9th International Colloquium on Grammatical Inference, September 22-24, 2008, St. Malo (France)

Learning context-sensitive languages from linear structural information

José M. Sempere
Departamento de Sistemas Informáticos y Computación
Universidad Politécnica de Valencia

http://www.dsic.upv.es/users/tlcc/tlcc.html

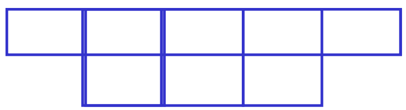
Outline

1. Introducing the Watson-Crick Finite Automaton (WKFA)
2. A representation theorem for languages accepted by WKFA
3. Characterizing local languages in WKFA
4. From local WKFA to local FA
5. The learning algorithm: correctness and complexity
6. Research in progress
7. Conclusions and Future Research

Introducing the WKFA (I)

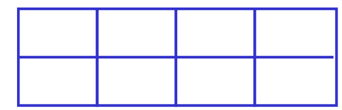
a sticker

$\begin{pmatrix} V^* \\ V^* \end{pmatrix}$

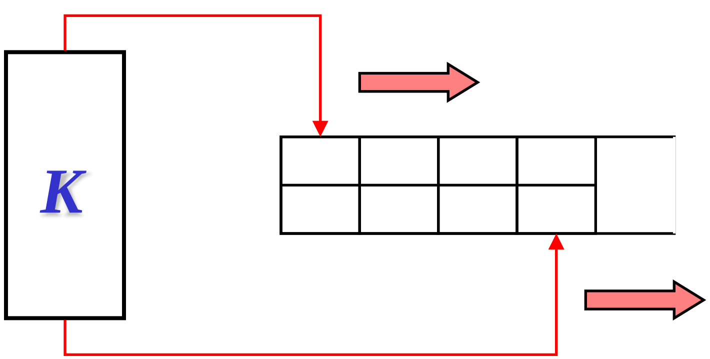


a complete molecule

$\begin{bmatrix} V^* \\ V^* \end{bmatrix}$



A Watson-Crick Finite Automaton (WKFA)



$$M = (V, \gamma, K, s_0, F, \delta)$$

V, K disjoint alphabets (symbols and states)

$\gamma \subseteq V \times V$ (symmetric relation of complementarity)

$s_0 \in K$ (initial state)

$F \subseteq K$ (final states)

$\delta : K \times \begin{pmatrix} V^* \\ V^* \end{pmatrix} \rightarrow \mathcal{P}(K)$ (transition function)

Introducing the WKFA (II)

Upper strand language $M = (V, \gamma, K, s_0, F, \delta)$

$$L_u(M) = \{ w_1 \in V^* : s_0 \begin{bmatrix} w_1 \\ w_2 \end{bmatrix} \overset{*}{\Longrightarrow} \begin{bmatrix} w_1 \\ w_2 \end{bmatrix} s_f \,,\ s_f \in F, w_2 \in V^*, \gamma(w_1) = \gamma(w_2) \}$$

$L_m(M)$ will denote the double stranded language accepted by M

REG ⊂ AWK(u) ⊂ CS

AWK(u) and CF are not comparable

$$L = \{ ww^r : w \in \Sigma^* \} \notin \text{AWK(u)}$$

$$L = \{ a^n b^n c^n : n \geq 0 \} \in \text{AWK(u)}$$

A Representation Theorem

Theorem (**Sempere, 2004**)

Every double stranded language accepted by an arbitrary WKFA is the result of the intersection between a linear language and an even linear one.

1. $s \rightarrow u_1\, s'\, u_2^r$ iff $s' \in \delta\left(s, \begin{bmatrix} u_1 \\ u_2 \end{bmatrix}\right)$
2. $s \rightarrow \#$ iff $s \in F$

1. $\forall\, (a,b) \in \gamma \quad S \rightarrow a\, S\, b$
2. $S \rightarrow \#$

An example

$\delta(q_0, \begin{bmatrix} a \\ \lambda \end{bmatrix}) = \{q_a\}$ $\delta(q_a, \begin{bmatrix} a \\ \lambda \end{bmatrix}) = \{q_a\}$ $\delta(q_a, \begin{bmatrix} b \\ a \end{bmatrix}) = \{q_b\}$ $\delta(q_b, \begin{bmatrix} b \\ a \end{bmatrix}) = \{q_b\}$

$\delta(q_b, \begin{bmatrix} c \\ b \end{bmatrix}) = \{q_c\}$ $\delta(q_c, \begin{bmatrix} c \\ b \end{bmatrix}) = \{q_c\}$ $\delta(q_c, \begin{bmatrix} \lambda \\ c \end{bmatrix}) = \{q_f\}$ $\delta(q_f, \begin{bmatrix} \lambda \\ c \end{bmatrix}) = \{q_f\}$

a	a	a	b	b	b	c	c	c
a	a	a	b	b	b	c	c	c

The linear grammar

$q_0 \rightarrow a\ q_a$
$q_a \rightarrow a\ q_a \mid b\ q_b\ a$
$q_b \rightarrow b\ q_b\ a \mid c\ q_c\ b$
$q_c \rightarrow c\ q_c\ b \mid q_f\ c$
$q_f \rightarrow q_f\ c \mid \#$

The even linear grammar

$S \rightarrow a\ S\ a \mid b\ S\ b \mid$
$c\ S\ c \mid \#$

$$L = \{\ a^n b^n c^n : n \geq 0\ \}$$

Local Testability in the Strict Sense

Let Σ be an alphabet and $k > 0$. We take $I_k, F_k \subseteq \Sigma^{\leq k-1}$ and $T_k \subseteq \Sigma^k$

We will say that a language L is k-testable in the strict sense if the following equation holds

$$L \cap \Sigma^{k-1}\Sigma^* = I_k\Sigma^* \cap \Sigma^*F_k - \Sigma^*T_k\Sigma^*$$

- Every k-testable language in the strict sense is regular for any $k > 0$
- The hierarchy of k-testable languages in the strict sense is infinite
- The class of k-testable languages in the strict sense will be denoted by k-LTSS
- The class of testable languages in the strict sense will be denoted by LTSS
- The class k-LTSS can be efficiently learned from positive data (García *et al.* 1990: Algorithm **KTSS**)

A reduction to regular languages (I)

From linear languages to even linear languages

Every linear grammar can be transformed into an even linear one

$A \to w$	$A \to w$
$A \to u\ B\ v$ with $\|u\| = \|v\|$	$A \to u\ B\ v$
$A \to u\ B\ v$ with $\|u\| < \|v\|$	$A \to u@^{\|v\|-\|u\|}\ B\ v$
$A \to u\ B\ v$ with $\|u\| > \|v\|$	$A \to u\ B\ v@^{\|u\|-\|v\|}$

Example

$S \to aAbb \mid aaBb$	$S \to a@Abb \mid aaBb@$
$A \to aAbb \mid \lambda$	$A \to a@Abb \mid \lambda$
$B \to aaBb \mid \lambda$	$B \to aaBb@ \mid \lambda$

A reduction to regular languages (II)

From even linear languages to regular languages (Sempere and García, 1994)

The σ transformation

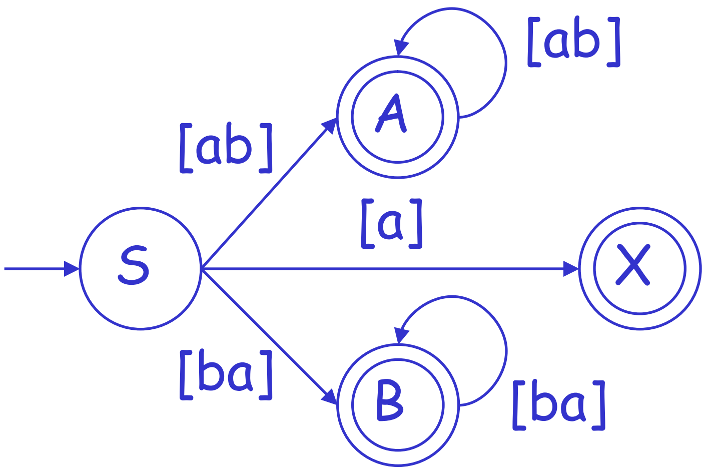
- $\sigma(\lambda) = \lambda$
- $(\forall\ a \in \Sigma)\ \ \sigma(a) = a$
- $(\forall a,b \in \Sigma)\ (\forall\ x \in \Sigma^*)\ \sigma(axb) = [ab]\ \sigma(x)$
- $\sigma(L) = \{\ \sigma(x) : x \in L\}$

If L is an even linear language then $\sigma(L)$ is regular

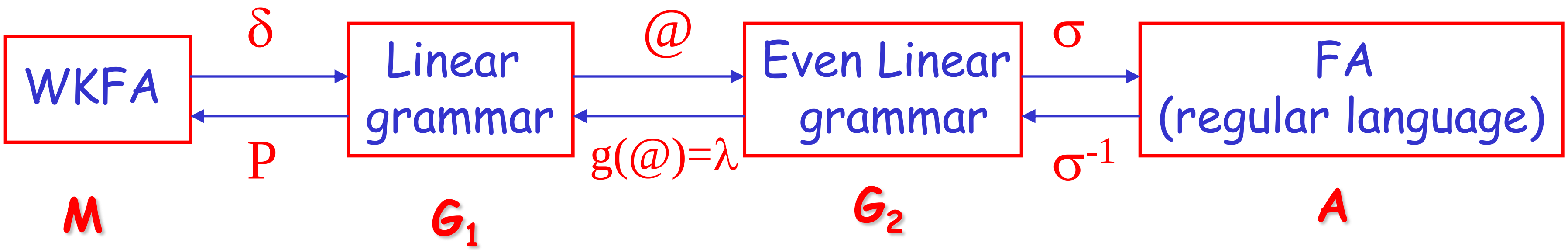
$\sigma^{-1}(L)$ can be deduced from $\sigma(L)$

Example

$S \rightarrow aAb \mid bBa \mid a$
$A \rightarrow aAb \mid \lambda$
$B \rightarrow bBa \mid \lambda$



Local testability in the double strand



We will say that $L_m(M)$ is in k-LTSS if $L(A)$ is in k-LTSS

- The hierarchy of testable languages $L(A)$ is inherited with respect to the languages $L_m(M)$

AWK_u^{KLTSS} will denote the class of upper strand languages accepted by WKFA with double stranded languages in k-LTSS

$\mathbf{L = \{ a^n b^n c^n : n \geq 0 \}}$ is in AWK_u^{2LTSS}

$\delta(q_0, \begin{bmatrix} a \\ \lambda \end{bmatrix}) = \{q_a\}$ $\delta(q_a, \begin{bmatrix} a \\ \lambda \end{bmatrix}) = \{q_a\}$ $\delta(q_a, \begin{bmatrix} b \\ a \end{bmatrix}) = \{q_b\}$ $\delta(q_b, \begin{bmatrix} b \\ a \end{bmatrix}) = \{q_b\}$

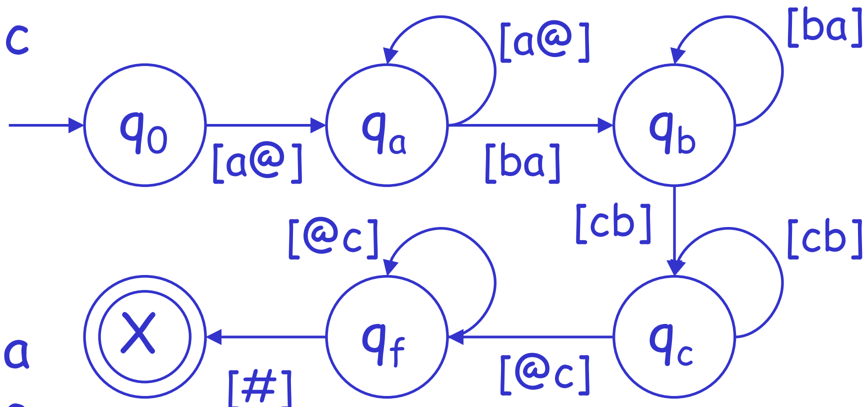
$\delta(q_b, \begin{bmatrix} c \\ b \end{bmatrix}) = \{q_c\}$ $\delta(q_c, \begin{bmatrix} c \\ b \end{bmatrix}) = \{q_c\}$ $\delta(q_c, \begin{bmatrix} \lambda \\ c \end{bmatrix}) = \{q_f\}$ $\delta(q_f, \begin{bmatrix} \lambda \\ c \end{bmatrix}) = \{q_f\}$

The linear grammar

$q_0 \rightarrow a\ q_a$
$q_b \rightarrow b\ q_b\ a \mid c\ q_c\ b$
$q_f \rightarrow q_f\ c \mid \#$

$q_a \rightarrow a\ q_a \mid b\ q_b\ a$
$q_c \rightarrow c\ q_c\ b \mid q_f\ c$

The finite automaton



The even linear grammar

$q_0 \rightarrow a\ q_a\ @$
$q_b \rightarrow b\ q_b\ a \mid c\ q_c\ b$
$q_f \rightarrow @\ q_f\ c \mid \#$

$q_a \rightarrow a\ q_a\ @ \mid b\ q_b\ a$
$q_c \rightarrow c\ q_c\ b \mid @\ q_f\ c$

2LTSS

The learning scheme

Input: A finite sample of linear structural duplicated strings S defined over Σ

Output: A WKFA *A* such that $S^+ \subseteq L_u(A)$

Method:

(1) $S_{ell} = ell(S)$
(2) $S_\sigma = \sigma(S_{ell})$
(3) A_r= **KTSS**(S_σ)
(4) $G_{ell} = \sigma^{-1}(A_r)$
(5) $G_{lin} = ell^{-1}(G_{ell})$
(6) $A = WKFA(G_{lin})$ where $\gamma = \{ (a,a) : a \in \Sigma \}$
(7) Return(*A*)

EndMethod.

(S^+ denotes the upper strand strings induced by S)

An example

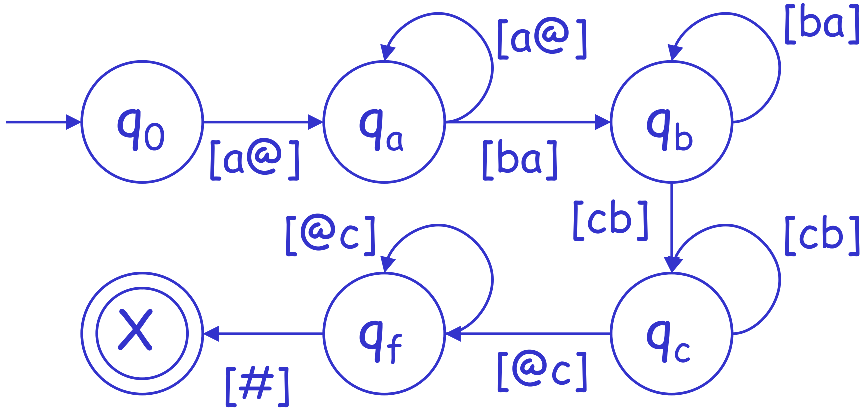
S = {(a(b(c((#)c)b)a)), (a(a(b(b(c(c(((#)c)c)b)b)a)a))) }

S_{ell} = { abc@#cba@, aabbcc@@#ccbbaa@@}

S_{σ} = { [a@][ba][cb][@c][#], [a@][a@][ba][ba][cb][cb][@c][@c][#] }

S^{+} = { abc, aabbcc}

A_r= KTSS(S_{σ}) (k=2)



Correctness and complexity

Proposition. The proposed algorithm runs in polynomial time with the size of the input sample S

Proposition. AWK_u^{KLTSS} is identifiable in the limit from only positive structural information

Research in progress
(generalizing the learning scheme)

Input: A finite sample of linear structural duplicated strings S defined over Σ

Output: A WKFA A such that $S^+ \subseteq L_u(A)$

Method:

(1) S_{ell} = ell(S)
(2) S_σ = $\sigma(S_{ell})$
(3) A_r = LearningRegPos(S_σ)
(4) G_{ell} = $\sigma^{-1}(A_r)$
(5) G_{lin} = $ell^{-1}(G_{ell})$
(6) A = WKFA(G_{lin}) where $\gamma = \{ (a,a) : a \in \Sigma \}$
(7) Return(A)

EndMethod.

Q: How does LearningRegPos characterize subclasses of *CS* ?

Conclusions

- New computation models influenced by DNA and cellular computing give a new representation space for Grammatical Inference (sticker systems, splicing systems, P systems, etc.)

- The new models allow the inference of larger language classes by using well known GI methods (in this case linear languages and reductions to regular ones)

Future research

- Is it possible GI "*in vitro*" or "*in vivo*" ?

- Is it enough the use of linear languages to infer larger classes of recursively enumerable languages ?

- Can duplication and complementarity be generalized as an input interface for GI ?