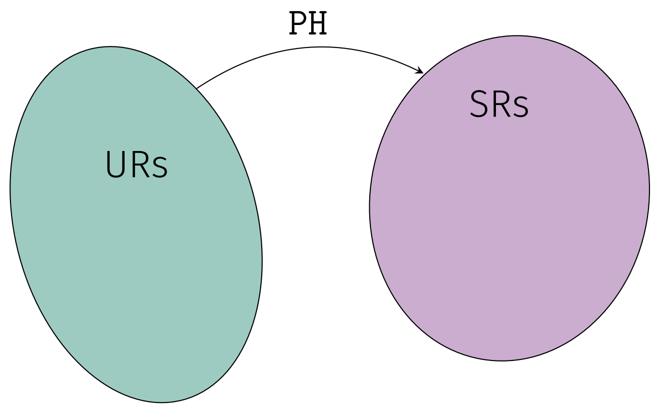
A subregular approach to the problem of learning underlying representations

Adam Jardine
Rutgers University

December 9, 2019 · Tel Aviv University

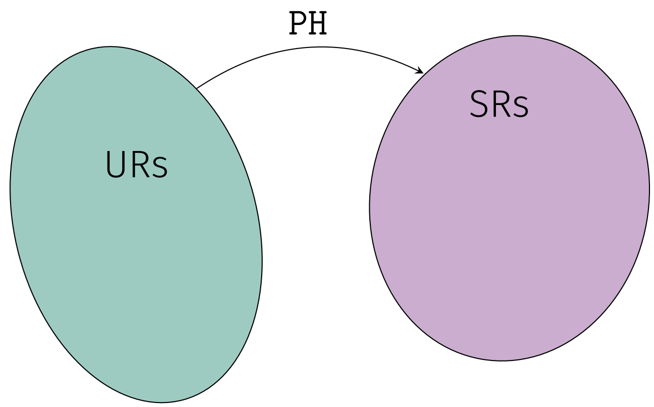
What is the nature of phonology?



What is the nature of:

- constraints on SRs? URs? (Halle, 1959; Prince and Smolensky, 1993; Gorman, 2013)
- maps from URs to SRs? (Chomsky and Halle, 1968; Johnson, 1972)
- relation between SRs and URs? (Hyman, 1970; Kiparsky, 1973; Kenstowicz and Kisseberth, 1977)

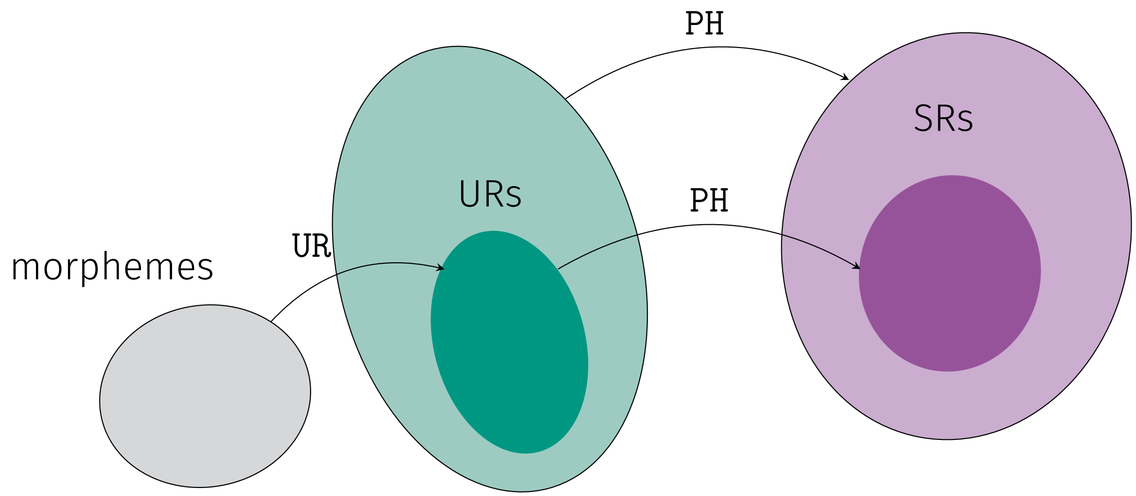
What is the nature of phonology?



What is the **computational** nature of **(learning)**

- **constraints on SRs?** URs? (Heinz, 2009, 2010)
- **maps from URs to SRs?** (Jardine et al., 2014; Chandlee and Heinz, 2018)
- relation between SRs and URs?

Learning URs and a grammar



- Computational restrictions on maps from URs to SRs provide avenue for **learning URs and a grammar**
- This includes **restrictions on relation between SRs and URs**

Learning URs and a grammar

- **Learning problem:** the **simultaneous inference** of URs and a grammar from SRs in a morphological paradigm

  (Tesar, 2014; Cotterell et al., 2015; Rasin et al., 2018)

- **Today:** the **subsequential** functions provide a structure that can solve this problem

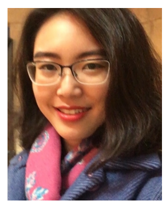
  (Mohri, 1997; Heinz and Lai, 2013; Jardine et al., 2014)

- More specifically, we'll look at **input strictly local (ISL)** functions (Chandlee and Heinz, 2018)

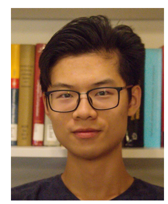
Learning URs and a grammar

- This poses further restrictions on the relationship between SRs and URs

- This is joint work with students at Rutgers



Wenyue Hua



Huteng Dai

- This is very much work in progress!

The learning problem

The learning problem

English plural:

CAT-PL	[kæts]
CUFF-PL	[kʌfs]
DEATH-PL	[dɛθs]
GIRL-PL	[gərlz]
CHAIR-PL	[tʃerz]
BOY-PL	[bɔɪz]
...	...

Analysis:

- A **map** from morphemes to URs

$$\begin{array}{lll} \text{CAT} & \rightarrow & /\text{kæt}/ \\ \text{PL} & \rightarrow & /\text{z}/ \\ \ldots & & \ldots \end{array}$$

- A **map** from URs to SRs

$$/\text{z}/ \rightarrow [\text{s}] \ / \ [-\text{voi}] \ \_\_\_$$

The learning problem

- M: finite set of morphemes $\{\textsc{cat}, \textsc{dog}, ..., \textsc{pl}\}$
- Σ: finite set of segments $\{\text{a, b, β, ..., z}\}$
- Learning targets:
  - lexicon function $\texttt{UR} : M^* \to \Sigma^*$

$$\begin{aligned} \texttt{UR}(\textsc{cat}) &= \text{kæt} \\ \texttt{UR}(\textsc{pl}) &= \text{z} \\ \texttt{UR}(\textsc{cat}\text{-}\textsc{pl}) &= \text{kætz} \\ \ldots & \quad \ldots \end{aligned}$$

  - phonology function $\texttt{PH} : \Sigma^* \to \Sigma^*$

$$\begin{aligned} \texttt{PH}(\text{kæt}) &= \text{kæt} \\ \texttt{PH}(\text{dɔgz}) &= \text{dɔgz} \\ \texttt{PH}(\text{kætz}) &= \text{kæts} \\ \texttt{PH}(\text{bnɪkz}) &= \text{bnɪks} \\ \ldots & \quad \ldots \end{aligned}$$

The learning problem

- Learning data is a finite sample of $\mathtt{PH} \circ \mathtt{UR}$

$w \in M^*$	$\mathtt{PH}(\mathtt{UR}(w))$
CAT-PL	kæts
CUFF-PL	kʌfs
DEATH-PL	dɛθs
GIRL-PL	gərlz
CHAIR-PL	tʃerz
BOY-PL	bɔɪz

The learning problem

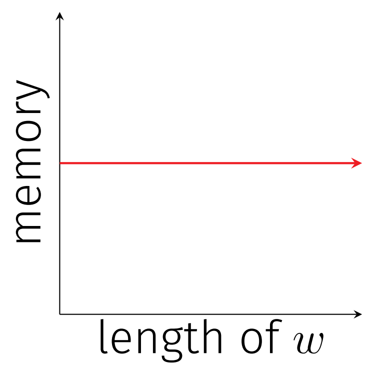
- **Problem:** identify `UR` and `PH` from a finite sample of $\texttt{PH} \circ \texttt{UR}$
- **Questions:** What is the nature of ...
  - `UR`
  - `PH`
  - $\texttt{PH} \circ \texttt{UR}$
  - the data sample

  ...such that learning is possible?

Subsequentiality and phonology

Subsequentiality and phonology

- Johnson (1972); Kaplan and Kay (1994): phonological maps are **regular**

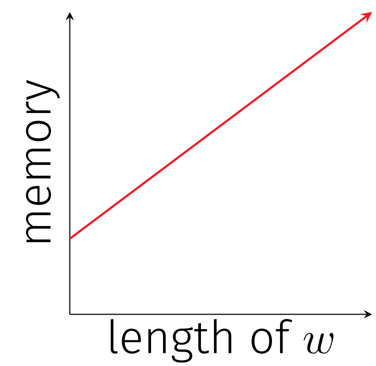


regular

E.g., directional harmony

$/\text{kaki-kæ}/ \mapsto [\text{kaki-ka}]$

$/\text{kiki-kæ}/ \mapsto [\text{kiki-kæ}]$



non-regular

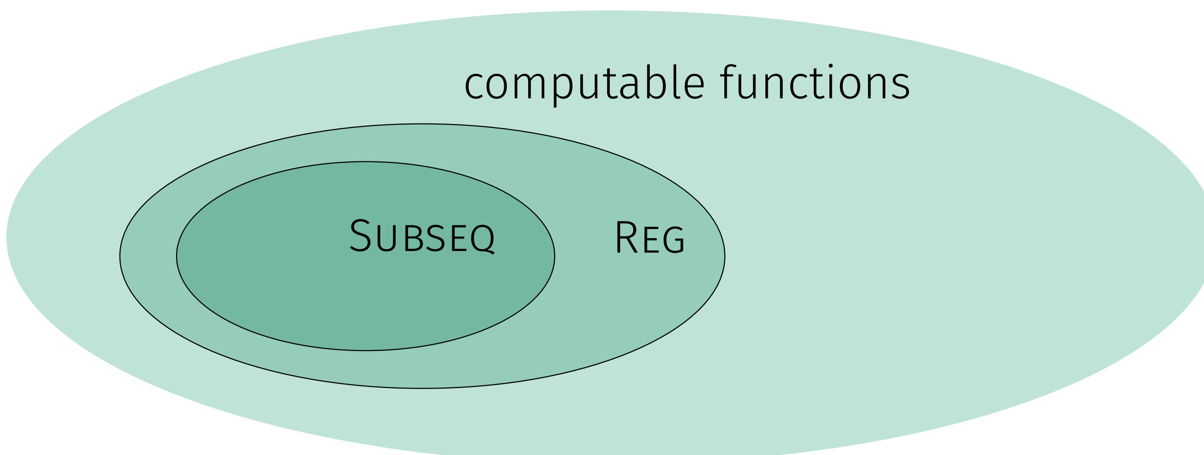
E.g., “majority rules” (Baković, 2000)

$/\text{kaka-kæ}/ \mapsto [\text{kaka-ka}]$

$/\text{kæka-kæ}/ \mapsto [\text{kækæ-kæ}]$

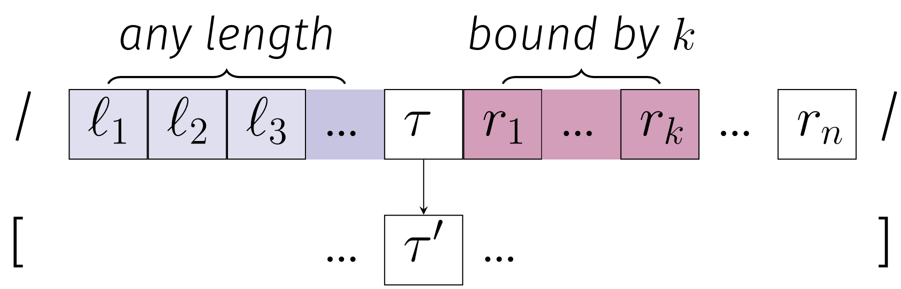
Subsequentiality and phonology

- Mohri (1997); Heinz and Lai (2013): Phonological maps are **subsequential**;
  - they are regular, **and**
  - they are **deterministic**



Subsequentiality and phonology

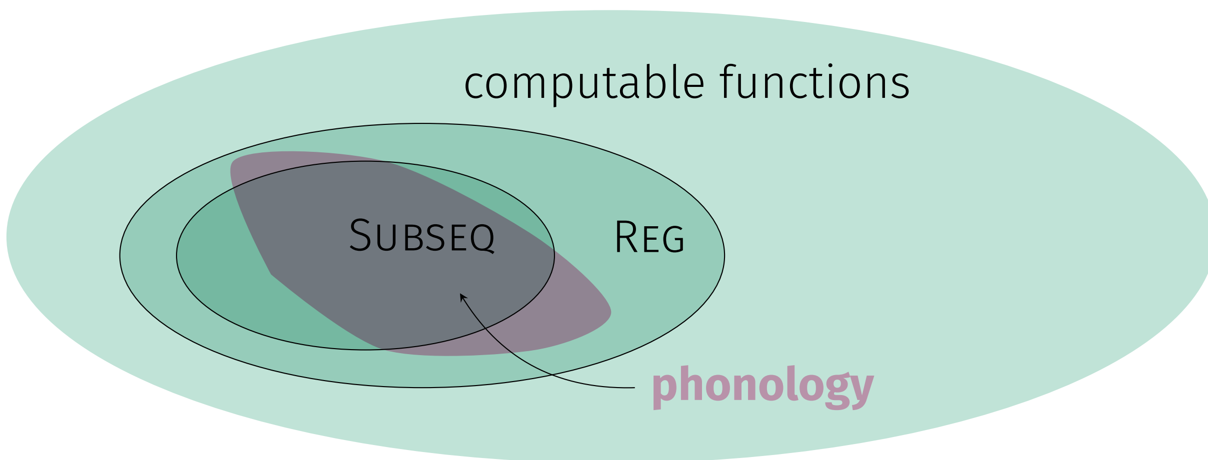
- **Subsequential:** output can be determined deterministically in one direction



- (Determinisic $\neq$ no optionality; Heinz in progress)

Subsequentiality and phonology

- The subsequentiality of phonology is empirically well-supported (Chandlee and Heinz, 2012; Heinz and Lai, 2013; Payne, 2017; Luo, 2017; Chandlee and Heinz, 2018)



- Though c.f. Jardine (2016); McCollum et al. (2017)

Subsequentiality and phonology

- The **longest common prefix (lcp)** is the longest initial sequence shared by a set of strings

$$\texttt{lcp}(\{aab, aaba, aac\}) =$$

Subsequentiality and phonology

- The **longest common prefix (lcp)** is the longest initial sequence shared by a set of strings

$$\texttt{lcp}(\{\underline{aa}b, \underline{aa}ba, \underline{aa}c\}) = aa$$

Subsequentiality and phonology

- The **longest common prefix (lcp)** is the longest initial sequence shared by a set of strings

$$\texttt{lcp}(\{\underline{aa}b, \underline{aa}ba, \underline{aa}c\}) = aa$$
$$\texttt{lcp}(\{bac, abc\}) =$$

Subsequentiality and phonology

- The **longest common prefix (lcp)** is the longest initial sequence shared by a set of strings

$$\texttt{lcp}(\{\underline{aa}b, \underline{aa}ba, \underline{aa}c\}) = aa$$
$$\texttt{lcp}(\{bac, abc\}) = \lambda$$

Subsequentiality and phonology

- Take a function f

$$\left.\begin{array}{lcl} f(tat) & = & tat \\ f(tatta) & = & tatta \\ f(tadta) & = & tadda \\ f(dtd) & = & ddd \\ f(tadtta) & = & taddta \\ & \dots & \end{array}\right\} t \;\rightarrow\; d \;/\; d \_\_ \text{ (simul.)}$$

Subsequentiality and phonology

- For f we define

$$f^p(w) \stackrel{\text{def}}{=} \texttt{lcp}(f(w\Sigma^*))$$

Subsequentiality and phonology

- For f we define

$$f^p(w) \stackrel{\text{def}}{=} \texttt{lcp}(f(w\Sigma^*))$$

- Let $\Sigma = \{a, d, t\}$

$$\begin{aligned} f^p(tad) = \texttt{lcp}(\{\ & f(tad), \\ & f(tada), \\ & f(tadd), \\ & f(tadt), \\ & f(tadaa), \\ & \ldots \qquad\quad \}) \end{aligned}$$

Subsequentiality and phonology

- For f we define

$$f^p(w) \overset{\text{def}}{=} \texttt{lcp}(f(w\Sigma^*))$$

- Let $\Sigma = \{a, d, t\}$

$$\begin{aligned} f^p(tad) = \texttt{lcp}(\{\ & f(tad) = tad, \\ & f(tada) = tada, \\ & f(tadd) = tadd, \\ & f(tadt) = tadd, \\ & f(tadaa) = tadaa, \\ & \ldots \qquad\qquad\qquad\qquad \}) \end{aligned}$$

Subsequentiality and phonology

- For f we define

$$f^p(w) \stackrel{\text{def}}{=} \texttt{lcp}(f(w\Sigma^*))$$

- Let $\Sigma = \{a, d, t\}$

$$\begin{aligned} f^p(tad) = \texttt{lcp}(\{\ & f(tad) = \underline{tad}, \\ & f(tada) = \underline{tad}a, \\ & f(tadd) = \underline{tad}d, \\ & f(tadt) = \underline{tad}d, \\ & f(tadaa) = \underline{tad}aa, \\ & \ldots \qquad\qquad\qquad \}) \end{aligned}$$

Subsequentiality and phonology

- For f we define

$$f^p(w) \stackrel{\text{def}}{=} \texttt{lcp}(f(w\Sigma^*))$$

- Let $\Sigma = \{a, d, t\}$

$$f^p(tad) = tad$$

Subsequentiality and phonology

- For f we define

$$f^p(w) \stackrel{\text{def}}{=} \texttt{lcp}(f(w\Sigma^*))$$

- Let $\Sigma = \{a, d, t\}$

$$\begin{aligned} f^p(tad) &= tad \\ f^p(tadt) &= \texttt{lcp}(\{\ f(tadt) = \underline{tadd}, \\ &\qquad f(tadta) = \underline{tadd}a, \\ &\qquad f(tadtd) = \underline{tadd}d, \\ &\qquad f(tadtt) = \underline{tadd}t, \\ &\qquad f(tadtaa) = \underline{tadd}aa, \\ &\qquad \ldots \qquad \}) \end{aligned}$$

Subsequentiality and phonology

- For f we define

$$f^p(w) \stackrel{\text{def}}{=} \texttt{lcp}(f(w\Sigma^*))$$

- Let $\Sigma = \{a, d, t\}$

$$f^p(tad) = tad$$
$$f^p(tadt) = tadd$$

Subsequentiality and phonology

- $f^p(w)$ is the **contribution** of w to any $f(wv)$

$$f(wv) = \overbrace{\boxed{a}\boxed{b}\boxed{a}\ \ldots\ \boxed{a}}^{f^p(w)}\boxed{b}\boxed{a}\ \ldots\ \boxed{b}$$

- f^p grows proportionally iff f subsequential...

$$\begin{aligned} m^p(\text{kækak}) &= \texttt{lcp}(\{\underline{\text{k}}\text{ækækæ},\ \underline{\text{k}}\text{akaka},\ldots\})) \\ &= \text{k} \end{aligned}$$

Subsequentiality and phonology

- For $w \in \Sigma^*$, the **environment function** is

$$f_w(v) \overset{\text{def}}{=} f^p(w)^{-1} f(wv)$$

$$f(wv) = \overbrace{\boxed{a}\,\boxed{b}\,\boxed{a}\,\ldots\,\boxed{a}}^{f^p(w)}\,\overbrace{\boxed{b}\,\boxed{a}\,\ldots\,\boxed{b}}^{f_w(v)}$$

Subsequentiality and phonology

- For $w \in \Sigma^*$, the **environment function** is

$$f_w(v) \stackrel{\text{def}}{=} f^p(w)^{-1} f(wv)$$

$$f(wv) = \overbrace{\boxed{a}\boxed{b}\boxed{a}\;...\;\boxed{a}}^{f^p(w)}\overbrace{\boxed{b}\boxed{a}\;...\;\boxed{b}}^{f_w(v)}$$

- Ex.

$$\begin{aligned} f_{tad}(ta) &= f^p(tad)^{-1} f(tadta) \\ &= (tad)^{-1}\underline{tad}da \\ &= da \end{aligned}$$

Subsequentiality and phonology

- For $w \in \Sigma^*$, the **environment function** is

$$f_w(v) \overset{\text{def}}{=} f^p(w)^{-1} f(wv)$$

$$f(wv) = \overbrace{\boxed{a}\boxed{b}\boxed{a}\boxed{\ldots}\boxed{a}}^{f^p(w)}\overbrace{\boxed{b}\boxed{a}\boxed{\ldots}\boxed{b}}^{f_w(v)}$$

- Ex.

$$\begin{aligned} f_{tad}(ta) &= f^p(tad)^{-1} f(tadta) \\ &= (tad)^{-1} tadda \\ &= da \\ f_{tat}(ta) &= f^p(tat)^{-1} f(tatta) \\ &= (tat)^{-1} \underline{tat}ta \\ &= ta \end{aligned}$$

Subsequentiality and phonology

- f is subsequential **iff it has finite environment functions**

w	$f_w(ta)$
a	ta
t	ta
aa	ta
at	ta
dt	ta
tt	ta
aaa	ta
$\ldots$	$\ldots$

w	$f_w(ta)$
d	da
ad	da
dd	da
td	da
aad	da
add	da
$\ldots$	$\ldots$

Subsequentiality and phonology

- f is subsequential **iff it has finite environment functions**

w	$f_w(ta)$
a	ta
t	ta
aa	ta
...	...

w	$f_w(ta)$
d	da
ad	da
dd	da
...	...

$$f_a = f_t = \ldots = f_{aaa} = \ldots = f_{tatat} = \ldots = f_{a/t}$$
$$f_d = f_{ad} = \ldots = f_{aad} = \ldots = f_{tatad} = \ldots = f_d$$

Subsequentiality and phonology

- Environment functions correspond to states in a **deterministic finite-state transducer** (Mohri, 1997)

- There are procedures for determining environment functions from positive data (Oncina et al., 1993; Chandlee et al., 2014; Jardine et al., 2014)

- If f's environment functions represent $k-1$ suffixes, f is **input strictly k-local (ISL$_k$)** (Chandlee, 2014; Chandlee and Heinz, 2018)

$$f \rightarrow \{f_{a/t}, f_d\}$$

Towards a solution

Towards a solution

- M: set of morphemes; Σ: finite set of segments
- lexicon function $\texttt{UR} : M^* \to \Sigma^*$
  phonology function $\texttt{PH} : \Sigma^* \to \Sigma^*$

- **Problem:** identify $\texttt{UR}$ and $\texttt{PH}$ from a finite sample of $\texttt{PH} \circ \texttt{UR}$

Towards a solution

- Example:

UR		PH
$r_1 \mapsto \text{tat}$	$s_1 \mapsto \text{ta}$	
$r_2 \mapsto \text{tad}$	$s_2 \mapsto \text{da}$	$\text{t} \rightarrow \text{d} \;/\; \text{d}\ \_\_$
$r_3 \mapsto \text{a}$	$s_3 \mapsto \text{a}$	

Sample of $\texttt{PH} \circ \texttt{UR}$

w	$\texttt{PH}(\texttt{UR}(w))$	w	$\texttt{PH}(\texttt{UR}(w))$	w	$\texttt{PH}(\texttt{UR}(w))$
$r_1 s_1$	tatta	$r_2 s_1$	tadda	$r_3 s_1$	ata
$r_1 s_2$	tatda	$r_2 s_2$	tadda	$r_3 s_2$	ada
$r_1 s_3$	tata	$r_2 s_3$	tada	$r_3 s_3$	aa

Towards a solution

- M: set of morphemes; Σ: finite set of segments
- lexicon function $\texttt{UR} : M^* \to \Sigma^*$
  phonology function $\texttt{PH} : \Sigma^* \to \Sigma^*$

- **Problem:** identify $\texttt{UR}$ and $\texttt{PH}$ from a finite sample of $\texttt{PH} \circ \texttt{UR}$
- What is the nature of …
  - $\texttt{UR}$
  - $\texttt{PH}$
  - $\texttt{PH} \circ \texttt{UR}$ …?

Towards a solution

- M: set of morphemes; Σ: finite set of segments
- lexicon function $\mathtt{UR} : M^* \to \Sigma^*$
  phonology function $\mathtt{PH} : \Sigma^* \to \Sigma^*$

- **Problem:** identify $\mathtt{UR}$ and $\mathtt{PH}$ from a finite sample of $\mathtt{PH} \circ \mathtt{UR}$
- The nature of ...
  - $\mathtt{UR}$
  - $\mathtt{PH}$
  - $\mathtt{PH} \circ \mathtt{UR}$ ...is **subsequential**

Towards a solution

Assumptions:

- $\texttt{UR}$ has **one environment function** ($= \texttt{UR}$)

$$\begin{aligned} \texttt{UR}_w(\textsf{CAT}) &= \text{kæt} \quad \text{for any } w \in M^*; \\ \texttt{UR}_w(\textsf{PL}) &= \text{z} \quad\ \ \ \text{for any } w \in M^*; \text{ etc.} \end{aligned}$$

Towards a solution

Assumptions:

- $\texttt{UR}$ has **one environment function** ($= \texttt{UR}$)

$$\begin{array}{lll} \texttt{UR}_w(\textsf{CAT}) = & \text{kæt} & \text{for any } w \in M^*; \\ \texttt{UR}_w(\textsf{PL}) = & \text{z} & \text{for any } w \in M^*; \text{ etc.} \end{array}$$

- $\texttt{PH}$ is ISL$_2$
- That is, its environment functions are of the form $\texttt{PH}_\sigma$, $\sigma \in \Sigma$

$$\Sigma = \{t, a, d\} \rightarrow \text{ possible env. functions are } \texttt{PH}_a, \texttt{PH}_t, \texttt{PH}_d$$

Towards a solution

Strategy:

- Two hypotheses `UR`′ and `PH`′
- We modify `UR`′ so it has one environment function
- We make the *opposite* change in `PH`′ to remain consistent with input data

The procedure

The procedure

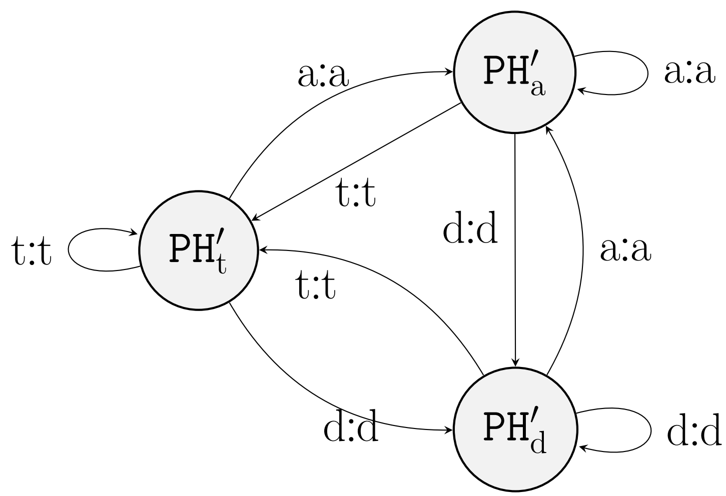
- Running example $D \subset \texttt{PH} \circ \texttt{UR}$

Sample of $\texttt{PH} \circ \texttt{UR}$					
w	$\texttt{PH}(\texttt{UR}(w))$	w	$\texttt{PH}(\texttt{UR}(w))$	w	$\texttt{PH}(\texttt{UR}(w))$
r_1s_1	tatta	r_2s_1	tadda	r_3s_1	ata
r_1s_2	tatda	r_2s_2	tadda	r_3s_2	ada
r_1s_3	tata	r_2s_3	tada	r_3s_3	aa

The procedure

- Initialize $\texttt{PH}'$ to the identity function

$$\texttt{PH}'(\text{tad}) = \text{tad},\ \texttt{PH}'(\text{tatta}) = \text{tatta},\ \texttt{PH}'(\text{tadta}) = \text{tadta}, \text{ etc.}$$



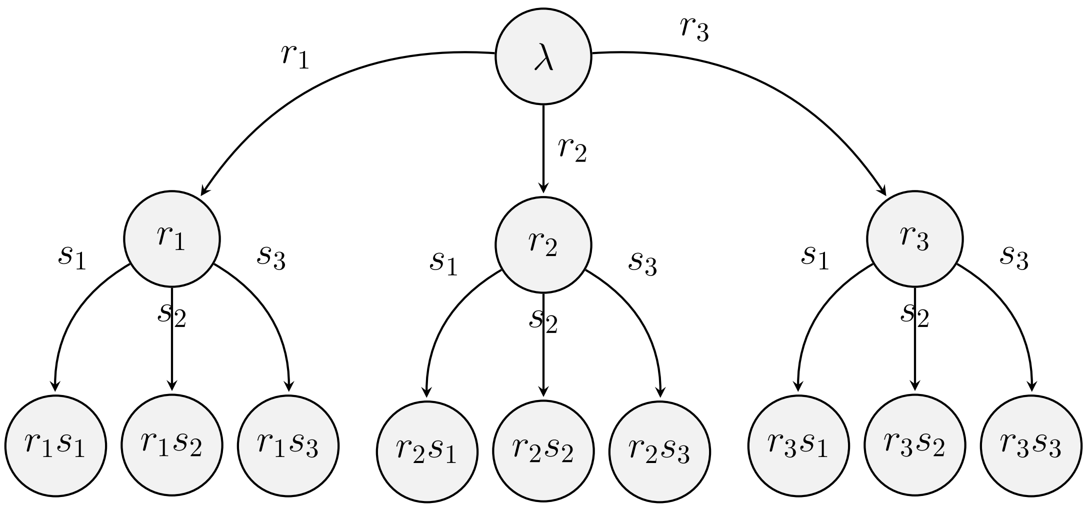
The procedure

- Initialize `UR`′ to a **prefix tree transducer** representing D

$r_1 s_1$	tatta	$r_2 s_1$	tadda	$r_3 s_1$	ata
$r_1 s_2$	tatda	$r_2 s_2$	tadda	$r_3 s_2$	ada
$r_1 s_3$	tata	$r_2 s_3$	tada	$r_3 s_3$	aa

The procedure

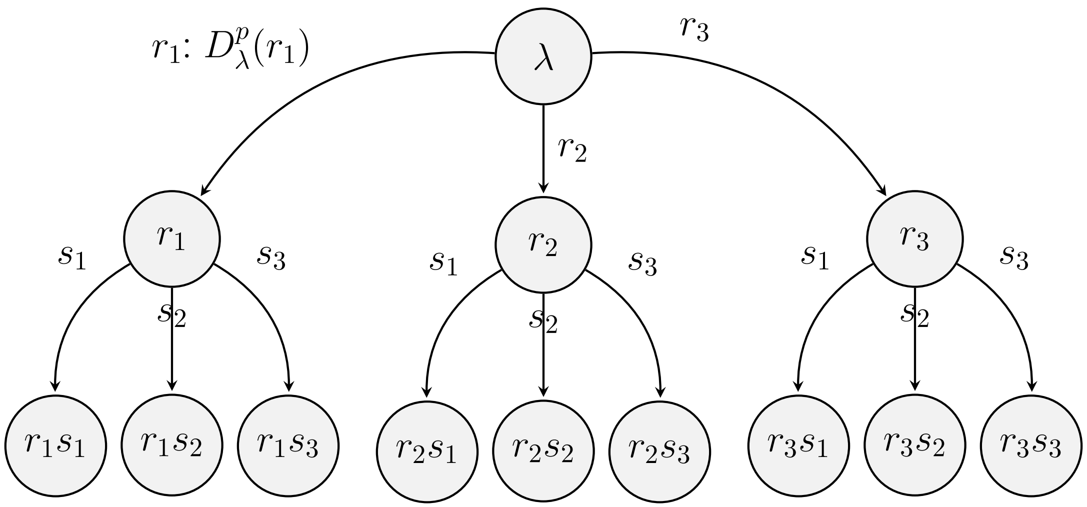
$r_1 s_1$	tatta	$r_2 s_1$	tadda	$r_3 s_1$	ata
$r_1 s_2$	tatda	$r_2 s_2$	tadda	$r_3 s_2$	ada
$r_1 s_3$	tata	$r_2 s_3$	tada	$r_3 s_3$	aa



The procedure

$m : D^p_w(m)$

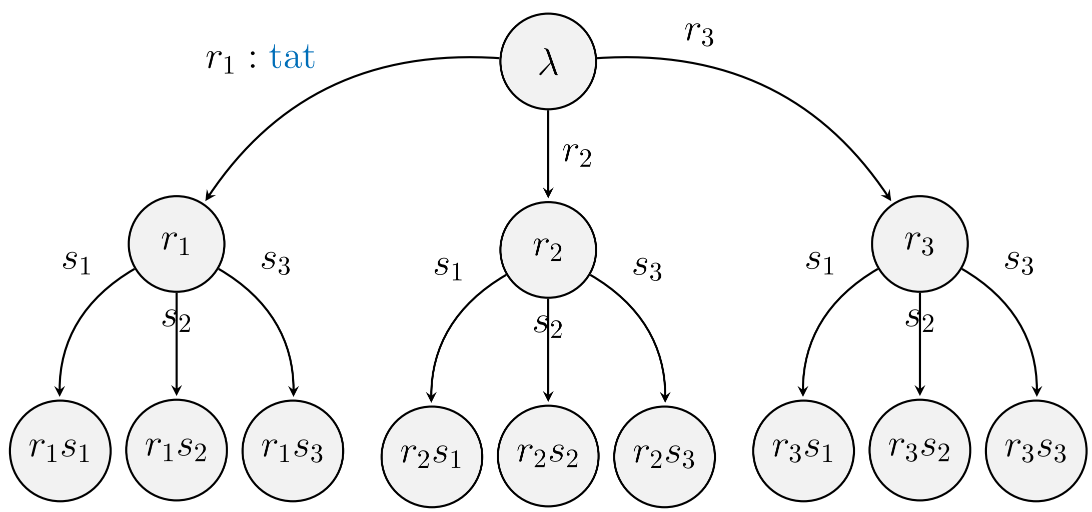
$r_1 s_1$	tatta	$r_2 s_1$	tadda	$r_3 s_1$	ata
$r_1 s_2$	tatda	$r_2 s_2$	tadda	$r_3 s_2$	ada
$r_1 s_3$	tata	$r_2 s_3$	tada	$r_3 s_3$	aa



The procedure

$m : D^p_w(m)$

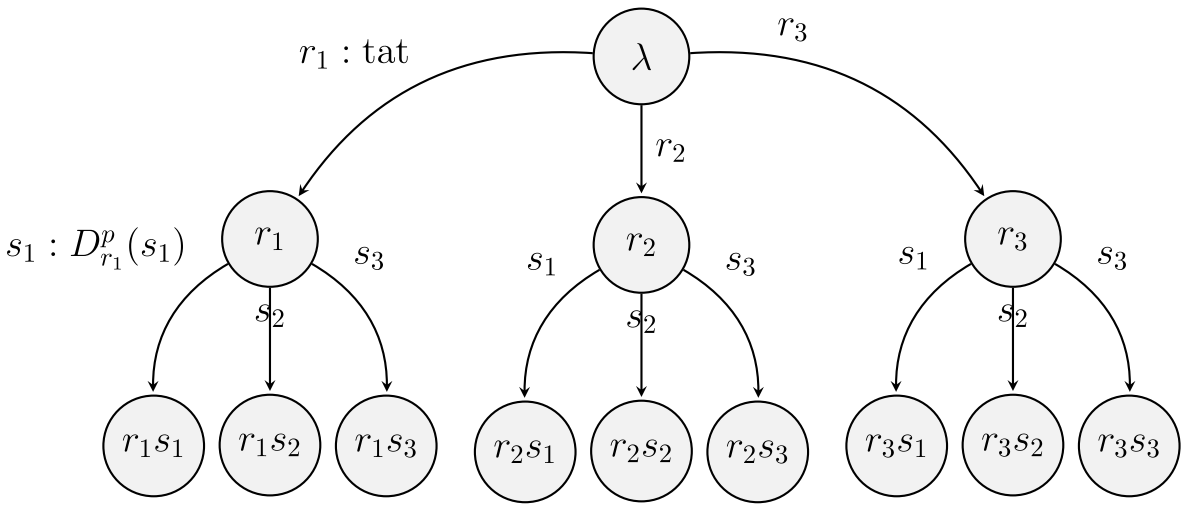
$r_1 s_1$	tatta	$r_2 s_1$	tadda	$r_3 s_1$	ata
$r_1 s_2$	tatda	$r_2 s_2$	tadda	$r_3 s_2$	ada
$r_1 s_3$	tata	$r_2 s_3$	tada	$r_3 s_3$	aa



The procedure

$m : D_w^p(m)$

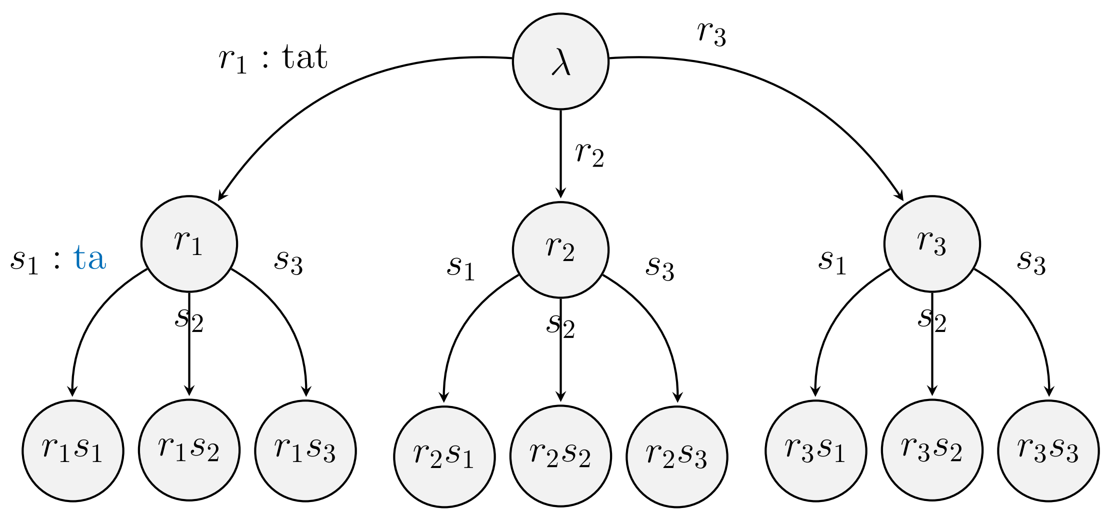
$r_1 s_1$	tatta	$r_2 s_1$	tadda	$r_3 s_1$	ata
$r_1 s_2$	tatda	$r_2 s_2$	tadda	$r_3 s_2$	ada
$r_1 s_3$	tata	$r_2 s_3$	tada	$r_3 s_3$	aa



The procedure

$m : D_w^p(m)$

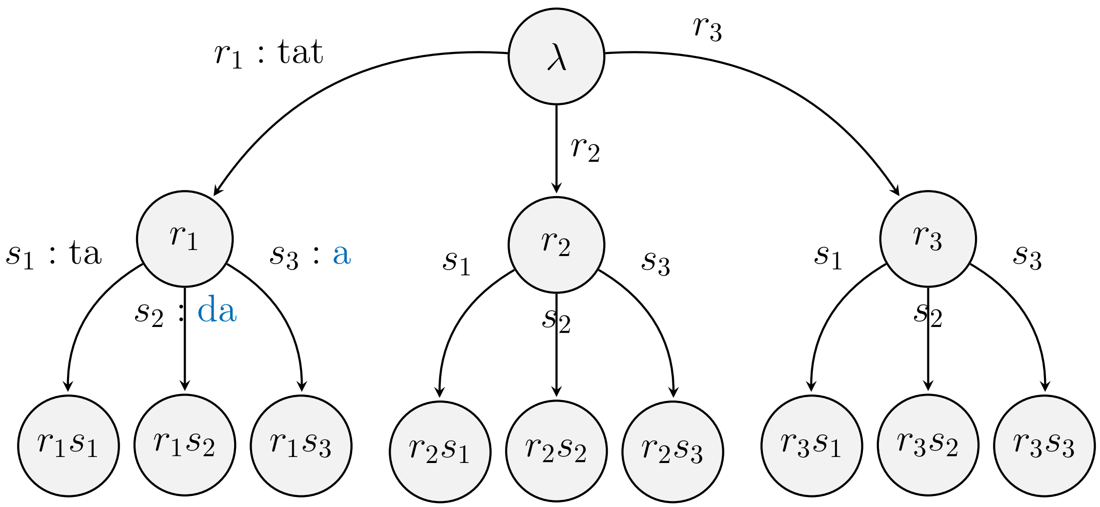
$r_1 s_1$	tat<u>ta</u>	$r_2 s_1$	tadda	$r_3 s_1$	ata
$r_1 s_2$	tatda	$r_2 s_2$	tadda	$r_3 s_2$	ada
$r_1 s_3$	tata	$r_2 s_3$	tada	$r_3 s_3$	aa



The procedure

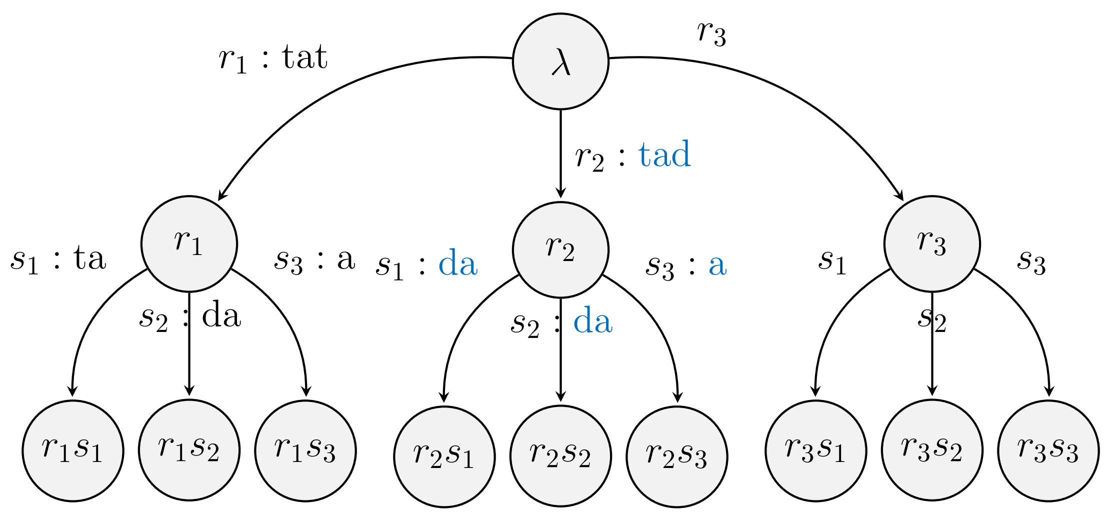
$m : D_w^p(m)$

$r_1 s_1$	tatta	$r_2 s_1$	tadda	$r_3 s_1$	ata
$r_1 s_2$	tatda	$r_2 s_2$	tadda	$r_3 s_2$	ada
$r_1 s_3$	tata	$r_2 s_3$	tada	$r_3 s_3$	aa



The procedure

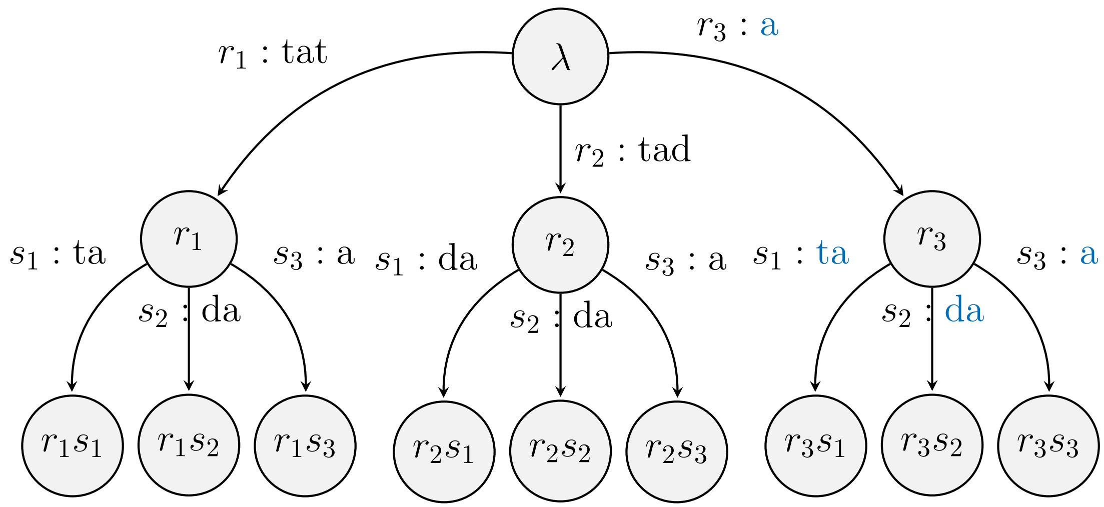
$m : D_w^p(m)$						
	$r_1 s_1$	tatta	$r_2 s_1$	tad da	$r_3 s_1$	ata
	$r_1 s_2$	tatda	$r_2 s_2$	tad da	$r_3 s_2$	ada
	$r_1 s_3$	tata	$r_2 s_3$	tad a	$r_3 s_3$	aa



The procedure

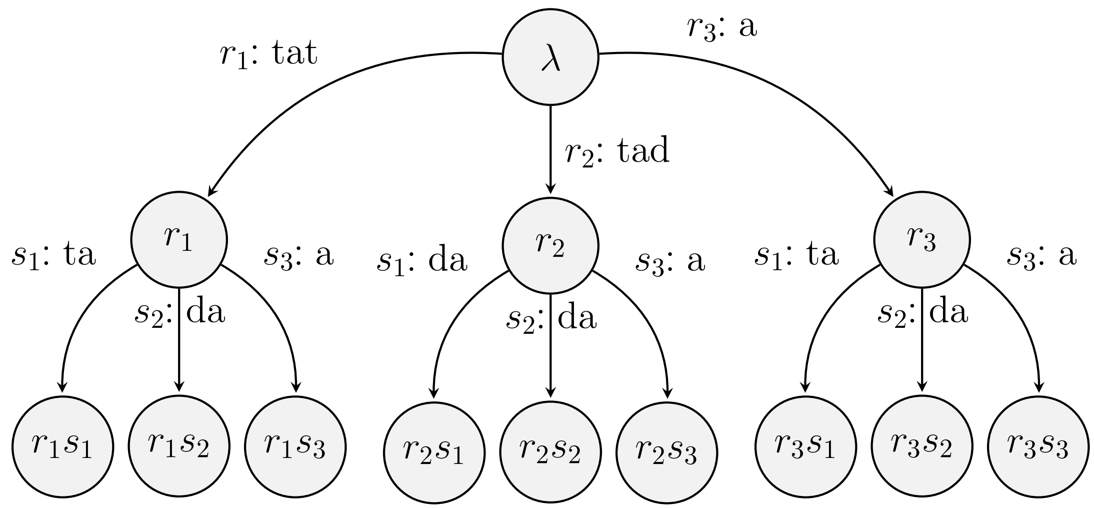
$m : D^p_w(m)$

$r_1 s_1$	tatta	$r_2 s_1$	tadda	$r_3 s_1$	a ta
$r_1 s_2$	tatda	$r_2 s_2$	tadda	$r_3 s_2$	a da
$r_1 s_3$	tata	$r_2 s_3$	tada	$r_3 s_3$	a a



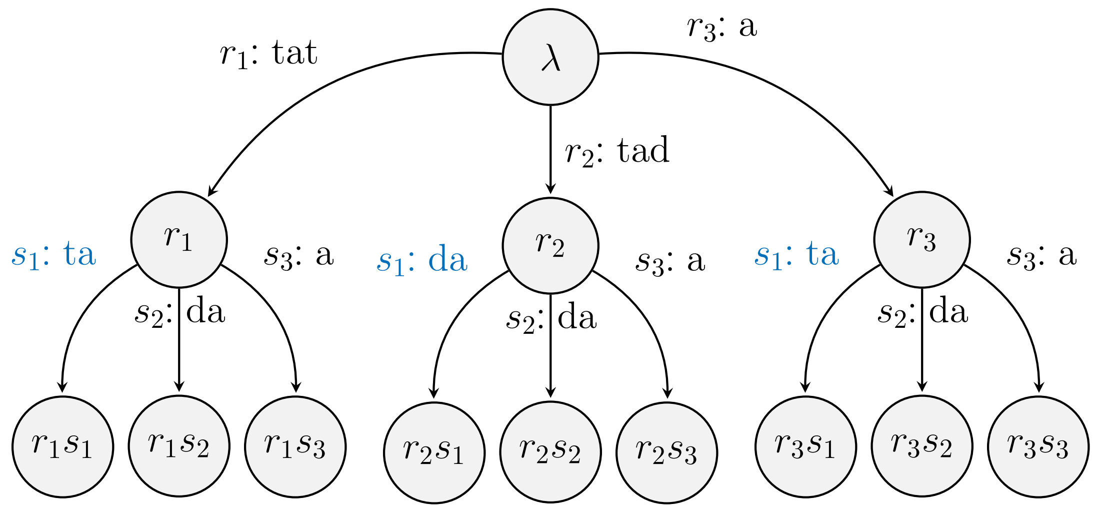
The procedure

r_1s_1	tatta	r_2s_1	tadda	r_3s_1	ata
r_1s_2	tatda	r_2s_2	tadda	r_3s_2	ada
r_1s_3	tata	r_2s_3	tada	r_3s_3	aa

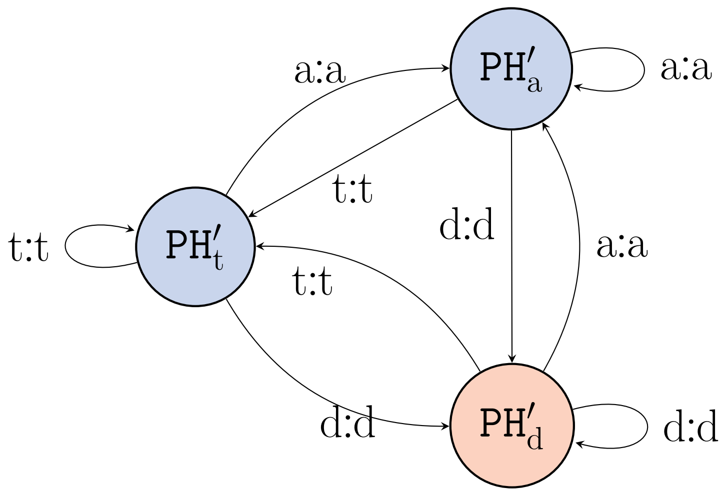


The procedure

r_1s_1	tatta	r_2s_1	tadda	r_3s_1	ata
r_1s_2	tatda	r_2s_2	tadda	r_3s_2	ada
r_1s_3	tata	r_2s_3	tada	r_3s_3	aa

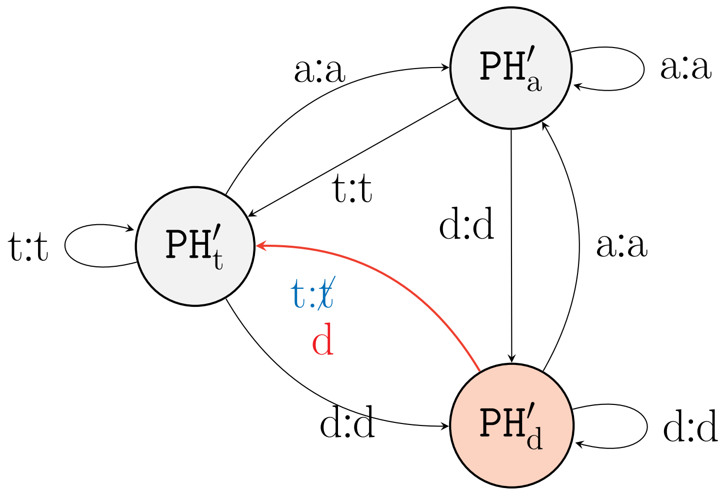


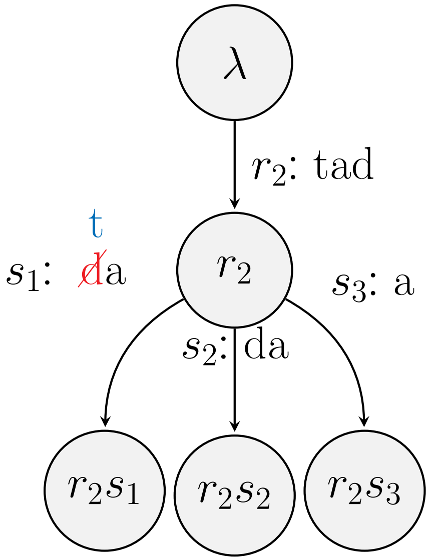
The procedure



ws_1	env.	s_1
r_1s_1	tat	ta
r_3s_1	a	ta
r_2s_1	tad	da

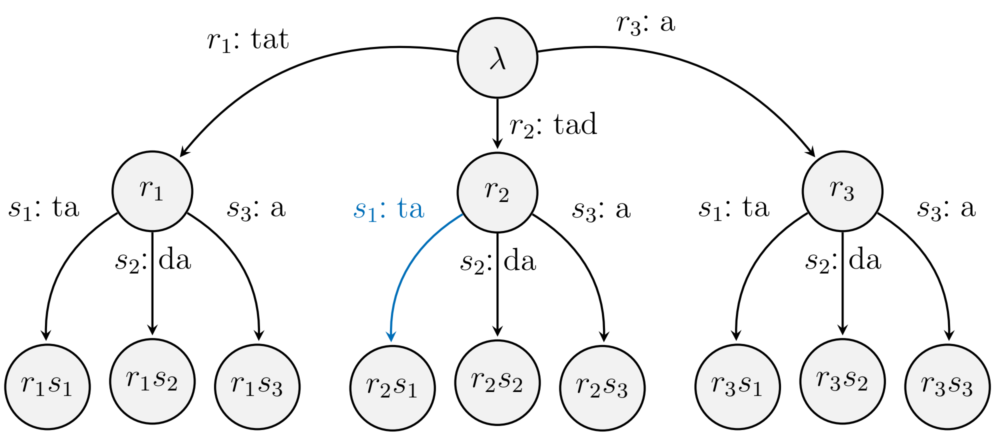
The procedure



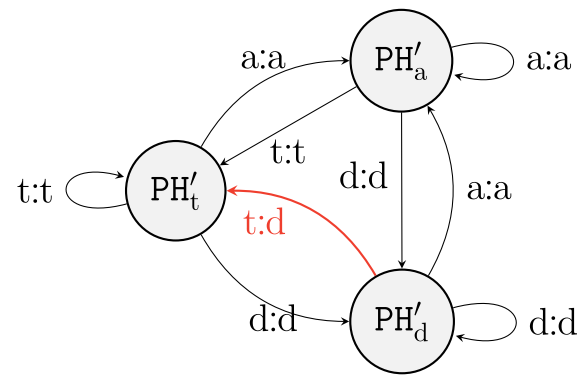


The procedure

UR′



PH′



$$\mathtt{UR}'(r_1 s_1) = \text{tatta} \qquad \mathtt{PH}'(\text{tatta}) = \text{tatta}$$
$$\mathtt{UR}'(r_2 s_1) = \text{tadta} \qquad \mathtt{PH}'(\text{tadta}) = \text{tadda}$$

The procedure – summary

$\mathtt{UR}$		$\mathtt{PH}$	Sample of $\mathtt{PH} \circ \mathtt{UR}$					
$r_1 \mapsto$ tat	$s_1 \mapsto$ ta		$r_1 s_1$	tatta	$r_2 s_1$	tadda	$r_3 s_1$	ata
$r_2 \mapsto$ tad	$s_2 \mapsto$ da	t $\rightarrow$ d / d __	$r_1 s_2$	tatda	$r_2 s_2$	tadda	$r_3 s_2$	ada
$r_3 \mapsto$ a	$s_3 \mapsto$ a		$r_1 s_3$	tata	$r_2 s_3$	tada	$r_3 s_3$	aa

- Correct $\mathtt{UR}'$ and $\mathtt{PH}'$ from a sample of $\mathtt{PH} \circ \mathtt{UR}$
- This dependent on the
  - the **subsequentiality** of $\mathtt{UR}$ and $\mathtt{PH}$
  - that $\mathtt{UR}$ maps one morpheme to one UR
  - that $\mathtt{PH}$ is $\mathbf{ISL_2}$

The procedure – summary

UR		PH	Sample of $\texttt{PH} \circ \texttt{UR}$					
$r_1 \mapsto$ tat	$s_1 \mapsto$ ta		r_1s_1	tatta	r_2s_1	tadda	r_3s_1	ata
$r_2 \mapsto$ tad	$s_2 \mapsto$ da	t $\rightarrow$ d / d __	r_1s_2	tatda	r_2s_2	tadda	r_3s_2	ada
$r_3 \mapsto$ a	$s_3 \mapsto$ a		r_1s_3	tata	r_2s_3	tada	r_3s_3	aa

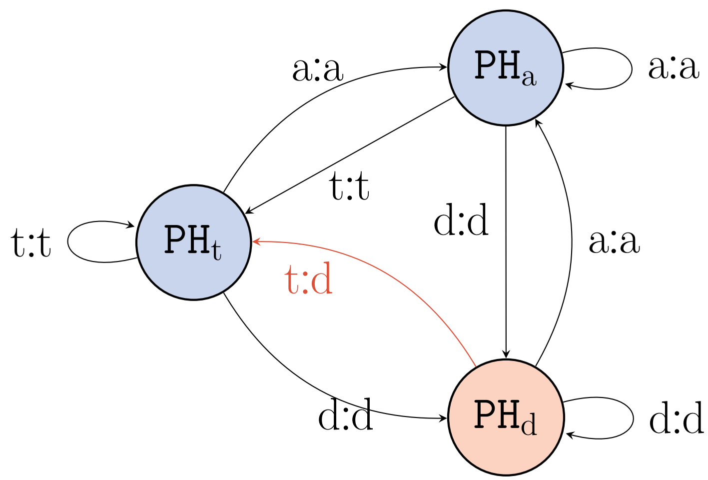
- A learner based on this procedure can learn ISL_2:
  - progressive assim./dissim.
  - regressive assim./dissim.
  - deletion
  - epenthesis
- This includes opacity (self-counterbleeding)
- So far we are limited to **single processes**

The procedure – summary

UR		PH	Sample of $\mathtt{PH} \circ \mathtt{UR}$					
$r_1 \mapsto$ tat	$s_1 \mapsto$ ta		$r_1 s_1$	tatta	$r_2 s_1$	tadda	$r_3 s_1$	ata
$r_2 \mapsto$ tad	$s_2 \mapsto$ da	t $\rightarrow$ d / d __	$r_1 s_2$	tatda	$r_2 s_2$	tadda	$r_3 s_2$	ada
$r_3 \mapsto$ a	$s_3 \mapsto$ a		$r_1 s_3$	tata	$r_2 s_3$	tada	$r_3 s_3$	aa

- It requires that the UR is **recoverable** from $\mathtt{PH} \circ \mathtt{UR}$
- What are the constraints on PH? On $\mathtt{PH} \circ \mathtt{UR}$?

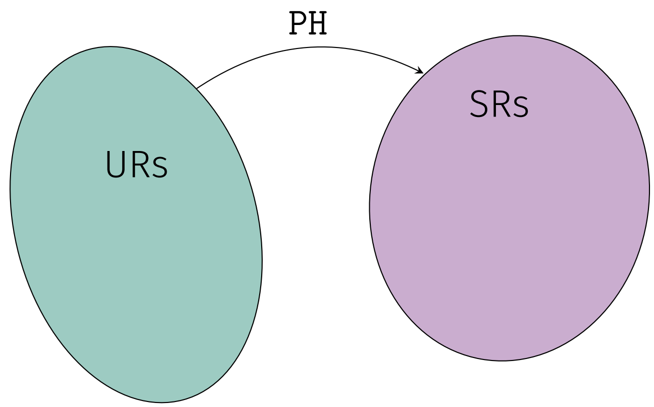
The procedure – summary



- Environment functions $\texttt{PH}_w$ must split into change and elsewhere functions
- Any change $\texttt{PH}$ makes must be seen at morpheme boundaries
- Formalizing these constraints on $\texttt{PH} \circ \texttt{UR}$ is work in progress

Discussion

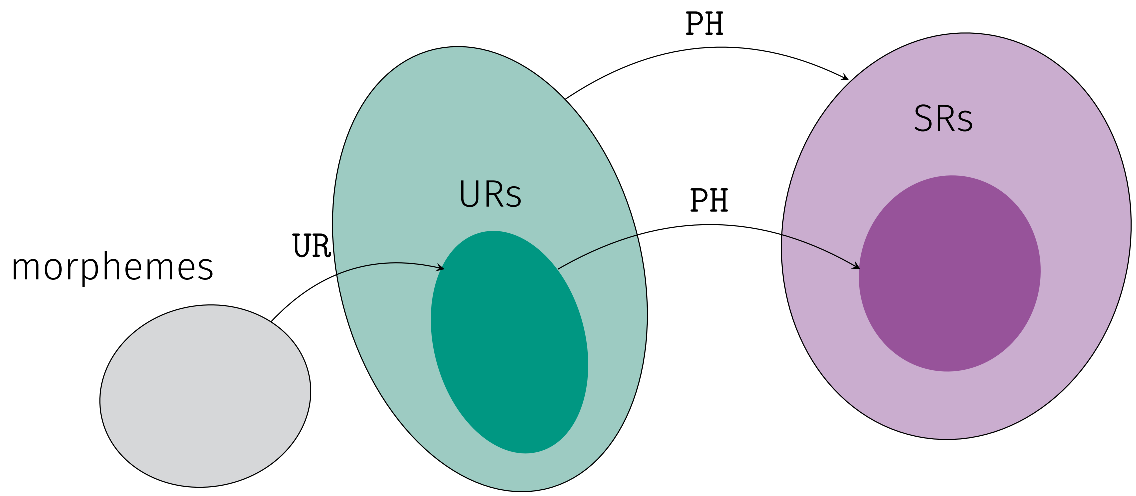
What is the nature of phonology?



What is the nature of:

- maps from URs to SRs?
- relation between SRs and URs?

What is the nature of phonology?



Assuming

- **subsequential** maps from URs to SRs, and
- a (relatively) **concrete** relation between SRs and URs

...allows for a procedure for learning URs and a grammar

Future work

- Formalizing “relatively concrete”

- Extending to cases in which `PH` is...
  - ISL_2
  - ISL_k for some k
  - in any subsequential class with a shared structure (Jardine et al., 2014)
  - **output**-strictly local (Chandlee et al., 2015)

- Extending to...
  - featural learning (Heinz and Koirala, 2010; Chandlee et al., 2019)
  - optional/gradient processes (Shibata and Heinz, 2019; Beros and de la Higuera, 2016)

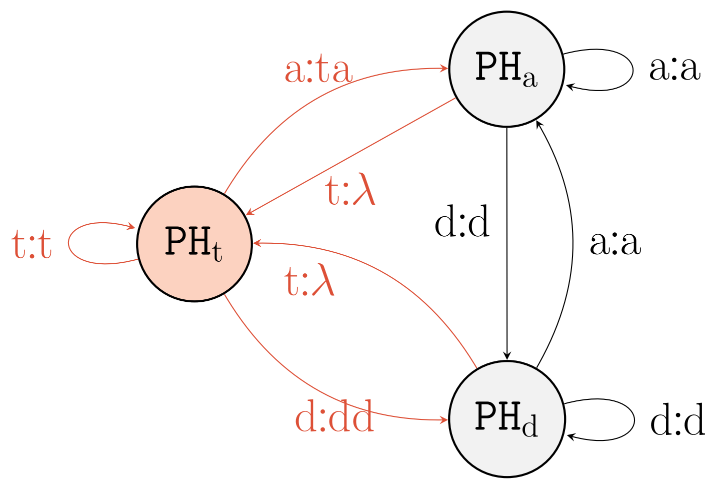
Acknowledgements

Thank you for having me!

…and many thanks to **Wenyue Hua** and **Huteng Dai**, attendees of the Rutgers/SBU/Haverford/Delaware subregular phonology workshop, **the Rutgers MathLing group**, an audience at NECPhon, and in particular **Jeff Heinz**, **Charles Reiss**, **Bruce Tesar**, **Adam McCollum**, and **Colin Wilson** for their insightful comments.

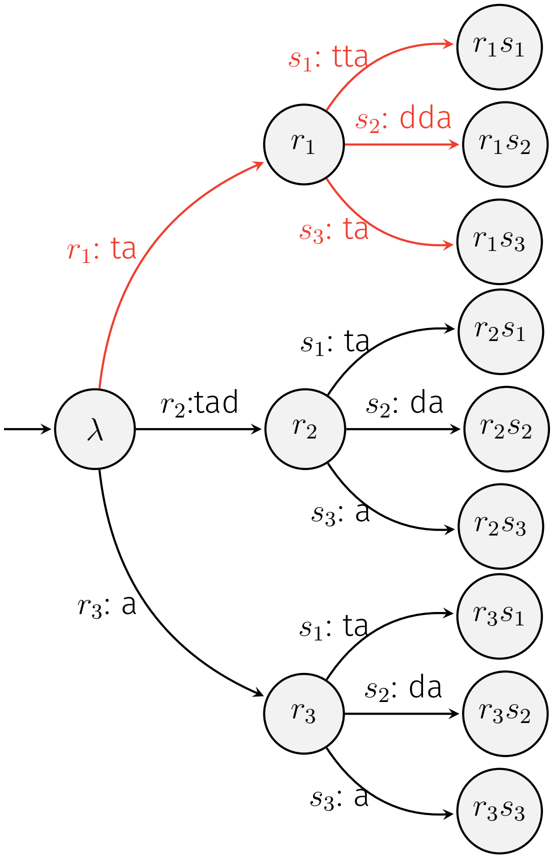
Appendix: Regressive assimilation

w	$\mathtt{UR}(w)$	$\mathtt{PH} \circ \mathtt{UR}(w)$
$r_1 s_1$	tatta	tatta
$r_1 s_2$	tatda	tadda
$r_1 s_3$	tata	tata
$r_2 s_1$	tadda	tadda
$r_2 s_2$	tadta	tadta
$r_2 s_3$	tada	tada
$r_3 s_1$	ata	ata
$r_3 s_2$	ada	ada
$r_3 s_3$	aa	aa

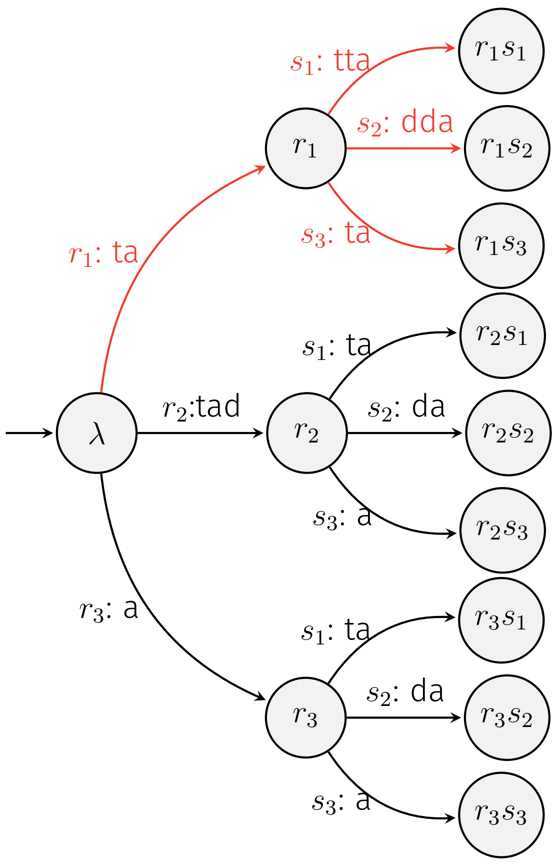


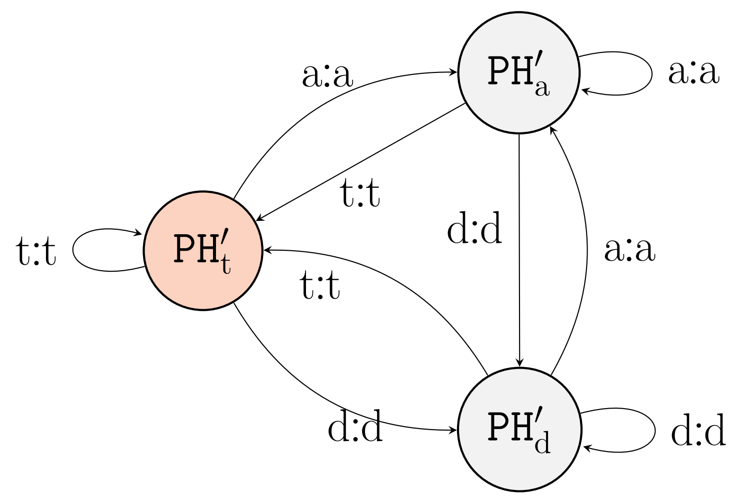
Appendix: Regressive assimilation

w	$\mathtt{UR}(w)$	$\mathtt{PH} \circ \mathtt{UR}(w)$
$r_1 s_1$	tatta	tatta
$r_1 s_2$	tatda	tadda
$r_1 s_3$	tata	tata
$r_2 s_1$	tadda	tadda
$r_2 s_2$	tadta	tadta
$r_2 s_3$	tada	tada
$r_3 s_1$	ata	ata
$r_3 s_2$	ada	ada
$r_3 s_3$	aa	aa

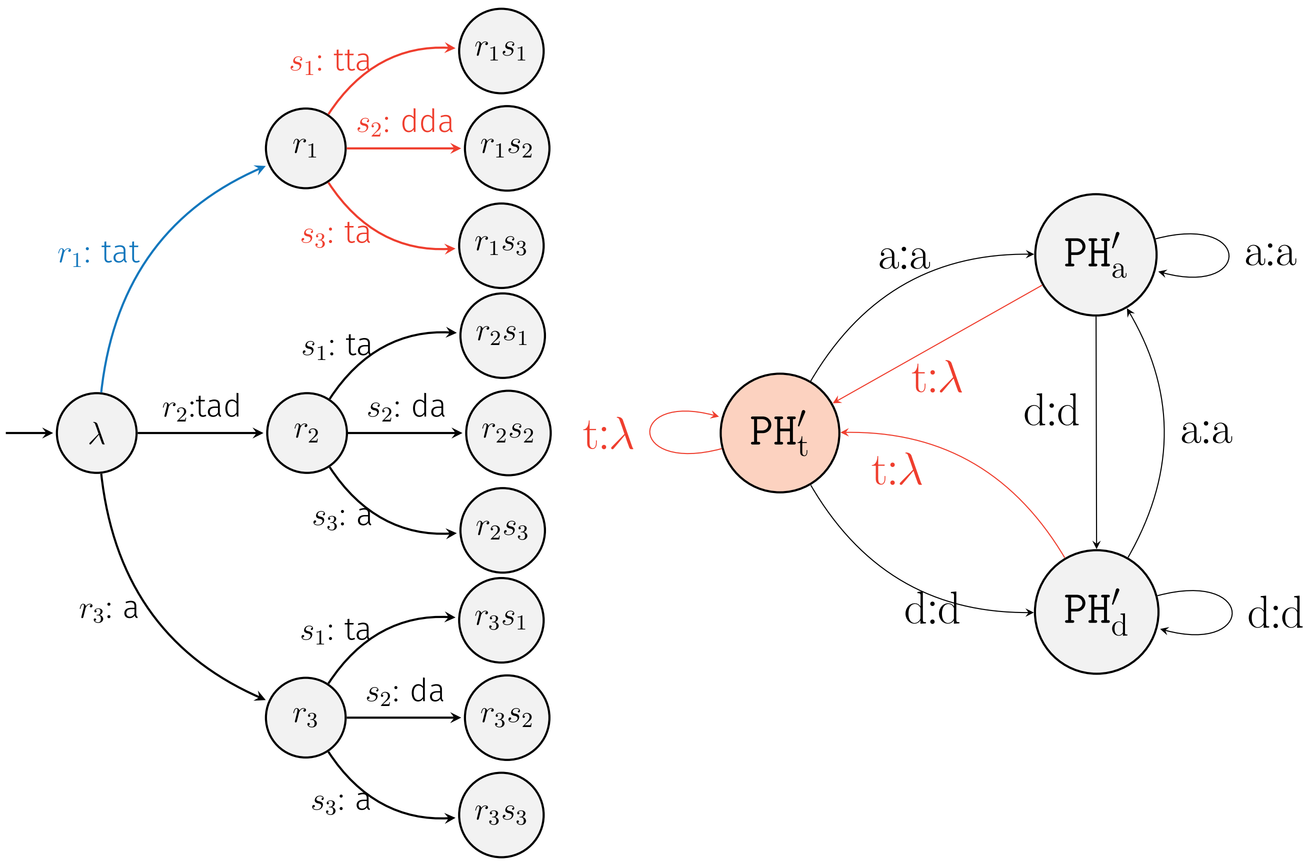


Appendix: Regressive assimilation





Appendix: Regressive assimilation



Appendix: Regressive assimilation