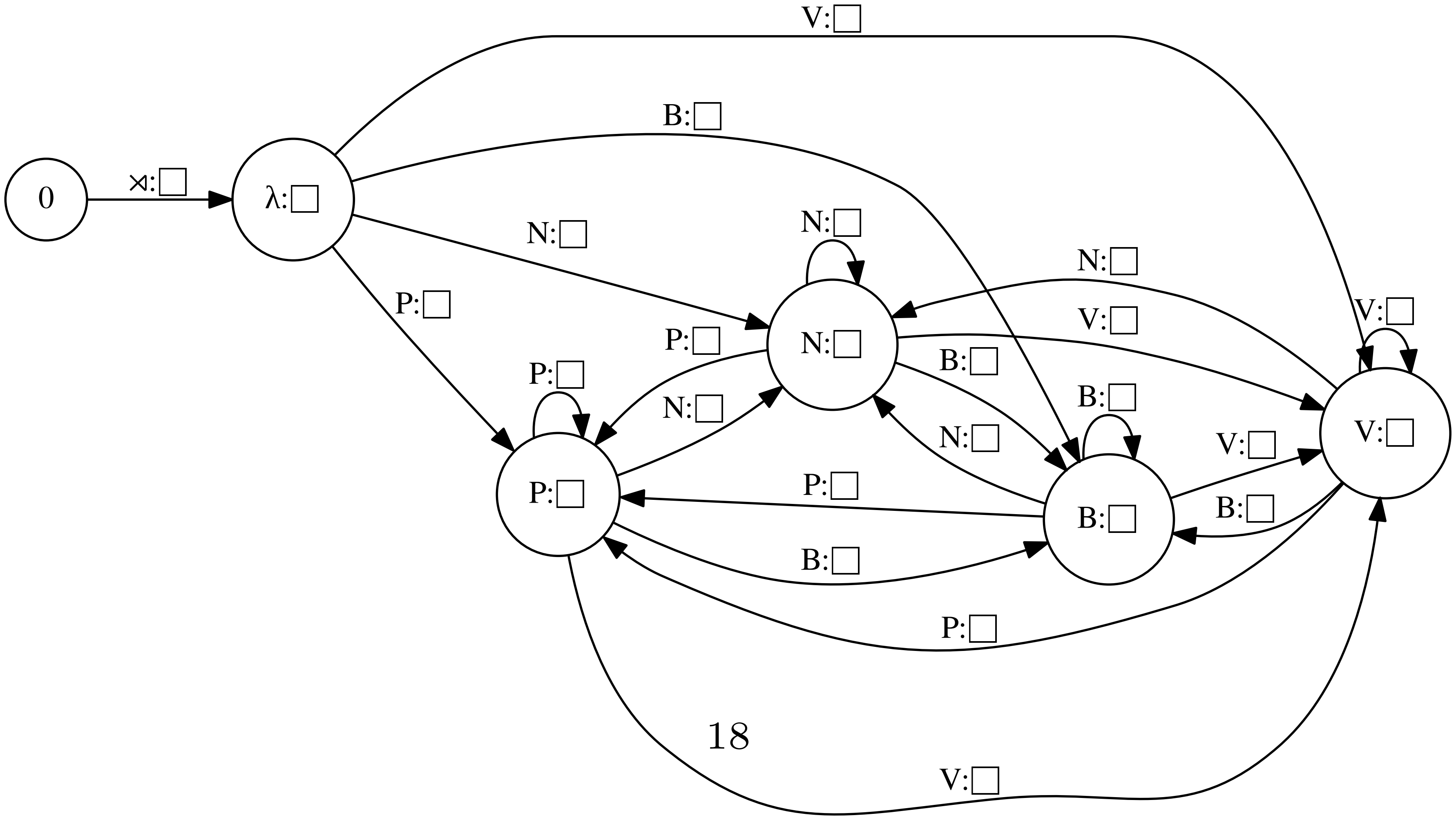
Very efficient learning of structured classes of subsequential functions from positive data

Adil Soubki, and Salam Khalifa

Structured Onward Subsequential Function Inference Algorithm (**SOSFIA**)

Data: A sample $S \subset \rtimes\Sigma^* \ltimes \times \Delta^*$, an output-empty DSFST $\tau_\square = \langle Q, q_0, q_f, \Sigma, \{\square\}, \delta \rangle$
Result: $\tau_\square$ as a DSFST with filled transitions

Structured Onward Subsequential Function Inference Algorithm (**SOSFIA**)



Structured Onward Subsequential Function Inference Algorithm (**SOSFIA**)

$F \leftarrow$ empty_Queue *(FIFO)*
Push$(F, (q_0, \lambda))$
mark(q_0)
while F *is not empty* **do**
  $(q, w) \leftarrow Shift\_First(F)$
  for $\sigma \in \Sigma \cup \{\rtimes, \ltimes\}$ *in lexicographic order* **do**
    for $\delta_i = (q, \sigma, \square, q') \in \delta$ **do**
      if *there exists* $\sigma' \neq \sigma$ *such that* $(q, \sigma', u, q'') \in \delta$ **then**
        Change δ_i to (q, σ, v, r), where $v = \texttt{min\_change}_S(w, \sigma)$
      else
        Change δ_i to (q, σ, λ, r)
      if q' *is not marked* **then**
        Push$(F, (q', w\sigma))$
        mark(q')
return $\tau_\square$;

The Theoretical Learning Result

- For every output-empty transducer $\tau_{\square}$, the SOSFIA strongly identifies $\tau_{\tau_{\square}}$ in polynomial time and data.
- It is worth mentioning that the algorithm presented in this paper returns a function even given incomplete data.

Demonstrations - Input Strictly Local Functions

- Tested the learner on three example ISL functions.
- They can be thought of as phonological ‘repairs’ that prevent a surface output sequence D⋉ (where D is a voiced obstruent).

Process	Rule
Final devoicing	$D \rightarrow T \; / \; \_\_ \ltimes$
Deletion	$D \rightarrow \emptyset \; / \; \_\_ \ltimes$
Epenthesis	$\emptyset \rightarrow V \; / \; D\_\_ \ltimes$

Demonstrations - Input Strictly Local Functions

- $\Sigma = \{D, T, N, V\}$
- D is a voiced obstruent (e.g., ‘b’, ‘d’, ‘z’, ‘g’)
- T is a voiceless obstruent (e.g., ‘p’, ‘t’, ‘s’, ‘k’),
- N is a sonorant consonant (nasal sounds and ‘l’ and ‘r’)
- V is a vowel.
- A data set of 1365 string pairs.
- The left projection of the data set is $\Sigma^{\leq 5}$
- Each string in the left projection was paired with an output string according to the target function.

Demonstrations - Input Strictly Local Functions

- The same output-empty transducer was used in all three test cases, as the DSFST for each rule only differs in terms of the output strings.

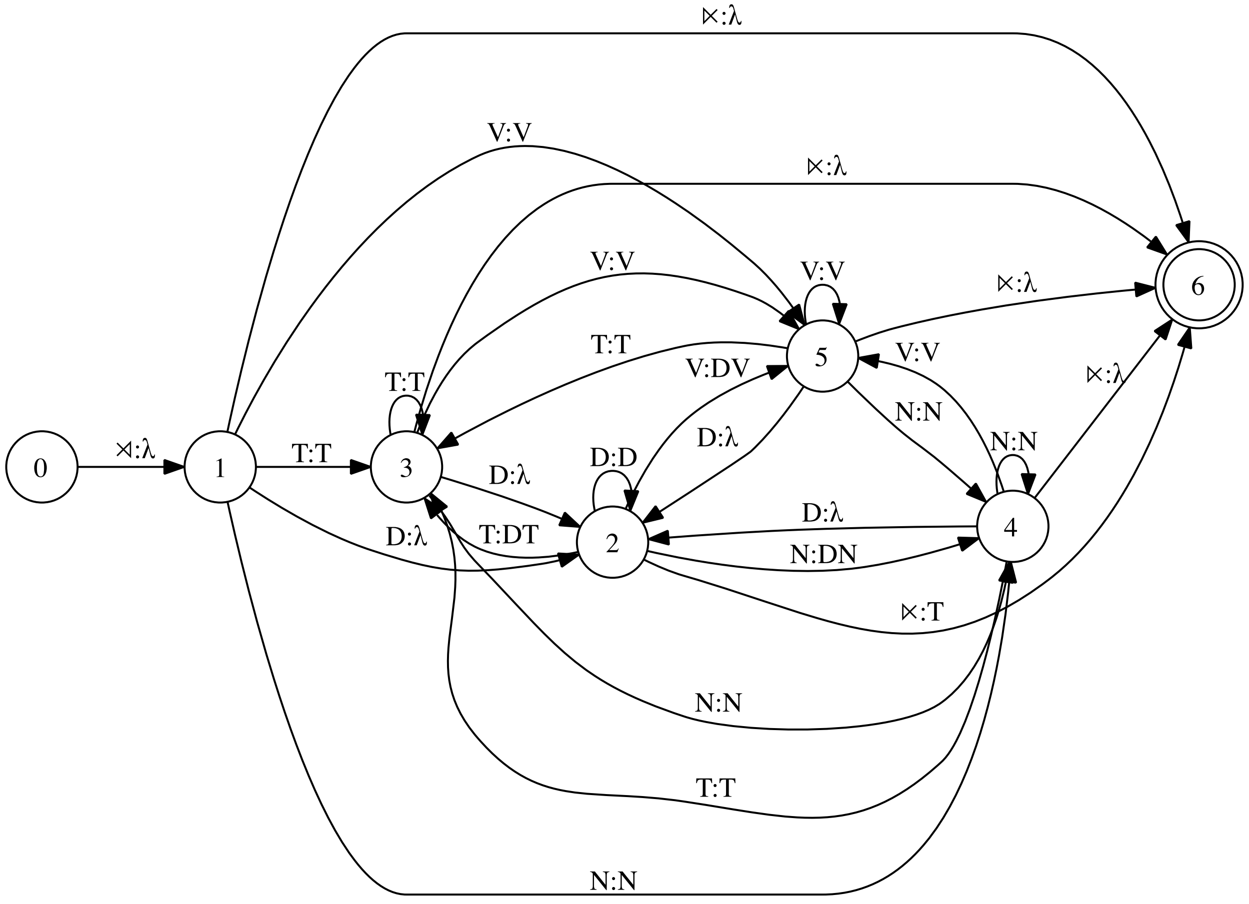


Figure 5: Final devoicing test case

Demonstrations - Non-ISL Phonological Processes

- These are ‘long-distance’ processes such as the sibilant harmony process attested in Samala.
- In this language, all sibilant sounds (e.g., ‘s’, ‘ʃ’) in a word must be the same as the rightmost one.

/hasxintilawaʃ / ↦ [haʃxintilawaʃ] *‘his former gentile name’*

Demonstrations - Non-ISL Phonological Processes

- $\Sigma = \{s, ʃ, t, a\}$
- A dataset of string pairs in which the left projection is $\Sigma^{\leq 4}$.
- Each of these strings was paired with one in which all sibilants (if any) assimilate to the rightmost one.

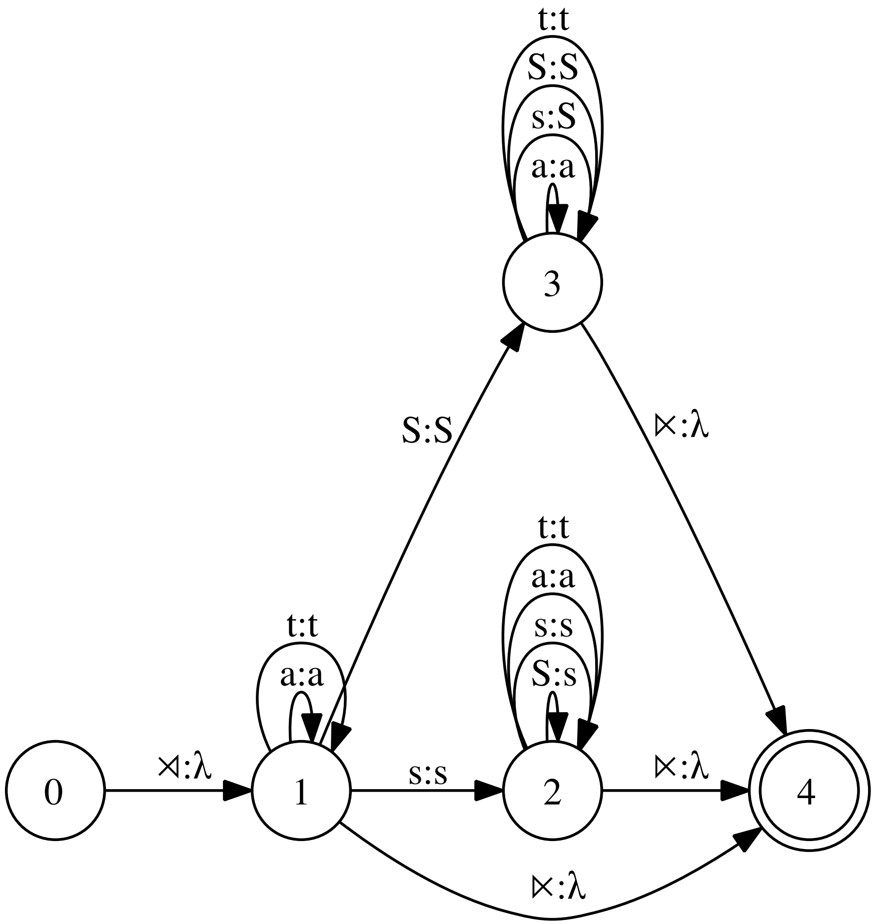


Figure 6: Sibilant harmony test case

Demonstrations - Non-ISL Phonological Processes

- This process is right subsequential; however, all this entails is that the input and output strings must be reversed to get the correct mapping. This has no effect on how the learner functions.
- The output-empty DSFST given to the learner includes separate states for words in which ‘s’ is the rightmost sibilant and words in which ‘ʃ’ is the rightmost sibilant (i.e., the *a priori* knowledge given to the learner identifies the set of segments involved in the process). This is akin to phonological theories in which certain classes of segments (here the sibilants) are represented on distinct tiers.

Demonstrations - Morphological Parsing

- The learner was tested on a morphology-to-phonology function in which the meanings of morphemes are mapped to their pronunciations.
- The data were from Swahili verbs, which show a series of prefixes indicating person, number, and tense.
- The morphological breakdown of *nimenipenda* ‘I have liked myself’

ni +	me +	ni +	penda
1-acc	perf	1st-nom	like

‘I have liked myself’

Demonstrations - Morphological Parsing

- The learner was given an empty transducer representing the possible morpheme orders and 90 pairs of the shape <*morpheme string, phoneme string*>:

1-nom+perf+1-acc+like, nimenipenda
3-nom+pres+1-acc+like, ananipenda
2-nom+perf+1-pl-acc+beat, umetupiga

Demonstrations – Similar tasks (Goldman 2022)

	Input		Output
Eng	give	IND;FUT;NOM(1,SG);ACC(3,SG,MASC);DAT(3,SG,FEM)	I will give him to her
Deu	geben	IND;FUT;NOM(1,SG);ACC(3,SG,MASC);DAT(3,SG,FEM)	ich werde ihn ihr geben
Tur	vermek	IND;FUT;NOM(1,SG);ACC(3,SG);DAT(3,SG)	onu ona vereceğim
Heb	נתן	IND;FUT;NOM(1,SG);ACC(3,SG,MASC);DAT(3,SG,FEM)	אתן אותו לה
Heb_{voc}	נָתַן	IND;FUT;NOM(1,SG);ACC(3,SG,MASC);DAT(3,SG,FEM)	אֵתֶּן אוֹתוֹ לָהּ

(a) Inflection examples

Demonstrations – Similar tasks (Goldman 2022)

		Average		Eng		Deu		Heb		$\text{Heb}_{vocalized}$		Tur	
		word	clause	word	clause	word	clause	word	clause	word	clause	word	clause
inflec.	LSTM	84.7 ±1.1	70.0 ±1.2	86.0 ±1.8	68.5 ±3.8	64.5 ±4.7	47.5 ±4.0	90.7 ±1.6	82.5 ±0.6	91.7 ±1.1	**70.0 ±1.2**	90.8 ±0.9	81.6 ±2.1
	DEEPSPIN	89.4 ±0.8	71.8 ±0.5	87.3 ±2.8	78.4 ±1.5	78.2 ±0.5	40.0 ±0.5	90.9 ±0.2	**86.1 ±0.7**	93.1 ±2.0	**71.7 ±0.7**	97.5 ±2.1	82.7 ±1.6
	TRANSDUCE	86.7 ±0.5	**78.9 ±0.4**	86.8 ±0.4	**85.4 ±1.1**	76.6 ±2.5	**71.5 ±1.3**	89.4 ±0.6	80.4 ±0.8	81.1 ±0.5	60.0 ±1.1	99.4 ±0.1	**97.2 ±0.5**
	MT5	NA	51.9 ±1.1	NA	70.7 ±1.7	NA	57.7 ±3.3	NA	48.0 ±3.3	NA	34.2 ±1.4	NA	48.7 ±1.7

Demonstrations – Similar tasks (Goldman 2022)

Morphology Without Borders: Clause-Level Morphology

Omer Goldman
Bar Ilan University
omer.goldman@gmail.com

Reut Tsarfaty
Bar Ilan University
reut.tsarfaty@biu.ac.il

Demonstrations – Similar tasks (Goldman 2022)

	Input		Output
Eng	IND;FUT;NOM(1,SG);ACC(3,SG,MASC);DAT(3,SG,FEM)	I will give him to her	
	IND;PRS;NOM(1,PL);ACC(2);DAT(3,PL);NEG		we don’t give you to them
Deu	IND;FUT;NOM(1,SG);ACC(3,SG,MASC);DAT(3,SG,FEM)	ich werde ihn ihr geben	
	IND;PRS;NOM(1,PL);ACC(2,SG);DAT(3,PL);NEG		wir geben dich ihnen nicht
Tur	IND;FUT;NOM(1,SG);ACC(3,SG);DAT(3,SG)	onu ona vereceğim	
	IND;PRS;PROG;NOM(1,PL);ACC(2,SG);DAT(3,PL);NEG		seni onlara vermiyoruz
Heb	IND;FUT;NOM(1,SG);ACC(3,SG,MASC);DAT(3,SG,FEM)	אתן אותו לה	
	IND;PRS;NOM(1,PL,MASC);ACC(2,SG,MASC);DAT(3,PL,FEM);NEG		אנחנו לא נותנים אותך להן
Heb_{voc}	IND;FUT;NOM(1,SG);ACC(3,SG,MASC);DAT(3,SG,FEM)	אֶתֵּן אוֹתוֹ לָהּ	
	IND;PRS;NOM(1,PL,MASC);ACC(2,SG,MASC);DAT(3,PL,FEM);NEG		אֲנַחְנוּ לֹא נוֹתְנִים אוֹתְךָ לָהֶן

(b) Reinflection examples

Demonstrations – Similar tasks (Goldman 2022)

	Input	Output	
Eng	I will give him to her	give	IND;FUT;NOM(1,SG);ACC(3,SG,MASC);DAT(3,SG,FEM)
Deu	ich werde ihn ihr geben	geben	IND;FUT;NOM(1,SG);ACC(3,SG,MASC);DAT(3,SG,FEM)
Tur	onu ona vereceğim	vermek	IND;FUT;NOM(1,SG);ACC(3,SG);DAT(3,SG)
Heb	אתן אותו לה	נתן	IND;FUT;NOM(1,SG);ACC(3,SG,MASC);DAT(3,SG,FEM)
Heb$_{voc}$	אֶתֵּן אוֹתוֹ לָהּ	נָתַן	IND;FUT;NOM(1,SG);ACC(3,SG,MASC);DAT(3,SG,FEM)

(c) Analysis examples