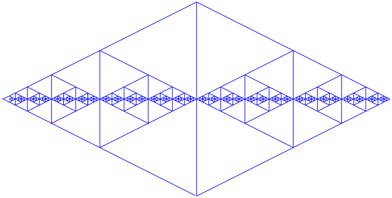
Regular Automata



Didier Caucal

CNRS / LIGM

University Paris - Est

France

The first level of the pushdown hierarchy

definition, characterizations, properties

The first level of the pushdown hierarchy

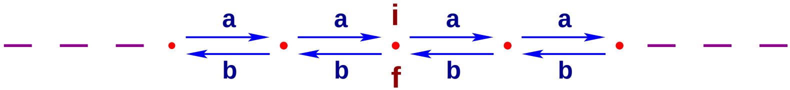
definition, characterizations, properties

An application: Eilenberg's recognizability

boolean algebras of context-free languages

Automaton $G \subseteq V \times L \times V \cup C \times V$

- oriented labelled graph
- finite set L of labels
- finite set C of colours
- $i, f \in C$ for the initial / final vertices

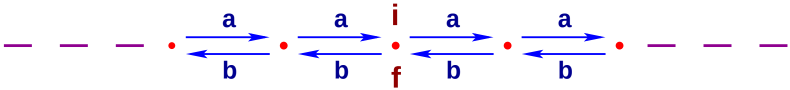


recognizing $L(G) = \{ u \in \{a,b\}^* \mid |u|_a = |u|_b \}$

Syntactical typed recursion

Automaton $G \subseteq V \times L \times V \cup C \times V$

- oriented labelled graph
- finite set L of labels
- finite set C of colours
- $i, f \in C$ for the initial / final vertices

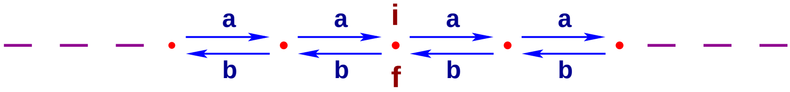


recognizing $L(G) = \{ u \in \{a,b\}^* \mid |u|_a = |u|_b \}$

Syntactical typed recursion

Automaton $G \subseteq V \times L \times V \cup C \times V$

- oriented labelled graph
- finite set L of labels
- finite set C of colours
- $i, f \in C$ for the initial / final vertices

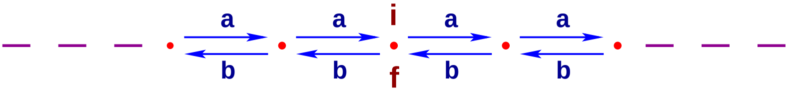


recognizing $L(G) = \{ u \in \{a,b\}^* \mid |u|_a = |u|_b \}$

Syntactical typed recursion

Automaton $G \subseteq V \times L \times V \cup C \times V$

- oriented labelled graph
- finite set L of labels
- finite set C of colours
- $i, f \in C$ for the initial / final vertices

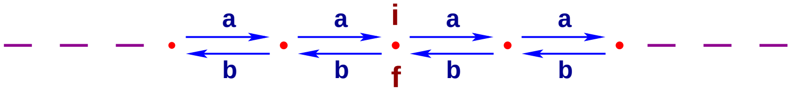


recognizing $L(G) = \{ u \in \{a,b\}^* \mid |u|_a = |u|_b \}$

Syntactical typed recursion

Automaton $G \subseteq V \times L \times V \cup C \times V$

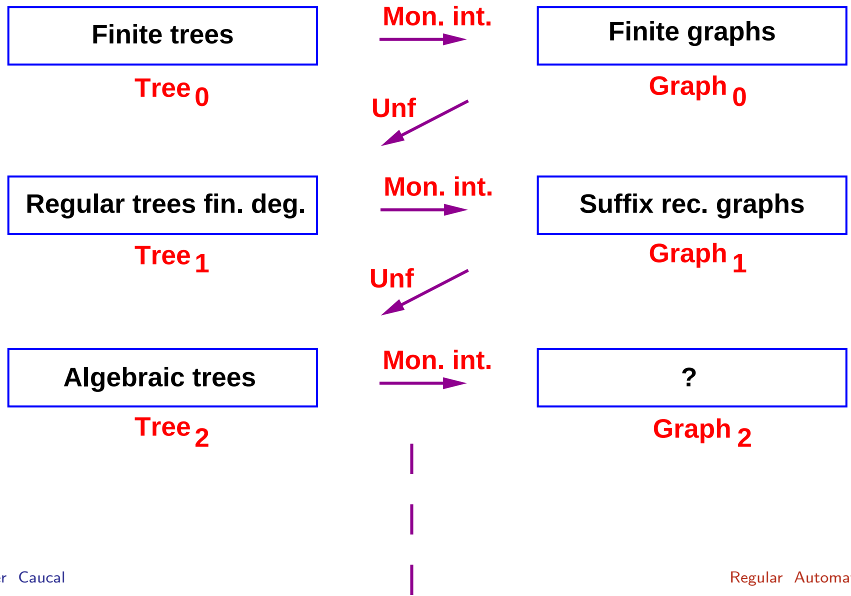
- oriented labelled graph
- finite set L of labels
- finite set C of colours
- $i, f \in C$ for the initial / final vertices



recognizing $L(G) = \{ u \in \{a,b\}^* \mid |u|_a = |u|_b \}$

Syntactical typed recursion

The pushdown hierarchy



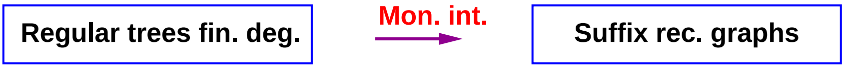
Property

The suffix recognizable graphs have

a decidable monadic theory

are preserved by monadic interpretation and
synchronization product with finite automata

recognize the context-free languages



Property

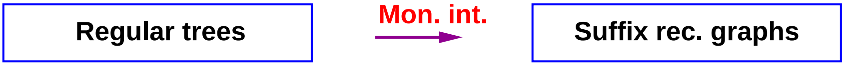
The suffix recognizable graphs have

a decidable monadic theory

are preserved by monadic interpretation and

synchronization product with finite automata

recognize the context-free languages



Property

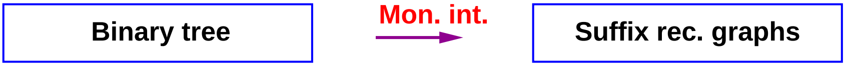
The suffix recognizable graphs have

a decidable monadic theory

are preserved by monadic interpretation and

synchronization product with finite automata

recognize the context-free languages



Property

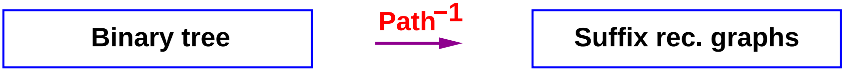
The suffix recognizable graphs have

a decidable monadic theory

are preserved by monadic interpretation and

synchronization product with finite automata

recognize the context-free languages



Property

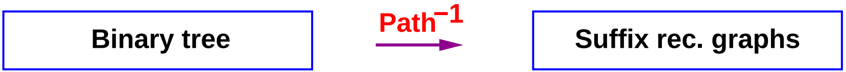
The suffix recognizable graphs have

a decidable monadic theory

are preserved by monadic interpretation and

synchronization product with finite automata

recognize the context-free languages



Inverse path function ?

Path functions

set Exp of path expressions

$$L \cup C \cup \{\varepsilon\} \subseteq Exp$$

for any u,v $\in$ Exp

$$u^{-1} , u \cdot v , u^{+} , \neg u , u \vee v , u \wedge v \in Exp$$

path $s \xrightarrow[G]{u} t$ for $u \in Exp$

$$s \xrightarrow{a} t \quad \text{for} \quad (s,a,t) \in G$$

$$s \xrightarrow{c} t \quad \text{for} \quad s = t \ \wedge \ (c,s) \in G$$

$$s \xrightarrow{\varepsilon} t \quad \text{for} \quad s = t$$

$$s \xrightarrow{u^{-1}} t \quad \text{for} \quad t \xrightarrow{u} s$$

$$s \xrightarrow{u \cdot v} t \quad \text{for} \quad \exists\, r\, (s \xrightarrow{u} r \ \wedge \ r \xrightarrow{v} t)$$

$$s \xrightarrow{u^{+}} t \quad \text{for} \quad s\, (\xrightarrow{u})^{+}\ t$$

$$s \xrightarrow{\neg\, u} t \quad \text{for} \quad \neg\, (s \xrightarrow{u} t)$$

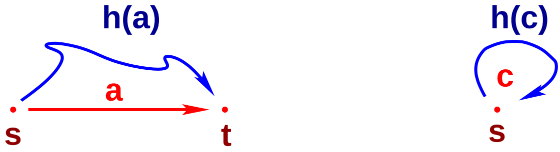
$$s \xrightarrow{u \vee v} t \quad \text{for} \quad s \xrightarrow{u} t \ \vee \ s \xrightarrow{v} t$$

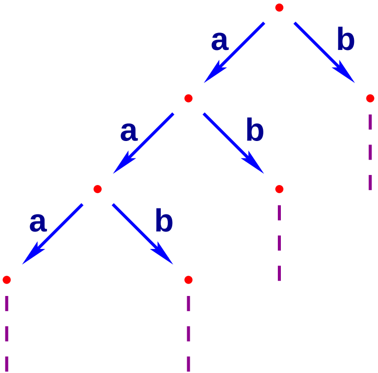
For instance $s \xrightarrow{\varepsilon\ \wedge\ a \cdot a^{-1}} t$ means that $s = t \wedge s \xrightarrow{a}$

Path function $h : L \cup C \longrightarrow Exp$

applied by inverse on a graph G

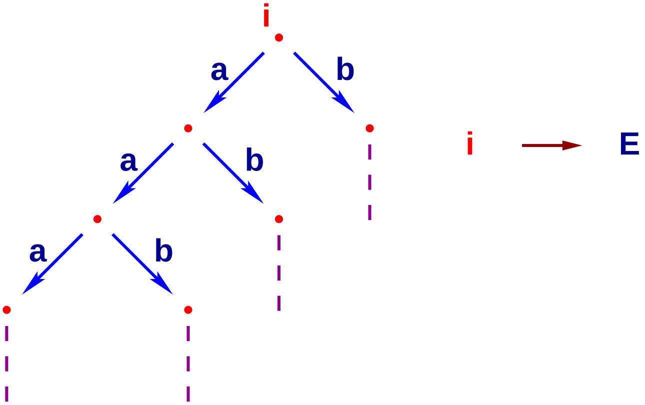
$$h^{-1}(G) = \{ (s,a,t) \mid s \xrightarrow[G]{h(a)} t \} \cup \{ (c,s) \mid s \xrightarrow[G]{h(c)} s \}$$





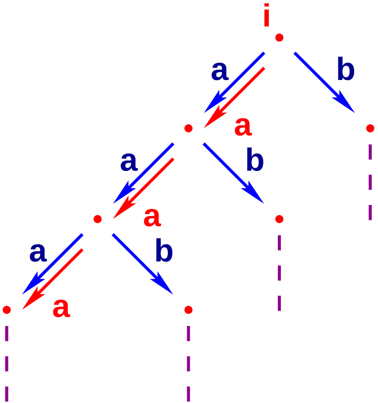
$$E = \varepsilon \wedge \neg(a^{-1}a \vee b^{-1}b)$$

$$F = \varepsilon \wedge (a^{-1})^* E\, a^*$$



$$E = \varepsilon \wedge \neg(a^{-1}a \vee b^{-1}b)$$

$$F = \varepsilon \wedge (a^{-1})^* E a^*$$

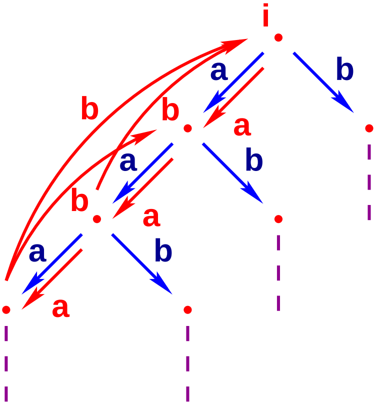


i $\longrightarrow$ E

a $\longrightarrow$ F a

$$E = \varepsilon \wedge \neg(a^{-1}a \vee b^{-1}b)$$

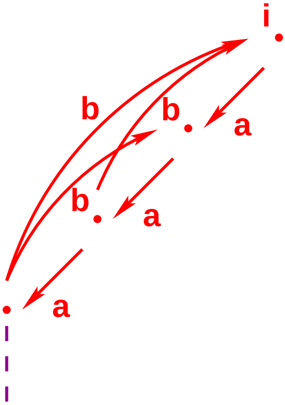
$$F = \varepsilon \wedge (a^{-1})^* E a^*$$



i	$\longrightarrow$	E
a	$\longrightarrow$	F a
b	$\longrightarrow$	$F\,(a^{-1})^{+}a^{-1}$

$$E \;=\; \varepsilon \;\wedge\; \neg(a^{-1}a \,\vee\, b^{-1}b)$$

$$F \;=\; \varepsilon \;\wedge\; (a^{-1})^{*}\,E\,a^{*}$$



$$E = \varepsilon \wedge \neg(a^{-1}a \vee b^{-1}b)$$

$$F = \varepsilon \wedge (a^{-1})^* E\, a^*$$

Particular path functions

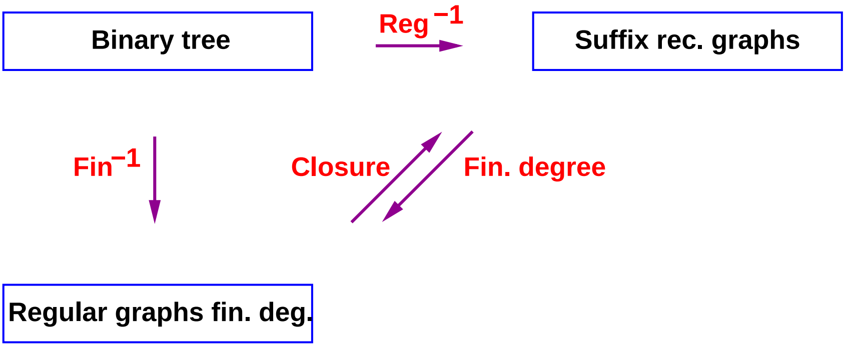
Regular substitutions

$$\text{Exp} = \text{set of regular expressions}$$

a^{-1} for $a \in L$; $u \cdot v$, u^+ , $u \vee v$ for $u,v \in \text{Exp}$

Finite substitutions

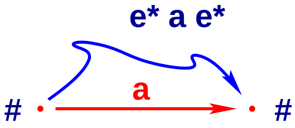
a^{-1} for $a \in L$; $u \cdot v$, $u \vee v$ for $u,v \in \text{Exp}$

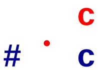

Binary tree
Reg⁻¹
Suffix rec. graphs
Fin⁻¹
Closure
Fin. degree
Regular graphs fin. deg.

Closure w.r.t. $e \in L$ and $\# \in C$

$$a \longrightarrow \# e^* a e^* \# \quad \text{for any } a \in L - \{e\}$$

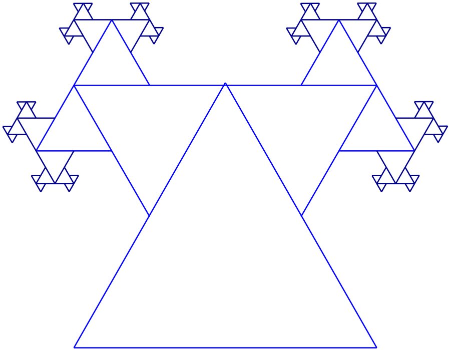
$$c \longrightarrow \# c \quad \text{for any } c \in C - \{\#\}$$





Regular automata

of finite degree



Regular automata of finite degree

- Decomposition by distance   Muller, Schupp 85

- Decompositions and graph grammars

  Courcelle 89, Caucal 92

- Suffix rewriting systems   Büchi 64

Regular automata of finite degree

- Decomposition by distance Muller, Schupp 85
- Decompositions and graph grammars

  Courcelle 89, Caucal 92
- Suffix rewriting systems Büchi 64

Regular automata of finite degree

- Decomposition by distance Muller, Schupp 85

- Decompositions and graph grammars

  Courcelle 89, Caucal 92

- Suffix rewriting systems Büchi 64

Regular automata of finite degree

- Decomposition by distance   Muller, Schupp 85
- Decompositions and graph grammars
  Courcelle 89, Caucal 92
- Suffix rewriting systems   Büchi 64

Pushdown automaton

- Stack letters $P = \{A, B, \ldots\}$
- States $Q = \{p, q, \ldots\}$
- Terminals $T = \{a, b, \ldots\}$
- Axiom $A\,p$
- Rules

$$R \left|\ \begin{array}{lllllll} A\,p & \xrightarrow{a} & A\,B\,p & \quad & B\,q & \xrightarrow{b} & q \\ B\,p & \xrightarrow{a} & B\,B\,p & \quad & B\,p & \xrightarrow{b} & q \end{array}\right.$$

Pushdown automaton

- Stack letters $P = \{A, B, \ldots\}$
- States $Q = \{p, q, \ldots\}$
- Terminals $T = \{a, b, \ldots\}$
- Axiom $A\,p$
- Rules

$$R \left| \begin{array}{lll} A\,p \xrightarrow{a} A\,B\,p & \qquad & B\,q \xrightarrow{b} q \\ B\,p \xrightarrow{a} B\,B\,p & \qquad & B\,p \xrightarrow{b} q \end{array} \right.$$

Pushdown automaton

- Stack letters $P = \{A, B, \ldots\}$
- States $Q = \{p, q, \ldots\}$
- Terminals $T = \{a, b, \ldots\}$
- Axiom $A\,p$
- Rules

$$R \left| \begin{array}{lllllll} A\,p & \xrightarrow{a} & A\,B\,p & \qquad & B\,q & \xrightarrow{b} & q \\ B\,p & \xrightarrow{a} & B\,B\,p & \qquad & B\,p & \xrightarrow{b} & q \end{array} \right.$$

Pushdown automaton

- Stack letters $P = \{A, B, ...\}$
- States $Q = \{p, q, ...\}$
- Terminals $T = \{a, b, ...\}$
- Axiom $A\,p$
- Rules

$$R \left| \begin{array}{lllllll} A\,p & \xrightarrow{a} & A\,B\,p & \quad & B\,q & \xrightarrow{b} & q \\ B\,p & \xrightarrow{a} & B\,B\,p & \quad & B\,p & \xrightarrow{b} & q \end{array} \right.$$

Pushdown automaton

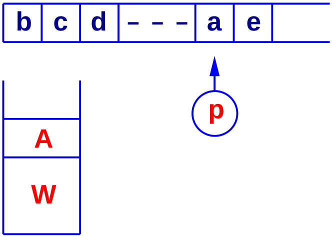
- Stack letters $P = \{A, B, \ldots\}$
- States $Q = \{p, q, \ldots\}$
- Terminals $T = \{a, b, \ldots\}$
- Axiom $A\,p$
- Rules

$$R \left| \begin{array}{lllllll} A\,p & \xrightarrow{a} & A\,B\,p & \qquad & B\,q & \xrightarrow{b} & q \\ B\,p & \xrightarrow{a} & B\,B\,p & \qquad & B\,p & \xrightarrow{b} & q \end{array} \right.$$

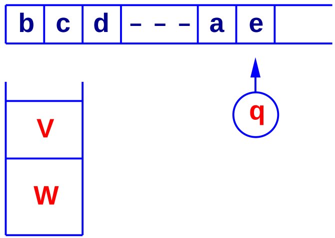
Pushdown automaton

- Stack letters $P = \{A, B, \ldots\}$
- States $Q = \{p, q, \ldots\}$
- Terminals $T = \{a, b, \ldots\}$
- Axiom $A\,p$
- Rules

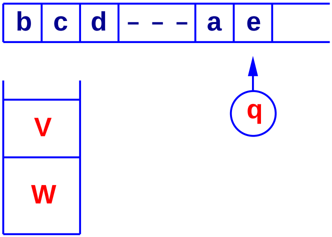
$$R \left| \begin{array}{lllllll} A\,p & \xrightarrow{a} & A\,B\,p & \quad & B\,q & \xrightarrow{b} & q \\ B\,p & \xrightarrow{a} & B\,B\,p & \quad & B\,p & \xrightarrow{b} & q \end{array} \right.$$



Applied rule $A\,p \xrightarrow{a} V\,q$



Applied rule $A\,p \xrightarrow{a} V\,q$



Applied rule $A\,p \xrightarrow{a} V\,q$

Transition $W\,A\,p \xrightarrow{a} W\,V\,q$

Pushdown graph $P^*.R$

$$\{\ WAp \xrightarrow{a} WVq \mid W \in P^* \wedge (Ap \xrightarrow{a} Vq) \in R\ \}$$

Accessible pushdown graph from the axiom

$$R \left| \begin{array}{lllllll} A\,p & \xrightarrow{a} & A\,B\,p & \quad & B\,q & \xrightarrow{b} & q \\ B\,p & \xrightarrow{a} & B\,B\,p & \quad & B\,p & \xrightarrow{b} & q \end{array} \right.$$

Pushdown graph $P^*.R$

$$\{ \; WAp \xrightarrow{a} WVq \mid W \in P^* \wedge (Ap \xrightarrow{a} Vq \,) \in R \; \}$$

Accessible pushdown graph from the axiom

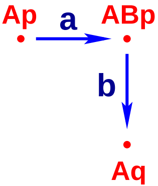
$$R \left| \begin{array}{llllll} A\,p & \xrightarrow{a} & A\,B\,p & B\,q & \xrightarrow{b} & q \\ B\,p & \xrightarrow{a} & B\,B\,p & B\,p & \xrightarrow{b} & q \end{array} \right.$$

Pushdown graph $P^*.R$

$$\{ WAp \xrightarrow{a} WVq \mid W \in P^* \wedge (Ap \xrightarrow{a} Vq) \in R \}$$

Accessible pushdown graph from the axiom

$$R \left| \begin{array}{lllllll} A\,p & \xrightarrow{a} & A\,B\,p & \quad & B\,q & \xrightarrow{b} & q \\ B\,p & \xrightarrow{a} & B\,B\,p & \quad & B\,p & \xrightarrow{b} & q \end{array} \right.$$

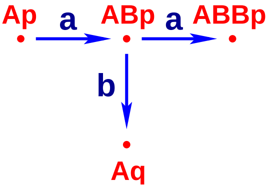


Pushdown graph $P^*.R$

$$\{\ WAp \xrightarrow{a} WVq \mid W \in P^* \wedge (Ap \xrightarrow{a} Vq\ ) \in R\ \}$$

Accessible pushdown graph from the axiom

$$R \left| \begin{array}{lllllll} A\,p & \xrightarrow{a} & A\,B\,p & \quad & B\,q & \xrightarrow{b} & q \\ B\,p & \xrightarrow{a} & B\,B\,p & \quad & B\,p & \xrightarrow{b} & q \end{array} \right.$$

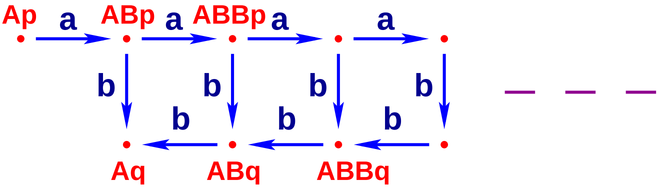


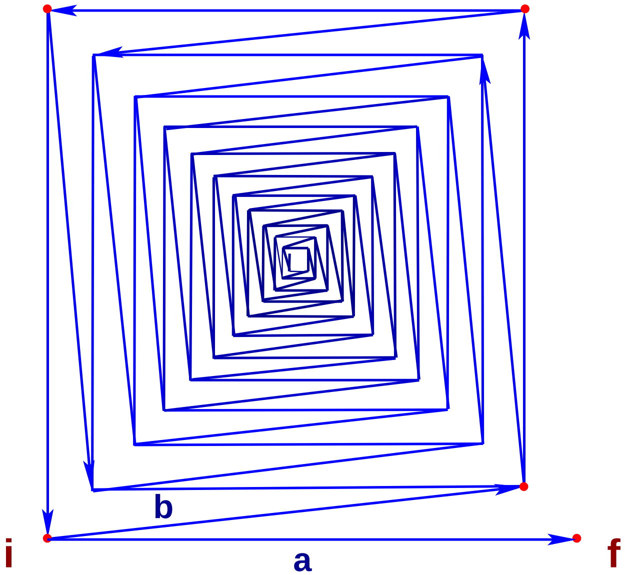
Pushdown graph P*.R

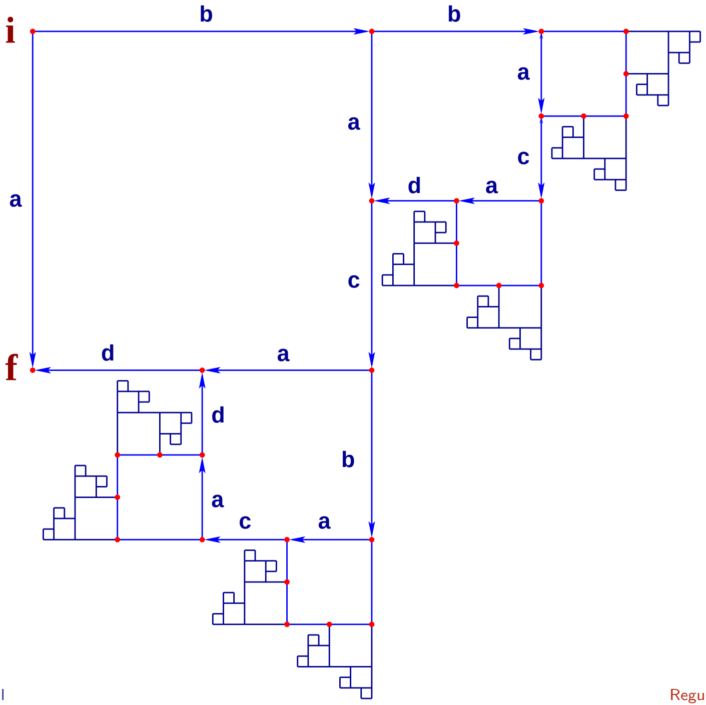
$$\{ WAp \xrightarrow{a} WVq \mid W \in P^* \wedge (Ap \xrightarrow{a} Vq) \in R \}$$

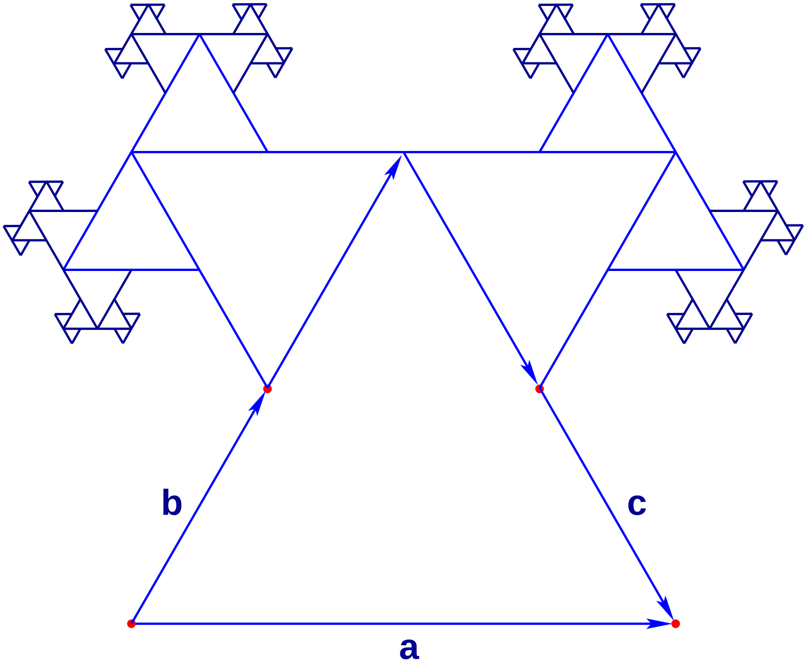
Accessible pushdown graph from the axiom

$$R \left| \begin{array}{llllll} A\,p & \xrightarrow{a} & A\,B\,p & B\,q & \xrightarrow{b} & q \\ B\,p & \xrightarrow{a} & B\,B\,p & B\,p & \xrightarrow{b} & q \end{array} \right.$$




b
i
a
f


i
b
b
a
a
a
c
d
a
c
f
d
a
d
b
a
c
a


b
c
a

Regular vertex set: $P^*(\,\mathrm{Dom}(R) \cup \mathrm{Im}(R)\,)$

By accessibility from an axiom c :

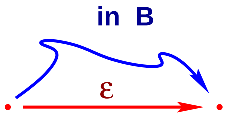
for languages L and B, the reduction of L by B is

$$L{\downarrow}B \;=\; L \,\cup\, (\{\, uw \mid \exists\, v \in B\ (uvw \in L)\ \}){\downarrow}B$$

Lemma Benois 69

$$L \ \textit{regular} \quad\Longrightarrow\quad L{\downarrow}B \ \textit{regular}$$

saturation method:



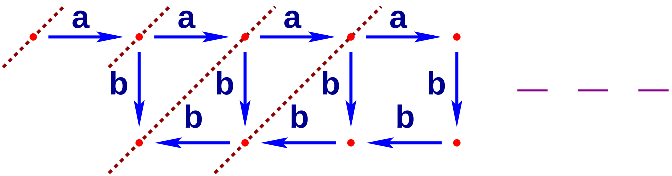
$$B \;=\; \{\, x\overleftarrow{x} \mid x \in P \cup Q \,\}$$

$$[c\,.\,(\{\, \overleftarrow{p}\ \overleftarrow{A}\ Uq \mid Ap \longrightarrow Uq \,\})^*]{\downarrow}B \;\cap\; P^*Q$$

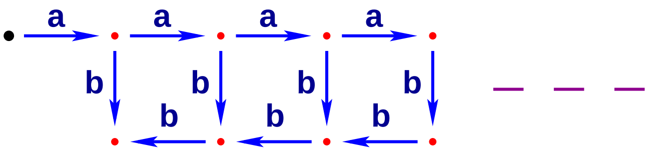
Decomposition by distance

Theorem Muller Schupp 1985

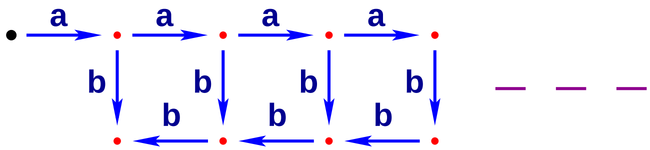
The accessible pushdown graphs are the rooted graphs of finite degree with a finite decomposition by distance



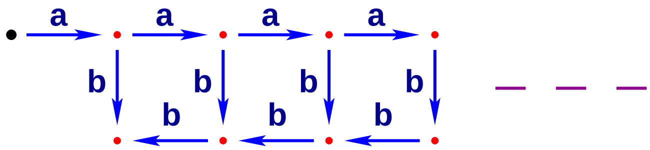
finite number of connected components

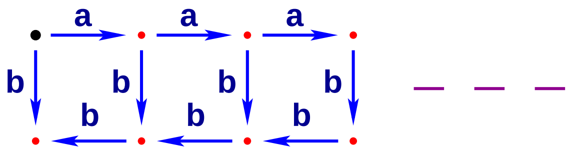


Connected components

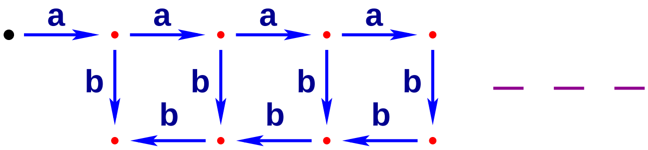


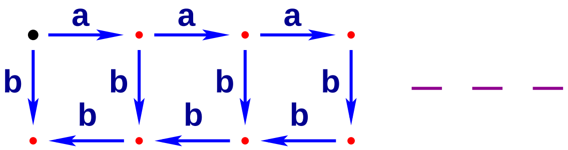
Connected components



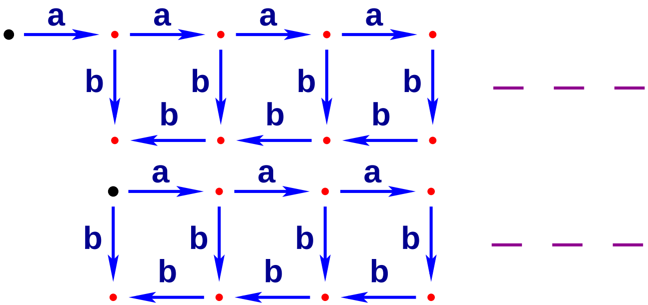


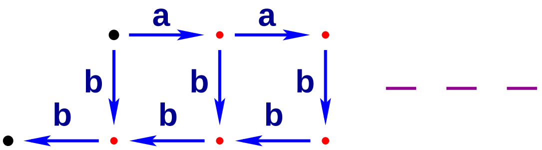
Connected components



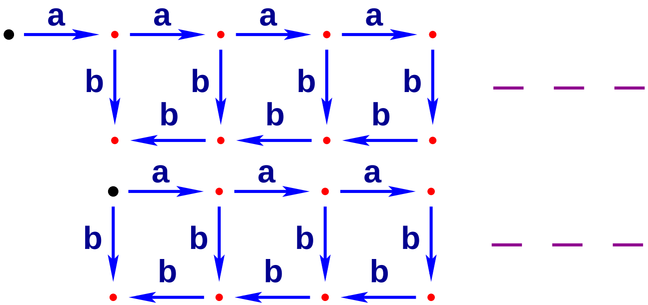


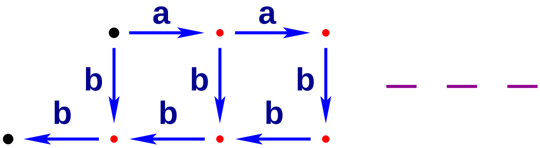
Connected components



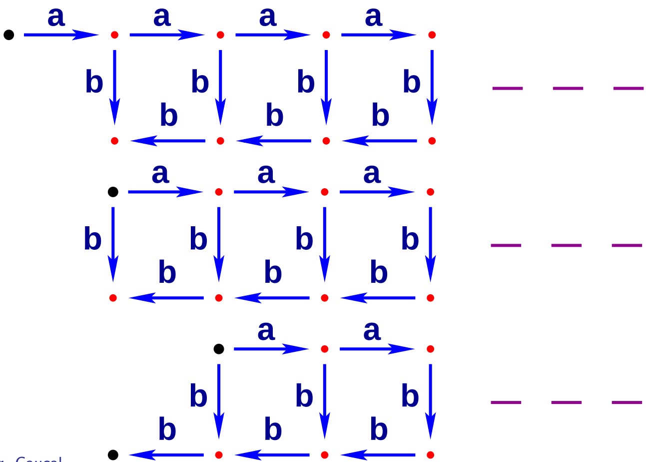


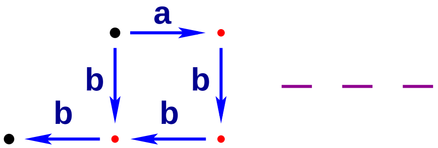
Connected components



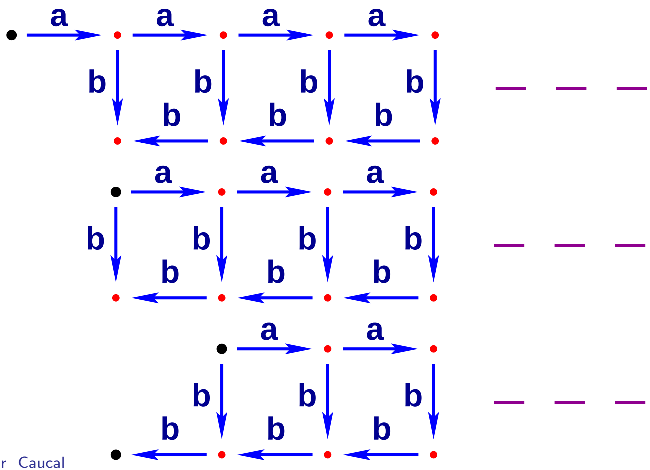


Connected components





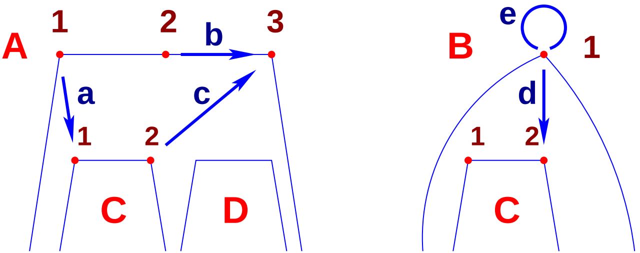
Connected components



$\Longleftarrow$: Finite decomposition by distance

A pushdown letter for each connected component

The vertices of each frontier are numbered by 1,2,...

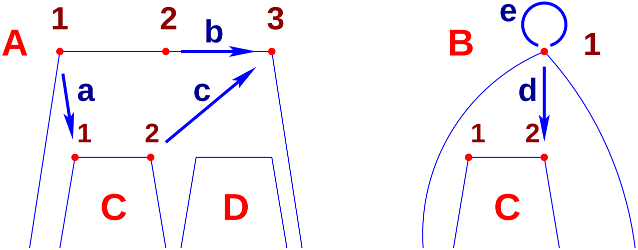


$A\,1 \xrightarrow{a} A\,C\,1$	$B\,1 \xrightarrow{d} B\,C\,2$
$A\,2 \xrightarrow{b} A\,3$	$B\,1 \xrightarrow{e} B\,1$
$A\,C\,2 \xrightarrow{c} A\,3$	$B\,C\,1 \xrightarrow{f} B\,1$

$\Longleftarrow$: Finite decomposition by distance

A pushdown letter for each connected component

The vertices of each frontier are numbered by 1,2,...



$$A\,1 \xrightarrow{a} A\,C\,1 \qquad B\,1 \xrightarrow{d} B\,C\,2$$

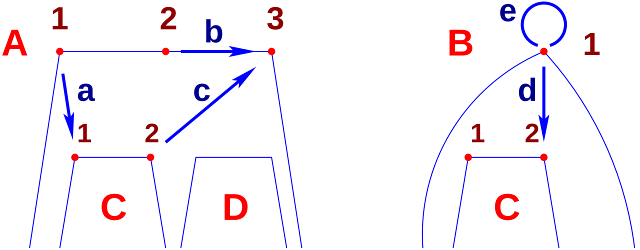
$$A\,2 \xrightarrow{b} A\,3 \qquad B\,1 \xrightarrow{e} B\,1$$

$$_{A}C\,2 \xrightarrow{c} 3 \qquad _{B}C\,1 \xrightarrow{f} 1$$

$\Longleftarrow$: Finite decomposition by distance

A pushdown letter for each connected component

The vertices of each frontier are numbered by 1,2,...



$$A\,1 \xrightarrow{a} A\,{}_{A}C\,1 \qquad B\,1 \xrightarrow{d} B\,{}_{B}C\,2$$

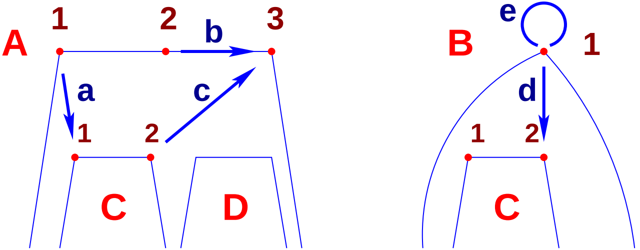
$$A\,2 \xrightarrow{b} A\,3 \qquad B\,1 \xrightarrow{e} B\,1$$

$$_{A}C\,2 \xrightarrow{c} 3 \qquad {}_{B}C\,1 \xrightarrow{f} 1$$

$\Longleftarrow$: Finite decomposition by distance

A pushdown letter for each connected component

The vertices of each frontier are numbered by 1,2,...



$$ {}_{X}A\,1 \xrightarrow{a} {}_{X}A\,{}_{A}C\,1 \qquad {}_{X}B\,1 \xrightarrow{d} {}_{X}B\,{}_{B}C\,2 $$

$$ {}_{X}A\,2 \xrightarrow{b} {}_{X}A\,3 \qquad {}_{X}B\,1 \xrightarrow{e} {}_{X}B\,1 $$

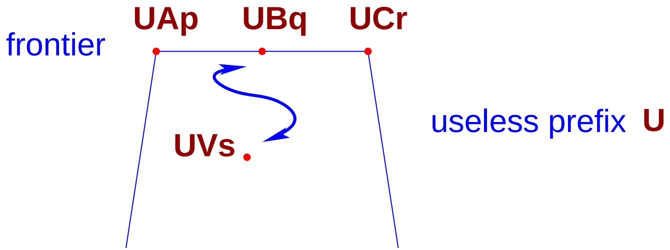
$$ {}_{A}C\,2 \xrightarrow{c} 3 \qquad {}_{B}C\,1 \xrightarrow{f} 1 $$

$\Longrightarrow$: Pushdown automaton (with an axiom)

Maximal length of the r.h.s. $m \le 3$

$A\,p \xrightarrow{a} q \qquad A\,p \xrightarrow{a} B\,q \qquad A\,p \xrightarrow{a} C\,B\,q$

Finite decomposition by length



Finite number of possible frontiers

$m \ge 3$: frontier depends only on suffixes $\le$ m-1

Finite decomposition by distance

Decomposition by distance

normal form for connected graphs of finite degree

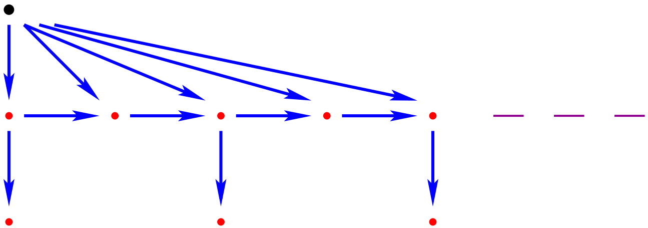
not appropriate for conn. graphs of infinite degree

for any $\mathsf{P} \subseteq \mathbb{N}$, the graph $\overrightarrow{\mathrm{P}}$

$\{\, \top \to \mathsf{n} \mid \mathsf{n} \geq 0 \,\} \cup \{\, \mathsf{n} \to \mathsf{n}{+}1 \mid \mathsf{n} \geq 0 \,\} \cup \{\, \mathsf{n} \to \text{-}\mathsf{n}\text{-}1 \mid \mathsf{n} \in \mathsf{P} \,\}$

is finitely decomposable by distance

$\overrightarrow{2\mathbb{N}}$:



Decomposition by distance

normal form for connected graphs of finite degree

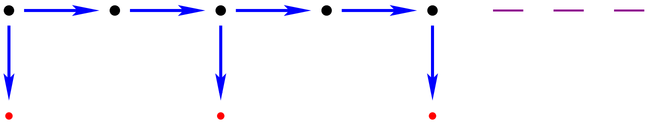
not appropriate for conn. graphs of infinite degree

for any $P \subseteq \mathbb{N}$, the graph $\overrightarrow{P}$

$\{\top \to n \mid n \geq 0\} \cup \{n \to n{+}1 \mid n \geq 0\} \cup \{n \to \text{-}n\text{-}1 \mid n \in P\}$

is finitely decomposable by distance

$\overrightarrow{2\mathbb{N}}$:



Decomposition by distance

normal form for connected graphs of finite degree

not appropriate for conn. graphs of infinite degree

for any $\mathsf{P} \subseteq \mathbb{N}$, the graph $\overrightarrow{\mathrm{P}}$

$\{\, \top \to \mathsf{n} \mid \mathsf{n} \geq 0 \,\} \cup \{\, \mathsf{n} \to \mathsf{n}{+}1 \mid \mathsf{n} \geq 0 \,\} \cup \{\, \mathsf{n} \to \text{-}\mathsf{n}\text{-}1 \mid \mathsf{n} \in \mathsf{P} \,\}$

is finitely decomposable by distance

$\overrightarrow{2\mathbb{N}}$:

Let G be any connected graph of finite degree

Let P,Q be any finite non empty vertex subsets

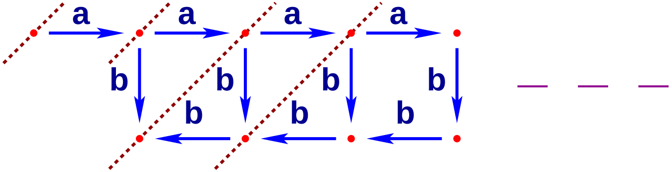
Property 1992

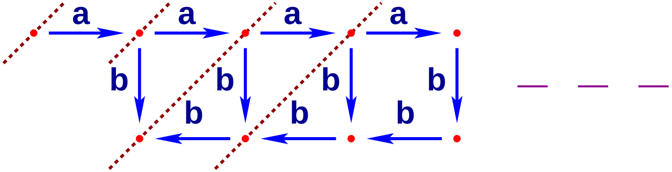
If G *has a finite decomposition* (?) *then*
G *has a finite decomposition by distance from* P

Corollary

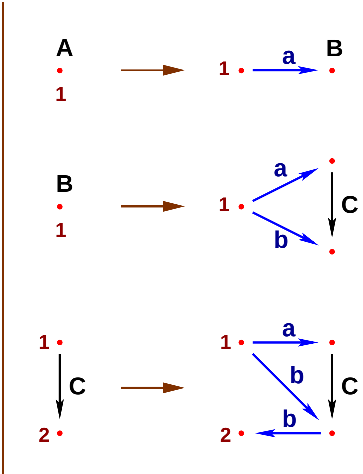
If G *has a finite dec. by distance from* P *then*
G *has a finite dec. by distance from* Q

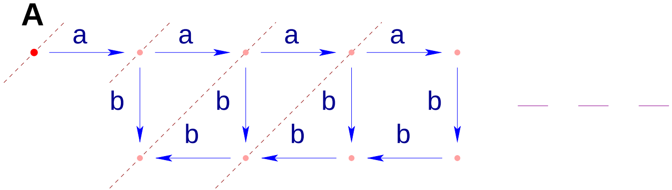
Finite decomposition ?


a
a
a
a
b
b
b
b
b
b
b

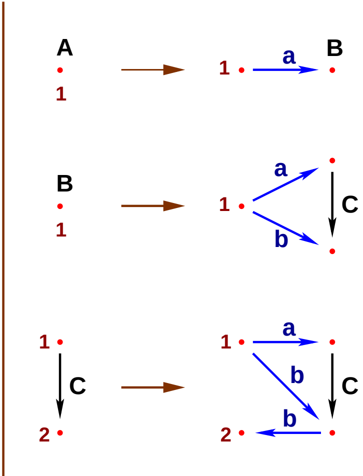


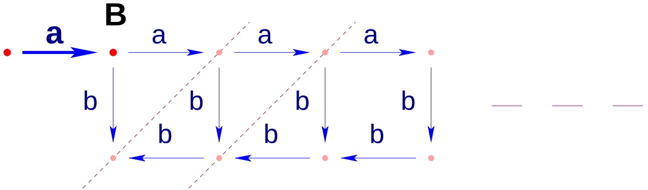
Deterministic graph grammar



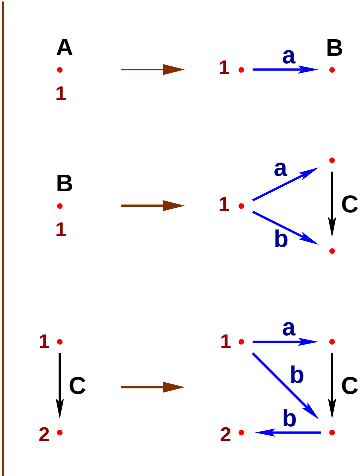


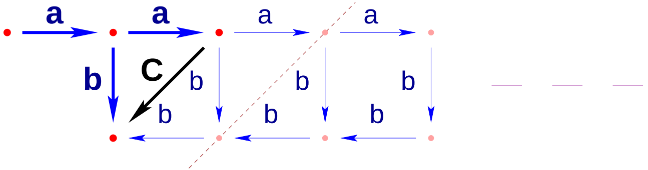
Deterministic graph grammar



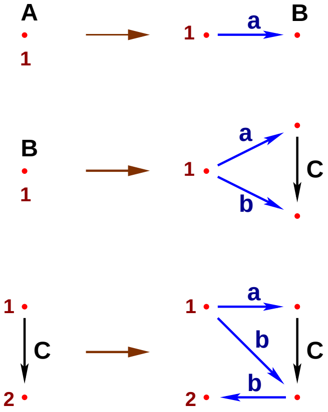


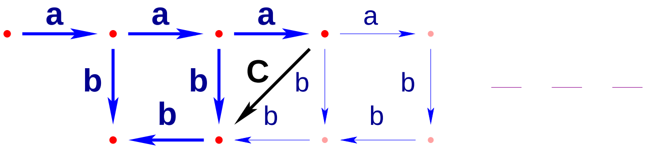
Deterministic graph grammar



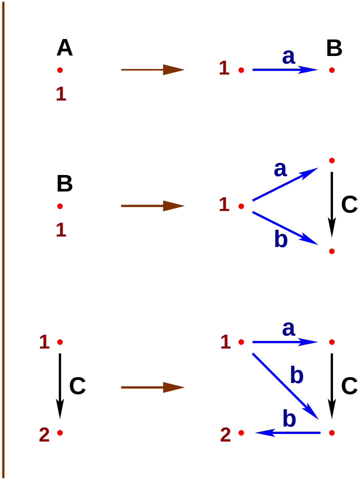


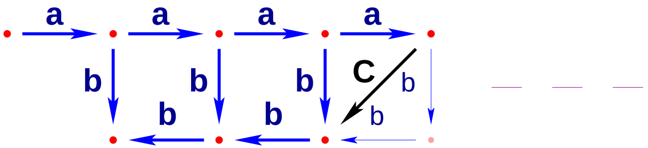
Deterministic graph grammar



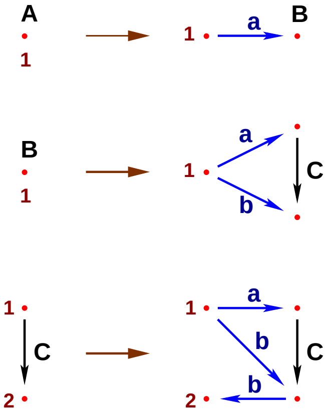


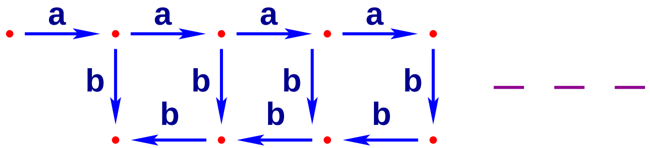
Deterministic graph grammar



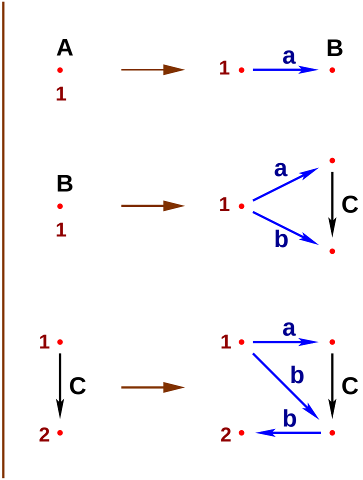


Deterministic graph grammar

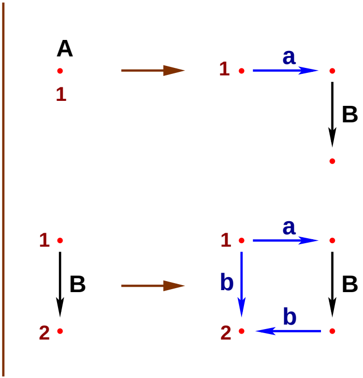




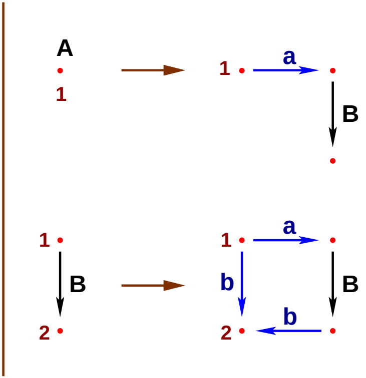
Deterministic graph grammar



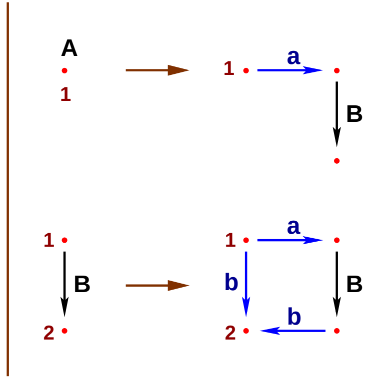
Another deterministic graph grammar

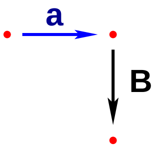


Another deterministic graph grammar

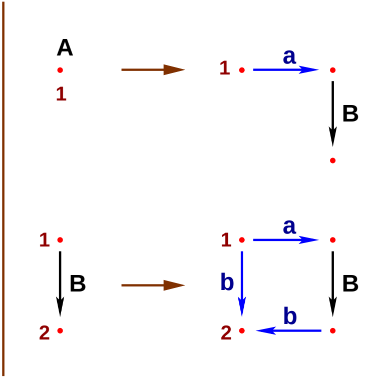


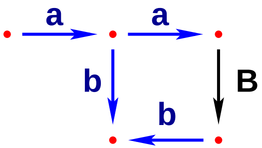
Another deterministic graph grammar



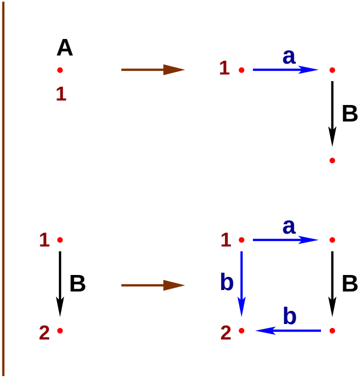


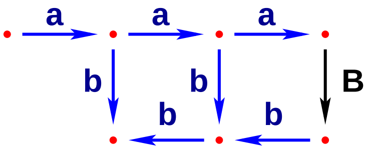
Another deterministic graph grammar



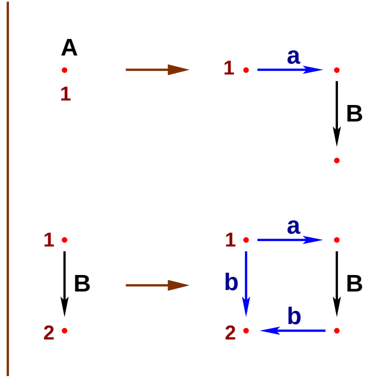


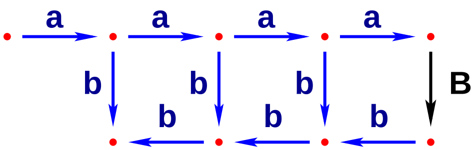
Another deterministic graph grammar



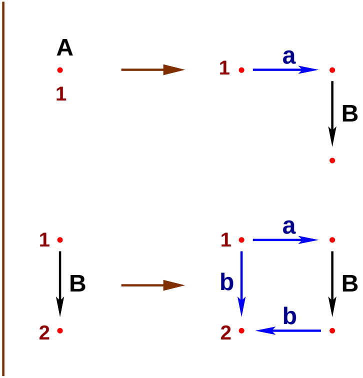


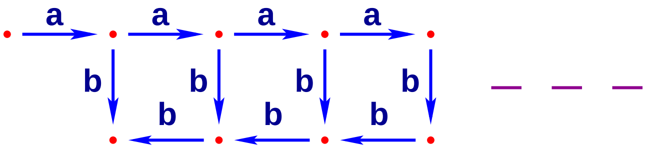
Another deterministic graph grammar



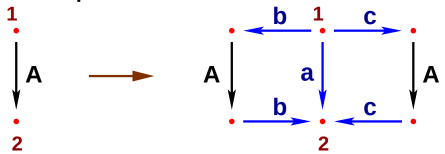


Another deterministic graph grammar

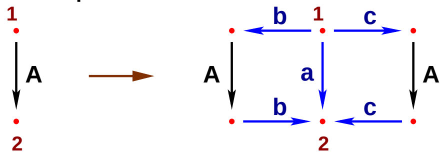


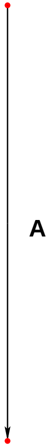


Another example

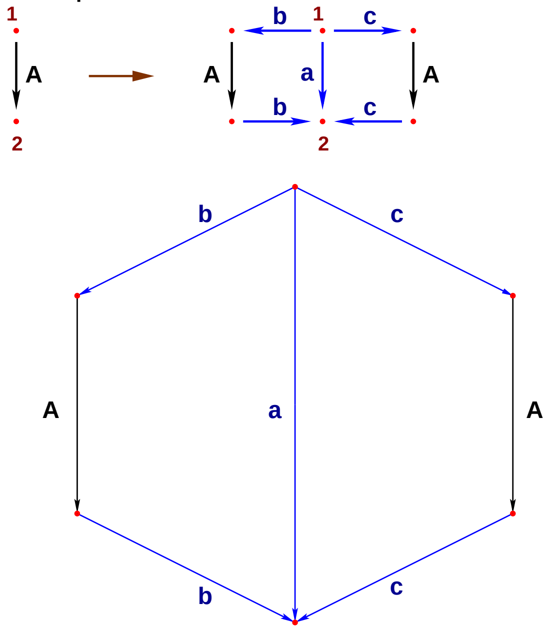


Another example

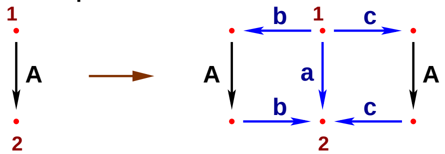


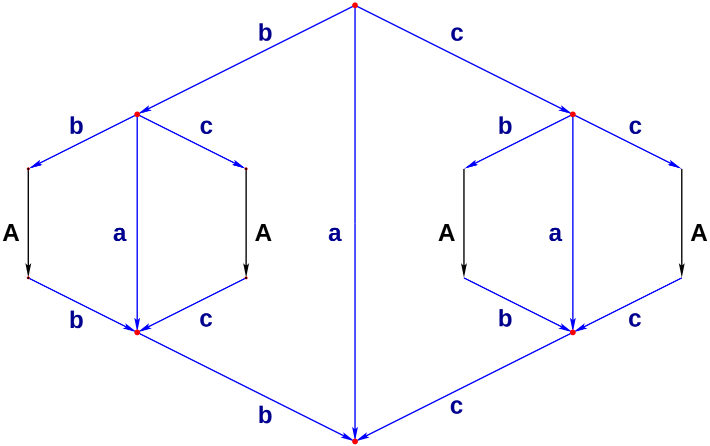


Another example

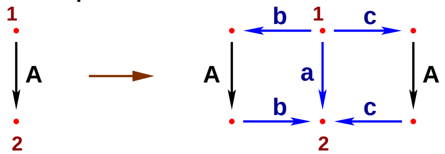


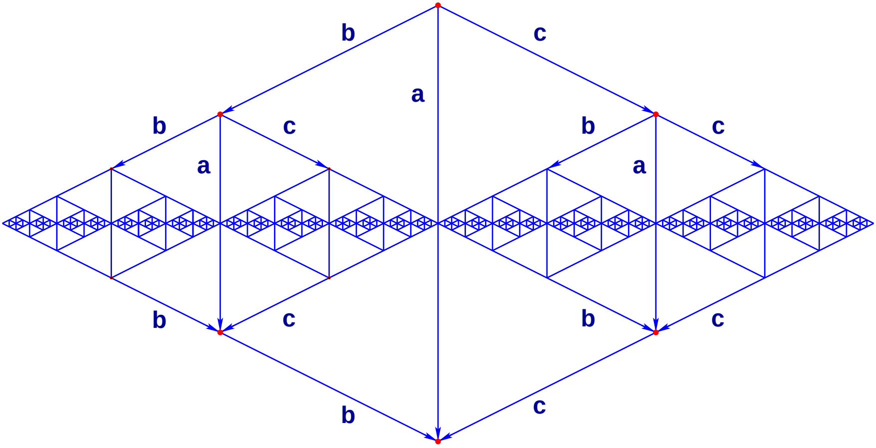
Another example





Another example

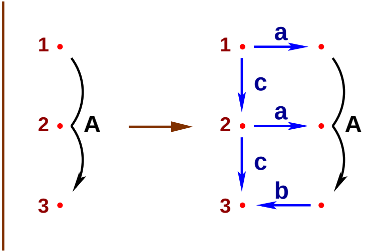




Regular graph

= graph generated by a deterministic graph grammar

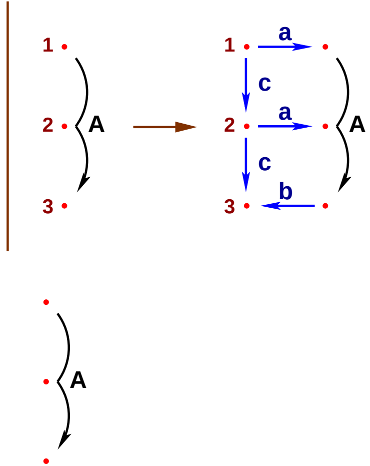
Non-terminal hyperarcs



Regular graph

= graph generated by a deterministic graph grammar

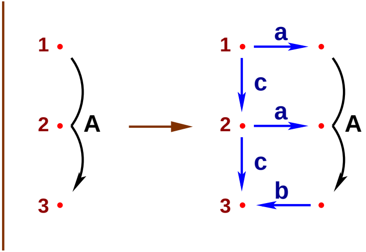
Non-terminal hyperarcs

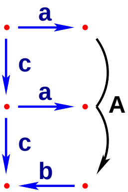


Regular graph

= graph generated by a deterministic graph grammar

Non-terminal hyperarcs

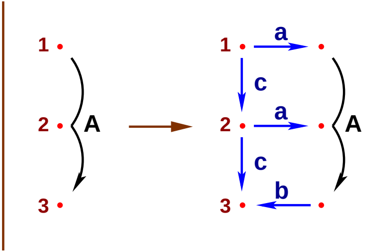


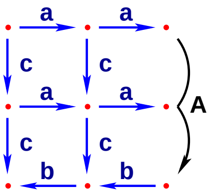


Regular graph

= graph generated by a deterministic graph grammar

Non-terminal hyperarcs

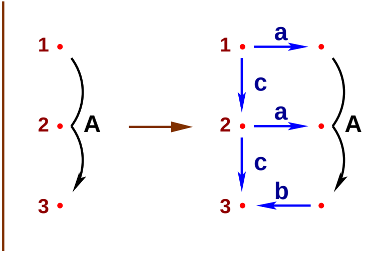


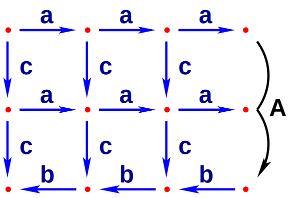


Regular graph

= graph generated by a deterministic graph grammar

Non-terminal hyperarcs

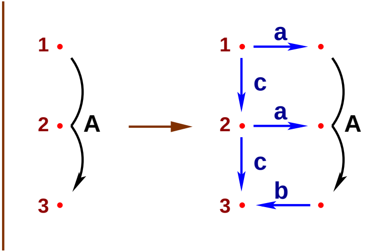


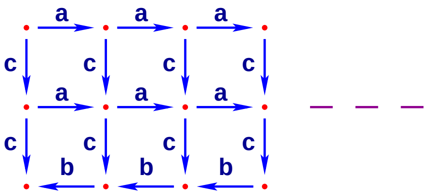


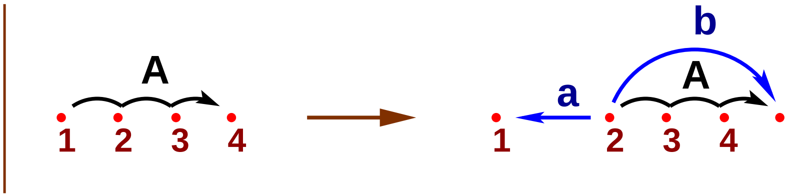
Regular graph

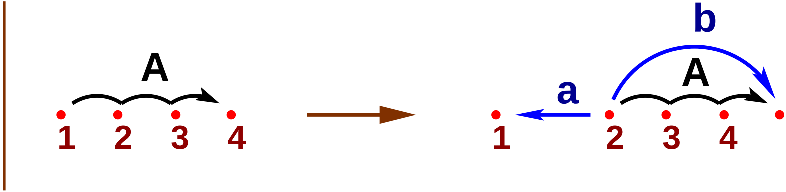
= graph generated by a deterministic graph grammar

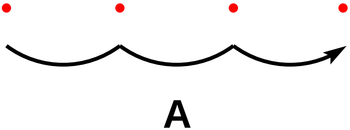
Non-terminal hyperarcs

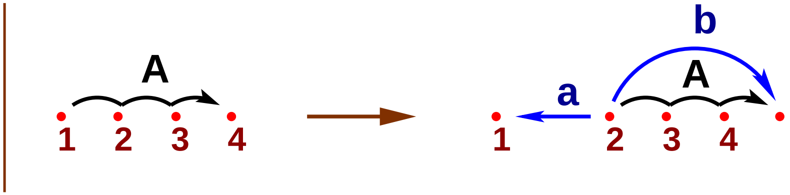


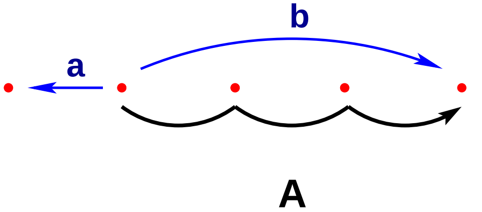


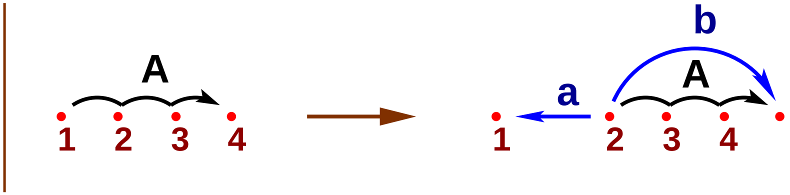

A
1
2
3
4
b
A
a
1
2
3
4

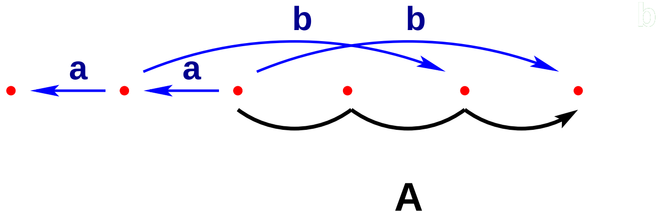

A
1 2 3 4
b
a
A
1 2 3 4

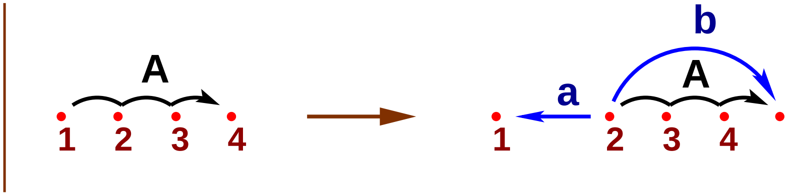

A

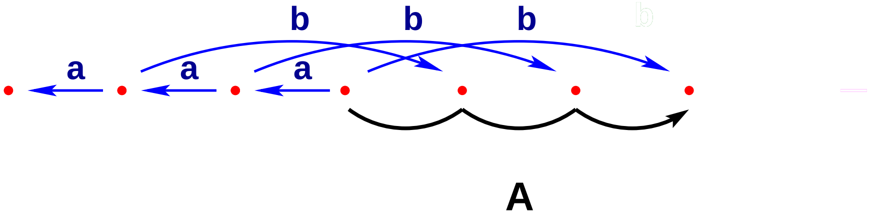

A
1 2 3 4
b
A
a
1 2 3 4

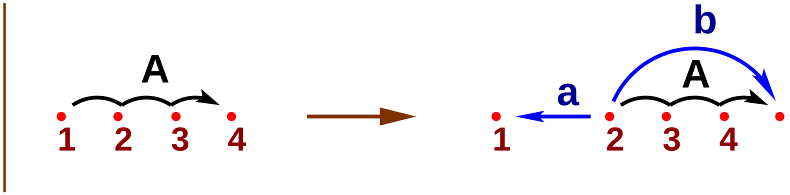

b
a
A

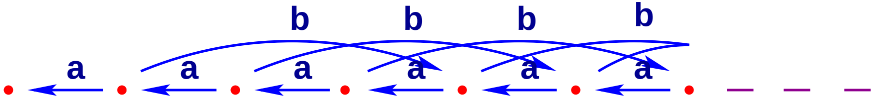

A
1
2
3
4
b
A
a
1
2
3
4

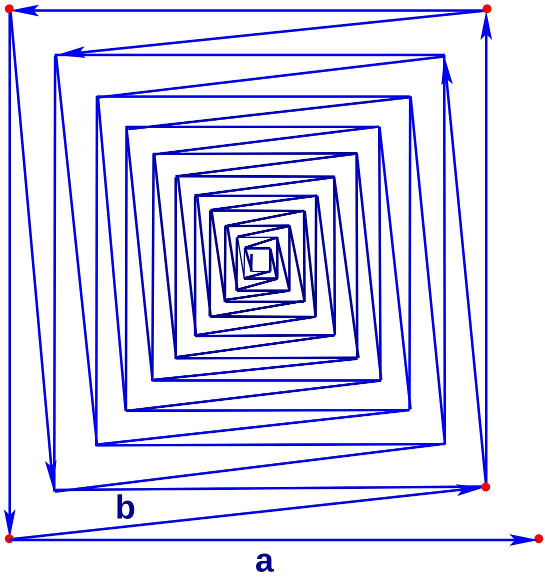

b
b
a
a
A


A
1 2 3 4
b
a
A
1 2 3 4


b b b
a a a
A


A
1
2
3
4
b
a
A
1
2
3
4


b
b
b
b
a
a
a
a
a
a

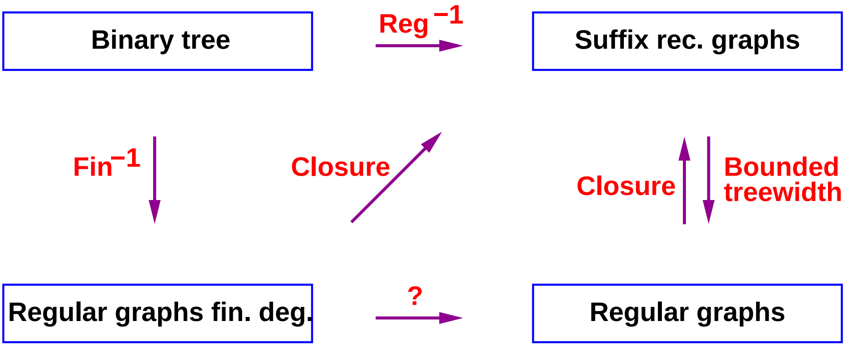

b
a

Proposition

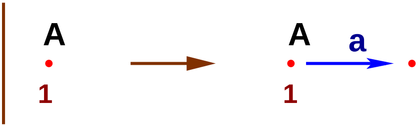
The accessible pushdown graphs are the rooted regular graphs of finite degree.

The connected components of the pushdown graphs are the connected regular graphs of finite degree.

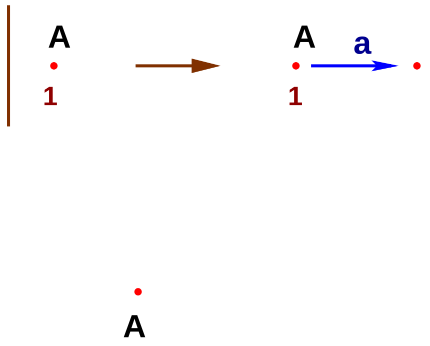
The regular restrictions of the pushdown graphs are the regular graphs of finite degree.


Binary tree
Reg −1
Suffix rec. graphs
Fin −1
Closure
Closure
Bounded treewidth
Regular graphs fin. deg.
?
Regular graphs

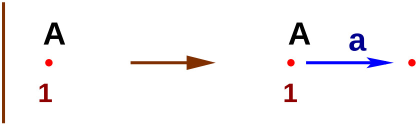
Regular graphs of infinite degree

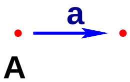


Regular graphs of infinite degree

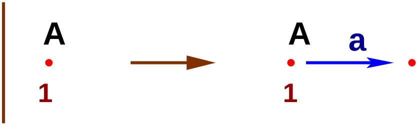


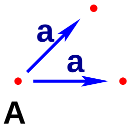
Regular graphs of infinite degree



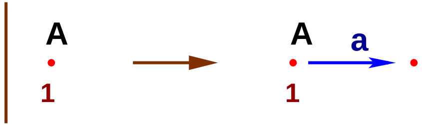


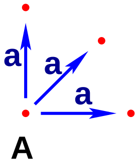
Regular graphs of infinite degree



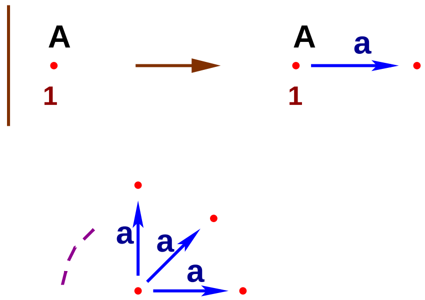


Regular graphs of infinite degree

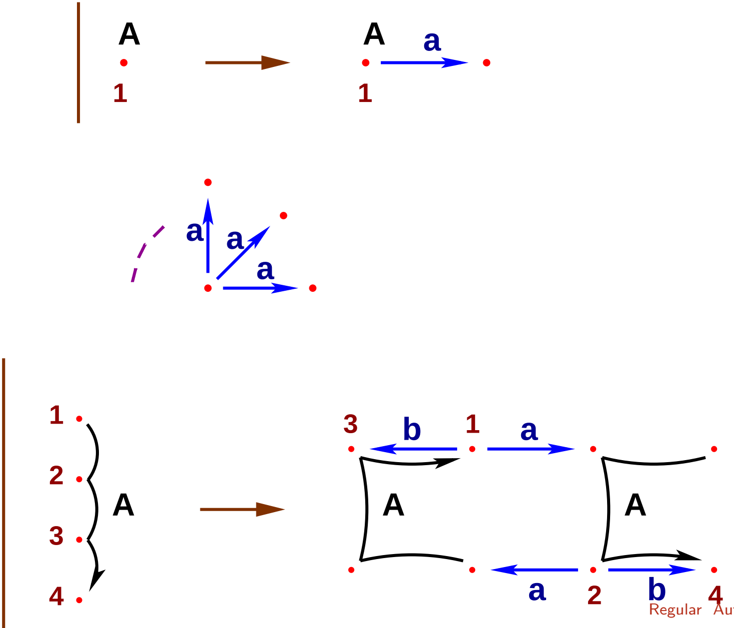


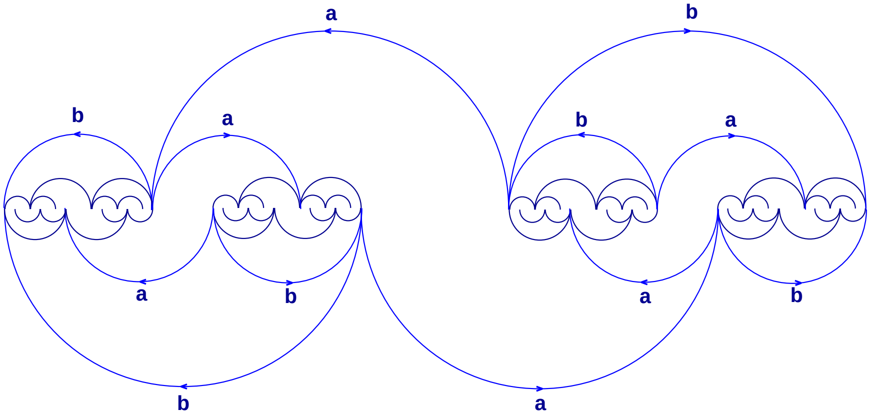


Regular graphs of infinite degree



Regular graphs of infinite degree




a
b
b
a
b
a
a
b
a
b
b
a

Suffix recognizable graphs ?

Theorem of Muller and Schupp

The accessible pushdown graphs are the rooted graphs of finite degree with a finite decomposition by distance

The ‘distance’ is a normal form

The accessible pushdown graphs are the rooted graphs of finite degree with a finite decomposition

The ‘pushdown automata’ is another normal form

Word rewriting system over an alphabet N

finite $R \subseteq N^* \times N^*$

rewriting $\longrightarrow_R \; = \; N^* R N^*$

$xuy \longrightarrow_R xvy$ for $(u,v) \in R$ and $x,y \in N^*$

derivation $\overset{*}{\longrightarrow}_R$ refl. trans. closure under o

suffix rewriting $\longrightarrow\!\!\!\shortmid_R \; = \; N^* R$

$xu \longrightarrow\!\!\!\shortmid_R xv$ for $(u,v) \in R$ and $x \in N^*$

suffix derivation $\overset{*}{\longrightarrow}\!\!\!\shortmid_R$

Proposition Büchi 1964

$\overset{*}{\longrightarrow}\!\!\!\shortmid_R(u) = \{ v \mid u \overset{*}{\longrightarrow}\!\!\!\shortmid_R v \}$ *regular language*

$\overset{*}{\longrightarrow}_R$ is regular preserving: for L regular

$\overset{*}{\longrightarrow}_R(L) = \{ v \mid \exists u \in L \, (u \overset{*}{\longrightarrow}_R v) \}$ *regular*

recognizable system

$R = \bigcup_i U_i \times V_i$ for U_i, V_i regular

Theorem

The suffix derivation $\overset{*}{\longrightarrow}_R$ *for* R *recognizable*
is the suffix rewriting $\longrightarrow_S$ *for some* S *recognizable*
hence is a regular binary relation on words

Benois's lemma

$$B \;=\; \{\, x\,\overleftarrow{x} \mid x \in N \,\}$$

$$\overleftarrow{uv} \;=\; \overleftarrow{v}\,\overleftarrow{u} \quad \text{for any} \quad u,v \in N^*$$

$$\{\, \overleftarrow{u}\,v \mid (u,v) \in R \,\}^* {\downarrow} B \;\cap\; \overleftarrow{N}^* N^* \;=\; \bigcup_i \overleftarrow{U_i} \times V_i$$

$$S \;=\; \bigcup_i U_i \times V_i$$

$$\overset{*}{\longrightarrow}_R \;=\; \longrightarrow_S$$

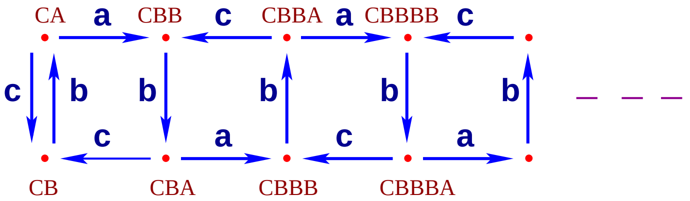
Labelled word rewriting system

$$R \left| \begin{array}{llll} A \xrightarrow{a} BB & & B \xrightarrow{b} A \\ BA \xrightarrow{c} B & & CA \xrightarrow{c} CB \end{array} \right.$$

Suffix transition

$$WU \xrightarrow{a} WV \quad \text{if} \quad (U \xrightarrow{a} V) \in R$$

Suffix transition graph $N^*.R$ accessible from CA



Theorem

The accessible suffix graphs are the rooted regular graphs of finite degree.

The connected components of the suffix graphs are the connected regular graphs of finite degree.

The regular restrictions of the suffix graphs are the regular graphs of finite degree.

Suffix recognizable graphs ?

Recognizable systems over N

$R \;\Big|\; U_i \xrightarrow{a_i} V_i$ with U_i, V_i regular over N

Suffix transition graph $N^* R \;=\; \bigcup_i N^*.(U_i \xrightarrow{a_i} V_i)$

$$= \bigcup_i \{\, wu \xrightarrow{a_i} wv \mid w \in N^*,\; u \in U_i,\; v \in V_i \,\}$$

Theorem 1996

The suffix recognizable graphs are

the regular restrictions of the suffix transition graphs of recognizable systems

$\bigcup_i W_i.(U_i \xrightarrow{a_i} V_i)$ for U_i, V_i, W_i *regular*

boolean algebra w.r.t. $N^* \times T \times N^*$

Graphs at level 1 of the pushdown hierarchy

internal representation:

recognizable suffix systems

external representation:

mon. interp.,...,inverse regular substitutions

from the infinite binary tree

geometrical representation: graph grammars

Graphs at level 2

graph grammars of level 2 ?

Recognizability

for regular automata

Proposition

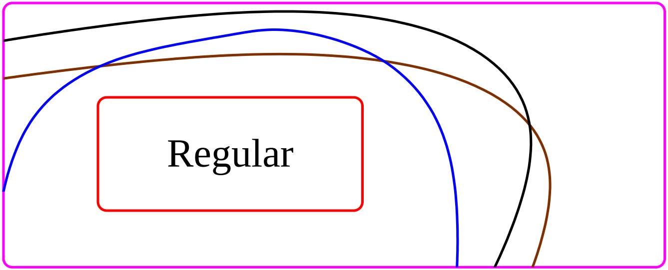
- *The deterministic regular automata of finite degree recognize the deterministic real-time context-free languages*
- *The deterministic regular automata recognize the deterministic context-free languages*
- *The context-free languages are preserved by union*
  but not by complementation and intersection
- *The deterministic context-free languages are preserved by complementation*
  but not by union and intersection

Synchronization of a regular automaton G

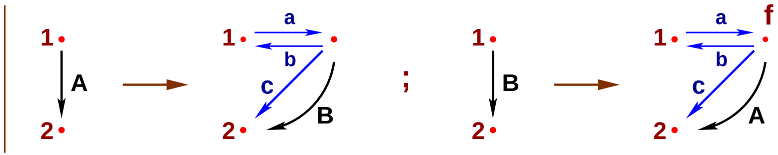
Family Sync(G) of context-free languages

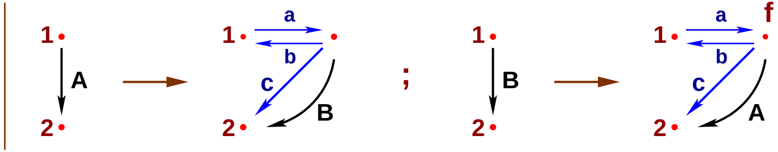
containing the regular languages $\subseteq$ L(G)

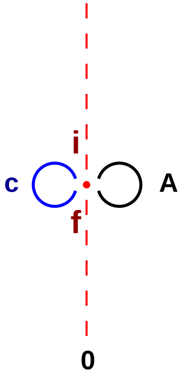
Sync(G) boolean algebra for G unambiguous

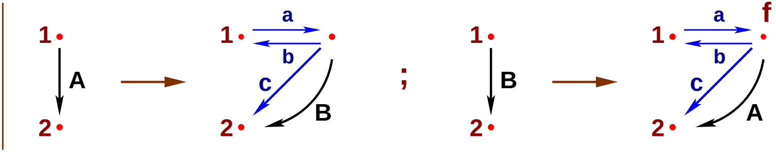


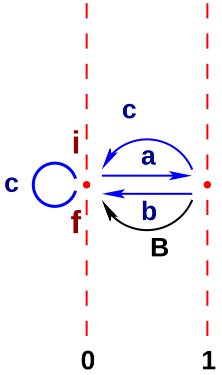
Unambiguous context-free languages

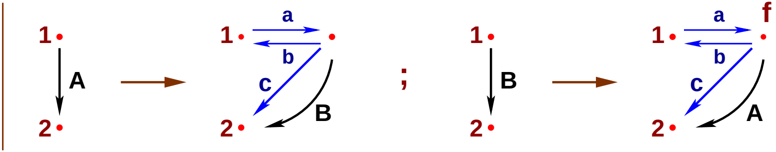

1
A
2
1
a
b
c
B
2
;
1
B
2
1
a
f
b
c
A
2

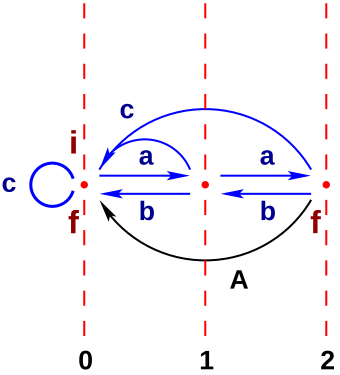

1
2
A
a
b
c
B
;
f

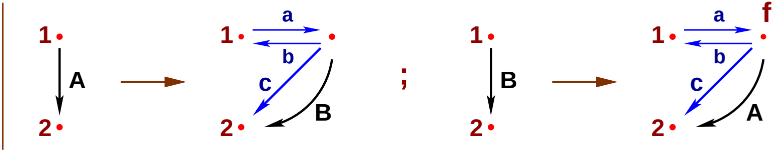

i
c
A
f
0

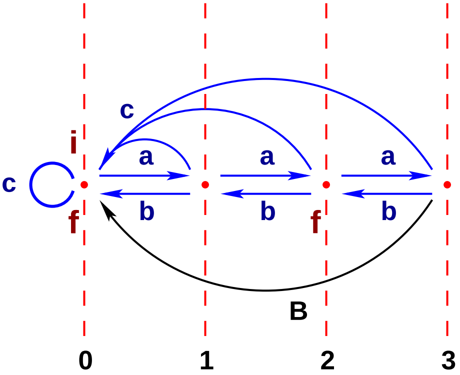

1
A
2
1
a
b
c
B
2
;
1
B
2
1
a
f
b
c
A
2

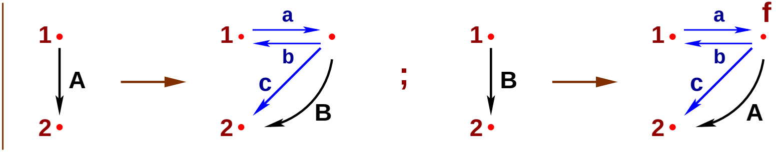

c
i
a
c
b
f
B
0
1

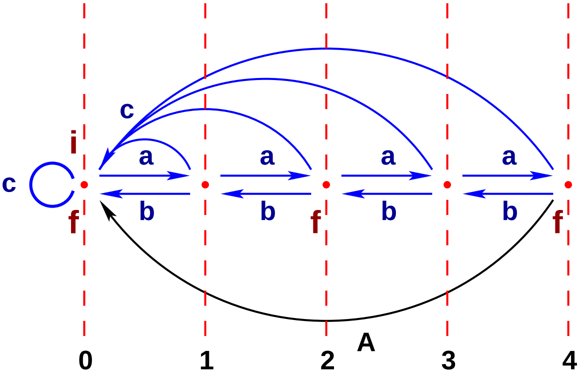

1
A
2
1
a
b
c
B
2
;
1
B
2
1
a
f
b
c
A
2

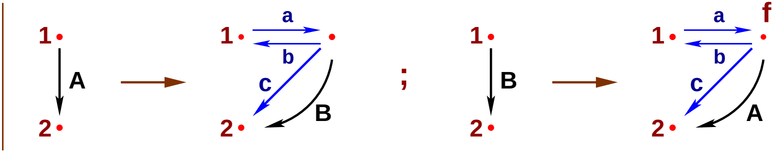

c
i
a
a
c
b
b
f
f
A
0
1
2

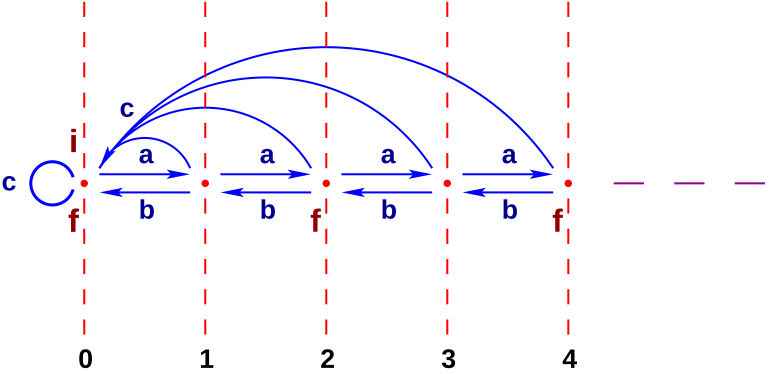

1
A
2
1
a
b
c
B
2
;
1
B
2
1
a
f
b
c
A
2


c
i
c
a
a
a
f
b
b
f
b
B
0
1
2
3


1
2
A
a
b
c
B
;
f


c
i
f
a
b
A
0
1
2
3
4



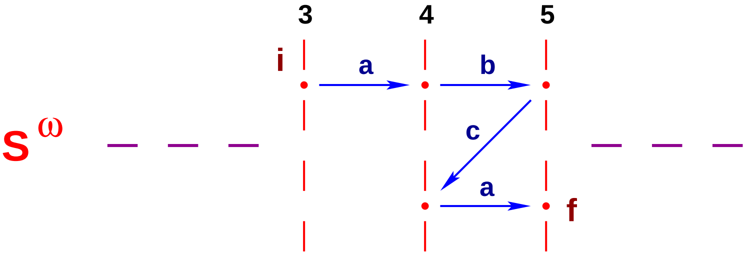


level of a vertex

Grammar S synchronized by R if

any accepting path of S^ω is synchronized by

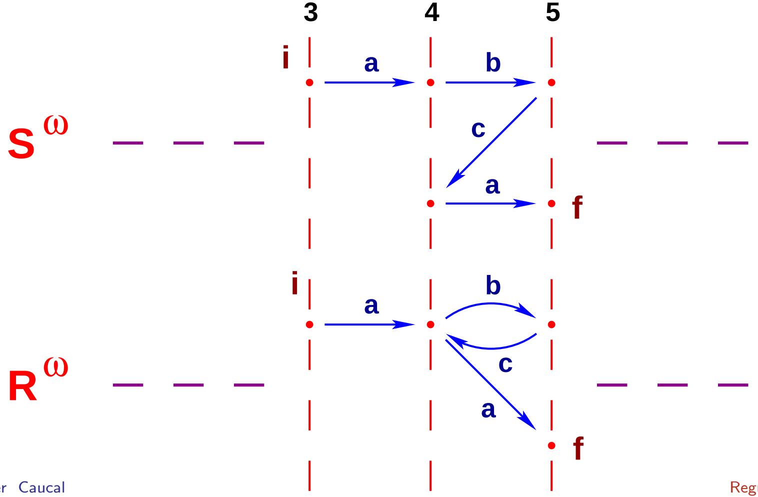
an accepting path of R^ω



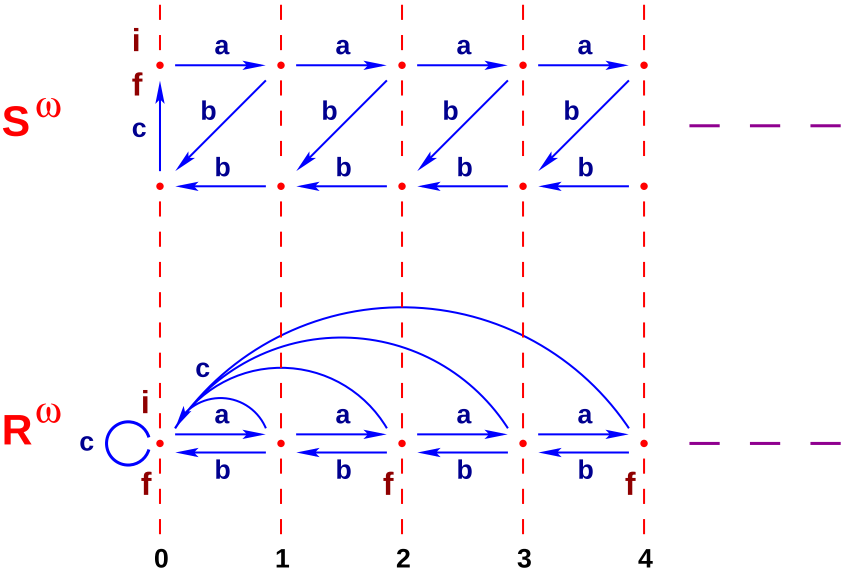
Grammar S synchronized by R if

any accepting path of S^ω is synchronized by

an accepting path of R^ω



S synchronized by R : $L(S) \subseteq L(R)$



Synchronized languages by R

$$\mathrm{Sync}(R) = \{ L(S) \mid S \text{ synchronized by } R \}$$

Theorem 08

$\mathrm{Sync}(R) = \mathrm{Sync}(S)$ *for* R, S *gen. the same graph*

Definition

For any regular automaton G

$$\mathrm{Sync}(G) = \mathrm{Sync}(R) \text{ for } R \text{ generating } G$$

Theorem 08 with Hassen

For any deterministic regular automaton G

Sync(G) *is an effective Boolean algebra w.r.t.* L(G)

For G deterministic, complete, of finite degree

Nowotka, Srba 07

Theorem 08

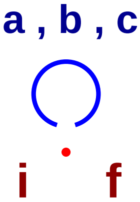
For any unambiguous regular automaton G

Sync(G) *is an effective Boolean algebra w.r.t.* L(G)

Corollary

For any unambiguous regular automaton G

Sync(G) *has a decidable inclusion problem*

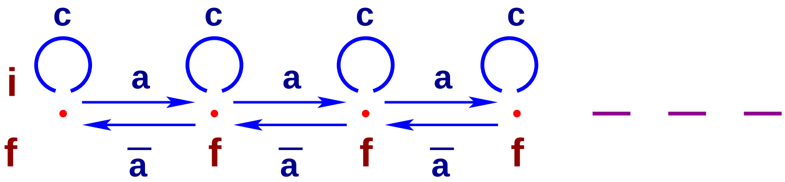


Family of regular languages

Finite automaton G

Family of regular languages included in $L(G)$

Deterministic regular automaton G

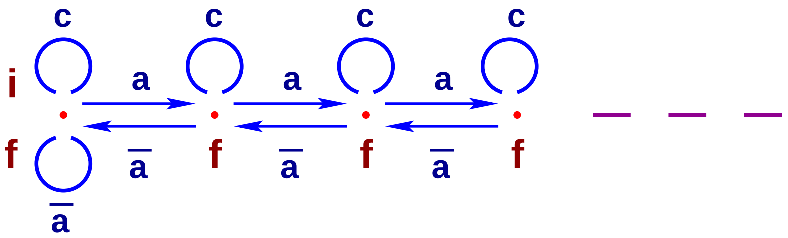


Family of input-driven languages

Mehlhorn 1980

contains the regular languages ⊆ L(G)

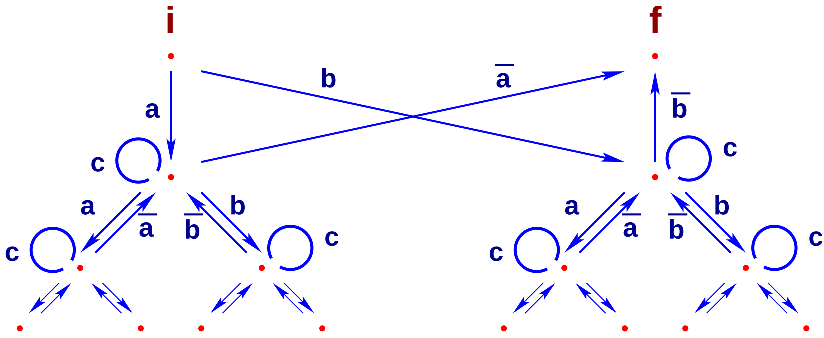
Complete deterministic regular automaton



Family of visibly pusdown languages

Alur Madhusudan 2004

contains all the regular languages



Family of balanced languages

Berstel Boasson 2002

Setback

The synchronization depends on

- graph grammars : vertex levels
- pushdown automata : configuration lengths

restricted and technical notion

simple and natural notion : morphism

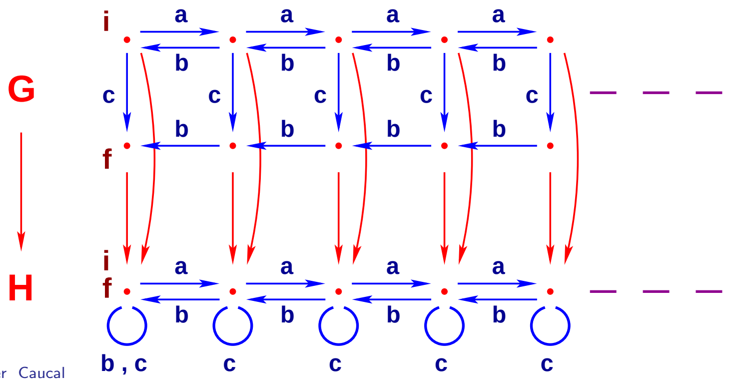
Automaton morphism $G \xrightarrow{h} H$

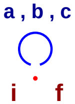
mapping $h : V_G \longrightarrow V_H$ such that $h(G) \subseteq H$ i.e.

$$s \xrightarrow{a}_G t \implies h(s) \xrightarrow{a}_H h(t)$$

$$c\,s \in G \implies c\,h(s) \in H$$

we say that G is reducible into H



any automaton is reducible into 

A morphism $G \xrightarrow{h} H$ is locally bounded if

there exists $b \geq 0$ such that for any $t \in V_H$

$$h^{-1}(t) \ = \ \{ \ s \in V_G \ | \ h(s) = t \ \}$$

is of cardinal at most b ; we write $G \xrightarrow{h}_{lb} H$

Recognizability / automaton family F

$$Rec_F(H) \ = \ \{ \ L(G) \mid G \in F \wedge G \longrightarrow_{lb} H \ \}$$

Family A of regular automata of finite degree

Theorem

For any unambiguous $H \in A$

$$\mathrm{Rec}_A(H) \text{ is a boolean algebra w.r.t. } L(H)$$

unique morphism for $G \longrightarrow H$ unambiguous

Synchronization / automaton family F

$$\mathrm{Sync}_F(H) = \{ L(G) \mid [G \longrightarrow_{lb} H] \in F \}$$

Theorem

For any unambiguous $\mathsf{H} \in \mathsf{A}$

$\mathsf{Rec}_{\mathrm{A}}(\mathsf{H}) \;=\; \mathsf{Sync}_{\mathrm{A}}(\mathsf{H})$ *is a boolean algebra* / $\mathsf{L}(\mathsf{H})$

$\mathsf{Sync}_{\mathrm{A}}(\mathsf{H}) \;\subseteq\; \mathsf{Rec}_{\mathrm{A}}(\mathsf{H})$: by definitions

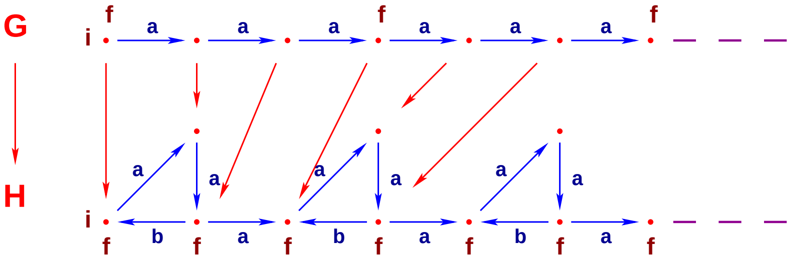
$\mathsf{Rec}_{\mathrm{A}}(\mathsf{H}) \;\subseteq\; \mathsf{Sync}_{\mathrm{A}}(\mathsf{H})$: corresponds to the

Key property

If $\mathsf{G} \longrightarrow_{\mathrm{lb}} \mathsf{H}$ *unambiguous with* $\mathsf{G}\,,\mathsf{H} \in \mathsf{A}$
then $[\mathsf{G} \longrightarrow \mathsf{H}] \in \mathsf{A}$

decomposition of $[G \xrightarrow{f} H]$

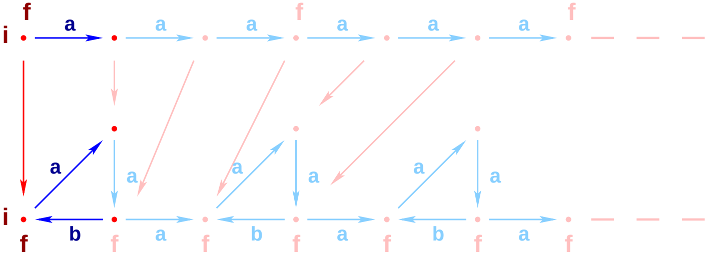
$=$ decomposition of $H + f^{-1}$



not by distance

decomposition of $[G \xrightarrow{f} H]$

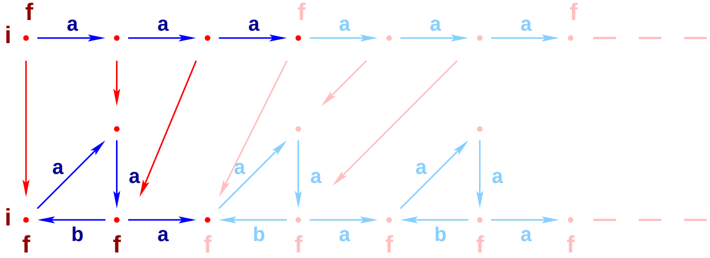
$=$ decomposition of $H + f^{-1}$



not by distance

decomposition of $[G \xrightarrow{f} H]$

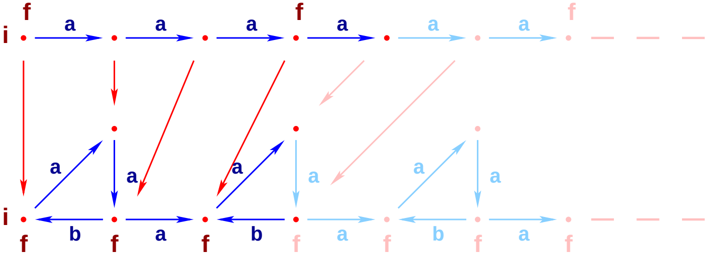
$= \;$ decomposition of $H + f^{-1}$



not by distance

decomposition of $[G \xrightarrow{f} H]$

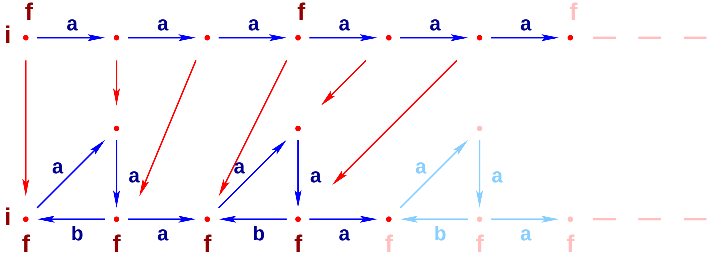
$=$ decomposition of $H + f^{-1}$



not by distance

decomposition of $[G \xrightarrow{f} H]$

$=$ decomposition of $H + f^{-1}$



not by distance

Conclusion

Level 2 of the pushdown hierarchy

- graph grammars
- recognizability
- synchronization

Conclusion

Level 2 of the pushdown hierarchy

- graph grammars
- recognizability
- synchronization

Conclusion

Level 2 of the pushdown hierarchy

- graph grammars
- recognizability
- synchronization

Conclusion

Level 2 of the pushdown hierarchy

- graph grammars
- recognizability
- synchronization