoritmo konformiĝas al la Markov'a difino de la "normala algoritmo". La diferenco estas nur tio, ke la ĵusa difino estas semiotika: ĉiu algoritmo estas koncepto el la aro K . La operacioj mem, el kiuj ĝi "konsistas", estas objektoj el la aro de referentoj R . La teksto t kaj la stringoj de tiu teksto, estas signoj, reprezentantaj la algoritmon, kaj, respektive, la operaciojn, en la donita lingvo.

Stringoj, signantaj operaciojn aŭ signifantaj algoritmojn, nomiĝas imperativaj stringoj. Ĉiuj aliaj stringoj nomiĝas deklarativaj. La distingon ni trovas esenca, ĉar en programaj tekstoj la du tipoj aperas miksite, sed havas diversajn semantikajn funkciojn kaj tial ili estas uzataj diverse. Multobla uzo de deklarativaj stringoj en la sama programo estas superflua, pro tio tiaj stringoj aperas en la teksto nur unu fojon, dum imperativaj stringoj aperadas en la teksto multfoje. Ĝenerale por deklarativaj stringoj ne gravas ilia ordo en la teksto, dum imperativaj stringoj devas eniri la tekston en fiksita ordo. Yiu ĉi rimarko helpas nin formale paroli pri la ekvivalenteco de tekstoj.

Por tekstoj la ekvivalenteco povas esti difinita diverse. Estas speciale interesa ilia koncepta ekvivalenteco.

Ni supozas, ke sur la konceptaro K estas donita rilato de ekvivalenteco. (Ekzemple algoritmojn ni konsideras ekvivalentaj en la Markova senco [Markov 1954].)

Tekstoj povas esti konsiderataj vicoj de subtekstoj. Du tekstoj t kaj t' estas koncepte ekvivalentaj, se oni povas disigi ilin je subtekstoj: $t = t_1 t_2 \dots t_n$ kaj $t' = t'_1 t'_2 \dots t'_n$, tiel, ke la subtekstoj t_i kaj t'_i estas koncepte ekvivalentaj. Du tekstoj (aŭ subtekstoj) t kaj t' estas koncepte ekvivalentaj, se ili diferencas nur je ordo de la enhavataj deklarativaj stringoj, aŭ se la fonoj $\psi(t)$ kaj $\psi(t')$ estas ekvivalentaj algoritmoj.

Laŭ la difino, algoritmoj transformas korpusojn al korpusoj. Oni povas noti

$$\alpha : X \rightarrow Y$$

kie α signas algoritmon, X signas ĝian eniran korpuson, kaj Y signas ĝian eliran korpuson. Estu donitaj rilatoj de ekvivalenteco \tilde{x} kaj \tilde{y} sur la korpusoj X kaj Y respektive. Ni diras, ke α konservas la klasojn de ekvivalenteco, se ĝi transformas ekvivalentajn tekstojn al ekvivalentaj, kaj nekvivalentajn al nekvivalentaj. Se la algoritmo α mem estas reprezentita en lingvo \mathcal{L} per teksto t (alifraze se $\alpha = \gamma(t)$; $t \in \mathcal{L}$) kaj ĝi konservas la klasojn de ekvivalenteco, do la kvaropon $\pi = (X, Y, t, \mathcal{L})$ ni nomas abstrakta procezilo. (Nia formala difino de la abstrakta procezilo konformiĝas al neformala difino de ordinara proceziloj -- tradukaj programoj, asembliloj kaj aliaj -- donita de Dömölki [Dömölki, Rajki 1973].)

Se $X \subset \mathcal{L}_x^+$ kaj $Y \subset \mathcal{L}_y^+$ estas korpusoj de tekstoj, reprezentitaj en lingvoj X kaj Y , kaj la rilatoj \tilde{x} kaj \tilde{y} estas la rilato de la koncepta ekvivalenteco, do la abstraktan procezilon $\pi = (X, Y, t, \mathcal{L})$ ni nomas lingva procezilo. La lingvojn \mathcal{L}_x , \mathcal{L}_y , \mathcal{L} ni nomas respektive ĝiaj enira, elira kaj progra lingvoj.

Ekzistas, certe, ankaŭ nelingvaj proceziloj. Ekzemple abstrakta ordigilo, kiu leksikografike ordigas stringojn en ajna teksto sur alfabeto A , ne estas lingva procezilo. La sekvanta parto de la prelego traktos pri la lingvaj proceziloj.

2. Algoritmaj transiroj en la semantika modelo

Ĉar nia difino de la abstrakta procezilo baziĝas sur semiotika difino de la nocio "algoritmo", ni povas klasifiki la lingvajn procezilojn laŭ la realigitaj de ili algoritmaj transiroj en la semantika modelo.

2.1. Tradukado kaj interpretado

Lingva procezilo povas transformi tekstojn tiel, ke ajna enira teksto estas koncepte ekvivalenta al tiu elira teksto, en kiu la procezilo ĝin transformas. Tiujn lingvajn procezilojn ni nomas abstraktaj tradukiloj.

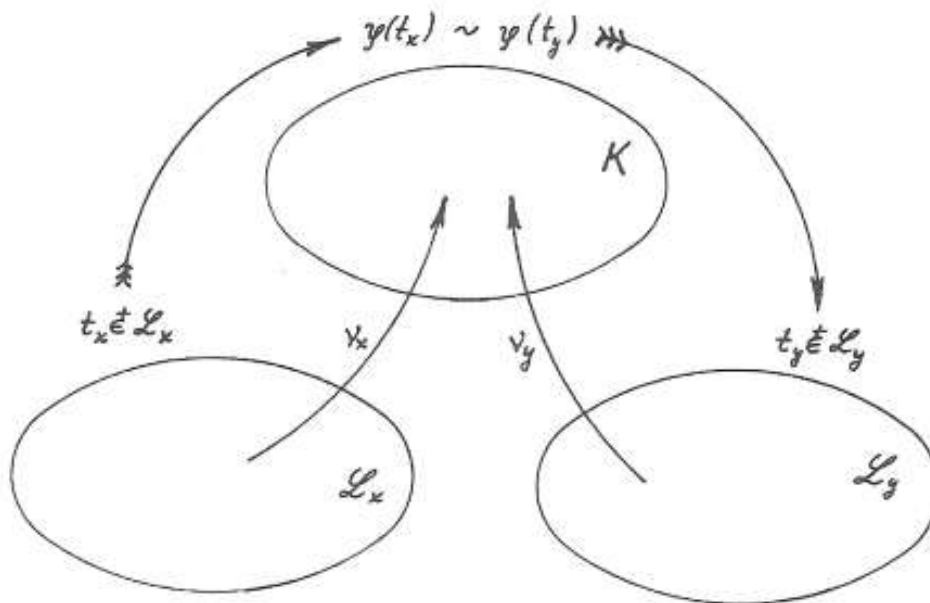


Fig. 2. Skemo de la tradukado

Do la tradukilo trovas laŭ la enira teksto $t_x \in L_x$ ĝian fonon $y(t_x)$, poste laŭ alia, ekvivalenta fono $y(t_y)$ ĝi trovas ĝian novan reprezenton $t_y \in L_y$ (Fig. 2.).

Ni rigardu abstraktan procezilon $\pi = (X, Y, t, \mathcal{L})$.
 Supozu, ke X estas korpuso sur algoritma enira lingvo \mathcal{L}_x
 kaj \approx estas la rilato de la koncepta ekvivalenteco.
 Supozu ankaŭ, ke $Y = \mathcal{R}(\mathcal{L}_x)$ estas la objekta areo de la
 enira lingvo. Ĉi-kaze la procezilon π ni nomas abstrakta
interpretilo. Interpretilo ŝanĝas ĉiun stringon de la
 enira teksto al ĝia referento (Fig. 3.):

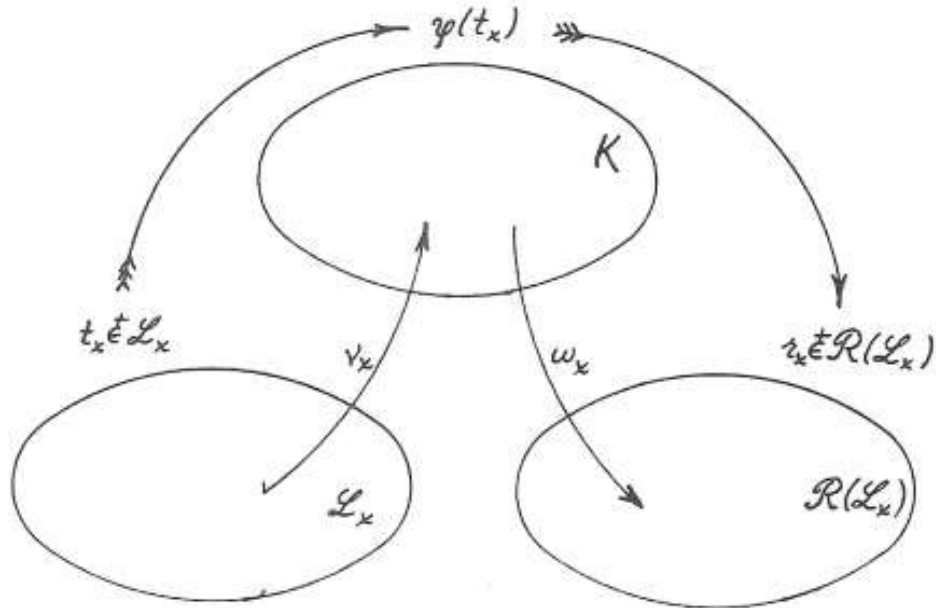


Fig. 3. Skemo de la interpretado

Estu la procezilo $\pi = (X, Y, t, \mathcal{L})$ interpretilo.
 La referentoj de stringoj de la enira lingvo \mathcal{L}_x estu
 finitaj vicoj de operacioj, reflektataj de algoritmoj
 al $\mathcal{S}(\mathcal{L})$. Ĉi-kaze la interpretilon π ni nomas abstrakta
mikroprocezilo (Fig. 4.):

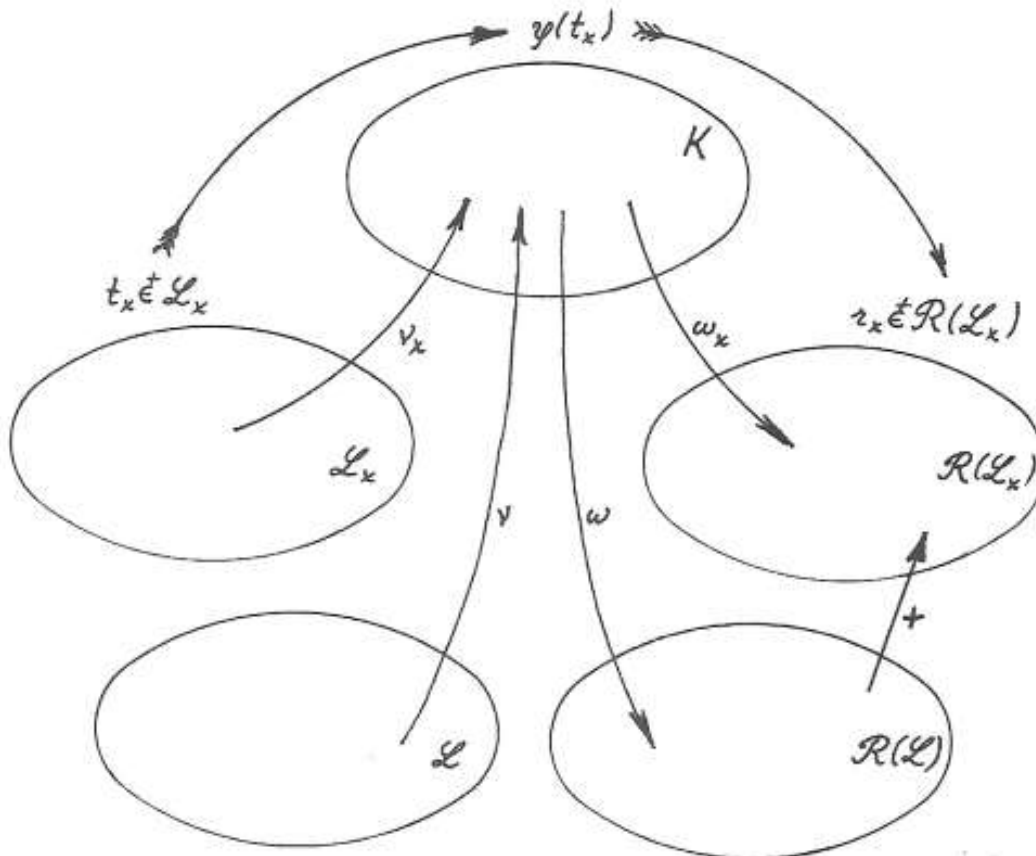


Fig. 4. Skemo de mikroprocezila interpretado

Sur la figuro 4. la "+"-markita sago signas, ke $\mathcal{R}(\mathcal{L}_x)$
 mem estas lingvo sur $\mathcal{R}(\mathcal{L})$, kiel alfabeto, alifraze:
 $\mathcal{R}(\mathcal{L}_x) \subset \{\mathcal{R}(\mathcal{L})\}^+$.

Estu la procezilo $\Pi = (X, Y, t, \mathcal{L})$ interpretilo, kaj la objekta areo $\mathcal{R}(\mathcal{L}_x)$ de la enira lingvo \mathcal{L}_x estu korpuso sur elira lingvo $\mathcal{L}_y \subset \mathcal{L}_x$. Ĉi-kaze la interpretilon ni nomas abstrakta makroprocezilo. La tekstoj el la objekta areo $\mathcal{R}(\mathcal{L}_x)$ nomiĝas makroj, kaj la stringoj de la enira lingvo \mathcal{L}_x nomiĝas makrovokoj.

Estas facile videbla, ke makroproceziloj estas kaj tradukiloj kaj interpretiloj. La figuro 5. montras la realigitan de ili algoritman transiron:

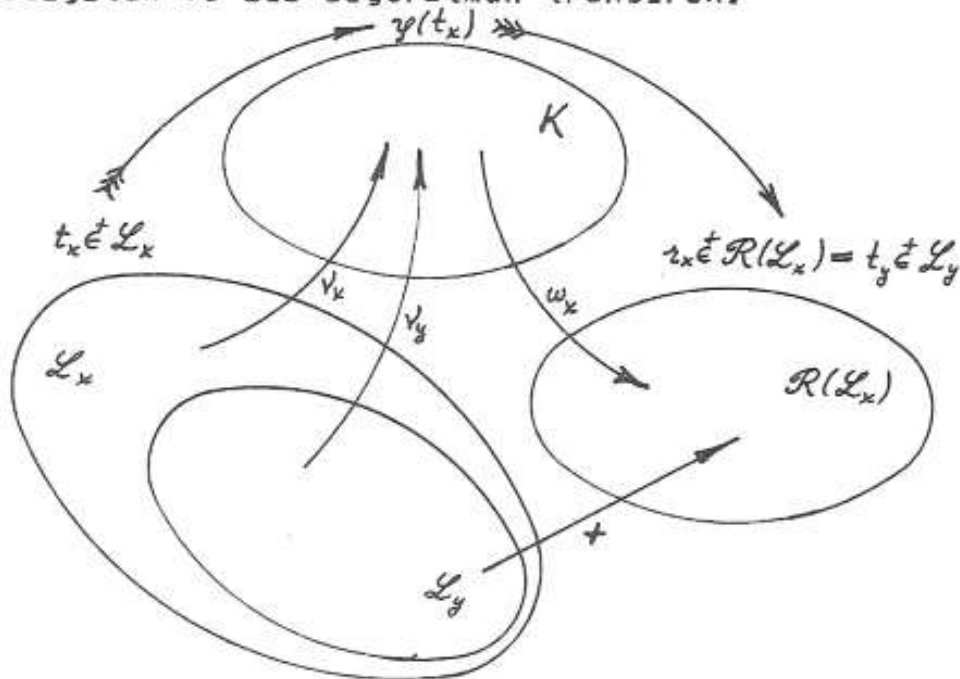


Fig. 5. Skemo de makroprocezila interpretado

Sur la figuro 5. la "+"-markita sago signas, ke $\mathcal{R}(\mathcal{L}_x)$ mem estas lingvo sur \mathcal{L}_y , kiel alfabeto, alifraze: $\mathcal{R}(\mathcal{L}_x) \subset \mathcal{L}_y^+$.

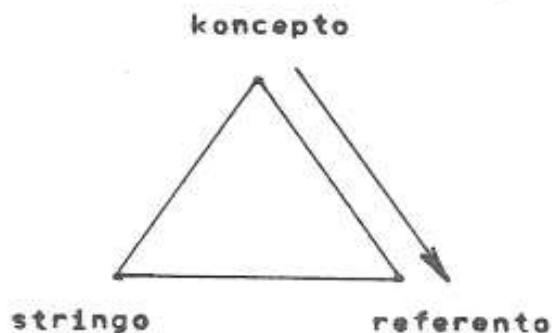
2.2. Konkretigo kaj abstrakcio

Makroprocezilo transformas sian eniran tekston $t_x \in L_x$ al elira teksto $t_y \in L_y$, dum

- $L_y \subset L_x$
- $R(L_x) \subset L_y^+$.

Makroprocezilo transformas de makrovokojn al makroj. La rilato de makrovokoj kaj makroj por \hat{g} estas donita "en \hat{g} generala formo", nomata makrodifino. (Per semiotika esprimo: makrodifino specifikas "semiozon", nome la signiĝon de la makrovoko.)

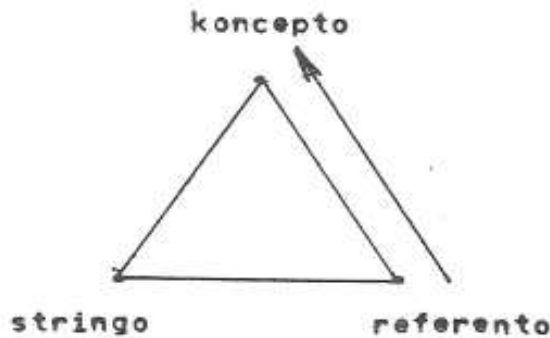
Ricevinte makrovokon, makroprocezilo memorfiksas \hat{g} generalan modelon de sia estonta agado, la algoritmon de la makroekstenco (transiro de la \hat{g} generala formo de makro al \hat{g} konkreta formo). Do, makroprocezilo enhavas algoritmon de konkretigo. Sur la skemo de Frege tio aspektas:



kie:

stringo	--	makrovoko
koncepto	--	makro en \hat{g} generala formo (abstraktaĵo)
referento	--	makro post makroekstenco (konkretaĵo).

Do makroekstenso reprezentas la algoritmigon de la "konkretigo". (La sama operacio estas en bazo de la programada lingvo REFAL [Turĉin 1974].) estas interesa la inverza operacio, la abstrakcio:



Necesas rimarki, ke ĉi tie la vortoj "konkretigo" kaj "abstrakcio" ne estas komprenendaj tiel, kiel ilin komprenas psikologistoj aŭ lingvistoj. Nin interesas la formala flanko de ĉi tiuj procezoj:

konkretigo -- en la REFALa senco,
abstrakcio -- malkonkretigo.

Ĉu eblas doni algoritmon de abstrakcio? Igi la komputilon formi koncepton surbaze de aro de objektoj kaj enkonduki novan signon en sian lingvon por tiu nova koncepto? La demando estas aktuala, kiam oni klopodas proksimigi maŝinajn lingvojn al homaj. Ĝi kondukas nin al tiaj esplorterenoj, kiel "artefarita intelekto", "rekono de figuroj".

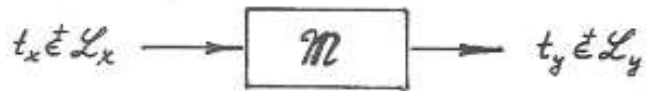
La problemo havas ankaŭ praktikan intereson. en la programada tekniko makroj estas uzataj por

- komfortigo de la programado,
- plialtigo de la lingva nivelo,
- transplanto de programoj,
- pliefektivigo de la programado (pere de la uzo de makrobibliotekoj), ktp.

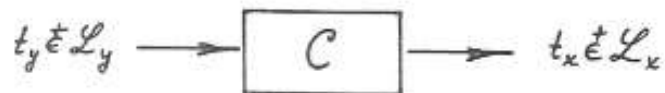
La makroj ekestas en la kapo de la programisto. Estas interese kompreni kaj algoritmiigi lian intelektan laboron. (Per semiotika vorto: ankaŭ tio estas algoritmiigo de semiozo, signiĝo, sed nun ĝi estas kontraŭdirekta, de iranta de la konkreta objektaro al la koncepto kaj poste al la signo.)

Ni formulos taskon. Necesas konstrui procezilon, plenumantan la inverzon de la makroekstenco. Tiun "mal-makroprocezilon" ni nomos konceptumilo.

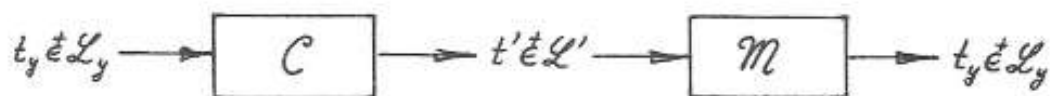
Ni havu makroprocezilon \mathcal{M} , transformantan eniran tekston $t_x \in \mathcal{L}_x$, skribitan en makrolingvo \mathcal{L}_x , al elira teksto $t_y \in \mathcal{L}_y$, skribita en baza lingvo \mathcal{L}_y :



Verŝajne estus troe postuli, ke konceptumilo \mathcal{C} rekreu la eniran tekston t_x surbaze de la elira:



Pro tio ni malfortigas la postulon jene (la esencon ja tio ne koncernas):



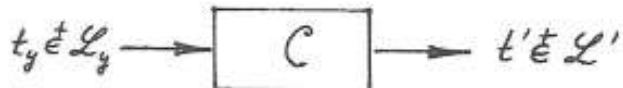
Evidenta kontentigo de la postulo per simpla C , kopianta sian eniran tekston (do la kazo de $t' = t_y$ kaj $\mathcal{L}' = \mathcal{L}_y$) ne estas interesa. Lingvo $\mathcal{L}' \supset \mathcal{L}_y$ devas esti makrolingvo.

Supoze, ke la origina teksto $t_x \in \mathcal{L}_x$ enhavis makrovokojn por kompakta noto de oftaj subtekstoj, konceptumilo povas labori tiel:

- 1) Diserigas la tekston je stringoj (en kazo de pli komplike strukturitaj tekstoj eble eĉ je vortoj kaj morfoj).
- 2) Analizas la ligojn de la teksteroj.
(Analizo de kontekstoj, interreferencoj ktp.)
- 3) Retrovas la "similajn" partojn en la teksto.
(Similajn kaj samajn. Por tio ĝi komputas ian mezuron de konstanteco de tiuj ligoj.)
- 4) Plenumas la ĝeneraligon kaj abstrakcion.
(Enkondukas makrodifinojn kaj laŭgrade larĝigas la lingvon \mathcal{L}_y , konstruadas do la makrolingvon \mathcal{L}' .)
- 5) Rekonstruas la tekston kompaktigita.

Tio ĉi ankoraŭ ne estas algoritmo, ĉar ne estas klara, ĉu eblas plenumi ĉion ĉi efektive aŭ ne. Ne klaras, kia "mezuro de konstanteco de ligoj" direktas la laboron de la konceptumilo. Ni precizigu la nocion.

Temas pri mezuro, kiu estas pli granda en la enira teksto t_y de la konceptumilo, ol en ĝia elira teksto. Ĝi dependas de la multaspececo en la tekstoj. La elira teksto de la konceptumilo estas pli kompakta ol la enira, do en ĝi la ligoj estas malpli konstantaj, se la meza multaspececo de la tekstoj en ĝi estas pli granda. Facile videblas, ke temas pri la sintakso:



tekstoj estas
pli multaspecaj
sintaksaj limigoj
estas malpli grandaj

Nun jam ni povas priskribi efektivan procezon de komputo de la sintakso, konsekvence algoritmon de konceptumilo:

- 1) Diserigu la tekston je stringoj.
- 2) Trovu la egalajn stringojn.
- 3) Se estas egalaj stringoj, trovu la egalajn parojn de intersekvaj stringoj.
- 4) Daŭrigu tion (trovu la egalajn triopojn, kvaropojn ktp.).
Transiru al 5), se en la teksto ĉiuj n -opoj diversas. (tio estas la maksimumo de la multaspececo.)
- 5) Se $n > 2$, tiam formulu makrovokojn por la egalaj $(n-1)$ -opoj, male transiru al 8).
- 6) Ŝanĝu en la teksto la $(n-1)$ -opojn al iliaj makrovokoj.
- 7) Konsiderante la rezultan tekston enira, transiru al 1).
- 8) Eldonu la tekston.

Tia konceptumilo laboras surbaze de plena egaleco de n -opoj (alifraze: n -longaj subtekstoj). Eblas konstrui ankaŭ tian konceptumilon, kiu baziĝas sur neplena egaleco, ekzemple permesante en la n -opoj "giĉetojn", en kiuj la egaleco ne estas kontrolata. Sekve makrovokoj formuliĝas kun parametroj. Tiaspecaj malsimpligoj de la konstruo rezultas aldonajn eblecojn por kompaktigi tekstojn per konceptumilo.

Konceptumilo, certe, povas labori per plena transrigardo de tekstoj, statistiko kaj decido sur tiu bazo pri la taŭgeco de tiuj aŭ aliaj makrovokoj. Tamen, estas senduba, ke tiu ĉi vojo estas malbona. En la laboron de la konceptumilo necesas enkonduki heŭristikajn principojn. La metodoj de programado de ludoj, kiel ekzemple ŝako, donos helpon [Botvinnik 1975], [Krinickij 1970].

3. Dinamikaj proceziloj

Se la konceptumilon eniras diversaj tekstoj $t_1 \in L_1, t_2 \in L_2, \dots$, ĝi eldonas diversajn tekstojn $t_1' \in L_1', t_2' \in L_2', \dots$, dum ankaŭ la lingvoj $L_1' \neq L_2' \neq \dots$ estas diversaj. La algoritmo de la konceptumilo, certe, konservas la klasojn de la koncepta ekvivalenteco, pro tio ĝi estas lingva procezilo. Kompare al "kutimaj" lingvaj proceziloj ĝi havas gravan distingan econ: ĝia elira lingvo dum ĝia laboro iĝas pli kaj pli vasta. Ĝeneraligante ĉi tiun econ ni enkondukas novan fakvorton.

Procezilon ni nomas dinamika, se dum ĝia laboro ŝanĝiĝas ĝia enira, aŭ elira, aŭ propra lingvo.

En ordinara makroprocezilo la enira lingvo ŝanĝiĝas ĉe ĉiu makrodifino, do ankaŭ makroproceziloj estas dinamikaj proceziloj. La makrodifinoj, kiujn la makroprocezilo trovas en sia enira teksto, montras al ĝi ĉiun vastigon de ĝia enira lingvo.

Dinamikan procezilon ni nomas pasiva, se la ŝanĝigoj de la enira, elira, aŭ propra lingvo estas por ĝi indikitaj de certaj stringoj, havantaj ĝuste tiun sencon.

Por rekoni la situaciojn, kiam lingvo devas ŝanĝiĝi, aktiva dinamika procezilo uzas iun mezuron, difinitan sur la teksto. En la kazo de konceptumilo la mezuro estas la multaspececo de la teksteroj, do ĝi radikiĝas en la sintakso. Se elekti iajn aliajn mezurojn, difiniĝas aliaj dinamikaj proceziloj. Kvankam teorie tio estas klara, ne estas facile eltrovi praktike interesajn kaj konstruktivajn difinojn.

Pli facila estas la afero en la kazo de pasivaj dinamikaj proceziloj. La aŭtoro difinis kaj realigis praktike du tipojn el tiu klaso. Sube ili estas koncize prezentitaj.

Malplena procezilo

Ĉiun lingvan procezilon eblas realigi malplena. Ekzemple, malplenan asemblilon oni povas konstrui jene: Oni komence limigas la eniran lingvon de la asemblilo tiel, ke ĝi instrukciojn tute ne havu, havu nur direktivojn. Oni konstruas ĝeneraligitan asemblilon, kapablan interpreti direktivojn, formi printliston, eldoni asemblitan programtekston. Tia asemblilo enhavas tabulon de simbolaj nomoj (komence malplenan), interpretilon de adresesprimoj, semantikajn klasojn de eblaj instrukcioj, do ĉion krom la eniraj instrukciaj mnemonikoj.

La enirajn instrukciajn mnemonikojn oni povas indiki al la asemblilo per speciala direktivo, difinanta instrukcion (ni ĉiam uzis por tiu celo specialigitan formon de la direktivo EQU), ekzemple

MOV! EQU 940

La asemblilo metas la nomon MOV kaj la valoron 40 en la simboltabulon, kaj scias (pro la krisigno) ke MOV estas nomo de instrukcio kun baza kodo 40, kaj ĝi apartenas al la 9-a semantika klaso de instrukcioj. Videblas, ke oni povas larĝigi la komence nur-direktivan eniran lingvon al la dezirata asemblila lingvo, kaj eĉ plu. La plej grava estas ĉi tio "kaj eĉ plu", ĉar havante tian asemblilon la uzanto povas programi ekzemple interpretilon de iu lingvo \mathcal{L} , iom pli altnivela ol la maŝinkoda nivelo, kaj dume "instrui" al la asemblilo respektivan simbolan lingvon \mathcal{L}' . Asemblinte la interpretilon, la asemblilo iĝas tradukilo de tiu lingvo \mathcal{L}' al la lingvo \mathcal{L} .

Por malplena procezilo la lingvolarĝigon eblas indiki ankaŭ per aliaj metodoj. Ni realigis ekzemple programon por aŭtomatigi difinon de komputilaj konfiguroj. La konfiguron oni devis priskribi en speciala lingvo, kiun povis difini la uzanto mem. La metodo de la lingvolarĝigo estis jena: se la uzanto "petis" nulon da iu objekto (ekzemple: "0 da magnetbenda storo"), tio signifis, ke peton de tia objekto la lingvo devos permesi (plue la enira teksto povos enhavi similajn stringojn). Se la petata kvanto estis alia natura nombro, tiam la stringo estis kontrolita, ĉu ĝi estas en la lingvo aŭ ne. La semantiko de la difinebla enira lingvo estis fiksita en la programo (same, kiel en malplena asemblilo), sed la uzanto povis elekti la nomojn de la peteblaj objektoj laŭ sia gusto.

El-semo-procezilo

Certajn procezilojn oni povas evolui aŭtomate. Se ekzemple la semantikaj klasoj de asemblilnivela lingvo estas multnombraj sed similaj, indas skribi programon, generontan la priskribon de tiuj klasoj. Anstataŭ la procezilo oni skribas "semon", enhavantan abstraktan priskribon de la instrukcioj kaj algoritmon de la kreskigo de la semo al preta procezilo.

La objekto estas tre dinamika (ŝanĝiĝas dum tiu ĉi "evoluo" ĝia propra lingvo), sed la dinamiko estas ĉiudetaŭle planita de la programisto. (Tiu ĉi principo estis realigita en aŭtokodilo de sistemo PROCESS 2/0 de komputilo R10 [Szőke 1978].)

La metodo certe ne estas universala, ĉar tiel realigebla programo devas havi "ĉelan" strukturon, kiel kreskaĵoj. Tamen, se la programplanisto povas antaŭvidi tian strukturon de la programo, tiam la realigo tute ne estas malfacila.

4. Konkludoj

Problemon, pro sia naturo malsimplan, eblas solvi nur malsimple. Priskribo de tia problemo povas esti simpla, se oni uzas riĉan kaj altnivelan lingvon. Tamen priskribo, ĝenerale, ankoraŭ ne estas solvo, ĉar la uzitaj lingvoelementoj mem, siavice, reprezentas novajn problemojn.

Dinamikaj proceziloj ebligas simple formuli kaj simplajn kaj pli komplikajn problemojn pere de larĝigo de la priskriba lingvo je novaj konceptoj. Ili do havigas al ni en tiu senco "vivajn" programadajn lingvojn.

Literature

[Balogh, Santa-Tóth, Szeredi 1979]

Balogh K., Santa-Tóth E., Szeredi P.: Logic based program design. Proceedings of the First Congress of the John von Neumann Computer Society, Dec. 1979. Szeged, Hungary.

[red. Bjórner 1980]

Redaktite de Bjórner D.: Abstract software specifications, 1979 Copenhagen Winter School proceedings -- Lecture notes in computer science, vol.86, Springer Verlag, 1980.

[Bjórner, Jones 1978]

Bjórner D., Jones C. B.: The Vienna Development Method: The Meta-Language -- Lecture notes in computer science, vol.61, Springer Verlag, 1978.

[Botvinnik 1975]

Ботвинник М.М.: О кибернетической цели игры. — Советское Радио, Москва, 1975.

[Brooker, Morris 1960]

Brooker R., Morris D.: An assembly program for a phrase structure language -- Comput. J. No 3 (1960) p.168

[Brown 1969]

Brown P. J.: A survey of macro processors -- Annual Review in Automatic Programming, vol. 6, part 2, pp. 37-88, Pergamon Press, 1969.

[Chomsky 1959]

Chomsky N.: On certain formal properties of grammars -- Inf. and Control, 1959, vol.2, No.2, pp. 137-167.

[Church 1956]

Church A.: Introduction to mathematical logic, Princeton 1956.

[Dahl, Dijkstra, Hoare 1972]

Dahl O.-J., Dijkstra E. W., Hoare C. A. R.: Structured programming -- Academic Press, 1972.

[Dömölki, Rajki 1973]

Dömölki B., Rajki P.: Microprogram implementation of high level languages. Proceedings of the First Hungarian Computer Science Conf., Székesfehérvár, May 21-26, 1973.

[Feldman 1966]

Feldman J O.: A formal semantics for computer languages and its application in a compiler-compiler -- Comm. ACM, vol.9, No. 1 (Jan. 1966), pp. 3-9.

[Feldman, Gries 1968]

Feldman J. A., Gries D.: Translator writing systems -- Comm. ACM, vol.11, No. 2 (Feb. 1968), pp. 77-113.

[Ferguson 1966]

Ferguson D. E.: Evolution of the meta-assembly program -- Comm. ACM, vol.9, No.3 (March 1966), pp. 190-196.

[Krinickij 1970]

Криницкий Н.А.: Равносильное преобразование алгоритмов и программирование. Советское Радио, Москва, 1970.

[Lecarme 1977]

Lecarme O.: Usability and portability of a compiler writing system -- Lecture notes in computer science, vol.47, Springer Verlag, 1977.

[Markov 1954]

Марков А.А.: Теория алгоритмов. -- Труды Математического института АН СССР им. Стеклова, том 42., Москва, 1954.

[Szöke 1978]

Сёке Л.: Применение макропроцессора для разработки саморазвивающихся программ. — Труды МЭИ, выпуск 386, стр. 69—73, Москва, 1978.

[Turčin 1974]

Турчин В.Ф.: Базисный РЕФАЛ. Описание языка и основные приемы программирования. (Методические рекомендации). ЦНИПИАСС, Москва, 1974.

[Wegner 1972]

Wegner P.: The Vienna Definition Language
-- ACM Comp. Surveys, 4, 1(1972), pp. 5-63.

[Zemanek 1966]

Zemanek H.: Semiotics and programming languages. Communications of the ACM v.9. no.3, March 1966, pp. 139-143.



SZŰKE László naskiĝis en 1943 en urbo Debrecen, Hungario. Diplomon de elektriĝeniero pri komputiloj ricevis en Moskvo en 1972. De tiu jaro konstante laboras en la hungara komputilprodukta firmao VIDEOTON kiel programar-evoluisto, nun gvidas kompililevoluan teamon.

En 1981 finis aspiranturon en Moskvo.

La publikita ĉi tie artikolo estas ekstraktita el lia kandidata disertaĵo, defendota ĉi-jare.

La plurcela ligilnivela proceduro PLP

Christian BERTIN

Resumo: Tiu referaĵo prezentas datentransigan proceduron kiu povas esti uzata por interŝanĝi datenojn inter plej diversmaniere ligitaj ekipaĵoj (ligitaj per maŝaro, ringo, buso aŭ radio) kaj tiu proceduro ofertas al la ekipaĵoj plurajn servklasojn (certigita datena flukvanto, senerara ligilorientita aŭ transakciorientita datentransigo).

La ideo de tiu proceduro venas de analizo de la avantaĝoj kaj de la malavantaĝoj de la proceduro X25 [1]. La proceduro X25 rekomdata de C.C.I.T.T por komunikado inter ekipaĵoj de uzantoj kaj publikaj pakkomutaj datenretoj konsistas el tri niveloj: la fizika nivelo (1), la ligila nivelo (2) kaj la paka nivelo (3). La ligila nivelo respondecas pri la detekto kaj la korekto de la transmisiaj eraroj kaj pri la reordigo de la ricevataj datenblokoj nomataj "framoj". La paka nivelo respondecas pri la estigo kaj detruo de la komunikvojoj, pri la multiplekso de pluraj komunikoj, pri la transigo de la datenoj de ĉiuj apartaj komunikoj kaj pri la rego de la datenfluo por ĉiuj apartaj komunikoj. Pro la divido de la taskoj inter la ligila kaj la paka niveloj, la paka nivelo disponas je nur unu servklaso: senerara ne certigita datena flukvanto. Aliflanke, X25 ne povas esti uzata en multpunkta reto kia estas buso, ringo aŭ radireto.

La nun proponata proceduro aparte taŭgas por transita proceduro (tio estas por komunikado interne de retoj inter la diversaj nodoj), por akceso al la estonta Numera Reto je Integritaj Servoj (NRIS) kaj por komunikado en lokaj retoj. Fakte tiu proceduro kreiĝis por komunikado en multpunktaj retoj kie pluraj servklasoj estas necesaj (por ekzemplo kiam oni deziras miksi parolfluojn kaj datenfluojn).

La nuna proceduro estas nur ligilnivela proceduro (nivelo 2) kaj ĝi konformas al la referenca modelo por aplikaĵoj en publikaj datenretoj nun difinata de C.C.I.T.T. kaj de I.S.O.

I. BLOKOJ DE LA PROCEDURO PLP

Ni ne precizigas kiel la blokoj estos limigitaj por transmisio, ni povos uzi HDLCan aŭ karaktran limigon, tio dependas nur de ĉiu aparta reto.

I-1. Indiko de Stacia Aktiviĝo (ISA) (opcia)

Tiu bloko estas sendata de stacio por sciigi al la aliaj stacioj ke ĝi jus aktiviĝis aŭ reinitiĝis (Fig. 1).

Parametroj de la bloko ISA:

- Origina Adreso (OA)
- Cela Adreso (CA = ĝenerala adreso)
- Bloka Tipo (BT = Indiko de Stacia Aktiviĝo)

Neniu respondo de iu ajn stacio estas atendata de la senda stacio. Tiu bloko ISA utilas kaze de reinitiĝo de stacio por pli rapide detruji ĉiujn estigitajn ligilojn kun aliaj stacioj. Tio ne estas centprocente fidinda, ĉar neniu kvitanco ekzistas; konsekvence, alia mekanismo estas necesa por eviti ke stacioj havantaj aktivajn ligilojn kun stacio kiu reinitiĝis ne perturbu la novajn ligilojn de tiu stacio.

Kaze de maŝara reto, la senda stacio estas ligita al nodo. Je ricevo de la bloko ISA, la komutilo detruos ĉiujn estigitajn ligilojn kun la senda stacio. Same kiel en multipunkta reto, la komutilo ne kvitanco tiun blokon ISA.

Kaze de ligitaj multipunktaj retoj, se la ekipaĵoj interligantaj la retojn ne distingas la ligilojn, la bloko ISA devas esti dissendata tra ĉiuj multipunktaj retoj por atingi tiujn staciojn kiuj detruos ĉiujn aktivajn ligilojn kun la senda stacio se ili havas. Se la ekipaĵoj interligantaj la retojn distingas la ligilojn, ilia konduto similas al tiu de komutilo de maŝara reto.

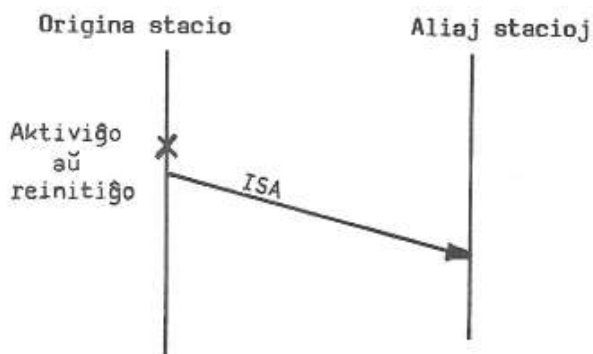


Fig.1. Aktiviĝo aŭ reinitiĝo de stacio.

1-2. Ligilestiga Peto (LEP) :

Tiu bloko estas sendata de stacio kiu deziras komuniki kun alia stacio.
Parametroj de la bloko LEP:

- Origina Adreso (OA)
- Cela Adreso (CA)
- Bloka Tipo (BT = Ligilestiga Peto)
- Origina Referenco (OR)
- Karakterizoj de la estigota ligilo:
 - o certigita datena flukvanto (sen flurego, sen erarkorekto sed kun erardetekto kaj certigita reordigo)
 - o ne certigita datena flukvanto (kun flurego, kun erara detekto kaj korekto, kun certigita reordigo kiel X25)
- Datena Flukvanto el la origino (certigita aŭ ne)
- Datena Flukvanto el la celo (certigita aŭ ne)
- Uzantaj Datenoj

Ligilestiga Peto (LEP) povos esti sendata nur post prokrasto t post la aktiviĝo aŭ reinitiĝo de stacio (tiu tempdaŭro estas duoble pli longa ol la tempdaŭro por trairi la reton), tio estas necesa por eviti ke respondo al ligilestiga peto sendita antaŭ la reinitiĝo ne alvenu post la sendo de la unua ligilestiga peto kiu sekvas la reinitiĝon (Fig. 2).

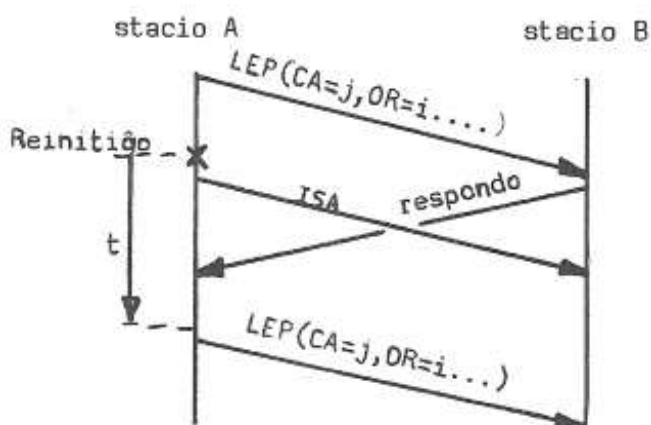


Fig.2. Reinitiĝo inter du ligilestigaj petoj.

Sekve de ligilestiga peto, tri kazoj povas estiĝi:

- neniu respondo venas (peto ne ricevita de la cela stacio aŭ cela stacio ne aktiva aŭ respondo kun transmisieraro). La origina stacio resendas sian ligilestigan peton unufoje aŭ plurfoje.
- respondo venas: akcepto de ligila estigo.
- respondo venas: rifuzo de ligila estigo.

Flanke de cela stacio, la ricevo de ligilestiga peto kun origina adreso (OA) kaj origina referenco (OR) identaj al tiuj de jam aktiva ligilo estigas la detruon de tiu aktiva ligilo kaj la akcepton de la nova peto. Tio estas necesa ĉar la cela stacio ne kapablas scii ĉu temas pri la ripeto de la sama peto aŭ pri nova peto (Fig. 3).

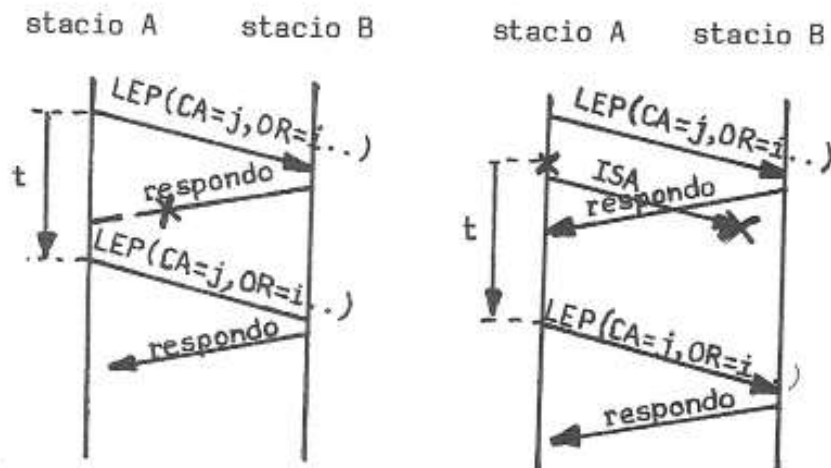


Fig.3. Ricevo de du ligilestigaj petoj (samaj originaj adresoj kaj originaj referencoj).

I-3. LigilEstiga Akcepto (LEA)

Tiu bloko estas la respondo de stacio kiu ricevis ligilestigan peton kaj kiu akceptas tiun novan ligilon (Fig. 4). Neniu respondo estas sendata post la ricevo de tiu bloko.

Parametroj de la bloko LEA:

- Origina Adreso (OA)
- Cela Adreso (CA)
- Bloka Tipo (BT = LigilEstiga Akcepto)
- Cela Referenco (CR = Origina Referenco de la LigilEstiga Peto)
- Origina Referenco (OR)
- Uzantaj datenoj

I-4. LigilEstiga Rifuzo (LER) :

Tiu bloko estas la respondo de stacio kiu ricevis ligilestigan peton kaj kiu rifuzas tiun novan ligilon (Fig. 4). Neniu respondo estas sendata post la ricevo de tiu bloko.

Parametroj de la bloko LER:

- Origina Adreso (OA)
- Cela Adreso (CA)
- Bloka Tipo (BT = LigilEstiga Rifuzo)
- Cela Referenco (CR = Origina Referenco de la LigilEstiga peto)
- Rifuzo Kaŭzo (RK = saturado, decido de supera nivelo..)
- Uzantaj datenoj.

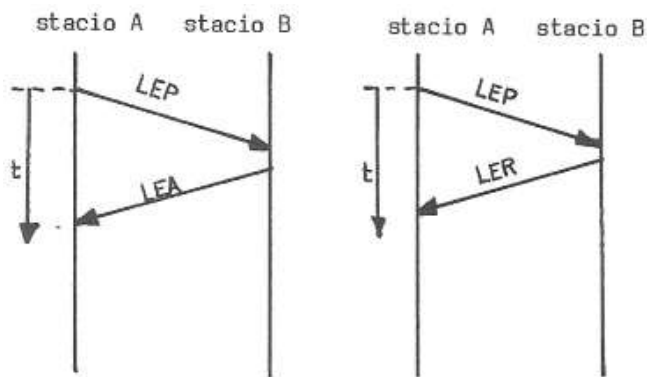


Fig.4. Ligilestigaj akcepto kaj rifuzo.

I-5. Datenoj (D)

Post ricevo aŭ sendo de ligilestiga akcepto, stacio rajtas sendi datenojn al la alia stacio per tiu ligilo en datenaj blokoj (Fig. 5).

Parametroj de la bloko D:

- Origina Adreso (OA)
- Cela Adreso (CA)
- Bloka Tipo (BT = Datenoj)
- Cela Referenco (CR)
- Senda Laŭvica Datenbloka Numero
- Indiko de eraro en la uzantaj datenoj (nur por certigita datena flukvanto)
- Indiko de lasta aŭ ne lasta datenbloko de mesaĝo
- Uzantaj Datenoj.

I-6. Datena Rego (DR) :

Tiu bloko ne uziĝas por la ligiloj kun certigita datena flukvanto. En la aliaj kazoj, ĝi uziĝas por kvitanco bone ricevatajn datenblokojn kaj por permesi al la alia stacio sendi pliajn datenblokojn (Fig. 5).

Parametroj de la bloko DR:

- Origina Adreso (OA)
- Cela Adreso (CA)
- Bloka Tipo (BT = Datena Rego)
- Cela Referenco (CR)
- Laŭvica numero de la datenbloko atendata de la origino
- Kredito (numero de la datenblokoj kiujn pretas ricevi la origino)

La resendado de datenbloko okazas post nuliĝo de tempumilo por atendi kvitancon (bloko DR) de datenbloko plej antaŭe sendita. Kaze de resendado de datenbloko post nuliĝo de tempumilo, nur la plej malnova datenbloko ankoraŭ ne kvitancita estas resendata.

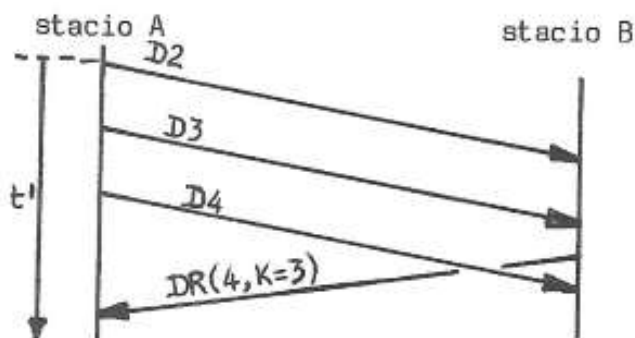


Fig. 5. Sendo de datenblokoj kaj datenregaj blokoj.

I-7. LigilDetrua Peto (LDP)

Tiu bloko uziĝas nur por detruo jam estigitan ligilon, ĝi distingiĝas de la bloko "ligilestiga rifuzo" por eviti miskomprenon kaze de reinitigo (Fig. 6).

Parametroj de la bloko LDP:

- Origina Adreso (OA)
- Cela Adreso (CA)
- Bloka Tipo (BT = LigilDetrua Peto)
- Cela Referenco (CR)
- Detrua Kaŭzo (silento de la alia flanko, peto de la origina abonanto, procedura eraro...)
- Uzantaj Datenoj

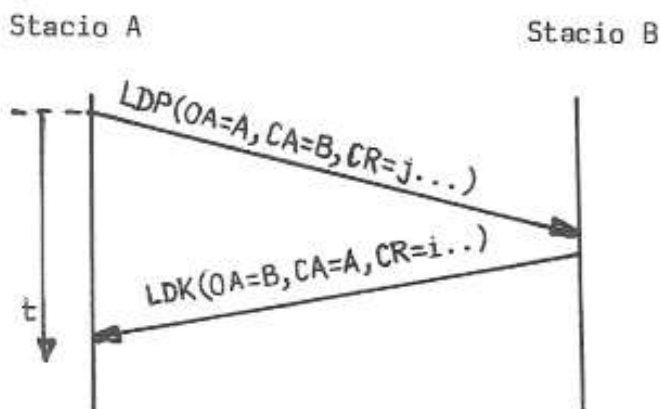


Fig.6. Ligildetrua peto kaj ligildetrua konfirmo.

La ligildetrua peto estas aŭtomate sendata post 5-minuta silento por iu ajn ligilo. Ne eblas nuligi ligilestigitan peton se ĝi estas jam sendita; por tion fari, ni devas atendi la respondon al tiu ligilestiga peto kaj sendi ligildetruan peton se la respondo estas pozitiva.

I-8. LigilDetrua Konfirmo (LDK) :

Tiu bloko estas la respondo al ligildetrua peto (LDP). Post ricevo de tiu konfirmo, la stacio povas reutiligi la referencon de la ligilo (Fig. 6). Neniu respondo estas sendata post ricevo de tiu bloko. Parametroj de la bloko LDK:

- Origina Adreso (OA)
- Cela Adreso (CA)
- Bloka Tipo (BT = LigilDetrua Konfirmo)
- Cela Referenco (CR)

1-9. Datagramo (DG) (opcia):

Tiu bloko uziĝas por la mallongaj transakcioj kun malsamaj korespondantoj. Malsame al la ligilestiga peto ĝi ne estigas ligilon inter la du stacioj (Fig. 7).

Parametroj de la bloko DG:

- Origina Adreso (OA)
- Cela Adreso (CA)
- Bloka Tipo (BT = Datagramo)
- Referenco de la datagramo
- Uzantaj datenoj

Se la senda stacio ne ricevas datagraman kvitancon antaŭ iu tempdaŭro post la sendo de la datagramo, ĝi resendas la datagramon.

I-10. Datagrama Kvitanco (DK) (opcia, ligita al DG) :

Tiu bloko estas generata de la cela stacio responde al datagrama bloko sendita de origina stacio (Fig. 7).

Parametroj de la bloko DK:

- Origina Adreso (OA)
- Cela Adreso (CA)
- Bloka Tipo (BT = Datagrama Kvitanco)
- Referenco de la ricevita datagramo

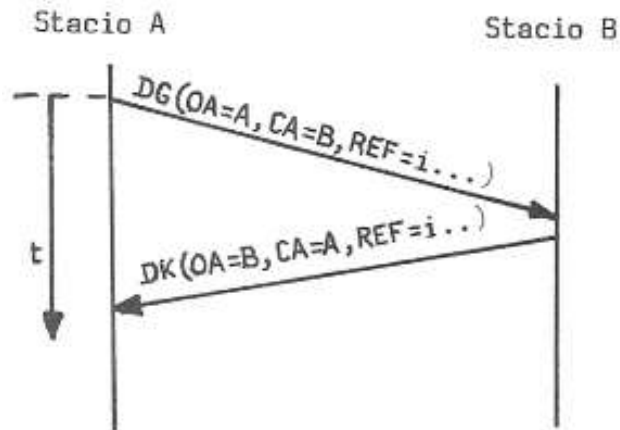


Fig.7. Interlanĝo de datagramo kaj de datagrama kvitanco.

Datagrama kvitanco povas ne esti generata se la datagramo ne estas ricevita de la cela stacio. Kiam datagrama kvitanco estas generita de la cela stacio, nenio pruvas ke la stacio kiu sendis la datagramon ricevos tiun datagraman kvitancon.

Se stacio ricevas duan datagramon de la sama stacio kun la sama referenco ĝi kvitanco ĝin kaj liveras ĝin al la supera nivelo kiu devas kapabli distingi la duoblaĵojn.

II. ŜIRMO DE LA BLOKOJ

La retaj informoj, t.e. cela adreso, origina adreso, bloka tipo, referenco kaj laŭvica datenbloka numero estas asociita al aparta kontrolsumo por povi kaze de detektita transmisieraro scii ĉu la eraro rilatas al la kapumo de la bloko aŭ al la uzantaj datenoj, en la dua kazo la uzantaj datenoj povas esti liverataj se la uzanto petis tion en la ligilestiga peto.

La tuta bloko estas ŝirmata per BlokKontrola Sekvenco (BKS) pro facileco (estus malpli facile ŝirmi nur la uzantajn datenojn).

III. LA TRIA NIVelo

La ligilnivela proceduro PLP liveras al la supera nivelo la informojn kiuj troviĝas en la kampoj nomataj "uzantaj datenoj". Por povi kompreni la enhavon de tiuj uzantaj datenoj, aŭ ni ĉiam uzas la saman trianivelan proceduron (por ekzemplo la paka nivelo de X25) aŭ ni por ĉiu nova ligilo diskutas kaj interkonsentas pri la aparta trianivela proceduro uzota por tiu nova ligilo, tiun duan solvon ni preferis.

La unua okbito de la uzantaj datenoj por iu ligilo aŭ datagramo kiuj atingas la trian nivelon estas la identigaĵo de la uzota trianivela proceduro. Tiu unua okbito povas troviĝi en la uzantaj datenoj de la ligilestiga peto aŭ de la unua datenbloko (por ligilo). Se la ricevanto ne komprenas tiun proceduron, restas al ĝi nur peti de la dua nivelo la ĉesigon de la komuniko aŭ la sendon de datagramo kiu informas pri la nekompreno de la trianivela proceduro.

BIBLIOGRAFIO

=====

- [1] La datentransiga proceduro X25 kun ĝenerala enkonduko pri proceduroj. SUK-Kursoteksto KJ04. C. BERTIN.

GLOSARO

=====

Akceso: Kapableco atingi per moviĝo aŭ ne kaj kapableco rilati kaj komuniki kun io por uzi ĝiajn servojn: akceso al komputoro, al datenbanko, al biblioteko.

BlokKontrola Sekvenco (BKS): Vico de bitoj kiu estas la rezulto de aparta operacio farita de la sendanto pri la transigotaj bitoj. Tiu vico de bitoj estas sendata kun la uzantaj bitoj por ebligi al la ricevanto detekti kaj eventuale korekti erarojn en la transigitaj uzantaj bitoj.

C.C.I.T.T.: Comité Consultatif International Télégraphique et Téléphonique. (Internacia Konsultiĝa Komitato pri Telefonio kaj Telegrafio).

Datagramo: Datenbloko kiu estas sen rilato kun iu ajn antaŭe sendita aŭ poste sendota bloko kaj kiu enhavas ĉiujn direktajn informojn necesajn por trairi reton kaj eliri el tiu reto. (aŭtonoma datenbloko).

H.D.L.C.: High level Data Link Control. (Altnivela Datena LigilRego). Nomo de familio de proceduro normigita de I.S.O kaj de aparta maniero por nekaraktra transmisio.

Initi: Fari diversajn agojn komence aŭ antaŭ aktiviĝo aŭ reaktiviĝo (por ekzemplo: seti variablojn, presi mesaĝon por funkciigistoj). Kaze de eraro, ekipaĵo povas iniciati reinitiĝon, ankaŭ

funkciigisto povas reiniti ekipaĵon.

I.S.O.: International Organization for Standardization. (Internacia Normiga Organizaĵo).

Multipleksi: Ebligi plurajn samtempajn komunikojn per unu ligilo.

Okbito: Mallongigo de okbitaĵo.

Stacio: Uzanta ekipaĵo plej proksima al la reto kiu sendas kaj ricevas la interesanĝatajn blokojn.

Tempumi: Atendi dum iu difinita periodo.

BIOGRAFIO



Christian BERTIN, inĝeniero pri datenprilaboro (INSA, Rennes, 1972) laboris ĉe IRIA pri la reto CYCLADES ĝis 1976, ekde 1976 laboras ĉe la CCETT (Centro Komuna Esplora pri Elsendado kaj Telekomuniko) en Rennes, Francio.

Rapida malkompono en Tejloran serion kaj solvo de Koŝi-problemo por sistemoj de diferencialaj ekvacioj, ordinaraj kaj kun partaj derivaĵoj

Aleksandro M. Gofen

Resumo:

Estas ekzamenata Koŝi-problemo por sistemoj de, ĝeneraldire, nelinearaj ekvacioj, solvataj rilate derivaĵojn laŭ tempo. Ĝis la lasta tempo oni kutime ne uzis la klasikan ideon por solvi ĉi tiun problemon per rekta kalkulo de Tejloraj koeficientoj helpe de laŭvica diferenciado de la dekstraj partoj de la ekvacioj, kiam oni konstruis algoritmojn por komputiloj. Tamen ekde apero de la ideo de Moore kaj Gibbons, plue nomata rapida Tejlora Malkompono /RTM/, kaj komputilprogramoj de Barton kaj kunaŭtoroj, integri ordinarajn diferencialajn ekvaciojn per la metodo RTM fariĝas 5-50 oble rapide, al almenaŭ per tiuj metodoj kutimaj, kiuj aproksimas la solvon kune kun ĝiaj derivaĵoj ./tiaj estas metodoj, aplikataj por „nerigidaj" /angle "not .stiff"/ problemoj/. Laŭ vidpunkto iom pli ĝenerala, ni pridiskutas en jena verko ĉi tiun metodon kaj vastigas ĝin por ekvacioj kun partaj derivaĵoj. Kurta listo de konataj aplikoj estas enhavata en la bibliografio, kaj estas proponataj kelkaj novaj: apliko al la metodo de Galorkin, netradicia solvo de Koŝi-problemo por ekvacioj kun partaj derivaĵoj en la sola punkto de spaco.

1. Enkonduke

Vasta klaso de nombraj metodoj por solvo de Koŝi-problemo

$$y' = f(y), \quad y|_{t=t_0} = y_0, \quad y = (y_1, \dots, y_m) \quad (1)$$

baziĝas sur aproksimo de la solvo per fragmento de Tejlora serio

$$y = \sum_{N=0}^{\infty} \frac{y^{(N)}}{N!} (t - t_0) \quad (2)$$

kiu prezentas la solvon en ĉirkaŭaĵoj de punktoj, apartenantaj al intervalo de integrado [8]. Kvankam oni povas kalkuli la koeficientojn de la serio (2) helpe de laŭvica diferenciado de ambaŭ partoj de la ekvacioj (1)

$$\begin{aligned} y' &= f \\ y'' &= \sum_{i=1}^m \frac{\partial f}{\partial y_i} y'_i \\ &\dots \dots \dots \end{aligned} \quad (3)$$

oni kutime ne uzis ĉi tiun klasikan ideon por konstrui algoritmojn komputilajn pro sekvaĵoj kialoj.

La unua estas tio, ke ĝeneralkaze oni ne scias algoritmon por kalkuli partajn derivaĵojn de f .

La dua kialo estas komplikeco de kalkulo de ĝenerala membro el la vico (3), kiu esprimiĝas helpe de ĝeneraligita formulo de Faa di Bruno 10 por N -a derivaĵo de Superpozicio de funkcioj

$$\begin{aligned} \frac{1}{N!} y^{(N+1)} &= \sum_{\left\{ \prod_{i=1}^m \prod_{j=1}^N \frac{1}{n_{ij}!} \left(\frac{1}{j!} y_i^{(j)} \frac{\partial}{\partial y_i} \right)^{n_{ij}} \right\}} f(y_1, \dots, y_m) \\ \sum_{i=1}^m \sum_{j=1}^N j n_{ij} &= N \end{aligned}$$

kie la sumigo estas farata tra ĉiuj diversaj mN -opoj de nenegativaj entjeroj n_{ij} , kontentigantaj la notitan egalecon.

Jam por $m=1$ la kvanto $p(N)$ da adiciatoj en tiu ĉi formulo rapide kreskas laŭ kresko de N , kio sekvas por grandaj N el asimptota formulo de Ramanujan [12]

$$p(N) \sim \frac{1}{4N\sqrt{3}} e^{\frac{\pi\sqrt{2/3}}{\sqrt{N}}}$$

kaj por la malgrandaj - el la tabelo 24.2 [13] de la funkcio $p(N)$:

$p(5)=7, \dots, p(30)=5604, \dots, p(50)=204226, \dots, p(100)=190569292, \dots$

Evidentiĝas, ke la fakto, ke dekstraj partoj de sistemoj, ordinare renkonteblaj en aplikoj, konsistas el funkcioj ne ĝeneralaj, sed elementaj aŭ specialaj, esence malpliigas ambaŭ nomitajn malfacilaĵojn. Ĝi permesas unue konstrui ĝeneralan algoritmon por kalkuli derivaĵojn de f , kaj due redukti la formulon de Faa di Bruno, en kiu plejparto da adiciatoj evidentiĝas similaj por tiu klaso de funkcioj.

Verŝajne unuafoje ĉi tion rimarkis Moore [1] kaj Gibbons [2]. La nombro da operacioj por kalkuli N -an derivaĵon de la solvo de la sistemo (1) proporcias al N^2 laŭ ilia metodo. Tiamaniere, aperas la nombra metodo por integri la Koŝi problemon kun kiu ajn alta ordo de aproksimo, bazata sur ekonomia senpera kalkulo de koeficientoj de la serio (2).

Ĉi tiu metodo estas realigita en komputil-programoj [3,4]. Kompara esploro de la rapido en la verko [3] montras, ke por ĉiuj provproblemoj la komputil-tempo de nova metodo evidentiĝis 5-50oble malplia, ol por la metodo de Runge-Kutta kaj aliaj metodoj por integrado kun fiksa ordo de aproksimo.

La ĉefa celo de la verko estas konstrui similan metodon por solvi ĝenerale nelinearajn evoluajn ekvaciojn kun partaj derivaĵoj. Tamen por klarigi la bazajn ideojn kaj malkompliki priskribon de novaj rezultoj, en la sekva parto ni pridis-kutos la ĉefan proceduron por ordinaraĵ diferencialaj ekvacioj laŭ vidpunkto iom pli ĝenerala, ol tiu en [1, 3, 4].

2. Priskribo de la algoritmo

La aŭtoroj de la verkoj [1-4] ne proponis specialan nomon por sia algoritmo. Ni kondiĉe nomu ĝin Rapida Tejlora Malkompono aŭ /RTM/.

Al ni necesos sekva

Difino 1: ni diros, ke vektor-funkcio $\{f_i(x_1, \dots, x_n)\}$, $i = 1, 2, \dots, m$, de $n \geq 1$ variantoj estas elementa rilate al la aro de variantoj

$$(x_{a_1}, x_{a_2}, \dots, x_{a_s}) \subset (x_1, \dots, x_n), \quad s \geq 1,$$

en la opo de $M \geq m$ funkcioj f_i , $i = 1, 2, \dots, M$, se ĉiuj f_i estas komponantoj de la solvo de ĉiu el jenaj S sistemoj

$$j = 1, 2, \dots, S \left\{ \frac{\partial}{\partial x_{aj}} f_i = R_{ij}(f_1, \dots, f_M), \quad i = 1, 2, \dots, M, \right. \quad (1)$$

kie R_{ij} estas raciaj funkcioj de nomitaj argumentoj.

Porm = n = 1. Ĉi tiu difino koincidas kun tiu en [1, 3, 4]. Ĝin kontentigas ĉiuj raciaj funkcioj (suplemento 2), kaj la funkcioj, tradicie konsiderataj kiel elementaj, ekzemple exp, ln, arctg, kaj ankaŭ specialaj funkcioj: integralaj logaritmo kaj sinuso, funkcioj de Bessel k.t.p.

Oni povas pruvi (suplemento 2), ke aro de elementaj vektor-funkcioj posedas jenajn proprecojn:

- se por du tiaj funkcioj estas difinita la superpozicio, la lasta ankaŭ estas elementa;
- se elementa vektor-funkcio havas la inversan, ankaŭ ĝi estas elementa.

Tiu ĉi aro tamen ne estas grupo, ĉar ne por ĉiu paro de ĝiaj elementoj ekzistas superpozicio, kaj ne por ĉiu elemento ekzistas la inversa. La ekzemplon de iu neelementa funkcio ni ne scias /vidu suplementon 2 pri ĉi tio/.

2.1. RTM por ordinaraj diferencialaj ekvacioj

Ni konsideru Koŝi-problemon

$$\left\{ \begin{array}{l} y_i' = f_i(y_1, \dots, y_m), \quad y_i \Big|_{t=t_0} = y_i^0, \quad i = 1, 2, \dots, m \end{array} \right. \quad (2)$$

RTM baziĝas sur du sekvaj proceduroj, transformantaj la problemon (2) al analogaj problemoj en spacoj kun dimensio ĝenerale pli alta ol m :

- 1^o. Transformo de la problemo (2) al formo

$$\begin{cases} y_i' = R_i(y_1, \dots, y_n, y_1', \dots, y_{i-1}'), & y_i \Big|_{t=t_0} = y_i^0 \\ i = 1, 2, \dots, n, \end{cases} \quad (3)$$

kie R_i estas raciaj funkcioj de la ĉiuj indikitaj argumentoj, kaj ĝenerale $n \geq m$. Ni rimarku, ke la dekstra parto de i -a ekvacio povas dependi de y_j' , $j < i$. Sistemojn de la tipo (3) ni nomos raciaj.

2^o. Transformo de la racia sistemo (3) al speciala sistemo, konsistanta el algebraj kaj diferencialaj ekvacioj, kiun oni nomas kanona.

Por difini kanonan sistemon, estas oportune konsideri specialan vicigitan aron de variantoj $V = \{v_{-q+1}, \dots, v_0, v_1, \dots, v_p\} = \{v_k\}$, kiu konsistas el $p+q$ variantoj kaj enhavas $y_1, y_2, \dots, y_n, y_1', \dots, y_n'$ kaj novajn variantojn a_j , kiujn ni nomos helpaj. Elementoj de V kun numeroj $k < 1$ estas ĉiuj y_i kaj tiuj a_j , kiuj ne dependas de t /konstantoj/. Elementoj de V kun pozitivaj indeksoj estas ĉiuj y_i' kaj a_j , estantaj eksplikitaj funkcioj de elementoj el V , implicate dependantaj de t .

Tiam nomiĝas kanona sistemo de la tipo

$$\begin{cases} y_i \Big|_{t=t_0} = y_i^0; & i = 1, 2, \dots, n \\ v_i = v_{k_i}^0 v_{l_i}; & k_i, l_i < i; \quad i = 1, 2, \dots, p, \end{cases} \quad (4)$$

kie „o” signifas iun el signoj +, -, x, /. Ni klarigu la sistemon (4). La dekstraj partoj de ĝiaj ekvacioj enhavas nur unu operacion /adicion, subtrahon, multiplikon aŭ dividon/ inter tiuj elementoj, por kiuj la ekvacioj jam estis difinitaj antaŭe. Oni povas nomi tian sistemon rekurente-difinita aŭ rekurenta.

Oni facile pruvos /suplemento 2/, ke la proceduro 1^o estas efike realigebla, se la dekstraj partoj de (2) estas

elementaj funkcioj rilate al ĉiuj variantoj. Evidentas, ke plenumi la proceduron 2^o oni povas ĉiam. La algoritmoj, kiuj faras tion, estas parto de programoj komputilaj, priskribitaj en [3, 4].

La ĉefa ideo kiel uzi la kanonan formon estas tio, ke por laŭvice kalkuli derivaĵojn de ordo 2, 3, ..., N de y_i ($i=1, 2, \dots, n$) oni povas apliki jenajn klasikajn formulojn por N - a derivato al ambaŭ partoj de la ekvacioj:

$$(P \pm Q)^{(N)} = P^{(N)} \pm Q^{(N)},$$

formulon de Leibniz /Lejbnic/

$$(PQ)^{(N)} = \sum_{i=0}^N P^{(i)} Q^{(N-i)} \quad (5)$$

kaj facile konkludeblan el (5) formulon por kvociento

$$\left(\frac{P}{Q}\right)^{(N)} = \frac{1}{Q} \left(P^{(N)} - \sum_{i=0}^{N-1} \binom{N-1}{i} \frac{P^{(i)}}{Q} Q^{(N-i)} \right), \quad (6)$$

(Ĉi tie ni konsideras ĉiun derivaĵon de ordo i kiel normitan

$$D_i = \frac{1}{i!} \left(\frac{d}{dt} \right)^i$$

por kiu anstataŭ la propreco de asocieco validas

$$D_i D_j = \frac{(i+j)!}{i! j!} D_{i+j}$$

Ĉe tio la komplikeco de kalkuloj por N - a kaj ĉiuj antaŭaj derivaĵoj laŭ metodo RTM estas jena. Por p ekvacioj de la kanona formo oni bezonas Np ĉelojn da komputilmemoro. Se nur $p_1 \ll p$ ekvacioj (4) estas nelinearaj, ne pli ol $P_1 N(N+1)/2$ multiplikoj, $P_1 N$ dividoj kaj $P_1 N(N+1)/2$ adicioj necesos [1].

Ni vidas, ke la nombro P estas grava por karakterizi la kvanton da kalkuloj. Sed cetere ĉi tiu nombro ne determiniĝas precize el la origina sistemo (2), ĉar dependas de tio, kiamaniere oni transformis sistemon (2) al kanona formo.

Pli detale ni pridiskutas ĉi tion en la parto 2.3 pri modifitaj kanonaj sistemoj. Ni nur rimarku ĉi tie, ke la kvanto P proksimume takseblas kiel kvanto da aritmetikaj operacioj, necesaj por kalkuli iamaniere ĉiujn dekstrajn partojn de la sistemo (1), kiu difinas elementajn dekstrajn partojn de (2). La pligrandigo de la kvanto P kontraŭ al la kvanto m en origina sistemo (2) estas laŭcela.

Reale, estu la unua ekvacio (2) ekzemple

$$y_1' = y_{m_1} y_{m_2} \cdots y_{m_k}$$

Por diferencii ĝin rekte necesas uzi Lejbnican formulon por k multiplikantoj

$$(P_1 P_2 \cdots P_k)^{/N/} = \sum P_1^{/n_1/} P_2^{/n_2/} \cdots P_k^{/n_k/},$$

$$n_1 + n_2 + \cdots + n_k = N$$

kio enhavas la kvanton da adiciatoj C_{N+k-1}^{k-1} /dunomiala koeficiento/. Sekve necesas plenumi $(k-1) C_{N+k+1}^{k-1}$ multiplikojn kaj C_{N+k-1}^{k-1} adiciojn.

Tamen se oni transformus ĉi tiun ekvacion, enkondukante helpajn variantojn

$$a_2 = y_{m_1} y_{m_2}, \quad a_i = a_{i-1} y_{m_i}, \quad i = 3, 4, \dots, k-1,$$

al kanona formo

$$\begin{cases} a_2 = y_{m_1} y_{m_2} \\ a_i = a_{i-1} y_{m_i}, \quad i = 3, 4, \dots, k-1, \\ y_1' = a_{k-1} y_{m_k}, \end{cases}$$

do, kvankam la kvanto da ekvacioj fariĝos $k-1$, oni povos aplikati al ĉiu el ili la formulon (5), kio bezonos nur $(k-1)(N+1)$ multiplikojn kaj samtiom da adiciojn.

Kaj nun ni transiru al

2.2. RTM por ekvacioj kun partaj derivaĵoj

Sekve de rimarkindaj proprecoj de la kanona sistemo (4) aplike al kalkulo de derivaĵoj por solvo de ordinaraĵ diferecialaj ekvacioj, aperas ideo apliki analogan algoritmon al ekvacioj kun partaj derivaĵoj. Ni konsideru kiel originan la Koŝi-problemon por sistemo de kvazaŭ-linearaj evoluaj diferecialaj ekvacioj

$$U_k' = \sum_{i=1}^{n_k} f_{ki}(x, U_1, \dots, U_m) D_{ki} + f_{ko}(x, U_1, \dots, U_m) \quad (7)$$

$$U_k \Big|_{t=t_0} = \varphi(x), \quad x \in E^{r-1}, \quad k = 1, 2, \dots, m.$$

Ĉi tie $U_k' = \frac{\partial U_k}{\partial t} = D_t U_k$, kie apartigita varianto t nomiĝas tempo, kaj D_{ki} signifas iujn partajn derivaĵojn de iuj U_k laŭ komponantoj de vektoro $x = (x_2, x_3, \dots, x_r)$. Ni nomas ilin spacaj; derivaĵojn laŭ ili - spacderivaĵoj. Kiel sekvas el [16], al la formo (7) estas transformebla iu ajn evolua sistemo.

Oni povas pruvi /suplemento 2/, ke se ĉiuj f_{ki} estas elementaj rilate al komponantoj de U , tiam la sistemo (7) estas transformebla al racia formo

$$\begin{cases} U_k' = R_k(U_1, \dots, U_n, D_{k1}, \dots, D_{kn_k}, U_1', \dots, U_{k-1}') \\ k = 1, 2, \dots, n; \quad n \geq m, \end{cases} \quad (8)$$

kie R_k estas raciaj rilate al ĉiuj nomitaj argumentoj. Ree, por skribi la kanonan sistemon, ni enkondukas aron de variantoj $V = \{v_{-q+1}, \dots, v_0, v_1, \dots, v_p\}$, kiu konsistas el ĉiuj U_k kaj iliaj derivaĵoj, partoprenantaj la sistemon (8), kaj el iuj variantoj a_j , kiujn ni nomas helpaj.

Kiel antaue, elementoj de V kun nepozitivaj indksoj estas ĉiuj U_k kaj iliaj spaderivaĵoj, kaj ankaŭ tiuj a_j , kiu ne dependas de t /kvankam eventuale dependas de x /.

Konforme al tio, la elementoj de V kun pozitivaj indksoj estas ĉiuj U_k' kaj a_j , implicate dependantaj de t . Tiam nomiĝas kanona sistemo de la tipo

$$\begin{cases} U_k|_{t=t_0} = \varphi_k(x) & ; \quad k = 1, 2, \dots, n, \\ v_i = v_{k_i} \circ v_{l_i} & ; \quad k_i, l_i < i ; \quad i = 1, 2, \dots, P. \\ & \circ \in \{+, -, \times, / \} \end{cases} \quad (9)$$

Nia celo estas trovi algoritmon por kalkuli derivaĵojn D_t^N de komponantoj de U . Kvankam la kanona sistemo (9) aspektas analoge al (4), almenaŭ ĉesto de spaderivaĵoj montras tion, ke laŭvica aplikado de nuraj operatoroj D_t^i al ekvacioj (9) ne sufiĉas por la starigita celo. Necesas ankaŭ apliki iujn operatorojn de miksa diferenciado laŭ (t, x) kaj memori la rezultojn. Kiujn nome - ni pridiskutos plue.

Kiel en la kazo de ordinaraj ekvacioj, fundamente gravas por pluj kalkuloj formuloj, analogaj al (5,6). Por skribi ilin, ni interkonsentu ĉiun normigitan operatoron de miksa diferenciado

$$\frac{1}{v! \alpha! \beta! \dots} \left(\frac{\partial}{\partial t} \right)^\nu \left(\frac{\partial}{\partial x_2} \right)^\alpha \left(\frac{\partial}{\partial x_3} \right)^\beta \dots$$

kaj la respondan derivaĵon signi helpe de la vektora potenco indico $\mathcal{M} = (\nu, \alpha, \beta, \dots)$, fiksinte unufoje por ĉiam la ordon de diferenciado: laŭ t, x_2, x_3, \dots, x_r . Aplikante operatoron $\mathcal{M} = (\nu, \alpha, \beta, \dots)$ al ambaŭ partoj de la ekvacioj, ni devas tamen memori, ke

$$\binom{P}{\mathcal{M}} (\mathcal{M}) = \frac{(\mathcal{M} + \mathcal{M}')^P}{\mathcal{M}! \mathcal{M}'!},$$

kie $\mathcal{M}' = \nu' \alpha' \beta' \dots$

Tiam la formuloj (5,6) fariĝos jenaj:

$$(P \pm Q)^{(n)} = P^{(n)} \pm Q^{(n)},$$

Lejbnica formulo

$$(PQ)^{(n)} = \sum_{0 \leq i \leq n} P^{(i)} Q^{(n-i)},$$

$$\left(\frac{P}{Q}\right)^{(n)} = \frac{1}{Q} \left(P^{(n)} - \sum_{0 \leq i \leq n} \left(\frac{P}{Q}\right)^{(i)} Q^{(n-i)} \right) \quad (10)$$

krom $\dot{Q} = 0$

/ni komprenas neegalecon inter vektoroj kiel aron de neegalecoj inter ĉiuj respondaj komponantoj/. Ni pruvu la lastan formulon.

Ĝia dekstra parto egalas al

$$\begin{aligned} & \frac{1}{Q} \left(P^{(n)} - \sum_{0 \leq i \leq n} \left(\frac{P}{Q}\right)^{(i)} Q^{(n-i)} + \left(\frac{P}{Q}\right)^{(n)} Q \right) = \\ & = \frac{1}{Q} \left(P^{(n)} - \left(\frac{P}{Q}\right)^{(n)} Q + \left(\frac{P}{Q}\right)^{(n)} Q \right) = \left(\frac{P}{Q}\right)^{(n)}, \end{aligned}$$

kion necesis pruvi. Ĉi tie ni subtrahis kaj aldonis unu adiciaton ĉe $\dot{Q} = 0$, kaj aplikis Lejbnican formulon (10).

Plue analogeco kun RTM por ordinaraj ekvacioj esence komplikiĝos. La kalkulon de derivaĵoj $D_t^N = (N, 0, 0, \dots)$ por $N = 1, 2, 3, \dots$ de nekonataj funkcioj oni plenumas en sola punkto $(t^0, x_2^0, \dots, x_r^0)$. Ĉe tio oni konsideras ekzistanta algoritmon, kiu ebligas kalkuli ĉiun spacierivaĵon de funkcioj φ_k kaj v_i por $i < 1$. Tia algoritmo reale evidentas, se, ekzemple, ĉiuj menciitaj konataj funkcioj estas reprezentataj kiel lineara kombino de kompleksaj eksponencialoj aŭ de aliaj funkcioj, por kiuj oni konas facilajn regulojn por derivado. Tamen ĝeneralkaze, kalkuli spacierivaĵojn estas preskaŭ tiel komplika problemo, kiel la origina /krom tio, ke la dimensio malpliĝos je 1/.

Sed la plej esenca komplikaĵo estas tio, ke por plialtigi indicon de tempderivaĵo D_t^N oni devas kalkuli pli kaj pli vastan aron de miksitaĵaj derivaĵoj, ĉe tio la aro havas sufiĉe komplikan strukturon. Ekzemple, oni devus konsideri ĉiujn komponaĵojn po N de diversaj diferencialaj operatoroj, ĉeestantaj en la sistemoj (8) aŭ (9). Tamen oni povas eviti ĉi tion kontraŭ ioma superfluo de kalkuloj.

Por plua pridiskuto utilas uzi izomorfismon inter la monoido de diferencialaj operatoroj kaj vektoroj el E^r havantaj entjerajn nenegativajn koordinatojn. Estu S la aro de operatoroj en la sistemo (9).

Troviĝos minimuma laŭ volumeno simplekso $\tilde{\sigma}_1$ kun verticoj $(0, 0, \dots, 0)$, $(1, 0, 0, \dots, 0)$, $(0, a_2, 0, \dots, 0)$, $(0, 0, a_3, \dots, 0)$, \dots , $(0, 0, \dots, a_r)$, kiu enhavas ĉiujn elementojn el S . Alivorte, se arbitra operatoro $(\nu, \alpha_2, \alpha_3, \dots, \alpha_r) \in S$, tiam

$$\frac{\nu}{1} + \frac{\alpha_2}{a_2} + \frac{\alpha_3}{a_3} + \dots + \frac{\alpha_r}{a_r} \leq 1, \quad (a_2, a_3, \dots, a_r > 0) \quad (11)$$

kaj simpleksvolumeno $a_2 a_3 \dots a_r / r!$ estas minimuma. Validas sekva Lemo: ajna komponaĵo po N diferencialaj operatoroj $\eta \in \tilde{\sigma}_1$ /responde, sumo de N vektoroj el $\tilde{\sigma}_1$ / estas enhavata en simplekso $\tilde{\sigma}_N$ simila al $\tilde{\sigma}_1$ kun la koeficiento N , t.e. en simplekso

$$N = \left\{ (\nu, \alpha_2, \alpha_3, \dots, \alpha_r) \mid \frac{\nu}{1} + \frac{\alpha_2}{a_2} + \frac{\alpha_3}{a_3} + \dots + \frac{\alpha_r}{a_r} \leq N \right\}.$$

Por prui tion sufiĉas sumigi N neegalecojn (11) por N iuj ajn punktoj el $\tilde{\sigma}_1$.

Nun ni pretas konsideri derivaĵkalkulan algoritmon. La kalkulo okazos laŭ pasoj tiel, ke je N -a paŝo oni kalkulos derivaĵojn D_t^N de ĉiuj U -komponentoj.

Ni nomu per A la aron de helpaj tempdependaj variantoj, kaj la aro $\{U_1, U_2, \dots, U_n\} = U$. Plue ni supozas, ke je la paŝo N ni jam kalkulis ĉiujn derivaĵojn $(n \in \tilde{\sigma}_{N-1})$ de variantoj el A , kaj ĉiujn derivaĵojn $(n \in \tilde{\sigma}_N \setminus (N, 0, 0, \dots))$ - ĉi tio estas simplekso sen „ĉapelo“ - de variantoj el U . Tiam, aplikante la formulojn (10) al la kanonaj ekvacioj (9), oni povas plenumi sekvajn etapojn:

1. Kalkuli derivaĵojn $D_t^N = (N, 0, 0, \dots)$ de funkcioj el U aplikante operatoron D_t^{N-1} .

Oni povas nomi tiun ĉi etapon „finkonstruo de ĉapelo“ de la simplekso, kaj la postajn etapojn - „finkonstruo de ĝia oblikva edro“.

2.0. Kalkuli ĉiujn derivaĵojn $(0, \alpha_2, \alpha_3, \dots, \alpha_r)$ el $\tilde{\sigma}_{N+1} \setminus \tilde{\sigma}_N$ por U kaj poste derivaĵojn de la sama formo el $\tilde{\sigma}_N \setminus \tilde{\sigma}_{N-1}$ por A .

2.1. Kalkuli ĉiujn derivaĵojn $(1, \alpha_2, \dots, \alpha_r)$ el $\tilde{\sigma}_{N+1} \setminus \tilde{\sigma}_N$ por U (aplikante operatorojn $(0, \alpha_2, \dots, \alpha_r)$), kaj poste derivaĵojn de la sama formo el $\tilde{\sigma}_N \setminus \tilde{\sigma}_{N-1}$ por A .

.....

2.N. Kalkuli ĉiujn derivaĵojn $(N, \alpha_2, \dots, \alpha_r)$ el $\tilde{\sigma}_{N+1} \setminus \tilde{\sigma}_N$ por U (aplikante operatorojn $(N-1, \alpha_2, \dots, \alpha_r)$), kaj poste derivaĵojn de la sama formo el $\tilde{\sigma}_N \setminus \tilde{\sigma}_{N-1}$ por A .

Por realigi ĉiujn menciitajn etapojn necesas, ke kiel rezulto de apliko de ĉiu bezonata operatoro al ambaŭ partoj de la ekvacioj (9) aperu nur tiaj derivaĵoj, kiujn oni jam kalkulis antaŭe. Garantias tion rekurenteco de la sistemo (9), la fakto, ke ĉiuj derivaĵoj en dekstraj partoj (9) estas enhavataj en la simplekso $\tilde{\sigma}_1$ kaj la lemo.

Tiamaniere, la N -a paŝo finiĝas, kalkulinte ĉiujn derivaĵojn el $\tilde{\sigma}_N$ por funkcioj A kaj tiujn el $\tilde{\sigma}_{N+1} \setminus (N+1, 0, 0, \dots)$ por U .

Lau matematika indukto, tio pruvas realigeblon de la proponita algoritmo por ĉiu $N > 0$.

Nun ni taksu kvanton da kalkuloj por ricevi derivaĵojn D_t^N de ĉiuj nekonataj funkcioj. Ni supozas, ke por tio necesas teni en komputilmemoro kaj kalkuli ĉiujn derivaĵojn el \mathcal{O}_N por ĉiu el p funkcioj de la sistemo (9). Do oni povas taksi la nombron da memorĉeloj per volumeno de la simplekso \mathcal{O}_N , t.e.

$$p \frac{a_2 a_3 \dots a_r}{r!} N^r$$

Plue ni supozas, ke por kalkuli ĉiun derivaĵon $\mathfrak{u} = (\nu, \alpha_2, \alpha_3, \dots)$ oni uzas la Lejbnican formulon (10), kiu postulas nombron da operacioj, egalan al nombro da diversaj vektoroj $\mathfrak{u} \in \mathcal{O}_N$, t.e. $\nu \alpha_2 \alpha_3 \dots \alpha_r$. Tiam la nombron da operacioj por unu funkcio

$$\sum_{(\nu, \alpha_2, \alpha_3, \dots, \alpha_r) \in \mathcal{O}_N} \nu \alpha_2 \alpha_3 \dots \alpha_r$$

ni taksas pere de la Dirichlet-integralo [11]

$$\iiint \dots \int x_1 x_2 \dots x_r dx_1 \dots dx_r = \frac{N^{2r} a_2^2 a_3^2 \dots a_r^2}{(2r)!}$$

$$\frac{x_1}{N} + \frac{x_2}{N a_2} + \dots + \frac{x_r}{N a_r} \leq 1; \quad x_1, \dots, x_r \geq 0$$

La finkonkludo estas jena: por kalkuli derivaĵon D_t^N (kaj kun ĝi ankaŭ ĉiujn derivaĵojn el \mathcal{O}_N de nekonataj funkcioj por la sistemo (9), oni bezonas $O(N^r)$ da memorĉeloj kaj $O(N^{2r})$ da operacioj.

Rimarko 1. Speciale ni konsideru la kazon de lineara sistemo $D_t u = Pu$. Ĉi tie $u = (u_1, u_2, \dots, u_m)$ kaj elementoj de la matrico $P = \left\| P_{ij} \right\|_{i, j=1}^m$ estas polinomoj kun konstantaj koeficientoj de diversaj operatoroj, kiuj operatoroj komutiĝas inter si.

Oni konsideras konataj la algoritmojn por kalkuli la operatorojn kaj ilian potencojn. Tiam la problemo kalkuli derivaĵojn D_t^N reduktiĝas al kalkulo de potencoj de la operatora matrico P .

Rimarko 2. Anstataŭ la simplekso σ_1 (formulo (11)), oni povas sammaniere konsideri regionon de spaco, kiu estas komunaĵo de iuj ajn simpleksoj, havantaj verticojn sur pozitivaj duonaksoj. Da la lemo kaj ĉiuj pridiskutitaj rezultoj validos ankaŭ por konveksaj poliedroj, enhavantaj ĉiujn derivaĵojn kaj iliajn projekciojn de la sistemo (7).

2.3. Modifita kanona sistemo

Oni jam menciis en la parto 2,1, ke la kanona sistemo (4) ne determiniĝas precize tiel rilate al la kvanto p da ekvacioj, kiel rilate al ilia formo. Pligrandiĝo de p kompare al la kvanto m da ekvacioj en origina sistemo (2) evidente sekvas el procezo de malkomplikiĝo de la dekstraj partoj. Fakte, la kanona formo (4, 9) ebligas pluan malkomplikigon. Kiel proponas profesoro M.L.Lidov, oni povus elimini la operacion divido el la sistemo (4), se ĉiun ekvacion de la formo $V = P/Q$ oni anstataŭigus per du sekvaj: $V = PW$ kaj $W' = -W^2Q'$, kie P , Q , kaj do ankaŭ Q' estas polinomoj de variantoj el la aro V . Tial aperas jena demando: Ĉu la opo da operacioj $\{+, -, \cdot, / \}$, permesitaj en la kanona formo (4,9), estas optimuma por kalkuloj laŭ la kanona formo?

Ni nomu sistemon de la formo (4,9) rilate al arbitra opo da operacioj $\{g_1, \dots, g_m\}$ kiel modifita kanona.

Ĉe kanona sistemo (4,9) ĉi tiu opo estas jena:

$$g_1(x, y) = x+y, \quad g_2(x, y) = x-y, \quad g_3(x, y) = xy, \quad g_4(x, y) = x/y.$$

La minimuma opo, evidente, estas la proponita far M.L.Lidov triopo da operacioj $\{+, -, \cdot\}$. Ĝi enhavas nur unu nelinearan operacion multipliko, pordiferencii kiun oni uzas Lejbnican formulon (5,10).

Ni traktu ĉi tiun formulon kiel ia mezuro de kvanto de kalkuloj. Kontraŭe al la minimuma opo, la maksimuma estas opo da ĉiuj analitikaj funkcioj de arbitra kvanto da variantoj. Tamen por diferencii tiajn funkciojn, oni devus uzi la formulon de Faa di Bruno. Tial ni interesiĝos nur pri tiaj operacioj, kiuj, simile al la aritmetikaj, kaŭzas aperon de granda parto da similaj termoj en ĉi tiu formulo. Plue ni vidos, ke por iuj elementaj funkcioj de unu varianto ekzistas formuloj, similaj laŭ komplikeco al Lejbnica formulo (5). Ni klarigu, ke uzado de tiaj formuloj por diferencii la modifitan kanonan sistemon ĝenerale kondukas al ekonomio de kalkuloj kompare al la kanona formo (4,9).

Efektive, se ia ekvacio $\mathbf{v} = \mathbf{f}_1(U)$ el sistemo de la tipo (4,9) enhavas kiel operacion la funkcion \mathbf{f}_1 , tiam por elimini ĉi tiun operacion almenaŭ necesas aldoni al la sistemo tiujn ekvaciojn, kiuj determinas \mathbf{f}_1 kiel elementan:

$$\left\{ \mathbf{f}'_i = R_i(\mathbf{f}_1, \dots, \mathbf{f}_n) \right., \quad i = 1, 2, \dots, n.$$

Tiamaniere anstataŭ unu ekvacio $\mathbf{v} = \mathbf{f}_1(U)$ aperas $n+1$:

$$\left\{ \begin{array}{l} \mathbf{v} = W_1 \\ W'_i = R_i(W_1, \dots, W_n) U' \end{array} \right., \quad i = 1, 2, \dots, n$$

Ĉe tio, se U' ĉeestas nenium el la antaŭaj ekvacioj de la kanona sistemo, oni devus prezenti ĝin kiel eksplicitan racian funkcion. La neceso malkomplikigi ĝin kaj ankaŭ ĉiujn R_i ĝenerale povas konduki al longa ĉeno de novaj ekvacioj, enhavantaj fin-fine nur unu aritmetikan operacion, eventuale nelinearan. Nome tial laŭcelas lasi la operacion divido kaj (kiel oni faras en [3,4]) ankaŭ iujn aliajn, por kiuj Moore proponas formulojn [1], kiel jenajn

$$\binom{p\alpha}{N} = \frac{1}{p} \sum_{i=0}^{N-1} \left(\alpha - \frac{i(\alpha+1)}{N} \right) \binom{p\alpha}{i}_p \binom{N-1}{N-1-i}_p, \quad N > 0$$

$$(e^P)^{(N)} = \sum_{i=0}^{N-1} \left(1 - \frac{i}{N}\right) (e^P)^{(i)} P^{(N-i)}, \quad N > 0$$

$$(\ln P)^{(N)} = \frac{1}{P} \left(P^{(N)} - \sum_{i=1}^{N-1} \left(1 - \frac{i}{N}\right) P^{(i)} (\ln P)^{(N-i)} \right), \quad N >$$

(Ĉiuj derivaĵoj estas normigitaj!). Komplikeco de ĉi tiuj kaj aliaj analogiaj formuloj en [1] proksimume egalas al tiu de (5, 6).

Fine ni rimarku, ke estas laŭcele permesi en dekstraj partoj de la kanonaj ekvacioj aperon de kiom ajn longaj linearaj esprimoj aŭ sumoj de duopaj produktoj. Ĉi tio ne pligrandigos la kvanton da kalkuloj, tamen malpliigos la nombron da necesaj komputilaj memor-ĉeloj. Sed lasi polinomon de indico $k > 2$ en la dekstraj partoj de la kanona sistemo jam ne laŭcelas. Ĉi tio sekvas el pridiskuto en la fino de la parto 2.1.

3. Aplikoj de algoritmo RTM

Plue ni konsideros Tejloran serion

$$\sum_{N=0}^{\infty} a_N (t - t^0)^N \quad (1)$$

ne nur kiel prezenton por solvo de ordinaraj diferencialaj ekvacioj, sed ankaŭ por solvo de Koŝi-problemo en fiksa punkto de spacotempo $(t^0, x_2^0, x_3^0, \dots, x_r^0)$ por ekvacioj kun partaj derivaĵoj. La dimensio r interesos nin nur kiel indico de kvanto da kalkuloj, kiuj necesas por ricevi la koeficientojn a_N , nome

$$O(N^{2r}) \quad (2)$$

Oni devas noti, ke la pritakso (2) neniel antaŭdeterminas efikecon de integrado por Koŝi-problemo en la donita intervalo, sed nur karakterizas komplikecon de kalkuloj por koeficientoj de Tejlora serio / aŭ de serio kun iaj aliaj funkcioj, kiel plue tie ĉi por metodo de Galorkin /. Koŝi-problemon, por kiu oni indikas intervalon de integrado kaj eraron ϵ , ni nomos konkretigita.

Por tiaj problemoj oni povas enkonduki la nocion valoro

de derivaĵoj kiel la kvanto inversa al nombro da elementoj el serio (1), kiujn oni ankoraŭ devas teni por reprezenti la solvon kun eraro ϵ . Do, por tiel nomitaj „rigidaj” problemoj /angla termino „stiff”/ tiu ĉi valoro estas malgranda. Tial oni devas ĉi-kaze plenumi grandan kvanton da kalkuloj. Eksperimente trovitan avantaĝon de la metodo RTM en [3,4] ĉe $r = 1$ oni povas klarigi tiel.

Ordinare pritakso de kvanto da kalkuloj por integrado per metodo kun fiksa ordo m de aproksimo estas

$$O\left(\epsilon^{-\frac{1}{m}}\right) \quad (3)$$

por ordinaraj diferencialaj ekvacioj. Sed por RTM ĝi estas $O\left(\ln \epsilon \cdot \epsilon^{2r}\right)$, kio sekvas el proprecoj de konverĝa Tejlora serio kaj el (2). Se $\epsilon \rightarrow 0$, tiam evidente la lasta pritakso kreskas malpli rapide, ol (3). Bedaŭrinde, ĉi tiu klarigo ne estas rigora.

Estante efektiva kalkulrimedo por Tejlora koeficientoj, RTM ebligas esplori ecojn de analitika solvo de Koŝi-problemo per rimedoj de la teorio de analitikaj funkcioj. Estas kurioza fakto: la serion (1), se ĝi diverĝas, oni povas sumigi per ia metodo de sumigo, kiu validas en ĉefa stela regiono /t.e. „korekti” elementojn de la serio per specialaj multiplikantoj/. Tiaj estas, ekzemple, metodoj de Lindelöf aŭ de Mittag-Leffler [14,15]. Tiam, se la solvo de Koŝi-problemo ne havas singularajn punktojn sur la intervalo mem de integrado, la funpunkto estas atingebla el la startpunkto per unu paŝo!

Tamen praktike ĉi tiu rezulto estas verŝajne senutila pro manko ĝeneralkaze de erartaksoj por sumigmetodoj, kaj pro malalta valoro de derivaĵoj, observita dum nombraj eksperimentoj ĉe komputilo por diverĝa serio (1).

3.1. Apliko de RTM al la metodo de Galorkin

Kalkulo per RTM de ĉiuj derivaĵoj en simplekso σ_N ebligas skribi malkomponon por solvo de Koŝi-problemo en punkto kiel

serio (1) aŭ pli ĝenerale, laŭ ĉiuj eblaj derivaĵoj
 $m = (\nu, \alpha, \beta, \dots)$

$$\sum_{m \in \mathcal{O}_N} U(m) (t - t^0)^\nu (x_2 - x_2^0)^\alpha (x_3 - x_3^0)^\beta \dots$$

Ĉi tio estas malkompono laŭ tre speciala sistemo de lineare nedependaj funkcioj (LND). Tamen scio de opo de derivaĵoj el \mathcal{O}_N egale ebligas malkomponi la solvon laŭ arbitra sistemo de LND-funkcioj.

Estu jam kalkulitaj derivaĵoj el la opo $\{m_k\} = \{(\nu, \alpha, \beta, \dots)\} = S \subset \mathcal{O}_N$, $k = 1, 2, \dots, K$. Kaj ankaŭ estu sistemo, havanta K LND-funkciojn φ_i , por kiuj determinanto $\|\varphi_i^{(m_k)}\|_{i,k=1}^K \neq 0$ en startpunkto $(t^0, x_2^0, \dots, x_r^0)$. /Se la aro $S = \{(\nu, 0, 0, \dots)\}$, $\nu = 0, 1, 2, \dots, K-1$, tiam la determinanto estas tiu de Vronskij/.

Laŭ la metodo de Galorkin, ni trovas proksimuman solvon kiel $\tilde{U} = \sum_{i=1}^K a_i \varphi_i$. El kondiĉo, ke K derivaĵoj de la aro S de la proksimumo \tilde{U} egalas al respondaj derivaĵoj de la preciza solvo U , oni ricevos linearan algebran sistemon

$$\sum_{i=1}^K a_i \varphi_i^{(m_k)} = U^{(m_k)}, \quad k = 1, 2, \dots, K$$

rilate al nekonatoj a_1, a_2, \dots, a_K kun determinanto, kiu ne egalas nulon en la punkto $(t^0, x_2^0, \dots, x_r^0)$ laŭ origina kondiĉo.

La problemo pri eraro de la trovita tiamaniere solvo \tilde{U} povas esti esplorita per observo de ekarto, aperanta pro substituo de \tilde{U} en la originan sistemon. Kutime oni elektas sistemon de LND-funkcioj konsiderante aprioran scion pri la solvo kaj tiel, ke oni povu facile diferencii ilin laŭ precizaj formuloj.

3.2. Pri elekto de paŝo kaj ordo de aproksimo

Laŭ oni jam notis, la eblo relative facile kalkuli derivaĵojn de la solvo por Koŝi-problemo ankoraŭ ne garantias efikecon de integrado.

Kontraŭe al metodoj de integrado por Koŝi-problemo kun fiksa ordo de aproksimo, por kiuj nur la paŝo variendas, uzante metodon RTM oni povas manipuli tiel la paŝon, kiel la ordon de aproksimo por ĉiu paŝo, aŭ entute solvi la problemon per unu paŝo. Tial aperas demando: kian strategion oni devas akcepti por elekto de la paŝo kaj ordo de aproksimo ĉe integrado de konkretigita Koŝi-problemo?

Ekzemple Moore [1] solvas ĉi tiun problemon, uzante heŭristikan pritakson (3), t.e. konsiderante, ke la kvanto da paŝoj m , kiun oni bezonas por integri kun eraro ϵ , egalas al $O(\epsilon^{-\frac{1}{m}})$. Tiam, ĉar kvanto da kalkuloj por ĉiu paŝo estas $O(m^{2r})$ laŭ (2), la totalakvanto da kalkuloj do egalas al $O(m^{2r}\epsilon^{-\frac{1}{m}})$.

La funkcio $f(x) = x^{2r} \epsilon^{-\frac{1}{x}}$ atingas minimumon ĉe $x = -\frac{\ln \epsilon}{2r}$.

Nome ĉi tiun rezulton por $r=1$ kaj sistemoj de ordinaraj diferencialaj ekvacioj ricevis Moore [1] kaj formulis kiel jenan „mnemonikan regulon“: la ordo m devas egali al kvanto da dekumaj ciferoj, kiujn enhavas mantisoj de nombroj en donita komputilo. Alivorte, ĉi tio estas la maksimuma diferenco inter ordoj de adiciatoj, kiujn adicii ankoraŭ estas senchave per la komputilo.

Tamen verŝajne oni ankoraŭ ne scias sufiĉe rigoran formuligon kaj solvon de la problemo.

4. Konkludoj

Ni rimarku iajn specifecojn de superskribita metodo RTM por integrado de Koŝi-problemo:

- por malpliigi eraron de la metodo, oni ne bezonas konverĝigi paŝon de integrado al nulo;

- por la metodo RTM ne aperas la demando pri kalkula stabileco kiel limigiteco de normoj por potencoj de operatoro.

Aldone por ekvacioj kun partaj derivaĵoj ni ankoraŭ notu, ke

- RTM neaplikeblas kutime por miksitaj problemoj, sed nur al Koŝi-problemo por evoluaj ekvacioj kun startaj kondiĉoj, prezentataj kiel analitikaj funkcioj;

- kvankam en aplikaj problemoj startaj kondiĉoj kutime venas kiel diskretaj, praktike ĉiam ili bezonas glatigan proceduron, kiun oni plenumas helpe de analitikaj agregatoj /polinomoj, kompleksaj eksponencialoj/. Do finfine, ni ricevos startajn kondiĉojn de la analitika formo, kaj samtempe la reguloj por diferencii ilin estas nekomplikaj;

- formale la metodo RTM permesas solvi la Koŝi-problemon en arbitra sola punkto de spaco ne uzante la aliajn. Tamen ne eblas plenumi la sekvan tempopaŝon ne ricevinte analitikan prezenton /analogan al la starta/ por la kuranta paŝo;

- RTM permesas dispareleligi kalkulojn ĉe multprocesora komputilo aŭ ĉe sistemo de komputiloj. Ĉiu procesoro povas solvi la problemon por aparta punkto /punktoj/ sendepende de la aliaj.

Malavantaĝo de la metodo RTM estas tio, ke la metodo ebligas trovi proksimuman solvon nur ĉe tiaj difinoj de proksimeco /metriko/, ĉe kiuj necesas egaleco /proksimeco/ ne nur inter la proksimuma kaj la preciza solvoj, sed ankaŭ inter ĉiuj iliaj respondaj derivaĵoj. Ne ĉiam tiaj metrikoj estas oportunaj. Ekzemple ne tute taŭgaj ili estas rilate al rigidaj problemoj, por kiuj oni kutime uzas metrikojn de spacoj C^1 , C , L^1 , L^2 . Ĉi tiu fakto estas esenca por tiuj problemoj.

Efikeco de la metodo RTM ankoraŭ estis kontrolita nur por ordinaraj diferencialaj ekvacioj [1-7, 18]. La konkludo de la aŭtoroj [3,4] pri tio, ke RTM pli bone ol la tradiciaj metodoj konsideras individuajn ecojn de la solvata problemo, validas ankaŭ por ekvacioj kun partaj derivaĵoj. Tamen plua esploro kaj kalkulaj eksperimentoj ankoraŭ restas farendaj.

Suplemento

Pruvoj por bazaj ecoj de elementaj funkcioj

Kompare al difino de elementeco donita en la verkoj [1,3,4],

la difino 1 formale vastigas la klason de elementaj funkcioj, enigante en ĝin ankaŭ vektor-funkciojn de multaj variantoj. La difino 1 permesas facile pruvi redukteblecon al la kanona formo por sistemoj kun elementaj dekstraj partoj, kaj ecojn de superpozicio kaj inversa funkcio.

Se elementa rilate al ĉiuj variantoj vektor-funkcio $\{f_i(x_1, \dots, x_n)\}$, $i = 1, 2, \dots, m$, kontentigas n sistemojn enhavantajn po M ekvaciojn

$$j = 1, 2, \dots, n \quad \left\{ \frac{\partial}{\partial x_j} f_i = R_{ij}(f_1, \dots, f_M); \quad i = 1, 2, \dots, M; \quad M \geq m \right.$$

tiam la dekstraj partoj de ĉi tiuj sistemoj esprimiĝas en la formo de matricio

$$\left\| \begin{array}{c} j = n \\ R_{ij} \\ i = M \\ i, j = 1 \end{array} \right\|$$

Tiamaniere, ĉiu racia funkcio $r/x_1, \dots, x_n$ en la opo da funkcioj $I_k(x_1, \dots, x_n) = x_k$, $k = 1, 2, \dots, n$, estas elementa rilate al ĉiuj variantoj, kaj ĉi tio esprimiĝas per matricio

$$\left(\begin{array}{cccccc} \frac{\partial r}{\partial x_1} & \frac{\partial r}{\partial x_2} & \frac{\partial r}{\partial x_3} & \dots & \dots & \frac{\partial r}{\partial x_n} \\ 1 & 0 & 0 & \dots & \dots & 0 \\ 0 & 1 & 0 & \dots & \dots & 0 \\ 0 & 0 & 1 & \dots & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & \dots & \dots & 1 \end{array} \right)$$

Sekvaj teoremoj argumentas la „grupajn“ ecojn por la aro de elementaj vektor-funkcioj.

Teoremo 1: Estu funkcio $Z_1(y_1, \dots, y_m)$ kaj vektor-funkcio $\{y_j(x_1, \dots, x_k)\}$, $j = 1, 2, \dots, m$, elementaj rilate al ĉiuj variantoj. Tiam ankaŭ la superpozicio $Z_1(y_1(x_1, \dots, x_k), \dots, y_m(x_1, \dots, x_k))$ estas elementa rilate al ĉiuj variantoj.

Pruvo. Estu elementeco de Z kune kun Z_2, \dots, Z_n esprimita per matrico $P = \left\| \left\| P_{ij} \right\| \right\|_{\substack{j=1 \\ i=1 \\ i, j=1}}^{j=m}$, kaj elementeco de vektorfunkcio $\{y_j(x_1, \dots, x_k)\}$ per matrico $Q = \left\| \left\| q_{jl} \right\| \right\|_{\substack{l=1 \\ j=1 \\ j, l=1}}^{l=k}$, $M \geq m$

Laŭ regulo por diferencii superpozicion

$$\frac{\partial z_i}{\partial x_1} = \sum_{j=1}^m \frac{\partial z_i}{\partial y_j} \frac{\partial y_j}{\partial x_1} = \sum_{j=1}^m P_{ij}(z_1, \dots, z_n) q_{j1}(y_1, \dots, y_M)$$

$i = 1, 2, \dots, n; \quad l = 1, 2, \dots, k.$

Kune kun ekvacioj respondantaj al la matrico Q , ni ricevas k sistemojn

$$l = 1, 2, \dots, k \quad \left\{ \begin{array}{l} \frac{\partial z_i}{\partial x_1} = \sum_{j=1}^m P_{ij}(z_1, \dots, z_n) q_{j1}(y_1, \dots, y_M) \\ \qquad \qquad \qquad i = 1, 2, \dots, n \\ \\ \frac{\partial y_j}{\partial x_1} = q_{j1}(y_1, \dots, y_M) \\ \qquad \qquad \qquad j = 1, 2, \dots, M \end{array} \right.$$

kies dekstraj partoj estas raciaj, ĉar ĉiuj P_{ij} kaj q_{j1} estas raciaj. Ĉi tio pravas la teoremon.

Oni povas skribi la matricon de ĉi tiuj sistemoj kiel

$$R = \begin{pmatrix} P \hat{Q} \\ Q \end{pmatrix},$$

kie $\hat{Q} = \left\| \left\| q_{jl} \right\| \right\|_{j, l=1}^m$. Ni rimarku, ke kvanto de ekvacioj,

difinantaj superpozicion, egalas al sumakvanto de ekvacioj en sistemoj, respondantaj al ĉiu vektor-funkcio de la superpozicio.

Teoremo 2. Se $\{y = \{y_1(x_1, \dots, x_m)\}\}$, $l = 1, 2, \dots, m$ prezentanta intermapon $E^m \rightarrow E^m$, estas elementa rilate al ĉiuj variantoj, do tia estas ankaŭ la inversa vektor-funkcio.

Aparte, por $m=1$ kaj intermapo $y_1(x) : E^1 \rightarrow E^1$,
 kontentiganta sistemon $y_i' = r_i(y_1, \dots, y_M)$, $i = 1, 2, \dots, M$, la
 inversa funkcio $Y_1(y_1)$ kontentigas la sistemon

$$\begin{cases} \frac{\partial Y_1}{\partial y_1} = \frac{1}{r_1(y_1, \dots, y_M)} \\ \frac{\partial y_i}{\partial y_1} = \frac{r_i(y_1, \dots, y_M)}{r_1(y_1, \dots, y_M)} \end{cases}, \quad i = 1, 2, \dots, M.$$

Ekzemple por funkcio $L = \ln x$, kontentiganta la sistemon

$$\begin{cases} L' = \frac{1}{x} \\ x' = 1 \end{cases},$$

la inversa funkcio $E(L)$ kontentigas la sistemon

$$\begin{cases} \frac{\partial E}{\partial L} = x \\ \frac{\partial x}{\partial L} = x \\ \frac{\partial L}{\partial L} = 1 \end{cases}$$

Ankaŭ ni rimarku, ke ĉiuj algebra, funkcioj estas elementaĵo.

Ni notas, ke en la verkoj [1,3,4], en kiuj oni uzas la
 nocion de la elementa funkcio nur de unu varianto, la aŭtoroj
 postulas, ke dekstraj partoj de reduktendaj al la kanona formo
 sistemoj estu superpozicioj de raciaj funkcioj de multaj vari-
 antoj kaj elementaj funkcioj de unu varianto.

La difino 1 ebligas pli simple formuligi ĉi tiun limigon
 /ne fortigante ĝin/ kaj tre facile pruvi redukteblecon al la
 kanona formo. Fakte en sekvaj du teoremoj ni pruvos la reduk-
 teblecon nur al la racia formo (proceduro 1^o), ĉar oni ĉiam
 povas transformi la racian sistemon al la kanona.

Teoremo 3: por ke la sistemo (2.2) de ordinaraj diferencialaj ekvacioj

$$\{y_i' = f_i(y_1, \dots, y_m)\}, \quad i = 1, 2, \dots, m$$

estu reduktebla al la racia formo, sufiĉas, ke la dekstraj partoj f_i estu elementaj.

Pruvo. Dank'al sia elementeco, kontentigu vektor-funkcio

$$\left\{ f_i(y_1, \dots, y_m) \right\}, \quad i = 1, 2, \dots, m \quad \text{sistemojn}$$

$$j = 1, 2, \dots, m \quad \left\{ \frac{\partial f_i}{\partial y_j} = R_{ij}(f_1, \dots, f_m) \right\}, \quad i = 1, 2, \dots, m, \dots, M.$$

Ni enkonduku novajn variantojn $z_i = f_i(y_1, \dots, y_m)$, $i = 1, 2, \dots, M$, kies derivaĵoj

$$z_i' = \sum_{j=1}^m \frac{\partial f_i}{\partial y_j} y_j' = \sum_{j=1}^m R_{ij}(z_1, \dots, z_M) y_j'$$

Tiam aro de variantoj $y_1, \dots, y_m, z_1, \dots, z_M$ kontentigas racian sistemon

$$\begin{cases} y_j' = z_j & , \quad j = 1, 2, \dots, m; \\ z_i' = \sum_{j=1}^m R_{ij}(z_1, \dots, z_M) y_j' & ; \quad i = 1, 2, \dots, M. \end{cases}$$

Nome ĉi tion oni devis prui.

Teoremo 4: por ke la sistemo (2.7) de kvazaŭlinearaj diferencialaj ekvacioj

$$u_k' = \sum_{i=1}^{n_k} f_{ki}(x, u_1, \dots, u_m) D_{ki} + f_{ko}(x, u_1, \dots, u_m)$$

$$k = 1, 2, \dots, m$$

estu reduktebla al la racia formo, sufiĉas, ke ĉiuj f_{ki} estu elementaj rilate al U_1, U_2, \dots, U_m .

Pruvo. Ni renomu ĉiujn f_{ki} kaj spacierivaĵojn D_{ki} respektive per $f_0, f_1, \dots, f_{n_1}, \dots, f_M$ kaj D_0, D_1, \dots, D_M tiel, ke la ekvacioj (2.7) fariĝas tiaj:

$$U'_k = \sum_{i=N_k+1}^{N_{k+1}-1} f_i(x, u_1, \dots, u_m)^{D_i} + f_{N_k}(x, u_1, \dots, u_m)$$

$$k = 1, 2, \dots, m.$$

Nun dank'al sia elementeco, kontentigu vektor-funkcio
 $\{f_i(x, u_1, \dots, u_m)\}$, $i = 0, 1, \dots, M$ sistemojn
 $j = 1, 2, \dots, m$ $\left\{ \frac{\partial f_i}{\partial u_j} = R_{ij}(f_0, \dots, f_N) \right.$; $i = 0, 1, \dots, N$; $N \geq M$.

Ni enkonduku novajn variantojn $v_i = f_i(x, u_1, \dots, u_m)$,
 $i = 0, 1, \dots, N$, kies derivatoj estas

$$v'_i = \frac{\partial v_i}{\partial t} = \sum_{j=1}^m \frac{\partial f_i}{\partial u_j} \frac{\partial u_j}{\partial t} = \sum_{j=1}^m R_{ij}(v_0, \dots, v_N) U'_j.$$

Tiam aro de variantoj $U_1, \dots, U_m, v_0, \dots, v_N$ kontentigas
 racian sistemon

$$\begin{cases} U'_k = \sum_{i=N_k+1}^{N_{k+1}-1} v_i^{D_i} + v_{N_k}, & k = 1, 2, \dots, m; \\ v'_i = \sum_{j=1}^m R_{ij}(v_0, \dots, v_N) U'_j, & i = 0, 1, \dots, N. \end{cases}$$

Ĉi tion oni devis pruvi.

Praktike por redukti sistemon al la kanona formo, laŭcelas
 unuigi procedurojn 1^o kaj 2^o en algoritmon, kiu procedas simile
 al kompililoj de algoritmaj lingvoj kiam ili programas komplikajn
 aritmetikan esprimon.

Fine ni notu, ke kvankam oni vaste uzas nocion de elementa
 funkcio en la verkoj [1-7] kaj en ĉi tiu, ni scias neniun
 ekzemplon de neelementa funkcio almenaŭ de unu varianto. Pruvi
 neelementecon verŝajne estas sufiĉe komplika problemo, kion hel-
 pas supozi la proksima laŭ senco teoremo de Hölder [17] :

la fama funkcio $\Gamma(x) = (x-1)\Gamma(x-1)$ povas esti solvo de
 neniuj ekvacio de formo $P(x, y(x), y^{(1)}, \dots, y^{(n)}) = 0$, kie
 P estas arbitra polinomo de ĉiuj menciitaj variantoj.

Dankoj

Mi dankas kandidaton de scienco V.A. Gordin pro la atento al ĉi tiu verko kaj utilaj konsiloj. Speciale mi esprimas profundan dankon al profesoro M.L. Lidov, kiu sugestis gravajn ideojn kaj helpis esence plibonigi la tekston. Kaj finfine mi dankas kandidaton de scienco V.S. Aroloviĉ, diligente korektintan la esperanta-version de ĉi tiu verko.

Literaturo

- [1] Moore R.E. Interval Analysis. Prentice-Hall, 1966.
- [2] Gibbons A. A Program for the Automatic Integration of Differential Equations Using the Method of Taylor Series. Computer J. 1960, 3, 108-111.
- [3] Barton D., Willers J.M., Zahar R.V.M. Taylor Series Methods of Ordinary Differential Equations - An Evaluation. In: Mathematical Software. Editor Rice J.R. Academic Press, 1971.
- [4] Barton D., Willers J.M., Zahar R.V.M. The automatic Solution of System of Ordinary Differential Equations by the Method of Taylor Series. Computer J. 1971, 14, N3, 243-248.
- [5] Rentrop P. A Taylor series Method for the Numerical Solution of Two point Boundary Value Problems. Numer Math. ,1979, 3, N4, 359-375.
- [6] Goldfine A. .Taylor Series Method for the Solution of Volterra Integral and Integro-Differential Equations. Math. Comput. 1977, 31, N139, 691-707.
- [7] Corliss G., Lowery D. Choosing a Stepsize for Taylor Series Methods for solving ODE's. J. Comput. and Appl. Math. 1977, 3, N4, 251-256.

- [8] Rakitszkij J.B., Usztyinov Sz.M., Csernoruckij I.G.
Csiszényije metodi resenia zsesztkih szisztem. Moszkva, 1979
- [9] Modern Numerical Methods for ordinary differential equations.
Oxford, 1976.
- [10] Good I.J. Annals of Mathematical Statistics, Vol.32, N2,
June 1961, 540-541.
- [11] Fihtengolc G.M. Kursz diferencialnovo i integralnovo
iszcsiszlenia. Tom 3. Moszkva, 1960.
- [12] Hardy G.H. "Ramanujan", London. Cambridge University
Press, 1940.
- [13] Abramowitz M., Stegun J.A. "Handbook of Mathematical
Funkctions"
U.S. Gov't Printing Office, 1964, . table 24.2, p.828.
- [14] Cooke R. Infinite matrices and sequence spaces. London,
1950.
- [15] Hardy G.H. Divergent series.
- [16] Courant R. Partial differential equations.
- [17] Gelfond A.O. Iszcsiszlenie konecsnih raznosztej. Moszkva
1967.
- [18] Nikulicsev J.V. Csiszlennij metod integrirovanija szisztem
obiknovennih diferencialnih uravnenij na osznove znaliti-
cseszko differencirovanija. Szbornik: "Optimizanija
dinamicseszkih szisztem". Novoszibirszk, 1979.

Biografio



Aleksandro M. Gofen naskiĝis je 1947a jaro en urbo Odeso. Je 1969a li finis la matematikan fakon de Odesa universitato. De 1978a li loĝas en Moskvo kaj laboras kiel malsupera scienca kunlaboranto en Hidrometeorologia Centro de U.S.S.R. Li okupiĝas precipe pri komputilprogramado, kaj ankaŭ pri kalkula matematiko.

Esperantiĝis li je 1980a.

Perkomputila optimumigo de urbega enlernejigo

d-ro inĝ. Peter BROCKÓ

RESUMO

Dum la urbega enlernejigo devus esti konsiderataj diversaj gepatraj vidopunktoj, kiujn la nuntempa enlernejiga sistemo, baziĝanta sur la lerneĵ-cirkaŭaĵa principo, nur malfrue kaj limigite povas plenumi. En Hungario la nun elformata ŝtata popolregistrado kreas la modernajn kondiĉojn informo-provizajn por apogo de urbega enlernejigo per matematikaj rimedoj.

Tiu ĉi artikolo konigas la formadon de la problemo, kiel transporta tasko de la lineara programado. Kunligite kun tio, ĝi donas efikan kaj tamen simplan procedon por relativa kaj absoluta mezurado de la gepatraj postuloj.

La komputila prilaborado eluzas la elastecon de la entenivo de la lernejoj kaj klasoj, krome per tio ĝi enportas daŭrajn heŭristikajn elementojn en la optimuman solvon de la lineara programada modelo.

I. LA AKTUALECO DE LA TASKO

En urbegoj la elekton el lernejoj povas motivi granda nombro da diversaj vidopunktoj. Decidanta povas esti la proksimeco de la lernejo, la loĝloko de geavoj, tiu fakto, ĉu la lernejo estas laŭvoje por la gepatrano iranta en sian laborejon aŭ ĉu la gepatrano precize ĉe lernejo transveturiliĝas, ktp. Pro la nuntempa mana enlernejiga sistemo oni devas simpligi ĉi tiun malsimplan demandon, tio estas oni alligas

al ĉiu lernejo difinitan cirkaŭaĵon. La limojn de la cirkaŭaĵoj oni difinas ankaŭ mane. Tial ofte okazas, ke gelernantoj el la najbarejo de unu lernejo devas viziti alian.

Ni povas konsideri enlernejigon, konvenan al gepatraj postuloj do esence kiel infrastrukturan problemon.

II. LA MATEMATIKA MODELO

La matematika modelo de la komputa sistemo optimumigas la enlernejigon per maksimuma kontentigo de diversaj vidopunktoj.

La senco de la uzata signado:

a_i nombro de geknaboj, apartenantaj al i -a postulgrupo

m nombro de postulgrupoj;

b_j entenivo de unuaj klasoj de j -a lernejo;

n nombro de lernejoj;

x_{ij} nombro de lernontoj el la i -a postulgrupo, vizitantaj la j -an lernejon;

c_{ij} pezo de la enlernejigo el i -a postulgrupo al j -a lernejo, difinita de gepatroj.

La matematika modelo:

$$\begin{aligned} \sum_{j=1}^n x_{ij} &= a_i & j &= 1, 2, \dots, m \\ \sum_{i=1}^m x_{ij} &= b_j & i &= 1, 2, \dots, n \\ \sum_{i=1}^m a_i &= \sum_{j=1}^n b_j & x_{ij} &\geq 0 \end{aligned}$$

$$\sum_{i=1}^m \sum_{j=1}^n c_{ij} x_{ij} \rightarrow \min$$

III. LA KOMPUTA PLANADA SISTEMO

La modelo por optimumigi la enlernejigon funkcias per tri datumaroj kaj la kvara datumaro, ricevita post ĝia prilaboro estas la fonto de la donotaj servoj. Datumaroj partoprenantaj en la prilaboro estas rezultoj aŭ fontoj po unu agadoserio. Alligiĝon de la agadoj unu al la alia montras la 1-a figuro, ni trarigardu laŭ tio la movon de la prilaborado.

1. KOMPONANTO Elekto de la enlernejigendaj lernontoj

Ni elskribas el la datumbazo de la popolregistrado la datumojn de tiuj geknaboj, kiuj inter la 1-a de septembro de la antaŭa jaro kaj la 31-a de aŭgusto de la kuranta jaro estas 6 jaraĝaj. Laŭ la vidopunkto de nia temo oni povus konsideri kiel urbegon ĉiun loĝlokon, kie almenaŭ en du lokoj ekzistas unua klaso, sed la modelo funkcias efike nur pri pli grandaj urboj, kie en multaj lokoj /minimume 5- 10/ estas unua klaso.

Oni diselektadas la datumojn de la lernontoj laŭ urboj kaj plue ni rakontos la agadon, plenumendan po urboj.

2. KOMPONANTO Esploro kaj mezurado de gepatraj postuloj

Al la adresoj ricevitaj el la popolregistrada datumbazo ni forsendas la sekvan gepatran informilon, konsistantan el du partoj:

- la unua parto por gepatroj enhavas
 - = adresojn kaj identigajn numerojn de lernejoj de la konkreta urbo;
 - = tipajn respondojn, kiuj motivas elekton el diversaj lernejoj;
 - = instrukcion por plenumo;

- la alia parto estas la respondo-poŝtkarto, resendota plenigite.

En instrukcio por plenigo oni devas skribi al gepatroj pri sekvontoj:

- oni elektu el lernejoj tiujn 9, kiuj estus por ili la plej konvenaj kaj oni enskribu ilin en rubrikojn de la respondo-poŝtkarto numeritajn ĝis 9 tiel, ke sur la unuan lokon oni skribu la identigon de tiu lernejo, kiu estas por ili la plej konvena kaj en la naŭan lokon tiun, kiu estas la plej malkonvena el la elektitaj;
- la elekton oni devas ankaŭ klarigi por enskribo de identaj numeroj de la konvenaj tipaj respondoj /2. figuro/.

La tipajn respondojn kunmetas komitato, kiu alligas al ili ankaŭ pezojn por la ena uzado.

La gepatroj enskribas la respondojn en la tabelon, trovigantan sur la respondo-poŝtkarto, parte jam per komputilo plenigitan /3. figuro/.

La ricevitajn tabelojn oni prilaboradas sekvamaniere:

- la relativajn pezojn oni transformas al matricoj de lernejo / persona numero;
- la absolutajn pezojn ni enlaboras en la relativajn tiel, ke ne ŝanĝiĝu la preferateco de lernejoj, specifitaj de gepatroj;
- ni grupigas la gelernontojn kun samaj aŭ preskaŭ samaj postuloj, per tio ni malgrandigas la person-numeran flankon de la matricoj, kiu per tio transformiĝas al flanko de postulgrupoj.

En ĉi tiu komponanto rezulte ni ricevas du datumarojn:

- datumaron de la postulgrupoj /vektoro \ddot{a} / :
- pezon de la enlernejigo el iu postulgrupo al iu lernejo /matrico C /.

3. KOMPONANTO Difino de la entenivo de la lernejoj, klasoj

La entenivo de la lernejoj, klasoj estas nerigida nombro, ĉar ĝi povas moviĝi inter tre larĝaj limoj. Oni povas ŝanĝi la nombrojn de la unuaj klasoj /ekzemple per kuntiro de pli maljunaj klasoj, per organizado de multŝanĝa lernado/, oni povas ŝanĝi ankaŭ la entenivon de certaj klasoj /ekzemple per instalado de suplementaj benkoj, per inteŝanĝo de malegale grandaj lernoĉambroj/.

La lernejoj en formularo verkita por tiu celo /4. figuro/ raportas la minimuman, optimuman kaj maksimuman entenivojn de la unuaj klasoj. Sur tiu bazo la modelanto formas tian serion da lernejaj entenivoj, ke la kondico $\sum a_i = \sum b_j$ plenumiĝu. Tiel oni difinas vektoron \ddot{b} .

4. KOMPONANTO Optimumigo

Oni devas plurfoje funkciigi la modelon plenumante la optimumigon, bone analizante la ricevitan rezulton, inter konkretaj baroj optimuman. Oni povas laŭcele elekte modifi la entenivon de lernejoj; per tio oni povas signife plibonigi la optimuman rezulton, ricevitan antaŭe.

Oni devas rimarkigi, ke kvankam komputante oni optimumigas modelon laŭ lineara programado, tamen la analizo de la rezultoj, kaj la modifo de la baroj, plenumita rezulte de la analizo estas tiaj manipuloj, kiuj finfine ŝanĝas la solvon de la tasko je heŭristika.

La ricevita solvo sciigas, kiom da lernontoj el iu postulo-grupo iros al iu lernejo /difino de matrico X /.

5. KOMPONANTO Kelkaj pli gravaj servoj

Al tiuj apartenas sciigilo presita por lernontoj, nomlistoj laŭ lernejoj, klasoj, statistika analizo de gepatraj postuloj.ktp.

La modelo nun estas en eksperimenta etapo. Por demonstraĵoj celoj ĝi estis efektivigata por urbo Miskolc /200 mil loĝantoj/ parte per vivaj, faktaj datumoj, parte - pro ekonomiaj kaŭzoj per datumoj, kreitaj per metodoj de stohastika simulo.

Ni optimumigis per program-pakaĵo "Mathematical Programing System" /Matematika Programada Sistemo/ de firmao Honeywell. Por alkonformigo de datumoj kaj por depreno de rezultoj de la program-pakaĵo ni verkis 16 aplikadajn programojn, en programadaj lingvoj COBOL, FORTRAN kaj GMAP.

Precipe favoraj estas la spertoj koneksaj kun tempo de la program-plenumado. La optimumigo mem postulas 5-6 minutojn da start-stop tempo, kiu post ŝanĝo delernejaj entenivo malkreskis je 50 %, ĉar la procezo de la optimumigo komenciĝis ne denove, sed de savita baza solvo de la antaŭa solvo.

Oni rimarkas, ke la modelo taŭgas ankaŭ por prilaborado de diversaj specialaj kazoj, ekzemple enlernejigo en lernejojn de mensmalsanuloj, en fakajn lernejojn, ktp.

IV. EBLOJ DE PLIEVOLUIGO DE SISTEMO

1. Regula difino de la gepatraj postuloj donas kredindan datumfonton al lerneĵ-dislokigaj decidoj, por apogo de kiuj ni povas formi ankaŭ komputan sistemon, kiuj per modernaj matematikaj metodoj /ekzemple analizo de grupoj/ povas helpi la lerneĵ-dislokigajn demandojn.

2. En pedagoga fakliteraturo larĝe konatas la kompensanta lerneja modelo, la esenco de kio mallonge sekvas: ni neglektu la tutmonde ĝenerale akceptitan sistemon de enlernejigo, kroĉiĝantan kun rigidaj aĝolimoj, kaj anstataŭ klasoj homogenaj laŭ aĝo oni aranĝu homogenajn laŭ korpa kaj mensa evoluitecoj.

Multaj poraj kaj kontraŭaj starpunktoj estas pri kompensanta lerneja modelo. Ĉi tie interrilate je nia temo ni devas atentigi pri du rilatoj:

- kompensanta lerneja modelo ebligas niveligon de signifa etat-ondado inter aĝogrupoj, kiu okazigas grandajn problemojn en lernejoj kaj estigas pro demografia ondado. Tiu ondado facile povas esti niveligota per ŝanĝado de mezuro de evoluiteco;
- la kompensanta lerneja modelo estas nesimpatia por fakuloj precipe pro tio, ĉar la subjektivaj faktoroj, la korpa kaj mensa evoluitecoj estas tre malfacile mezureblaj.

Tiurilate estigo de popolregistrado, apliko de person-numero malkovras senlimajn novajn eblojn, ĉar ĝi ebligas sekvetecon de la sanitara situacio, de la historio de korpa kaj mensa evoluoj. Tiel la juĝo de la mezuro de evoluiteco baziĝas ne sur medicina ekzameno, testo, plenumita en iu konkreta tempomomento, sed sur bazo de serio da tiaj ekzamenoj, plenumitaj ekde la naskiĝo kaj pro tio ĝi estas kvalite pli preciza.

3. Ankaŭ grandan literaturon havas en pedagogio problemo de la perdo de infanaĝaj talentoj.

Inter infanoj estas multe da "mirakl-infanoj", sed inter plenkreskuloj jam malmulte da elstaraj individuoj. Kaŭzo de tiu fakto estas, ke la meza socia cirkauaĵo malfavore efikas al evoluo de talentaj infanoj.

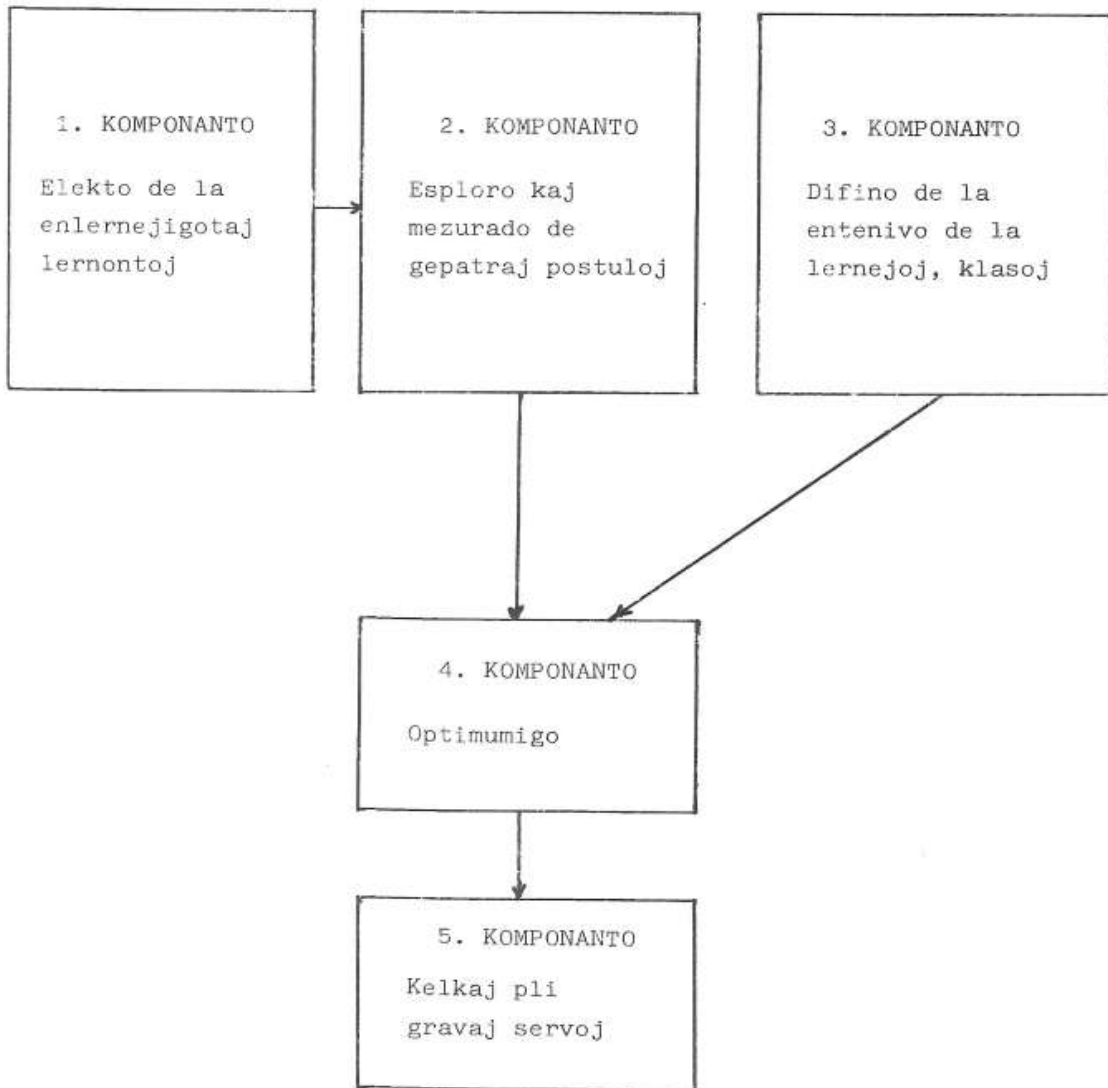
Ĉi tiun situacion malprofitan el la vidopunkto de la tuta socio oni povus signife malgrandigi per optimumigo de enlernejigo en fakajn lernejojn. La matematika modelo de ĉi tiu tasko jam pretas, ĝi eĉ estas jam farita por demonstraĵoj, sed la reala funkciado postulas plilarĝigojn de datumaro similajn al 2-a punkto pri la evoluado.

V. ATENDEBLAJ AVANTĜOJ DE LA ENKONDUKO DE LA PERKOMPUTILA ENLERNEJIGO

Realigo de proponata sistemo alportus la sekvajn avantaĝojn:

- ĝi signife plifaciligus la malkovron de la enlernejigotaj infanoj;
- ĝi ebligus la egalnivelan ekspluatadon de lernejen-
entenoj;
- ĝi ebligus konsideri krom la gepatraj vidopunktoj
ankaŭ la pli altajn vidopunktojn, kiuj realigus per
modelo unuece en kadro de urbo kaj tiamaniere ankaŭ
tutlande pli unuece;
- helpus likvidon de veturado de infanoj al
lernejoj, tiamaniere plimalaltigus la laŭvoĵajn
danĝerojn por la infanoj;
- kreus bazon por provi homigenigi la unuajn klasojn ne
laŭ aĝo, sed laŭ evoluiteco, kaj okaze de sukceso
ebligus ĝian enkondukon;
- kreus la kondiĉojn por prepari la perkomputilan enlernejigon
en fakajn lernejojn;
- per multflankaj informoj ĝi apogus la lernej-dislokigajn
decidojn.

Surbaze de la atendeblaj avantaĝoj ni povas konstati, ke la proponata enlernejiga sistemo rezultas efikan, bone utiligeblan sistemon kun multaj rektaj kaj nerektaj avantaĝoj ankaŭ sen popolregistrado.



Identa numero	Tipa respondo	La ena uzata pezo
1.	La lernejo estas pli proksima 01 100 metroj	50
2.	La lernejo estas pli proksima 01 200 metroj	40
⋮		
7.	La lernejo estas pli proksima 01 1000 metroj	0
⋮		
12.	Gis lernejo estas unu haltejo per tramo	30
⋮		
17.	Gis lernejo estas 5 haltejoj per tramo	20
⋮		
73.	Irante al la laborejo de gepatroj, la lernejo estas laŭvoje	50

2. figuro Tipaj respondoj klarigantaj elekton el lernejoj

Personnumero	Kioma-vice konvenas la lernejo?								
	1	2	3	4	5	6	7	8	9
Identaj numeroj de la tipaj respondoj klarigantaj la konvenon									

3. figuro Tabelo por mezurado de la gepatraj postuloj

Nomo, identiga numero kaj
adreso de la lernejo

Nombro de la ekmoveblaj unuaj klasoj en lernejo 19 /

en minimuma kazo

en la optimuma kazo

en la maksimuma kazo

Entenivo de la ekmoveblaj unuaj klasoj laŭ ordo,
difinita antaŭe:

Identigo de la klaso	Entenivo de la klaso		
	minimuma	optimuma	maksimuma
A			
B			
⋮	⋮	⋮	⋮
n			

subskribo de la direktoro

4. figuro Formularo enhavanta nombron kaj entenivajn
datumojn de la unuaj klasoj

	Identigoj de la postul-grupoj	
Identigoj de la lernejoj	Rezulto da la relativaj kaj absolutaj pezoj, donitaj al lernejoj de gepatroj matrico C	Entenivo de la lernejoj vektoro b
	Nombro de lernontoj de unuopaj postul-grupoj vektoro a	

5. figuro Skiza distribucia tabelo de la tasko

Biografio

d-ro inĝ. Peter BROCKZKÓ /1950.
Budapeŝto, Hungario/ estas
inĝeniero-ekonomisto pri
datumprilaborado /1973. Leningrad,
Sovetunio/ kaj fakekonomisto pri
komputado /1978. Budapeŝto/. Ĝis
1980 li laboris en popolregistrado,
poste en nutraĵ-industria
komputilcentro. Li doktoriĝis
priapliko de matematikaj metodoj en
popolregistrado /1978. Budapeŝto/.

La Informsystemo SELGEM:
Pritakso de Bibliografiaj Spertoj

de Jim CUSHING

Resumo:

Oni prezentas la ĉefajn trajtojn de le informsystemo SELGEM, komputilprogramaro disvolvita far la Smithsonian Institution en Vaŝingtono, kaj priskribas la celojn de bibliografia projekto ellaborita en la Centro por Maresplorado de la Instituto "Ruder Bošković" en Zagrebo, Jugoslavio, kaj uzanta la sistemon SELGEM kiel ĉefan programaron. Estas analizate, kiomgrade SELGEM kontentigas la bezonojn de la projekto kaj kiuj estas ĝiaj fortoj kaj malfortoj.

The SELGEM Information System:
an Evaluation of Bibliographic Experience

by Jim CUSHING

Abstract:

The main features of the SELGEM information management system of computer programs developed by the Smithsonian Institution in Washington, D.C., are presented. The goals of a bibliographic project elaborated at the Center for Marine Research in the "Ruder Bošković" Institute in Zagreb, Yugoslavia, and using the SELGEM system as its primary software support, are described. The strengths and weaknesses of SELGEM are analyzed, as well as the degree to which it satisfies the needs of the project.

1. Enkonduko

SELGEM estas ĝenerala informsystemo por komputila manipulado de rikordaroj kaj datumbazoj. Origine disvolvite far la Smithsonian Institution en Vaŝingtono, unuavice por prizorgado de iliaj natursciencaj muzeaj kolektoj, ĝi estas uzata nun en diversaj universitatoj kaj institutoj en pluraj landoj.

En la Instituto "Ruder Bošković" ĝi konsistas el 20 programoj, kiuj, kiam aplikataj laŭ diversaj kombinoj kaj sinsekvoj, povas plenumi multajn taskojn, kiaj: prizorgado de datumbazoj (aldonado, forigado, kaj ŝanĝado de datumoj), produktado de raportoj laŭ formatoj difinitaj far la uzantoj, elekta listigado de la enhavo de rikordaroj, presado de adres-etikedoj, produktado de kompletaj indeksoj (ekz., similaj al tiuj ĉe la fino de lernolibro), retrovado de datumoj surbaze de la plej diversaj logikaj kriterioj.

La systemo estas skribita en la programada lingvo COBOL kaj antaŭsupozas alireblon al tri-kvar magnetobendaj aparatoj (al diskaparatoj, samkiel al la normalaj presilo, kart-legilo kaj -truilo. La plej multaj programoj uzeblas en memoro de 50 000 silaboj (t.e., 50 K); kelkaj postulas ĝis 96 000.

2. Celoj de la projekto BIBJAD*

Le scienca literaturo rilate al esploroj en kaj pri la Adriatika Maro konstante pliampleksiĝas. Pro la grandaj

* "Adriatika Maro" en la serbokroata lingvo "Jadransko more"; tial le mallongigo BIBJAD ("Bibliografio pri la Adriatika Maro").

problemoj spertitaj, klopodante iel teni ĝisdatajn informojn en facile atingebla kaj manipulebla formo, la Centro por Maresplorado (CIM) en Zagrebo decidis en la aŭtuno de 1979 fari paŝojn al la starigo de komputilizita bibliografio de fakaj artikoloj pri oceanografio kaj maresplorado, kun ĉefa fokuso je la Adriatika Maro.

2.1. Informkategorioj

Estis dezirate, ke enestu en la bibliografio almenaŭ la jenaj informkategorioj por ĉiu artikolo:

- a) aŭtoro(j)
- b) titolo
- c) nomo de la gazeto
- ĉ) volumo, numero/jaro de apero
- d) komenca paĝnumero.

Krom tiuj, la ĉi-subaj estis konsiderataj utilaj kaj laŭeble enmetindaj:

- e) instituto, universitato, aŭ alia institucio de la aŭtoro(j)
- f) originala lingvo
- g) internacia gazetkodo
- ĝ) nombro de patoj
- h) ŝlosilvortoj por laŭtema retrovado
- ĥ) notoj de la ekzamenkomisiono, se temas pri artikolo de kunlaboranto en la Instituto "Ruder Bošković"
- i) komentoj.

2.2. Dezirataj sistemkapabloj

Esenca inter la necesaj kvalitoj estis simple la kapablo listigi la enhavon de la bibliografio laŭ diversaj vicordoj:

* La pritaksoj de la komisiono rilate al la kvalito de artikolo havas iugradan influon je la salajro de la kunlaboranto.

- laŭ aŭtoro
- laŭ jaro de apero
- laŭ titolo
- laŭ gazeto
- k.s.

Ankaŭ tre necesa estus la eblo retrovi kaj ekstrakti difinitajn specojn de artikolo (ekz., ĉiujn artikolojn pri elektrokemio, pri mar-biologio, k.s.).

En iu ajn nova projekto estas malfacile antaŭvidi ĉiujn estontajn postulojn. Sekve estis ege dezirinda la kapablo daŭre allasi kromajn, novajn informkategoriojn sen troa perturbado de la sistemstrukturo. Krome, kelkaj informkategorioj principe dezirataj devis resti komence nenigigitaj pro simpla manko de laborfortoj.

Estis klare al la planantoj de BIBJAD, ke almenaŭ en la komenco la bezonoj de la Instituto ne postulis interaktivajn komputilkapablojn -- la pli tradicia aropa prilaborado -- (angle: batch processing) sufiĉos.

3. Bazaj kapabloj de la sistemo SELGEM

Post prikonsidero de diversaj alternativoj, la planantoj decidis lanĉi eksperimentan projekton kun SELGEM kiel la ĉefa ĉenero. Ilin aparte impresis ĝia ĝenerala fleksebleco, kiu vidiĝos de la sekvaj teoriaj ekzemploj de aplikoj, por kiuj SELGEM aparte taŭgas.

3.1. Prizorgado de gazetabonoj

Certa universitato respondecas abonadon por pluraj sciencaj gazetoj. Kelkaj estas monataj, kelkaj kvaron-jaraj, aliaj jaraj. SELGEM prizorgas informojn koncerne la abonantojn (ekz., ĝis kiu dato validas la abono? kiom da ekzempleroj de ĉiu gazeto iras al kiuj adresoj? ktp), kaj presas adresetikedojn por ekspedado. De tempo al

tempo oni sendas informojn pri novaj publikaĵoj al elektitaj abonantoj, surbaze de iliaj anoncigitaj interesoj (t.e., kemio al kemiistoj, fiziko al fizikistoj, ktp.).

3.2. Datumbazo de agrikulturistaj spertoj

La direktoro de internacia agrikultura organizo ricevas help-petojn el diversaj evolulandoj. Pere de SELGEM li traserĉas sian datumbazon kaj ekstraktas la nomojn kaj adresojn de tiuj homoj, kiuj - ekzemple - estas diplomitaj agronomoj, havas multe da sperto en la kultivado de rizo, jam antaŭe loĝis almenaŭ ses monatojn en eksterlando, parolas la francan lingvon, kaj estas 25- ĝis 35-jaraj.

3.3. Katalogo de zoologiaj specimenoj

Marzoologo estas kolektanta dum la daŭro de pluraj kuroj preskaŭ mil specimenojn da mara faŭno. Por ĉiu unuopa specimeno li notas 20 ĝis 60 informparametrojn (ekz., profundecon kaj temperaturon de akvo, pH-on, grandecon kaj pezon de la oĵo, genron kaj specion, kolekt-lokon, k.s.). Li ne havas la samajn parametrojn por ĉiuj specimenoj, nek scios anticipe kion precize li trovos. Li volas transformi ĉiujn datumojn al komputile manipulebla formo kaj tiam efektiviigi longan serion de logikaj ekstraktoj: a) ĉiuj okazoj de Carcinus mediterraneus kiam la profundeco estas pli ol du metroj; b) Ĉiuj okazoj de Bleneus pavo kiam la specimeno entenis ksenobiotan substancan en hepata enzimo, sed nur se la specimeno estis prenita el akvo pli varma ol certa minimumo; c) Ĉiuj okazoj de Palaemon serratus, pli ol du centimetrojn longaj, sed nur kiam la specimeno estis kolektita malpli ol cent metrojn de iu bordo; ktp. Krom alfabeto listigo de ĉiuj

datumoj laŭ genro kaj specio, li volas ankaŭ listojn laŭ kelkaj aliaj sinsekvoj: profurdeco, pezo en gramoj, k.s., kaj por tiuj ĉi listoj li volas elpresi nur certajn parametrojn -- genron kaj specion, profundecon, kaj akvo-temperaturon -- nenion plian.

Kreiva aplikado de la SELGEM-a programaro povus ebligi ĉion ĉi-supre priskribitan, kaj ankoraŭ multon alian.

4. La strukturo de SELGEM

Por pli bone kompreni kiel oni aplikas SELGEM, estas utile iom scii pri datumstrukturo en SELGEM-a rikordaro. Ni konsideru la ekzemplon de malgranda biblioteka kartaro.

4.1. De kartaro al rikordaro

Ĉiu unuopa karto rilatas al unu libro, kaj ĉiu libro havas katalognumeron, kiu unikigas ĝin en la kolekto. Sur ĉiu karto troveblas pluraj informeroj: aŭtoro, titolo, jaro de eldono, nomo de la eldoninto, nombro de paĝoj, mallonga priskribo de la enhavo, nombro de ilustraĵoj, k.s. En SELGEM-a rikordaro, la tuta informo rilatanta al unu libro enirus tion, kio nomiĝas SELGEM-a dokumento. Tiu ĉi dokumento havus dokumentan numeron, kiu unikigus ĝin en la rikordaro (analogie al la katalog-numero de la libro), kaj en nia ekzemplo oni povas koncepti la dokumenton kiel ekvivalenta al karto en kartaro. La specifaj informeroj (aŭtoro, titolo, jaro de eldono, k.s.) konsistigas kategoriojn ene de SELGEM-a dokumento. Kiam la informo destinita al certa kategorio estas iom ampleksa -- ekz., kiam temas pri resumo aŭ priskribo -- pluraj apartaj SELGEM-aj linioj povas ekzisti en unu kategorio.

4.2. Rikordara strukturo

Ĉiu SELGEM-a dokumenta numero entenas ok ciferojn, ĉiu kategoria numero tri ciferojn, kaj ĉiu linia du. En ĉiu linio estas spaco por precize 64 karakteroj da informo (t.e., numeroj, literoj, interpunkciaj signoj, k.s.). La kombino de dokumenta numero, kategoria numero, linia numero, kaj informo konsistigas unu komputilan rikordon, kaj tiu rikordo estas ĉiam precize 77 karakterojn longa. Vi povas koncepti la komputilrikordon kiel unu SELGEM-a linio de informo, kune kun la linia, kategoria, kaj dokumenta numeroj kiuj unikigas ĝin en la rikordaro. Ekzistas maksimume 99 linioj ene de iu ajn difinita kategorio, 999 kategorioj por iu ajn dokumenta numero, kaj 99 999 999 dokumentaj numeroj en iu ajn rikordaro. Sekve, iu ajn difinita SELGEM-a rikordaro povas enteni ĉirkaŭ 6.33×10^{14} silabojn da informo, kio sufiĉas por la plej multaj normalaj celoj.

4.3. Programaj interrilitoj

Sekvas kelkaj tipaj programaj sinsekvoj, kune kun klarigo pri la celo de la programa kunligado.

- a) SELUPD - SELKEY - SELLST (Oni ĝisdatigas rikordaron, disordigas ĝin laŭ nova vicordo, kaj tiam listigas ĝin)
- b) SELKEY SELRPV (Oni disordigas rikordaron laŭ nova vicordo kaj listigas ein laa speciala formato indikita far la uzanto mem)
- c) SILEXT SELKEY - SELLST (Oni ekstraktas certajn rikordojn el rikordaro surbaze de difinitaj logikaj kriterioj, disordigas ilin, kaj listigas ilin)

ĉ) SELKWX - SELKEY SELRPV (Oni kreas la interan rikordaron por laŭtema indekso, disordigas la rikordaron, kaj presas la indekson laŭ formato difinita far la uzanto).

5. La apliko de SELGEM: pritakso de rezultoj

5.1. La atingoj

Ke la anoncitaj celoj de BIBJAD estis atingitaj pere de SELGEM, ekzistas neniu dubo. Tamen tio certe ŝuldiĝas parte al la relativa modesteco de la celoj mem. Oni tute facile ricevas nun listojn laŭ ĉiuj deziritaj vicordoj, kaj ankoraŭ kelkaj novaj. La ekstraktaj kapabloj estas kontentigaj, eĉ multe pli vastaj ol normale bezonataj. Same rilate al la fleksebleco de la sistemo, niaj impresoj devas esti principe pozitivaj. La kapablo produkti indeksojn laŭ difinitaj vortoj aŭ konceptoj ne estis nepre postulo komence de la projekto, kaj la facileco de indeksado pere de SELGEM estis agrabla surprizo. La raportscriba programo (SELRPV) ebligas grandan variecon de raportformatoj, difinitaj fare de la uzantoj mem.

5.2. La mankoj

Se kompari la sistemon al iu teoria idealo, tamen, ĝi montras kelkajn mankojn.

Unue, necesas konfesi, ke la programoj ne estas tute senereraj. La verkintoj en Smithsonian Institution faras sian eblon por korekti ĉiujn erarojn, pri kiuj ili ekscias, sed neniel formale garantias la fidindecon de la programaro.

Pri eventualaj problemoj respondecas do ĉefe la uzantoj mem*.

Due, la relative granda diverseco de regaj kartoj, sen evidenta interna unueco, nenecese komplikigas la aplikadon de la programoj. Ekzemple, en SELKEY (la disordig-prepara programo) por komuniki kampon uzotan dum la disordigo oni devas indiki la komencan kaj finan karaktropoziciojn (ekz., '1534' signifus 20-karaktran kampon komenciĝantan en la 15-a pozicio kaj finleantan en la 34-a). Dume, en SELRPV (la raportskriba programo) oni indiku por uzota kampo la komencan karaktropozicion kaj la longecon de la kampo (do '1534' estus 34-karaktra kampo komenciĝanta ja en la 15-a pozicio sed finiĝanta en la 48-a). Tia manko de interna reguleco estas la fonto de multaj eraroj ĉe nespertaj uzantoj.

Trie, en SELKES (la disordiga programo) oni estas devigata disordigi ĉiam laŭ la plena longeco (200 karaktroj) de la ŝlosilkampo, eĉ kiam -- kiel normale okazas -- sufiĉas uzi efektive multe malpli longan ŝlosilkampon. Tio multe plialtigas la koston de disordigoj, krom se oni anstataŭigas SELKES-on per la propraj disordigaj algoritmoj (kiel oni efektive faris en la Instituto "Ruder Bošković").

Laste, almenaŭ unu el la programoj -- SELEXT (la ekstrakta programo) -- ampleksas tiom da instrukcioj (ĉirkaŭ 2500) kun tiom komplika logiko, ke modifoj fare de uzantoj fariĝas signife malfacilaj. Por normalaj uzantoj ĝi estas praktike netuŝebla elemento. Se ĝi kontentigas la bezonojn de la uzanto, bele; se ne -- domaĝe.

* Menciindas, ke laŭ la spertoj de la aŭtoro la malmultaj eraroj neniel kaŭzas misan kalkuladon aŭ manipuladon, sed simple malebligas la plenan aplikon de la sistemo kiel difinita en la dokumentaĵoj.

5.3. Poziciaj relativeco kaj absoluteco

Unu kurioza fadeno trairanta la sistemstrukturon estas la ekzisto de du malsamaj manipul-konceptoj: la relativeco kaj la absoluteco poziciaj. En la plej multaj programoj regas ĉi-lasta. Nome, se la uzanto volas, ni diru, disordigi certan rikordaron laŭ alfabeto vicordo li devas precizigi la absolutan pozicion de la ŝlosilkampo ene de la unuopaj rikordoj, ekz., de la 33-a ĝis la 42-a karaktropozicio -- ĉiam. Tamen por almenaŭ unu gravega programo (SELEXT), volante ekstrakti informojn el rikordaro oni indiku - devige -- la relativan pozicion de la ekstraktota vorto (t.e., karaktro-vico), ekz., "se la kvina vorto de maldekstre egalas al 'PUDINGO', elskribu ĉi tiun dokumenton". Tiuj ĉi du apartaj aliroj al manipulado necesigas tre zorgan planedon de rikordar-strukturo, se oni ne volas resti poste kun informo praktike ne-ekstraktebla (aŭ nedisordigebla).

6. Konkludoj

Estas banalaĵo, sed vere, ke neniu sistemo estas perfekta. La grandaj fleksebleco kaj ĝeneraleco de SELGEM estas fortoj parte kompensantaj la malavantaĝojn cititajn en Sekcio 5. La fakto, ke Smithsonian Institution disponigas la sistemon senpage al senprofit-celaj organizoj estas faktoro neniel neglektinda de planantoj serĉantaj iun taŭgan sistemon por siaj inform-manipulaj bezonoj.

Al komputiluzantoj en arope prilabora medio aŭ kun nur modestaj interaktivaj kapabloj, SELGEM estas tute

esplorinda ilo montrante plurajn avantaĝojn. Dum du-jara eksperimenta aplikado al bibliografiej taskoj en la Instituto "Ruder Bošković", ĝi pruvis sian esencan utilon.

Literaturo

1. Creighton, Reginald A. kaj Crockett, James J.:
"SELGEM: A System for Collection Management", 50 p.,
aŭgusto, 1971, Smithsonian Institution ("SELGEM: Sistemo
por Kolekto-prizorgado").
2. Creighton, Reginald A. kaj Packard, Penelope: "Computer
Assisted Information Management: Retrieval", 48 p.,
junio, 1975, Smithsonian Institution ("Komputil-Helpata
Inform-manipulado"). Tiu ĉi amplekse priskribas la
retrov- kaj ekstrakt-kapablojn de SELGEM.
3. Pliajn informojn kaj ampleksajn priskribojn oni povas ricevi,
skribante al: James J. Crockett
Manager, Collection Management Division
Office of Computer Services
Smithsonian Institution
Washington, D.C. 20560
USONO.

Demandojn pri la funkciado de SELGEM en la Instituto
"Ruder Bošković" oni direktu al:

Jim Cushing
CIM IRB
p.p. 1016
41000 Zagreb
JUGCSLAVIO
(Tel: 041/272-611 - int. 431).

NOTO: La terminologio uzata ĉi-referaje baziĝas sur la
kvin-lingva vortaro komputil-teknika eldonita en
Budapeŝto en 1972, kaj kies Esperantan sekcion
redaktis Antal MÜNNICH.

La antaŭa kaj nuntempa apliko de la
komput-tekniko en la Hungara Poŝto

Árpád HARMATI

La apliko de la komput-tekniko en la Hungara Poŝto komenciĝis en 1960. La komencajn aplikojn karakterizis tio, ke la komputiloj enpenetris en la entreprenan laborprocezon. Sed ili estis nur unuopaj taskoj, kiuj estis ligitaj unuavice kun eksped-prilabora, fakturada, kontrolada agado de komunikad-teknologiaj procezoj, kiuj estis centrigite plenumblaj.

Lastatempe aperis komputilaj sistemoj, plenumantaj pli malsimplajn komputojn kaj decid-antaŭpreparojn, unuavice sur la kampo de ekonomiko, librotenado, teknika projektado.

La cefa parto de la komput-teknika rimedaro de la Poŝto estas ĉe PSZSZI /Komput-teknika kaj Organizada Instituto de la Poŝto/, kiu plenumas la aranĝas centran prilaboradon. PSZSZI por solvi la komput-teknikajn taskojn de la Poŝto, unue funkciigis 5 proprajn komputilojn. En tiun ĉi nombron ne estas enkalkulataj la programeblaj kalkuliloj, kaj la malgrandaj maŝinoj aplikataj por specialaj celoj /ekzemple por regado de leterdirektanta konversilo/.

La plej malnova komputilo de PSZSZI, funkciigita ekde 1969, estas la Honeywell-2000, kiu ankoraŭ nun funkcias sekure, sed kontinue okazas la transporto de ĝiaj taskoj sur komputilojn de Unueca Komputil-Sistemo /UKS/ de socialismaj landoj. Ĉar la komputilo jam amortiziĝis, oni devas kalkuli je esenca difektiĝo de ĝia sekureco. La taskojn, eĉ se nur alternative, sed oni devas transporti sur UKS-komputilojn.

En PSZSZI en 1976 komencis funkcii du R-30-tipaj sam-konstruitaj komputiloj, kun centra storkapacito 256 kilosilaboj.

La maŝinoj krom kelkaj novaj prilaboradoj, progresive transprenas la taskojn de Honeywell-2200. Konsidere la transportadon de la taskoj, ĝis alveno de nova komputilo, ne estas ebleco solvi esence novajn taskojn per tiu ĉi komputilaro. La PDP-8, funkcianta en la Poŝta Eksperimenta Instituto /PKI/ kaj la TPA-tipa komputilo funkcianta en la Radio-Televida Teknika Direkcio de Poŝto, solvas esploradajn-planadajn taskojn, sed la konfiguracio de tiuj du komputiloj nur limigite taŭgas por solvi la aktualajn taskojn.

Ĉar la poŝta esploro-planado utiligas la teknologian datumaron, en la estonto estos pli konvena per alia metodo solvi la provizadon per komput-teknikaj iloj de la aplikada kampo.

La sekura funkciigo kaj kvalita perfektigo de la nune funkciantaj komputilaj sistemoj, kaj la realigo de la plej urĝaj taskoj - inter limigitaj eblecoj - plenumebblas plej konvene per la rekonstruo de komputilaro de PSZSZI kaj per la decentralizita evoluigo de datum-fiksado.

La prilaboroj, kunligitaj kun la bazinstitutaj taskoj, kaj la ekzistantaj, kaj la ellabendaj sistemoj bezonas komputilon, havantan grandan centran kaj fonan storon.

La celo kaj motivoj de estigo de la sistemo

La teknologiajn procezojn de la telekomunikaj liveradoj kaj la organizan ordon fundamente determinas

- la altgrada laŭterena malcentraliziteco
- la multspecaj servoj
- la ret-karakteraj ekspluatadaj sistemoj.

El tiuj sekvas, ke en la poŝto la aplikadon de la komput-tekniko oni devas evoluigi en la direkton de dividita prilabora sistemo, kiu ebligas centran utiligadon de informoj kaj sekvas

la organizan dispartigon. Nune funkcias sola komputil-centro, en kiu oni devas kontentigi fundamente ĉiujn informajn dezirojn de la poŝto. La plimulto de la komput-teknikaj bezonoj de la mezgradaj organizoj de poŝto, ankoraŭ ne estas kontentigataj. La poŝtaj organoj ĝis nun faris plurajn unuopajn pasojn - salajro-prilaborado, aĉeto de malgranda maŝino - sed tiuj en perspektivo kondukas al dispartigo de fortoj, kaj rezultas izolitan solvodon de lokaj taskoj.

Pro tio urĝe necesas elkonstrui la decentralitan komputil-reton, antaŭplanitan en la komput-teknika projekto de la Poŝto. Tiu reto zorgos pri la inform-prilaboradaj taskoj de la meza poŝtaj organoj.

La elformadon de la decentraligitaj datum-kolektantaj kaj - prilaborantaj komputilcentroj ĉe la opaj poŝtdirekcioj oni realigas post ĉirkaŭ 2-3 jaroj, sed plej malfrue ĝis la fino de 1984. La fundamenton de la komputilcentroj formas la TPA-1140-tipaj komputiloj, produktitaj fare de la KFKI. /tio estas Centra Fizika-Esplor Instituto de Hungario/. Tiu maŝintipo kongruas kun la PDP 11/40 - komputiloj de la firmao "Digital Equipment Computer" /mallongige DEC/.

La sistemo konsistas el vaste aparataro kaj programaraj iloj, havas multnombrajn periferierojn kaj riĉan programaron. La produktintoj emfazis la datumtransportajn aplikojn. Al la sistemo TPA-1140 produktita fare de KFKI, adapteblas ĉiu kutima periferio. Estas ebleco funkciigi malproksimajn terminalojn, kaj kunlabori kun centra komputilo. La disponeblaj operaciaj sistemoj estas la sekvaj:

FOBOS /similas al RT - 11/

DOS-RV /similas al RSX- 11 M/

Por la utiligantoj estas disponeblaj programlingvoj: BASIC, FORTRAN-IV, DIBOL /MIDIBOL/, DEC-FORM /FORM-11/. La parentezitaj programlingvoj signifas lingvojn servantajn la saman celon dum la DOS-RV, la senparentezaj dum FOBOS. Dum la superrigardo

de DEC-FORM /okaze de operacia sistemo RSX-11 M/ estas ebleco funkciigi komputilon TPA-1140 kun terminaloj por datumfiksadaĵoj funkcioj. El socialismaj landoj oni povas aĉeti magnetdiskojn kiel fonstoron kun maksimuma kapacito $3 + 3 = 6$ megasilaboj, kaj magnetbendon kun kapacito 23 megasilaboj /kaj kun 64 kilosilaboj/sekundo transporta rapideco/ por la TPA-1140.

Nun ni esploru kiel okazas la adaptado de la komputilo/j/ al la poŝta reto? La komputiloj /TPA-1140/ de la malcentralizigitaj komputil-centroj teknike estos kunligitaj kun la nova centra komputilo funkciigonta post la rekonstruado /laŭplane post ĉirkaŭ 1-2 jaroj/ de komputiloj de PSZSZI.

La malcentralizigitaj komputiloj estos terminaloj por la distanca pakeda prilaboro /remote batch/ de la centra komputilo. La kondiĉo de alkroĉado trans la datumkomunika linio, estas la sinkronigita aliĝada ebleco flanke de la granda komputilo, kaj programo kiu manipulas la terminalon IBM 2780-tipan. Sur la flanko de TPA-1140 bezonatas sinkroniga linio-adaptilo, komunikada aritmetika elektado kaj konkuranta programo /emulilo/. La aparataraj elementoj haveblas por la TPA-1140/.

Estas planita akiri du ekzemplerojn da dufadenaj datumkomunikaj linioj, kiuj ebligas rapidecon 1200-4800 bito/sekundo, inter la TPA-1140 kaj la centra komputilo funkcionta en PSZSZI.

Por transporti datumojn sufiĉas 8 laborlokoj, sed eĉ kelkaj mezgradaj poŝtaj organoj pro la granda kvanto de la fiksendaj /datumoj ni planas 12-16 laborlokojn. Dum la elektado de periferieroj-tipoj ni konsideris aparatojn devenantajn el la socialismaj landoj. La KFKI solvis la adaptadon de pluraj periferieroj, devenantaj el okcidentaj landoj /ekzemple grandkapacitaj magnetdiskoj/. Samtempe okazis planado ankaŭ de kompletigaj aparatoj al funkciigo de komputil-centroj /tajpilo, fajroalarmilo, klimatoregulilo/.

La ekonomiaj motivoj de la sistem-elformigo

La elformigo de la datumkolektadaj kaj prilaboradaj centroj povas havi diversajn rezultojn. Gravas la fakto mem, ke la komput-tekniko elpaŝas el la ĝis nuna malvasta aplikado. Por la plej grandaj poŝtaj organoj ĝi fariĝas natura kaj atingebla laŭ la deziroj /bezonoj/.

La elformiĝinta reto povas formi bazon de efektiva poŝta komputilreto, kiu konektiĝas al granda laborprodukta centra maŝino, kaj al kiu oni povas konekti diversajn funkciigadajn objektojn. Tio estas esenca vidpunkto, ke la reto-elformigo komenciĝas per tia komputil-familio, kiu enhavas diversgrandajn maŝinojn.

Tiu maŝintipo apartenas al kategorio de minikomputiloj, por kiu pli kaj pli karakteriza estas la tendenco de prez-malgrandilo. Konsekvencas el la avantaĝoj, ke la plievoluigo de la reto ankaŭ baziĝas sur tiu komputil-familio. Oni povas konsideri je la plej granda avantaĝo tiun fakton, ke finiĝos la nuntempa komputil-luado, kies grandeco kreskus en la sekvontaj jaroj.

La enkonduko de la priskribitaj sistemoj rezultigas la kvantan kaj kvalitan pliboniĝon de la poŝtaj-telekomunikaj servoj, disvastiĝantajn en la tuta lando, kaj kontribuas al la kresko de la efikeco de la Hungara Poŝto.

Biografio



Mi naskiĝis en 1955. En 1977 mi finis la Budapeŝtan Elektroindustrian Teknikan Altlernejon, la fakon prikomputilan. Mi laboras pri kontaktigo de komputiloj unu kun la alia, precipe el la vidpunkto de etkomputilo TPA-1140.

Laboroj de la komputil-statistika divizio de la
Esplorinstituto pri la komputtekniko kaj Automatizado
de la Hungara Scienca Akademio, krociĝantaj al kurac-
biologiaj demandoj

Dr-o István Ratkó

dr-o Mihály Ruda

En nia prelego ni prezentas mallongan resumon pri komputilsistemo, patronanta medicin-sciencajn esplorojn daŭrantajn en ĉi-lastaj jaroj en la Esplorinstituto pri la Komputtekniko kaj Automatizado de la Hungara Scienca Akademio.

1. Hospitala malsaniĝaj esploroj en Hungario

En Hungario hospitala indik-kolektado daŭras ekde ĉirkaŭ 1870. Homogena statistika liverado de indikoj estas de post 1953 en niaj hospitaloj. Komence la indikoj estis prilaboritaj manmode. En 1967 jam estis aplikita Hollerith-maŝino, sed la prilaboro estis tro malrapida kaj pezmova.

En Hungario la unua komputila datum-prilaborado okazis en 1973-74. Jam en tiu prilaborado ankaŭ nia divizio partoprenis. La sampla entenis pli ol 600 mil vartadajn okazojn. La tasko lanĉis multajn interesajn komputteknikajn kaj matematikajn problemojn [1].

La diversaj inform-liveradoj enhavis unuavice la sekvajn faktorojn:

- la laŭaĝajn, laŭseksajn, laŭprofesiajn kaj laŭloĝlokajn distribuojn de la pacientoj
- la prizorgitecon de certaj teritorioj
- la disvastiĝon de diversaj malsanoj
- la dividiĝadon de la vartada tempo inter certaj teritorioj, medicinaj fakoj

Ankaŭ la komparadoj kun la pli fruaj patrolandoj kaj internaciaj indikoj havas signifan rolon en la taksado.

2. La informa sistemo SIS77

Pro la granda dimensio kaj la komplikeco de la hospitala malsaniĝa esploro oni ne povas kontentigi la informajn pretendojn per kelkaj antaŭfiksitaj statistikaj tabeloj.

Ni solvis tiun problemon helpe de elasta demondanta sistemo. Elformigon de ĝenerala kaj elasta demondanta sistemo necesigis la ŝanĝemo de la pretendoj de la malsaniĝa esploro.

La hungarlando hospitala malsaniĝa esploro fariĝis kontinua ekde 1974. Por la kontinua prilaborado de la indikoj necesiĝis bone uzebla statistika programpaketo. La Higiena Ministerio pri tiu laboro komisiis nian divizion.

Tiel ni pretigis la informan sistemon SIS77 /statistika informa sistemo 1977/[2].

Sistemo SIS77 konsistas el memstaraj programoj. Krom la pretigo de statistikaj tabeloj la sistemo plenumas kontroladojn, kodadojn kaj elektadajn taskojn.

La postaj taskoj aperas en taksado de komplikaj logikaj esprimoj respektive de entjer-valoraj funkcioj [3,4]. La sistemo enhavas ankaŭ artajn rikordaro-konstruadajn eblecojn.

La programoj de la sistemo estas ĝeneralcelaj proceduroj parametrizeblaj. La prilaboradojn plenumas ne ĉi tiuj programoj, sed la generitaj programoj fare de ili. La specialaj programoj plej adekvatoj al la donita tasko, re-kaj refoje generiĝas.

Tio garantias la efikan funkciadon de la sistemo.

La pligrandigo de la efikeco necesigis la pluevoluigon de SIS77. La rezulto de nia evoluiga laboro estas la informa sistemo SIS79/GENERA [5,6], pri kiu nun ni ne parolas, ĉar ĝi povus esti la objekto de memstara prelego.

3. Registramo de infarkto

Mortadaj statistikoj kaj de Hungario kaj de la evoluantaj landoj unusence montras la signifon de la cirkuladaj malsanoj /precipe de la Akuta kormuskola infarkto, mallongige AKI/.

En Hungario la cirkuladaj malsanoj estas respondecaj pro pli ol duono de la mortadnombro. Homoj mortintaj sekve de AKI nombras pli ol dek procentojn de la tutnombro.

En la aktiva perlabora aĝo la frekvenco kaj la efiko de AKI, kaŭzanta invalidecon eĉ pli urĝas lukton kontraŭ la malsano. Pro tio organizis la tielnomaton Registramon de Infarkto /mal-longe RI/ en 1970, en la sud-Peŝta teritorio, laŭ la propono de la Monda Higiena Organizaĵo. La manstorado de indikoj de RI estas pezmova. Tiel la metoda datumliverado estas plenumbla nur per granda administracia aparato. La rezultoj je nia dispono nur 4-5 jaroj post la komenco de la kolektado de indikoj. Tio necesigis la ellaboradon de la komputilsistemo de RI. RI donas infor-

mojn al diversaj niveloj de la higiena provizado kaj estraro:

- 1/ al la distrikta kuracisto pri siaj pacientoj
- 2/ al la hospitaloj de la sud-Peŝta teritorio
- 3/ al la estraj ĉefkuracistoj de la sud-Peŝtaj distriktoj
- 4/ por sciencaj esploradaj celoj

Ni raportis pri tiu laboro inter aliaj en [7].

4. La komputila registrado de la kor-operaciitoj

La celo de la sistemo estas:

- a/ La registrado de malsanuloj atendantaj por operacio.
- b/ La pretigo de tiuj listoj pri malsanuloj atendantaj, kiuj helpas la inviton por operacio, kaj kiuj listoj kontentigas donitajn logikajn kondiĉojn.
- c/ La registrado de la jam operaciitoj al iakaŭze ne operaciotoj /ekzemple la mortintoj/.
- d/ Pretigo de diversaj statistikoj.

Por kontentigi tion, estas necesaj la sekvaj datumfolioj:

- la folio de atendantoj por operacio /FAO/
- inspektada folio antaŭ operacio /IFAO/
- operacia resigna folio /ORF/

La celo de FAO estas la kolektado de tiuj gravaj datumoj pri la atendantaj malsanuloj, kiuj determinas la bazon de la envoko kaj donas inforwon pri la kompleteco de la esplorado.

La celo de IFAO estas, ke dum la envoko ni konsideru la novan enregistradon elformitan surbaze de inspektado, la sekvan templimon de la operacio kaj la kompletecon de la antaŭ-operacia esplorado. Krome ĝi malpliigu la nombron de misaj envokoj por operacio, per la anticipa filtra esploro.

La celoj de ORF estas; ke

- ĝi donu signon al la komputilo pri tiuj, kiujn la kuracistoj envokis, por ke la komputilo ne elskribu ilin denove, kiel envokotojn, kaj tiel la komputilo ne pertruĥu la ordon de la envokado.
- anonco de la ŝanĝoj. Se iu datumo pri operacio de la atendantaj malsanuloj ŝanĝiĝas, tiam ORF indikas tiun ŝanĝon al la komputilo.
- ĝi ordonu la konservadon /arkivgadon/ de la datumoj de operaciitoj aŭ neoperaciotoj el ia ajn kaŭzo.

- la komputilo ricevu pri tiuj malsanuloj, kiujn oni devas envoki iakaŭze pli malfrue.
- ĝi forstreu el la atendada listo kaj samtempe ĝi konservu certajn datumojn.

De tempo al tempo ni donas statistikojn kaj listojn pri atendantaj malsanuloj konsidere specialajn vidpunktojn.

Ni funkciigas la sistemon tra terminalo, certaj partoj de ĝi funkcias dialoge.

5. Niaj plujaj planoj

Nia celo estas prikonstrui en la sistemon GENERA krom la datumo-manipuladoj kaj priskribaj statistikaj elementoj uzitaj el la hospitalaj esploroj, taskojn matematikajn statistikajn kun akcento.

Kvankam ekzistas matematikaj statistikaj programoj- kiel ekzemple la BMDP- sed nia celo estas la realigo de tiuj sistemoj en pli elasta kaj etendebla formo.

Laŭ nia opinio la sistemo GENERA garantias bonan bazon por la organizado de prilaboro de grandaj kaj komplikaj datumsistemoj.

En la temocirklo malsano-registrado nia nova imago estas la ebleco de la aplikado de la rapidege disvastiĝantaj person-komputiloj en nia tagoj. Ĉi tiuj aparatoj havas senĉese kreskantan plenumokapablecon kaj fidindecon, samtempe iliaj prezoj kontinue malpliiĝas. En nia divizio daŭras la esploro de modelo TRS-80 kuneksperimenta karaktero. La kuracista-biologia apliko de la person-komputiloj estas memkomprenebla. Ne nur pro tio, ĉar kelkaj firmoj /ekzemple Hewlett-Packard/ roligas siajn aparatojn ankaŭ en instrumenta konfiguracio kun akcento, sed ankaŭ pro tio, ĉar esploro de la biologiaj modeloj -pro la komplikeco de la modeloj- pretendas kontinuan, dialogan funkciadon, kio estas multfoje pezmovi en grandaj komputilcentroj. La prezo de la person-komputiloj estas tiom, ke ankaŭ esplorgrupo disponanta je pli modestaj oblecoj povas aĉeti ilin. Ties aplikon oni povas imagi ankaŭ po unu hospitala klaso, dum la malsanulo-akcepto, malsanulo-sekvado, ktp.

- [1] J.Garádi,A.Krámli,I.Ratkó,M.Ruda: La apliko de statistikaj kaj komput-teknikaj metodoj en hospitalaj malsaniĝaj esploroj /hungare/, MTA SZTAKI, Studoj, 35/1975
- [2] M.Ruda: Statistical Information System with Health Service Application, 4-th Winterschool of Visegrad on the theory of Operating Systems, Szentendre, 1978
- [3] I.Ratkó: On optimization problems of logical expressions in programming languages, Colloquium on mathematical logic in programming, Salgótarján, 1978
- [4] I.Ratkó: Kelkaj rimarkoj pri la statistika datum-prilaboro, 1-a konferenco apliko de esperanto en la scienco kaj teĥniko, 1981, Žilina
- [5] A.Krámli,M.Ruda,P.Kerékfy: Large sample size statistical information system for HWB, Data analysis and informatics, North-Holland, 1980.
- [6] P.Kerékfy: GENERA -A program generater system-Progress in cybernetics and system research, Data Base Design, Hemisphere P.C., 1980.
- [7] I.Ratkó: A data base management system for patients suffering from acut myocardial infarction Progress in cybernetics and sytem research, Health care systems, Hempisphere P.C. 1980.

LISTO DE LA AŬTOROJ

=====

Jen la nomoj kaj la adresoj de la personoj kiuj sendis referaĵon eldonitan en tiu kajero, multaj sendis pere de la komputista grupo de HEA en Budapest.

<p>ANDRJUŠČENKO V. M. SOVETUNIO pere de komputista grupo de HEA.</p> <p>ANGYAL József Budapest, V. Március 15 tér 1 HUNGARIO NOVEX</p> <p>Ing. BERTIN Christian CCETT VRE/TDP B.P.: 1266 2, rue de la Mabilais F-35013 RENNES-CEDEX FRANCUJO</p> <p>D-ro Ing BRO CZÓ Péter HUNGARIO pere de komputista grupo de HEA</p> <p>CSÁS ZÁR Gyula BHG Híradástechnikai Vállalat, Budapest P. F. 2. H-1509 HUNGARIO</p> <p>CUSHING Jim Podbrežnica, blok XIV, stan 2 YU-41410 VELIKA GURICA JUGUSLAVIO</p> <p>ENDRÉDI Csanád Budapest, Akácfa u. 13. H-1204 HUNGARIO</p> <p>dro FISCHER Rudolf-Josef Institut für Medizinische Informatik und Biomathematik der Westfälischen Wilhelms-Universität Münster D-4400 MÜNSTER Hüfferstr. 75 FED. RESP. GERMANIO.</p>	<p>Agr. GERŐ Péter Budapest Galgóczy út 5-7/c H-1125 HUNGARIO</p> <p>GOFEN Aleksandro M. SOVETUNIO pere de komputista grupo de HEA</p> <p>HARMATI Árpád HUNGARIO pere de komputista grupo de HEA</p> <p>Komputista grupo de HEA H-1368 BUDAPEST P. F. 193 HUNGARIO</p> <p>Ing. NĚMEC vladimír Zvolenská 5/96, vlčince 3 CS-01008 ŽILINA ĈEĤOSLOVAKIO</p> <p>D-ro RATKÓ István H-1112. BUDAPEST Gulyás U. 19. HUNGARIO</p> <p>D-ro. SHERWDUD Bruce Arne Computer-based Education Research Lab. University of Illinois at Urbana- Champaign 252 Eng. Res. Lab. 103 S. Mathews. USA-URBANA, ILLINOIS 61801 USONO</p> <p>Ing. SZÉKELY Peter EGER Kolacskovszky L. u. 2/a H-3300 HUNGARIO</p> <p>SZŐKE László HUNGARIO pere de komputista grupo de HEA.</p>
---	--

VANDEN BEMPT Marc
Duisburgsestraat 12
B-5030 LEUVEN-HEVERLEE
BELGIO

WITKAM A. P. M.
BSD/Automation Technology b. v.
P. O. Box 8348
NL-35U3RH UTRECHT
NEDERLANDO

D-rino ZÁRDA Sarolta
HUNGARIO
pere de komputista grupo de HEA

TABELO DE LA ENHAVO

=====

Antaŭparolo. Pl.

La cibernetiko kaj la socio. Peter SZÉKELY. p6.

La ĉapelitaj literoj en informatiko. Rudolf-Josef FISCHER.
p13.

Komputila traktado de esperanta teksto. Bruce Arne SHERWOOD.
p21.

Esperanto kiel pontlingvo en komputilretoj - mankoj kaj
rimedoj. A. P. M. WITKAM. p42.

La studo de la komputil-devenaj lingvoj el la lingvistika
vidpunkto. V. M. ANDRJUŜĈENKO. p54.

Per niaj homaj vortoj ni parolu al la maŝino. Péter GĚRO. p85.

La subrutino de eligiloj en la operaciad-sigtemo por regado de
teknologiaj procezoj. Vladimír NĚMEC. p97.

HSZR-1 Retokonstrua kaj -desegna programo. József ANGYAL.
p102.

Disdono de la ŝarĝo inter telefon-subcentralaj manipulantoj.
Gyula CSÁSZÁR. p116.

Aŭtomata telekomunika sistemo regata de komputilo IBM S/7,
instalita ĉe Hungara Meteologia Organizaĵo. Czanád ENDRÉDI.
p135.

Modula programado. Marc VANDEN BEMPT. p144.

La mezurado de la programaro. Sarolta ZÁRDA. p154.

Dinamikaj, proprecoj de programaraj proceziloj. Lázsló SZŐKE.
p162.

La plurcela ligilnivela proceduro PLP. Christian BERTIN. p194.

Rapida malkompono en Tejloran serion kaj solvo de Koŝi-
problemo por sistemoj de diferencialaj ekvacioj, ordinaraj kaj
kun partaj derivaĵoj. Aleksandro M. GOFEN. p204.

Perkomputila optimumigo de urbega enlernejigo. Péter BROCKÓ.
p232.

La informsisitemo SELGEM: pritakso de bibliografiaj spertoj.
Jim CUSHING. p245.

La antaŭa kaj nuntempa apliko de la komput-tekniko en la Hungara Poŝtoŝofico. Arpád HARMATI. p256.

Laboroj de la komputil-statistika divizio de la Esplorinstituto pri la komputtekniko kaj Aŭtomatizado de la Hungara Scienca Akademio, krociĝantaj al kurac-biologiaj demandoj. D-ro István RATKÓ, d-ro Mihály RUDA. p261.

Listo de la aŭtoroj. p266.

Tabelo de la enhavo. p268.