Realizable and Context-Free Hyperlanguages

Hadar Frenkel, CISPA Helmholtz Center for Information Security, Germany

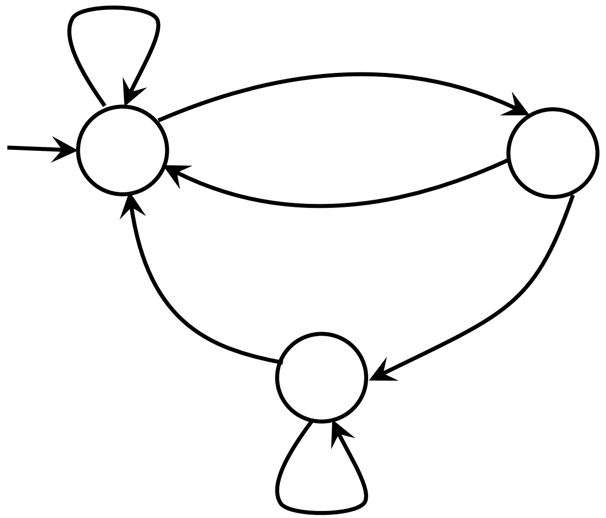
Sarai Sheinvald, Braude College of Engineering, Israel

CISPA HELMHOLTZ CENTER FOR INFORMATION SECURITY

BRAUDE College of Engineering, Karmiel

Hyperproperties [Clarkson & Schneider '10]

Standard Properties: behavior of the traces of the system



"Every request is eventually granted"

Property = a set of traces. LTL, Regular expressions, …

Hyperproperties: behavior of the system in its entirety

"**For every** trace with high-security signals,
there exists a trace in which they are unobservable"

Hyperproperty = a set of sets of traces. HyperLTL

In this talk

Finite-Word Hyperautomata

- Hyperautomata
- Realizability of hyperlanguages

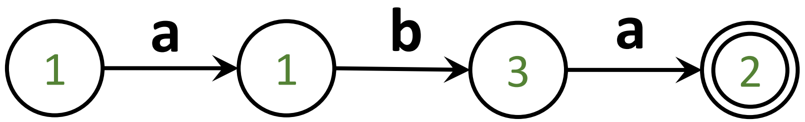
Context-Free Hypergrammars

- Hypergrammars
- Synchronous hypergrammars
- Emptiness and membership problems for hypergrammars

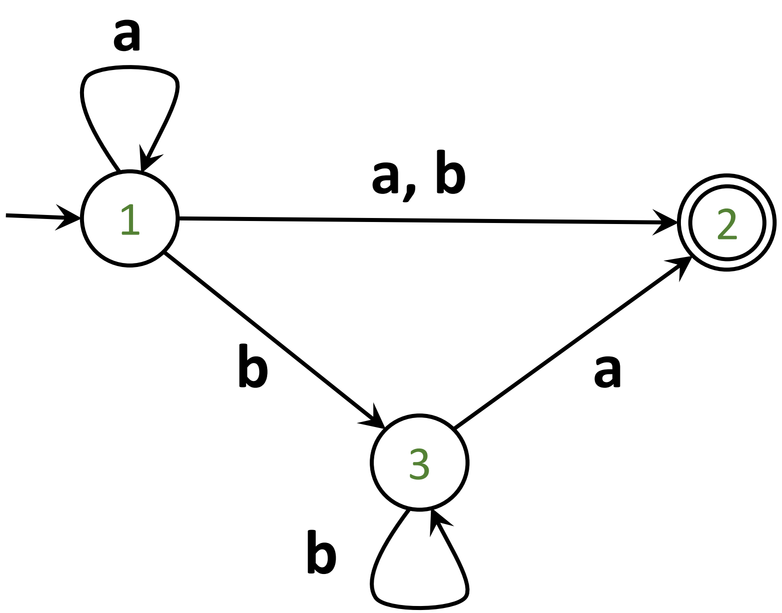
Finite-word automata

NFA: non-det finite word **automaton**

Runs: on **words**



Accepts a word **w** if **w** can reach an accepting state



The **language** of an NFA **A**: the set of all words that **A** accepts

NFA: regular languages

Hyperautomata [Bonakdarpour & Sheinvald '21]

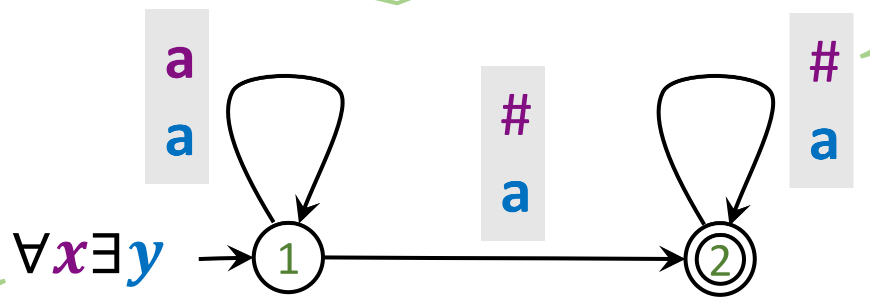
NFH: non-det finite word **hyperautomaton**

Runs: on **assignments** to the **variables**

x ← a a # #
y ← a a a a

Underlying NFA **A**

padding



Quantification condition α

An NFH accepts a **language** L
if L satisfies α w.r.t. A

“For every word there exists a longer word”

Hyperlanguage: all infinite languages over {a}

$\{L \mid L$ is infinite$\}$

Set of sets of words

NFH: regular hyperlanguages

Hyperautomata

Can express regular hyperproperties:

Noninference: replacing high-security commands with dummy value does not affect the low-security observable data.

$$\begin{matrix}\forall x \\ \exists y\end{matrix}\begin{pmatrix} \boldsymbol{low, high} \\ \boldsymbol{low, high \rightarrow low^{dummy}} \end{pmatrix}^*$$

Realizability

NFH A → Hyperlanguage $L_1, L_2, \ldots$

Realizability: Hyperlanguage $L_1, L_2, \ldots$ → NFH A

With which quantification condition?

Def: $\mathcal{L}$ = $L_1, L_2, \ldots$ is $\boldsymbol{\alpha}$**-realizable** if there is an $\boldsymbol{\alpha}$**-NFH** for $\mathcal{L}$

We study the basic case of singleton hyperlanguages: $\mathcal{L} = \{L\}$

- Various types of L
- Realizability and unrealizability results for various α

In this talk

Finite-Word Hyperautomata

- Hyperautomata ✓
- Realizability of $\{L\}$
  - Finite \ infinite L
  - Ordered L
  - Regular L

Context-Free Hypergrammars

- Hypergrammars
- Synchronous hypergrammars
- Emptiness and membership problems for hypergrammars

Realizability of $\{L\}$

Simple α does not suffice

$\forall x$ $\boxed{\text{A}}$: if L is accepted then also $L' \subset L$ $\Rightarrow$ **not ∀-realizable**

$\exists x$ $\boxed{\text{A}}$: if L is accepted with $x \leftarrow w$ then also L' for $w \in L' \cap L$ $\Rightarrow$ **not ∃-realizable**

Realizability of $\{L\}$: finite L

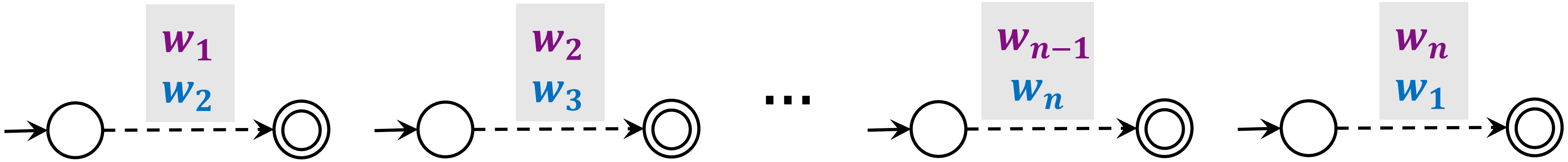
Simple α does not suffice

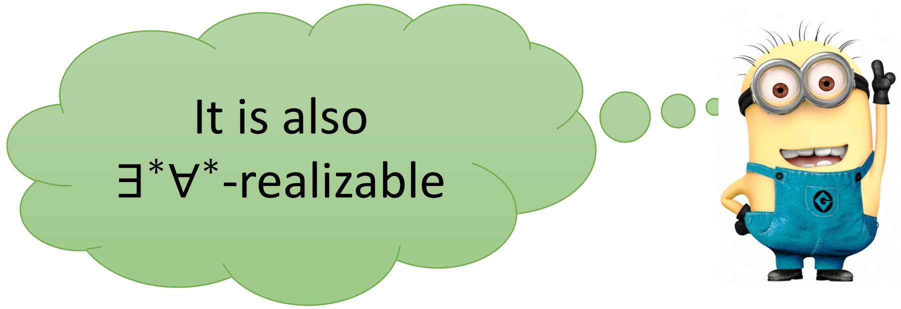
$\forall x$ A : if L is accepted then also $L' \subset L$ $\Rightarrow$ **not $\forall$-realizable**

$\exists x$ A : if L is accepted with $x \leftarrow w$ then also L' for $w \in L' \cap L$ $\Rightarrow$ **not $\exists$-realizable**

<u>**If L is finite then $\{L\}$ is $\forall\exists$-realizable:**</u> $L = \{w_1, \dots w_n\}$

$\forall x \exists y$





(Un)Realizability of $\{L\}$: infinite L

Simple α does not suffice

<u>$\{L\}$ is **not $\forall\exists$-realizable**</u>: Suppose that $\forall x \exists y$ $\boxed{A}$ accepts L

$\forall x$	w_1	w_2	…	w_i	…	w_{j-1}	
$\exists y$	w_2	w_3		w_{i+1}		w_j	$= w_i$

$\{w_i, w_{i+1}, \dots\} \in \mathcal{L}\{A\}$

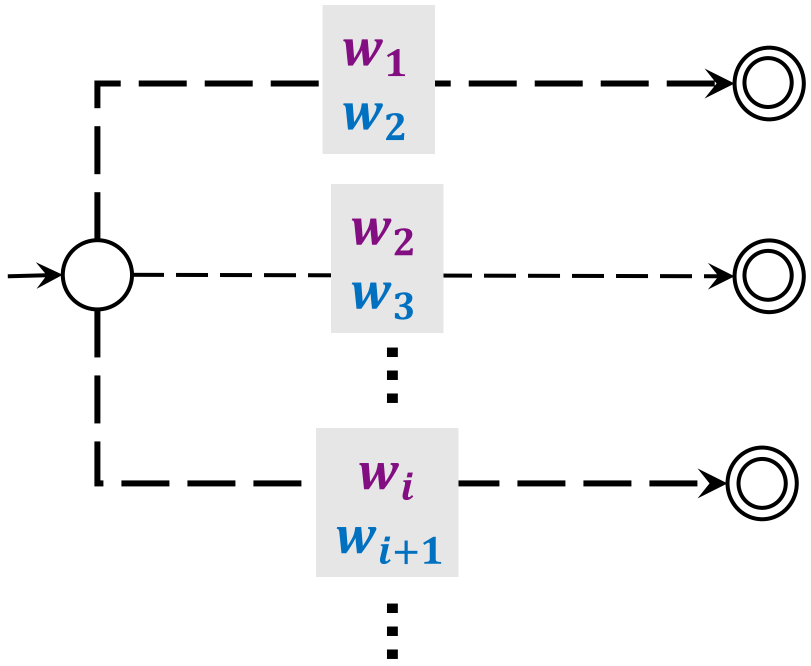
$\{w_i, w_{i+1}, \dots, w_j\} \in \mathcal{L}\{A\}$

It is also not $\exists^*\forall^*$-realizable

Realizability of $\{L\}$: Ordered L

Def: L is **ordered** if:

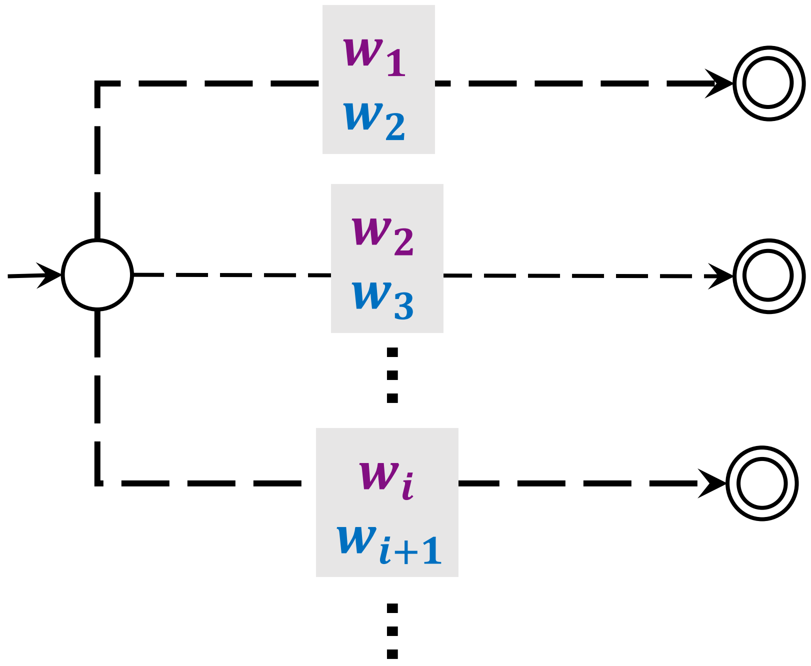
$L = \{w_1, w_2, \dots\}$ and there exists an NFA A_L



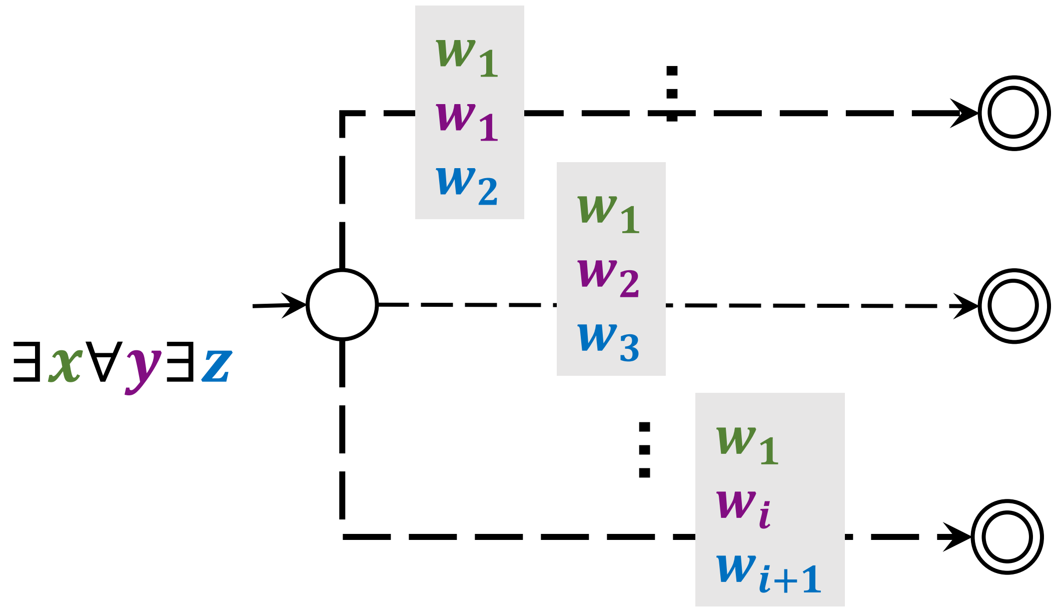
Realizability of $\{L\}$: Ordered L

<u>Def</u>: L is **ordered** if:

$L = \{w_1, w_2, \ldots\}$ and there exists an NFA A_L:

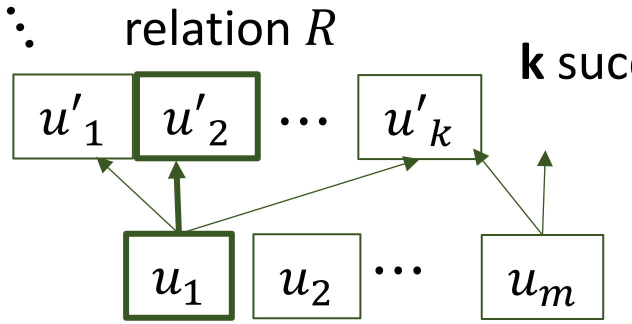


<u>If L is ordered then $\{L\}$ is $\exists\forall\exists$ -realizable</u>**:**



Realizability of $\{L\}$: Partially Ordered L

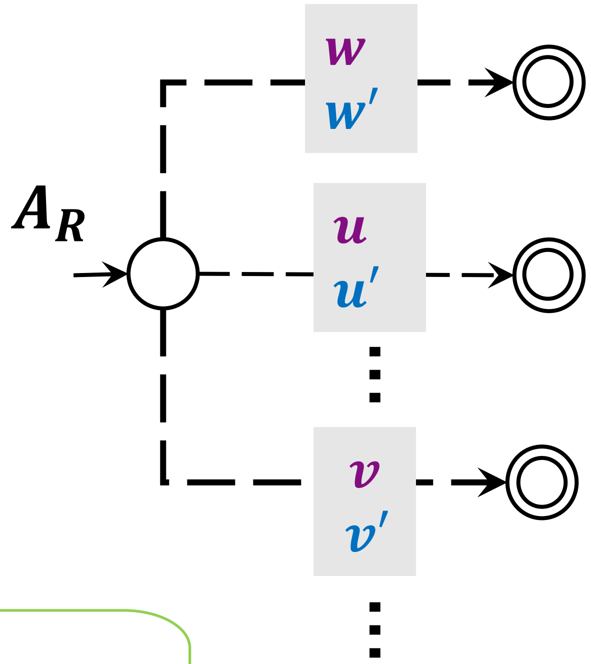
Def: L is **(m,k)-ordered** if: $L = \{w_1, w_2, \dots\}$



k succesors for each word

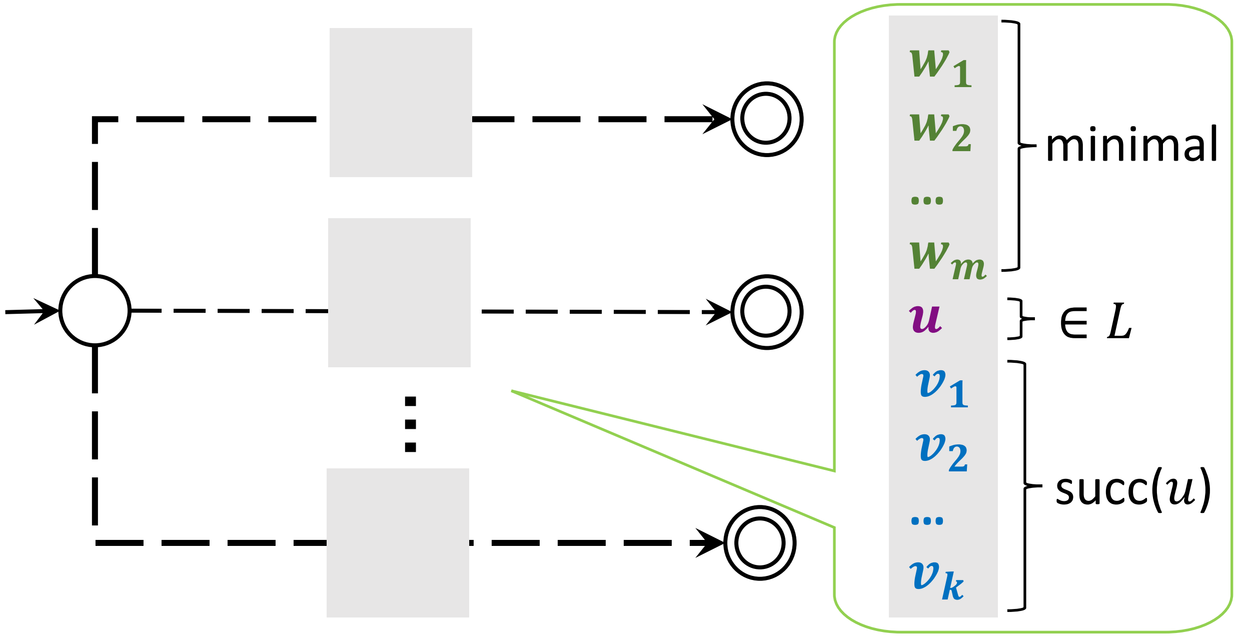
m minimal elements

There exists an NFA A_R for R



If L is (m,k)-ordered then $\{L\}$ is $\exists^m\forall\exists^k$-realizable:

$\exists x_1 \dots x_m \forall y \exists z_1 \dots z_k$



Realizability of $\{L\}$: Regular L

<u>If L is regular then $\{\boldsymbol{L}\}$ is **(m,k)**-ordered and $\exists^{m}\forall\exists^{k}$ -realizable</u>

∃∀∃ Realizable proof: automatic structures

<u>m:</u>

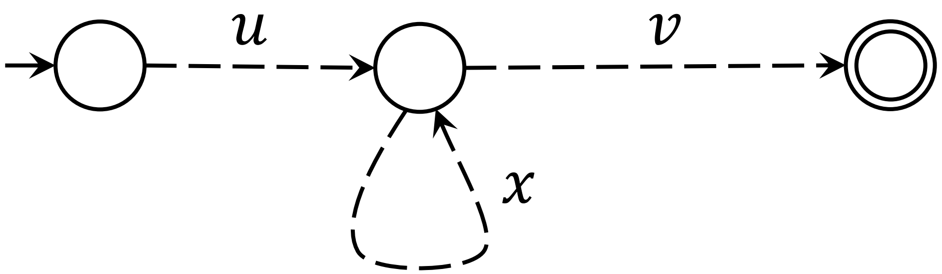
Minimal elements - simple paths to accepting states

$uv \in \mathrm{Min}$

<u>k:</u>

Successors words - one additional simple cycle

$u\boldsymbol{x}v \in succ(uv)$



In this talk

Finite-Word Hyperautomata

- Hyperautomata
- Realizability of $\{L\}$
  - Finite \ infinite L
  - Ordered L
  - Regular L

Context-Free Hypergrammars

- Hypergrammars
- Synchronous hypergrammars
- Emptiness & membership

Context-Free Grammars (CGF)

Terminal word

Derivation rule

Derives: **words**

(initial) variable

$S \to a\,S\,b \mid \epsilon$

$S \Rightarrow \underline{a\,S\,b} \Rightarrow a\,\underline{a\,S\,b}\,b \Rightarrow a\,a\,b\,b$

A terminal word **w** is in the **language** of a CGF **G**
if **w** can be derived from the initial variable

Context-Free Hypergrammars

CFHG: context-free **hypergrammar**

Underlying CFG **G**

$$\forall x \exists y \quad S \rightarrow \genfrac{}{}{0pt}{}{a}{a} \; S \; \genfrac{}{}{0pt}{}{b}{b} \;\Big|\; A$$

$$A \rightarrow \genfrac{}{}{0pt}{}{\#}{a} \; A \; \genfrac{}{}{0pt}{}{\#}{b} \;\Big|\; \genfrac{}{}{0pt}{}{\#}{a} \; \genfrac{}{}{0pt}{}{\#}{b}$$

Quantification condition α

padding

Context-Free Hypergrammars

CFHG: context-free **hypergrammar**

Derives: **assignments** to the **variables**

$\mathbf{x} \leftarrow$ a a # # b b

$\mathbf{y} \leftarrow$ a a a b b b

A GFHG accepts a **language** $\boldsymbol{L}$ if $\boldsymbol{L}$ satisfies $\boldsymbol{\alpha}$ w.r.t. $\boldsymbol{G}$

Underlying CFG **G**

Quantification condition $\boldsymbol{\alpha}$

$$\forall x \exists y \quad S \rightarrow \begin{matrix}a\\a\end{matrix} \; S \; \begin{matrix}b\\b\end{matrix} \;\Big|\; A$$

$$A \rightarrow \begin{matrix}\#\\a\end{matrix} \; A \; \begin{matrix}\#\\b\end{matrix} \;\Big|\; \begin{matrix}\#\\a\end{matrix} \; \begin{matrix}\#\\b\end{matrix}$$

padding

"For every word of type $a^n b^n$ there exists a longer word"

Hyperlanguage: all infinite languages $\subseteq \{\mathrm{a}^{\mathrm{n}}\mathrm{b}^{\mathrm{n}} | \mathrm{n} \in \mathbb{N}\}$

Set of sets of words

Context-Free Hypergrammars

$$\forall x \exists y \quad S \rightarrow \begin{matrix} a \\ a \end{matrix} \; S \; \begin{matrix} b \\ b \end{matrix} \;\Big|\; A$$

$$A \rightarrow \begin{matrix} \# \\ a \end{matrix} \; A \; \begin{matrix} \# \\ b \end{matrix} \;\Big|\; \begin{matrix} \# \\ a \end{matrix} \; \begin{matrix} \# \\ b \end{matrix}$$

Context-Free Hypergrammars

$$\forall x \exists y \quad S \rightarrow \binom{a}{a}\, S\, \binom{b}{b} \;\Big|\; A$$

$$A \rightarrow \binom{\#}{a}\, A\, \binom{\#}{b} \;\Big|\; \binom{\#}{a}\binom{\#}{b}$$

$$S \Rightarrow \underline{\binom{a}{a}\, S\, \binom{b}{b}} \Rightarrow \binom{a}{a}\underline{\binom{a}{a}\, S\, \binom{b}{b}}\binom{b}{b} \Rightarrow \binom{a}{a}\binom{a}{a}\,\underline{A}\,\binom{b}{b}\binom{b}{b} \Rightarrow \binom{a}{a}\binom{a}{a}\underline{\binom{\#}{a}\, A\, \binom{\#}{b}}\binom{b}{b}\binom{b}{b} \Rightarrow \binom{a}{a}\binom{a}{a}\underline{\binom{\#}{a}\binom{\#}{b}}\binom{b}{b}\binom{b}{b}$$

$$\binom{a}{a}\binom{a}{a}\binom{b}{a}\binom{b}{b}\binom{\ }{b}\binom{\ }{b}$$

Context-Free Hypergrammars

$$\forall x \exists y \quad S \rightarrow \begin{bmatrix} a \\ a \end{bmatrix} S \begin{bmatrix} b \\ b \end{bmatrix} \;\Big|\; A$$

$$A \rightarrow \begin{bmatrix} \# \\ a \end{bmatrix} A \begin{bmatrix} \# \\ b \end{bmatrix} \;\Big|\; \begin{bmatrix} \# \\ a \end{bmatrix} \begin{bmatrix} \# \\ b \end{bmatrix}$$

$$S \Rightarrow \begin{bmatrix} a \\ a \end{bmatrix} S \begin{bmatrix} b \\ b \end{bmatrix} \Rightarrow \begin{bmatrix} a \\ a \end{bmatrix} \begin{bmatrix} a \\ a \end{bmatrix} S \begin{bmatrix} b \\ b \end{bmatrix} \begin{bmatrix} b \\ b \end{bmatrix} \Rightarrow \begin{bmatrix} a \\ a \end{bmatrix} \begin{bmatrix} a \\ a \end{bmatrix} A \begin{bmatrix} b \\ b \end{bmatrix} \begin{bmatrix} b \\ b \end{bmatrix} \Rightarrow \begin{bmatrix} a \\ a \end{bmatrix} \begin{bmatrix} a \\ a \end{bmatrix} \begin{bmatrix} \# \\ a \end{bmatrix} A \begin{bmatrix} \# \\ b \end{bmatrix} \begin{bmatrix} b \\ b \end{bmatrix} \begin{bmatrix} b \\ b \end{bmatrix} \Rightarrow \begin{bmatrix} a \\ a \end{bmatrix} \begin{bmatrix} a \\ a \end{bmatrix} \begin{bmatrix} \# \\ a \end{bmatrix} \begin{bmatrix} \# \\ b \end{bmatrix} \begin{bmatrix} b \\ b \end{bmatrix} \begin{bmatrix} b \\ b \end{bmatrix}$$

at the middle of words leads to *asynchronization*

Synchronous Context-Free Hypergrammars

Easy solution:

hypergrammar G: $\forall \boldsymbol{x} \exists \boldsymbol{y}$ G

G $\cap$ $\begin{matrix} \Sigma^* \cdot \{\#\}^* \\ \Sigma^* \cdot \{\#\}^* \end{matrix}$

Avoid # at the middle of the word

a	a	#	#	b	b
a	a	a	b	b	b

Result: only the synchronous part of G

Can we define a hypergrammar that is **inherently** synchronous?

Synchronous Context-Free Hypergrammars

Can we define a hypergrammar that is **inherently** synchronous?

$S \to AB$

w

$A \to$

a	#
a	b
a	b

w'

$B \to$

#	#
#	#
c	c

$S \Rightarrow$

w

a	#
a	b
a	b

w'

#	#
#	#
c	c

$\mathrm{Rgt}(w) \subseteq \mathrm{Lft}(w')$

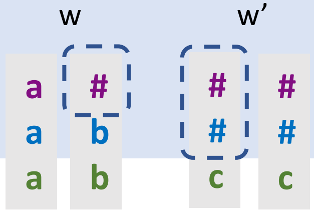
set of indices in which w contains # on the right

set of indices in which w' contains # on the left

Avoid # at the middle of the word

a	a	#	#	b	b
a	a	a	b	b	b

Synchronous Context-Free Hypergrammars



$Rgt(w) \subseteq Lft(w')$

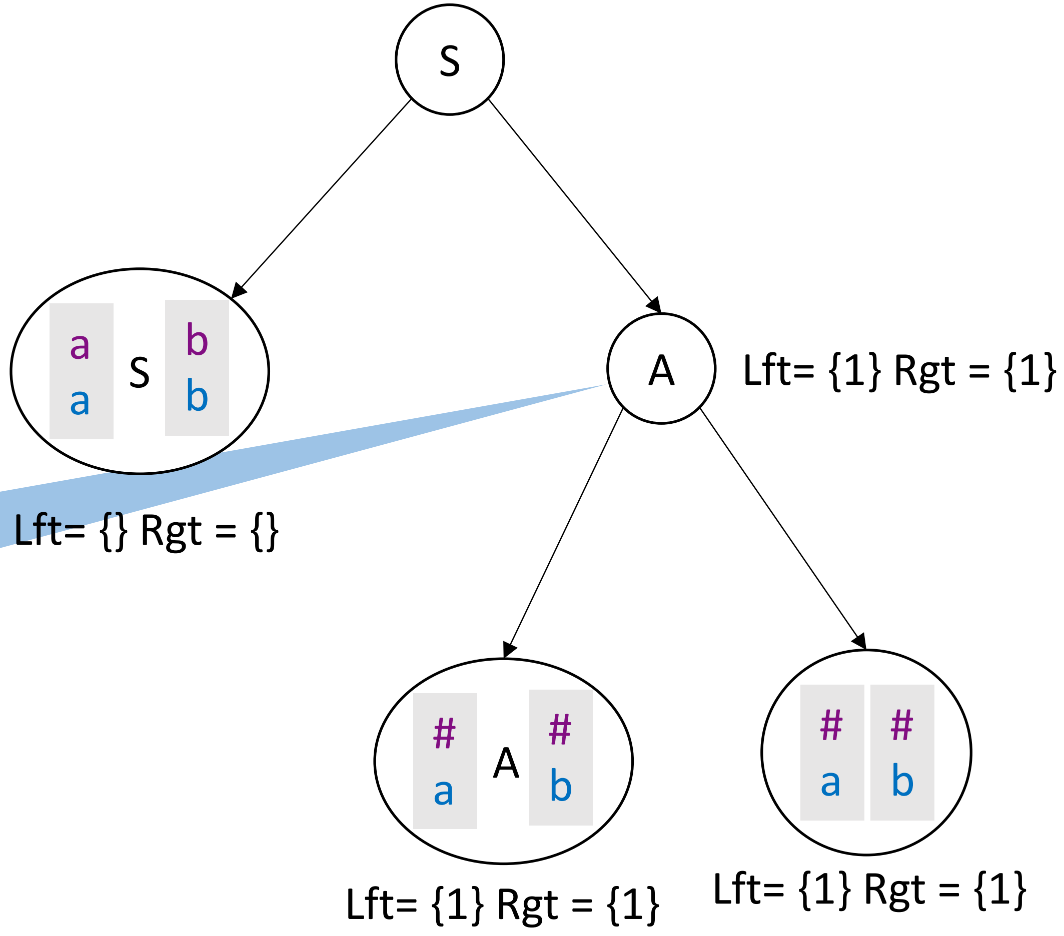
Can we define a hypergrammar that is **inherently** synchronous?

$$S \rightarrow \begin{bmatrix} a \\ a \end{bmatrix} S \begin{bmatrix} b \\ b \end{bmatrix} \mid A$$

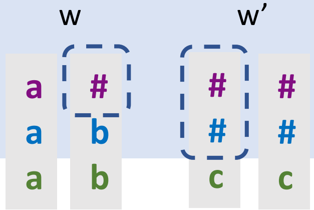
$$A \rightarrow \begin{bmatrix} \# \\ a \end{bmatrix} A \begin{bmatrix} \# \\ b \end{bmatrix} \mid \begin{bmatrix} \# \\ a \end{bmatrix} \begin{bmatrix} \# \\ b \end{bmatrix}$$

$$\mathrm{Lft}(X) = \bigcap_{X \rightarrow \alpha} \mathrm{Lft}\,(\alpha)$$

$$\mathrm{Rgt}(X) = \bigcup_{X \rightarrow \alpha} \mathrm{Rgt}\,(\alpha)$$



Synchronous Context-Free Hypergrammars



Rgt(w) ⊆ Lft(w')

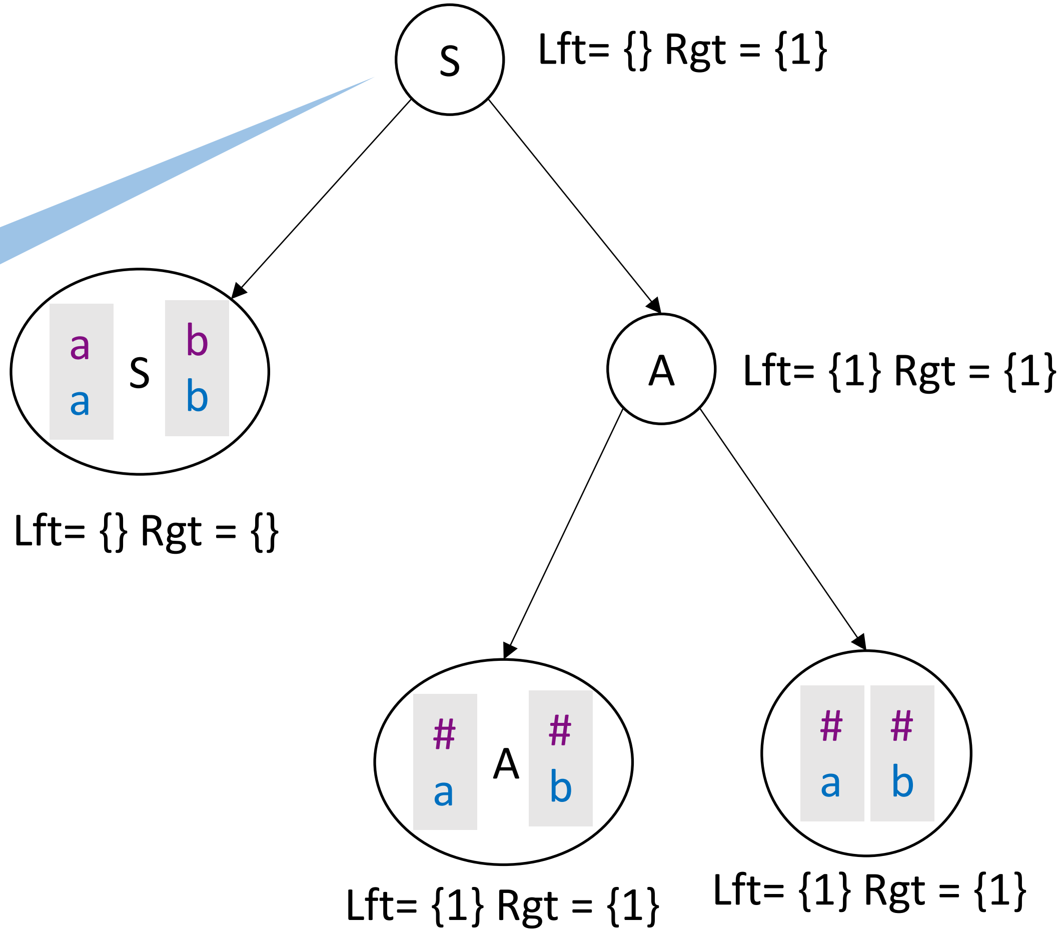
Can we define a hypergrammar that is **inherently** synchronous?

$$S \rightarrow \begin{matrix} a \\ a \end{matrix}\; S \;\begin{matrix} b \\ b \end{matrix} \;\Big|\; A$$

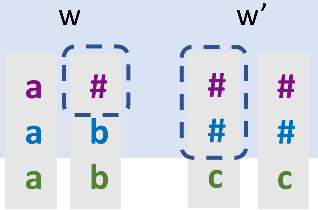
$$A \rightarrow \begin{matrix} \# \\ a \end{matrix}\; A \;\begin{matrix} \# \\ b \end{matrix} \;\Big|\; \begin{matrix} \# \\ a \end{matrix}\;\begin{matrix} \# \\ b \end{matrix}$$

$$\mathrm{Lft}(X) = \cap_{X\rightarrow\alpha}\, \mathrm{Lft}\,(\alpha)$$

$$\mathrm{Rgt}(X) = \cup_{X\rightarrow\alpha}\, \mathrm{Rgt}\,(\alpha)$$



Synchronous Context-Free Hypergrammars

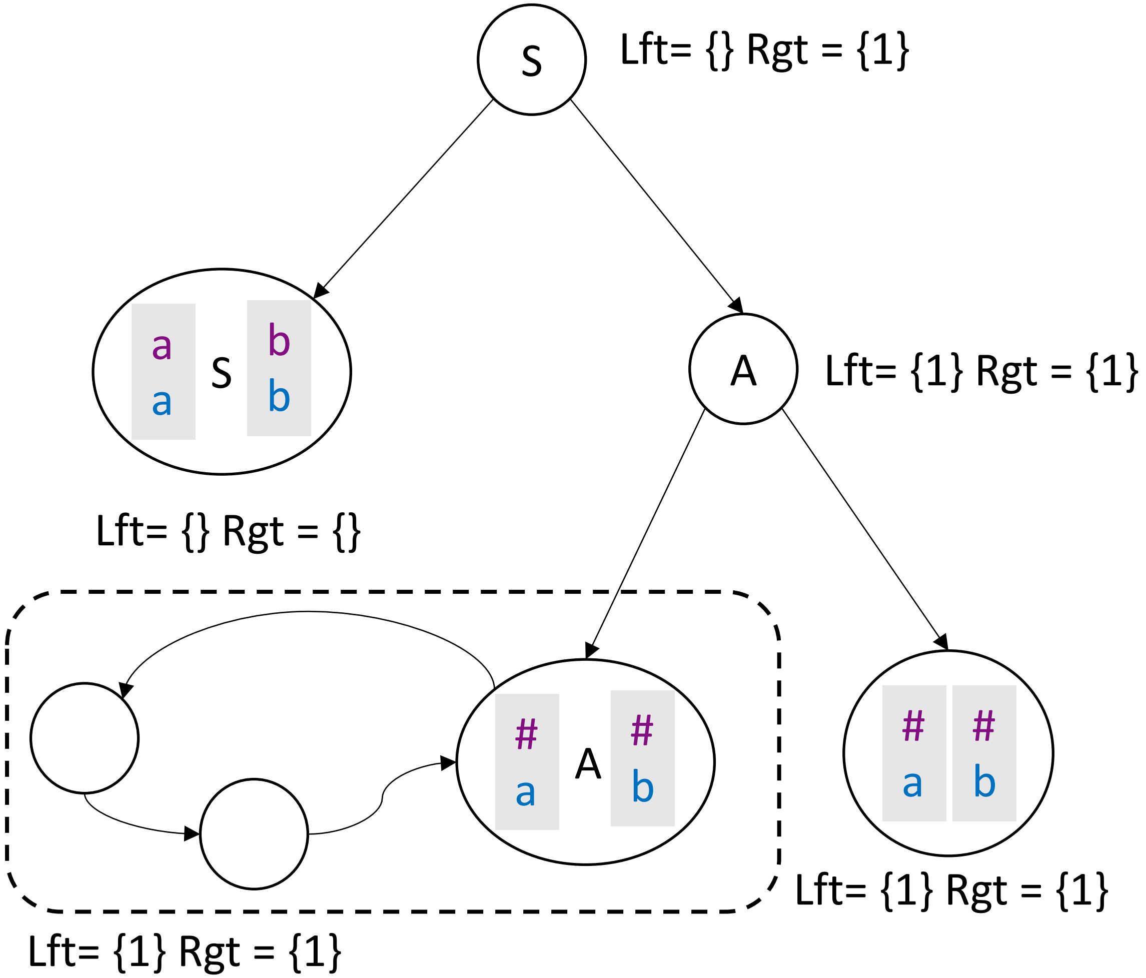


Rgt(w) ⊆ Lft(w')

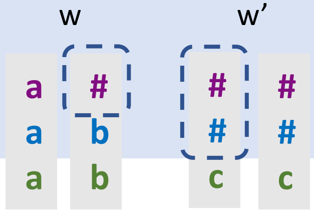
Can we define a hypergrammar that is **inherently** synchronous?

$$S \rightarrow \begin{matrix} a \\ a \end{matrix} \; S \; \begin{matrix} b \\ b \end{matrix} \;\Big|\; A$$

$$A \rightarrow \begin{matrix} \# \\ a \end{matrix} \; A \; \begin{matrix} \# \\ b \end{matrix} \;\Big|\; \begin{matrix} \# \\ a \end{matrix} \; \begin{matrix} \# \\ b \end{matrix}$$



Synchronous Context-Free Hypergrammars



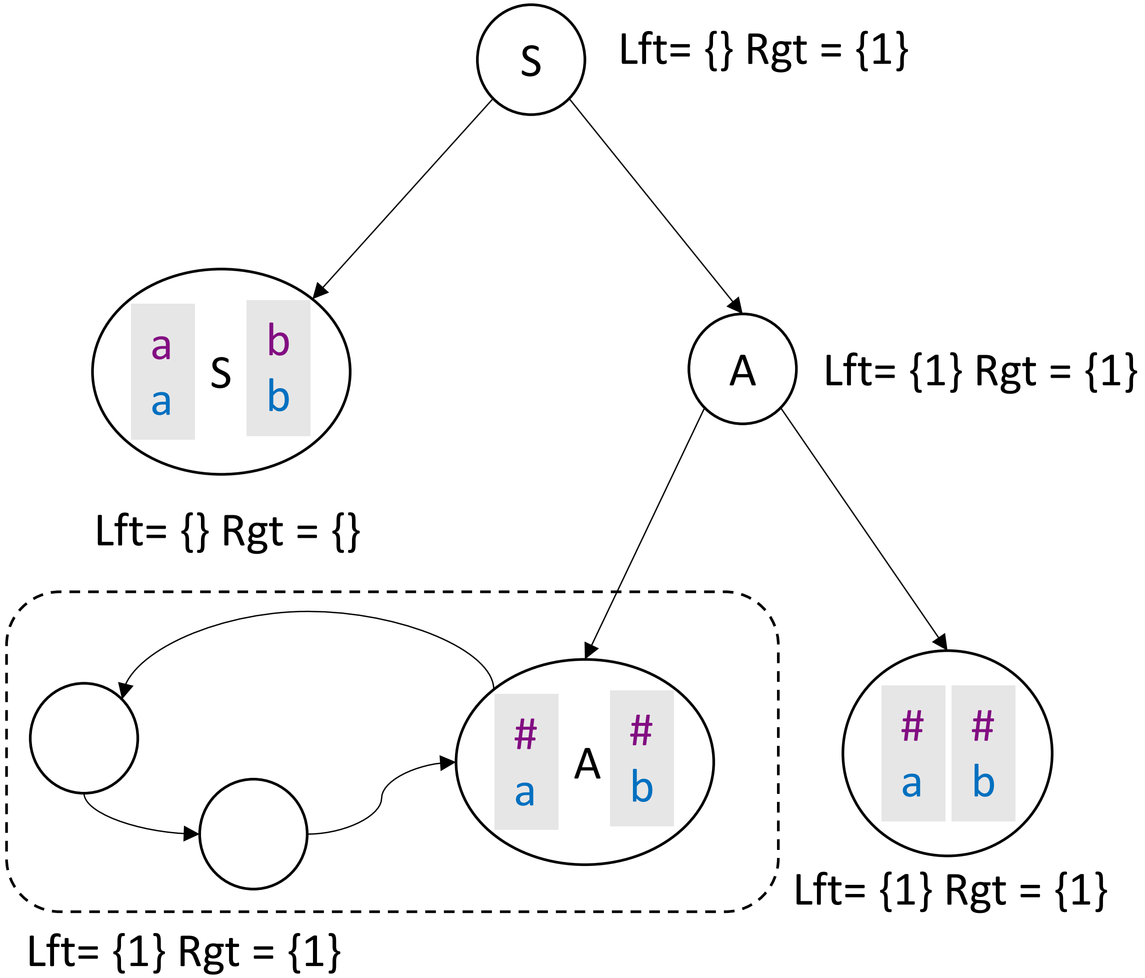
Rgt(w) $\subseteq$ Lft(w')

Can we define a hypergrammar that is **inherently** synchronous?

$$S \rightarrow \begin{matrix}a\\a\end{matrix}\ S\ \begin{matrix}b\\b\end{matrix} \mid A$$

$$A \rightarrow \begin{matrix}\#\\a\end{matrix}\ A\ \begin{matrix}\#\\b\end{matrix} \mid \begin{matrix}\#\\a\end{matrix}\ \begin{matrix}\#\\b\end{matrix}$$

$$\text{Rgt}(S) = \{1\} \not\subseteq \text{Lft}\left(\begin{matrix}b\\b\end{matrix}\right) = \{\}$$



In this talk

Finite-Word Hyperautomata

- Hyperautomata
- Realizability of $\{L\}$
  - Finite \ infinite L
  - Ordered L
  - Regular L

Context-Free Hypergrammars

- Hypergrammars √
- Synchronous hypergrammars √
- Emptiness & membership

Emptiness: $\forall^*$syncCFHG

$\forall x \forall y$ $\boxed{G}$: if L is accepted then also $L' \subset L$ $\Rightarrow$

G is not empty iff is a singleton language $\{w\} \in \mathcal{L}(G)$ $\Rightarrow$ $\begin{matrix} w \\ w \end{matrix} \in \boxed{G}$

$$\forall \boldsymbol{x} \forall \boldsymbol{y} \quad S \rightarrow \begin{matrix} a \\ a \end{matrix} \; S \; \begin{matrix} b \\ b \end{matrix} \;\Big|\; A$$

$$A \rightarrow \cancel{\begin{matrix} \# \\ a \end{matrix} \; A} \quad \begin{matrix} \# \\ b \end{matrix} \;\Big|\; \cancel{\begin{matrix} \# \\ a \end{matrix} \; \begin{matrix} \# \\ b \end{matrix}}$$

Check emptiness of the underlying grammar

Same proof also works for $\exists\forall^*$

Emptiness: Undecidable for $\forall^*$CFHG

Reduction from Post correspondence problem

1	2	3
a	ab	bba
baa	aa	bb

$\Longrightarrow$

a##	ab	bba
baa	aa	bb#

3 2 3 1

bba ab bba a

bb aa bb baa

$\forall x \forall y$ S $\rightarrow$

a## / baa	S	|	a## / baa	|
ab / aa	S	|	ab / aa	|
bba / bb#	S	|	bba / bb#	

$x \leftarrow$ bba ab bba a

$y \leftarrow$ bb aa bb baa

Same proof also works for $\exists\forall^*$

Regular Membership

EXPTIME for $\exists^*$(sync)CFHG

$$L \in \exists x_1 \cdots \exists x_k \; \boxed{\mathrm{G}}\,?$$

$$\boxed{\mathrm{G}} \cap$$

$$A_L \quad \times \quad \cdots \quad \times \quad A_L$$

Undecidable for $\forall^*$(sync)CFHG

Reduction from the universality problem of CFG

$\boxed{\mathrm{G}}$ is universal $\Leftrightarrow$

$\Sigma^* \subseteq \boxed{\mathrm{G}} \Leftrightarrow$

$\Sigma^* \in \forall \boldsymbol{x} \; \boxed{\mathrm{G}}$

Questions?

Hyperautomata

- Realizability of $\{L\}$ for
  - Finite \ infinite L
  - Ordered L
  - Regular L

Hypergrammars

- Synchronous hypergrammars
- Emptiness $\forall^*$, $\exists\forall^*$
  [in the paper: $\exists^*$, $\exists^*\forall^*$]
- Regular membership $\exists^*$, $\forall^*$
  [in the paper: finite membership]