Basics about algebraic approach to languages

Marc Zeitoun

LSV, ENS Cachan – LaBRI, U. Bordeaux – CNRS – INRIA

GAMES-EPIT Spring School 2011

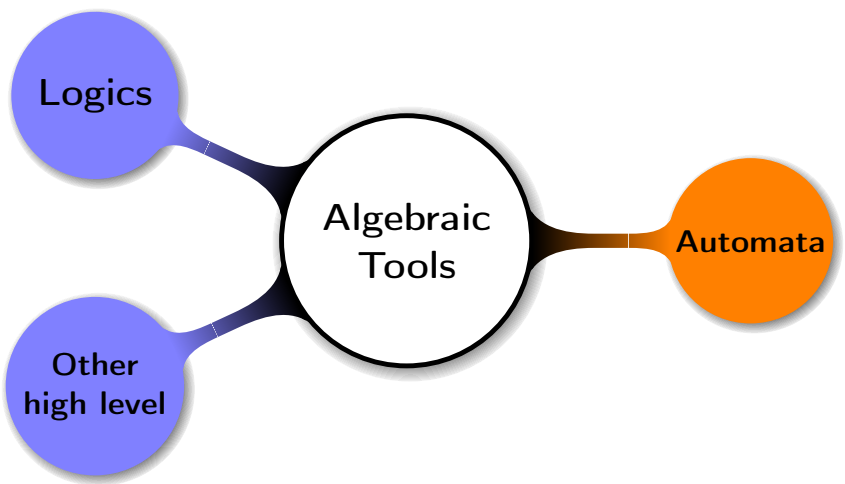
Carcans Maubuisson
May 23–27, 2011

Motivation

- Word languages are used in computer science for many purposes.
- Natural models for
  - (text) programs,
  - inputs of programs,
  - behaviors of programs,
  - etc.
- Several formalisms to define regular sets of words: logics, regexp, automata...

How to compare their expressive power?

The talk shows examples exploiting connections with algebra to answer this question.



Motivation: some examples

- Main question in this talk: expressiveness.

  Typical questions:
- Compare the expressive power of logic $\mathcal{L}_1$ and logic $\mathcal{L}_2$.
- Is temporal logic with past and future operators more expressive than pure future temporal logic?

Motivation: some examples

- Main question in this talk: expressiveness.

  Typical questions:
- Compare the expressive power of logic $\mathcal{L}_1$ and logic $\mathcal{L}_2$.
- Is temporal logic with past and future operators more expressive than pure future temporal logic?
- Is temporal logic with only unary operators weaker?
- Are FO($<$) and LTL on words equally expressive?
- Is a given regular language L expressible in first-order logic?
- A polynomial is a language of the form $\bigcup_{\text{finite}} B_0^* a_1 B_1^* \cdots a_k B_k^*$.

  Given a regular language L, is it decidable if
  - L is a polynomial?
  - L is a finite boolean combination of polynomials?

Link between logic and algebra for regular languages

- A language $L \subseteq A^*$ is recognized by a monoid M if there is a morphism $\varphi : A^* \to M$ such that
$$L = \varphi^{-1}(\varphi(L))$$
- In other words: it is enough to look at φ image to decide membership to L.

Thm. Kleene-Büchi

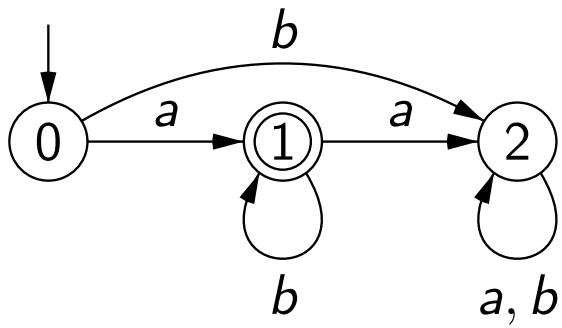
For a language $L \subseteq A^*$, it is equivalent

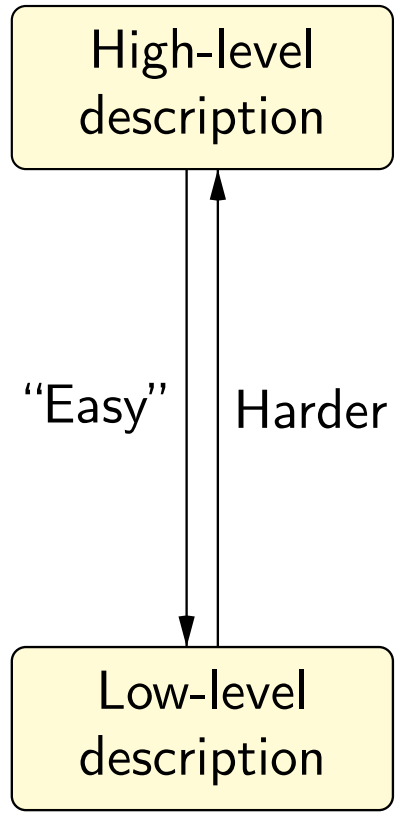
- to be recognized by a finite automaton,
- to be definable by a regular expression,
- to be definable in Monadic Second Order logic,
- to be recognized by a finite monoid.

Regular languages and monoids

Language L, Regular exp ab^*

Minimal automaton $\mathcal{A}_{\mathsf{min}}(L)$





Regular languages and monoids

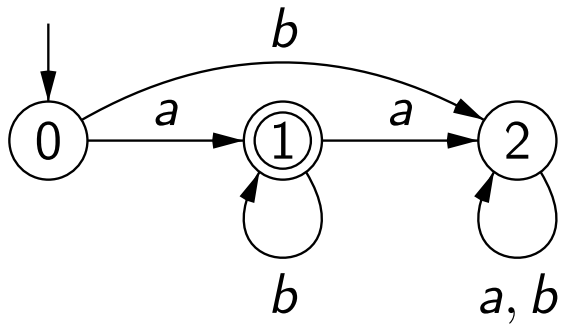
Language L, Regular exp ab^*

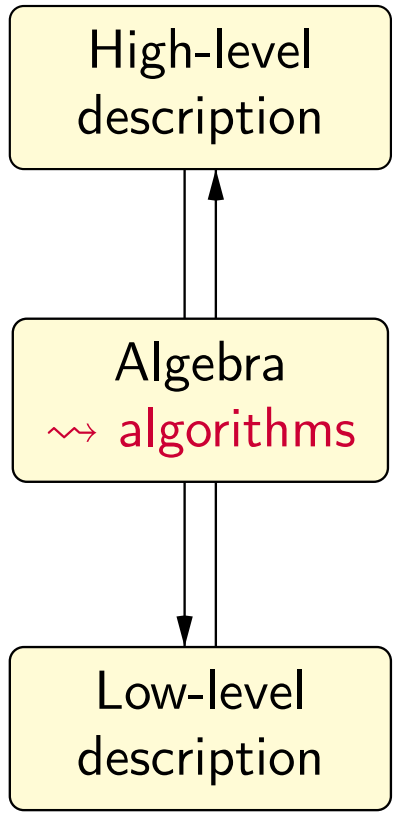
Syntactic monoid $M(L)$ (finite, [K56])

	0	1	2
a	1	2	2
b	2	1	2

$\{\varepsilon, a, b, a^2\}$
$ab = a, ba = a^2,$
$b^2 = b$

Minimal automaton $\mathcal{A}_{\min}(L)$





High-level vs. algebraic properties: prominent results

Language L	$\mathcal{A}_{\mathsf{min}}(L)$	$Synt(L)$	Logical definability
Star-free	Counter-free	A	$\mathrm{FO}(<), \mathrm{LTL}$

[Sch65,McN-P71,K68]

Piecewise testable	Very weak $+\ldots$	J	$\mathrm{Bool}(\Sigma_1)$

[S75,Th87]

$\biguplus_f \prod$ non ambiguous	2-way part. ord.	DA	$\mathrm{FO}_2(<)$, $\mathrm{UTL}, \Sigma_2 \cap \Pi_2$

[Sch76,SchThV01,ThW98,EVW97,PW97]

Loc. threshold testable	Forbidden patterns	ACom $*$ LI	$\mathrm{FO}(+1)$

[S85,ThW85,BP89]

Some tools

- Syntactic monoid of a regular language.
- Monoid structure, in terms of ideals.
- Basic and more advanced pumping arguments.

Outline

Syntactic monoids

- Notation: Monoid $(M, \cdot, 1)$, with $\cdot$ associative, 1 neutral.
- A^* free monoid over A.
- Morphism $\varphi : M \to N$.
- If $L \subseteq A^*$, its syntactic congruence $\sim_L \subseteq A^* \times A^*$ is defined by

$$u \sim_L v \text{ iff } \forall x, y \in xuy \in L \iff xvy \in L.$$

- It is indeed a congruence: equivalence relation compatible with concatenation.

$$u \sim_L v \implies [x(uw)y \in L \Leftrightarrow x(vw)y \in L] \implies uw \sim_L vw$$

- Syntactic monoid $M(L) = A^*/\sim_L$ and syntactic morphism $\eta_L : A^* \to M(L)$.

Thm. (Myhill)

A language is regular if and only if its syntactic congruence has finite index.

Syntactic monoids

- Say that M_1 divides M_2 if M_1 is a quotient of a submonoid of M_2.

Syntactic monoid and recognition

Monoids recognizing L are exactly those which are divided by $M(L)$.

Syntactic monoids

- Say that M_1 divides M_2 if M_1 is a quotient of a submonoid of M_2.

Syntactic monoid and recognition

Monoids recognizing L are exactly those which are divided by $M(L)$.

Proof

- η_L recognizes L since $\eta_L^{-1}(\eta_L(L)) = L$:

$$u \in \eta_L^{-1}(\eta_L(L)) \;\Rightarrow\; \eta_L(u) \in \eta_L(L) \;\Rightarrow\; \exists v \in L, u \sim_L v \;\Rightarrow\; u \in L.$$

Syntactic monoids

- Say that M_1 divides M_2 if M_1 is a quotient of a submonoid of M_2.

Syntactic monoid and recognition

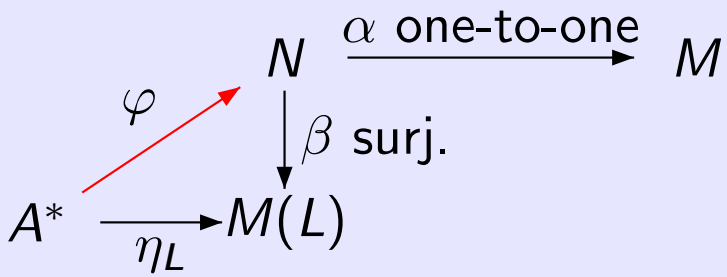
Monoids recognizing L are exactly those which are divided by $M(L)$.

Proof

- η_L recognizes L since $\eta_L^{-1}(\eta_L(L)) = L$:

$$u \in \eta_L^{-1}(\eta_L(L)) \;\Rightarrow\; \eta_L(u) \in \eta_L(L) \;\Rightarrow\; \exists v \in L, u \sim_L v \;\Rightarrow\; u \in L.$$

- If $M(L)$ divides M:



Then $\alpha \circ \varphi$ recognizes L.

Syntactic monoids

- Say that M_1 divides M_2 if M_1 is a quotient of a submonoid of M_2.

Syntactic monoid and recognition

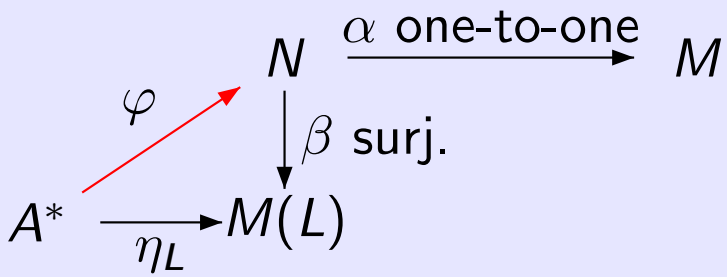
Monoids recognizing L are exactly those which are divided by $M(L)$.

Proof

- η_L recognizes L since $\eta_L^{-1}(\eta_L(L)) = L$:

$$u \in \eta_L^{-1}(\eta_L(L)) \;\Rightarrow\; \eta_L(u) \in \eta_L(L) \;\Rightarrow\; \exists v \in L, u \sim_L v \;\Rightarrow\; u \in L.$$

- If $M(L)$ divides M:



  Then $\alpha \circ \varphi$ recognizes L.

- Conversely, if $\varphi : A^* \to M$ recognizes L, then $\varphi(u) = \varphi(v)$ implies $u \sim_L v$. Therefore there is a surjective morphism β from $\varphi(A^*)$ into $M(L)$.

Syntactic monoids

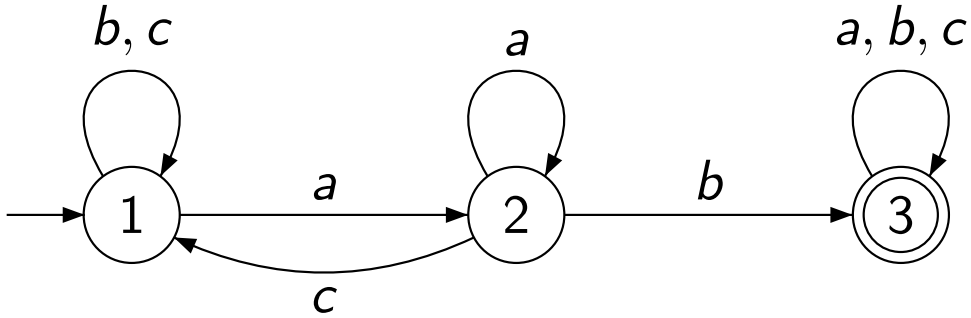
- $M(L)$ is the monoid of transitions of the minimal automaton of L.
  Each word induces a mapping $Q \to Q$ (hence $|M(L)| \leqslant Q^Q$).
  $\Rightarrow$ Algorithm to compute $M(L)$ (see Example next slide).
- In particular, we have $M(A^* \setminus L) = M(L)$

Syntactic monoids

- $M(L)$ is the monoid of transitions of the minimal automaton of L.
  Each word induces a mapping $Q \to Q$ (hence $|M(L)| \leqslant Q^Q$).
  $\Rightarrow$ Algorithm to compute $M(L)$ (see Example next slide).
- In particular, we have $M(A^* \setminus L) = M(L)$
- $M(L_1 \cap L_2)$ and $M(L_1 \cup L_2)$ divide $M(L_1) \times M(L_2)$.
- Consequence: properties defined by identities are preserved through boolean combinations.
- For instance, if $M(L_1)$ and $M(L_2)$ are commutative, then so are
  - $M(L_1 \cap L_2)$,
  - $M(L_1 \cup L_2)$
  - $M(L_1 \setminus L_2)$.

Syntactic monoid: example

- Example: let $A = \{a, b, c\}$ and $L = A^* abA^*$.
- The minimal automaton of L is the following:



Generators

	1	2	3
a	2	2	3
b	1	3	3
c	1	1	3

Relations

$$aa = a = ca, \quad ac = cb = cc = c$$
$$bb = b = bc, \quad ab = bab = 0$$

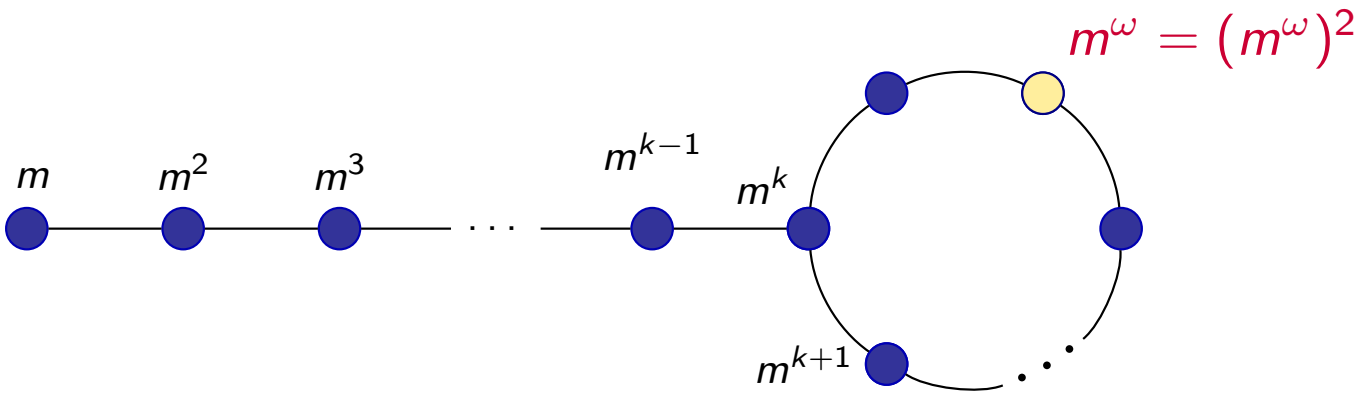
Elements

	1	2	3
$* 1$	1	2	3
$* a$	2	2	3
$* b$	1	3	3
$* c$	1	1	3
$* ab$	3	3	3
ba	2	3	3

Background on finite monoids

Idempotents in monoids

- An idempotent is an element $e \in M$ such that $e = ee$.
- Any $m \in M$ has a single power which is idempotent, denoted m^ω.



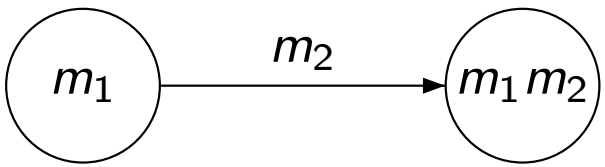
- Since the loop has at most $|M|$ elements, $m^{|M|!} = m^\omega$.

Ideal theory of monoids: Green's relations

- Automata exhibit some structure: graph representation, strongly connected components, DAG of scc, etc.
- One can extract such kind of information also out of a monoid M.

Ideal theory of monoids: Green's relations

- Automata exhibit some structure: graph representation, strongly connected components, DAG of scc, etc.
- One can extract such kind of information also out of a monoid M.
- The right action of M on itself yields a labeled graph $\mathcal{R}(M)$.



  Strongly connected components of $\mathcal{R}(M)$ are called $\mathcal{R}$-classes.
- Formally, u and v are $\mathcal{R}$-equivalent, written $u \mathcal{R} v$ if
  1. There exists $x \in M$ such that $v = ux$, and
  2. There exists $y \in M$ such that $u = vy$.
- Or equivalently, if $uM = vM$: u and v generate the same right ideal.

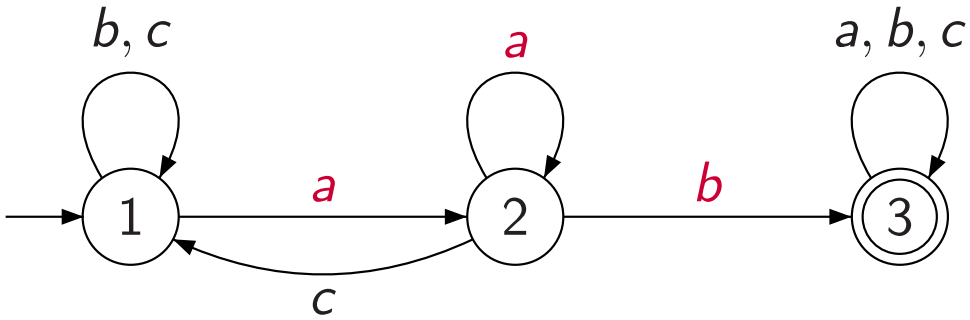
Green's relations

- $u \mathcal{R} v$ if $uM = vM$,
- $u \mathcal{L} v$ if $Mu = Mv$,
- $u \mathcal{J} v$ if $MuM = MvM$,
- $\mathcal{H} = \mathcal{R} \cap \mathcal{L}$.

Green's relations

- $u \mathcal{R} v$ if $uM = vM$,
- $u \mathcal{L} v$ if $Mu = Mv$,
- $u \mathcal{J} v$ if $MuM = MvM$,
- $\mathcal{H} = \mathcal{R} \cap \mathcal{L}$.

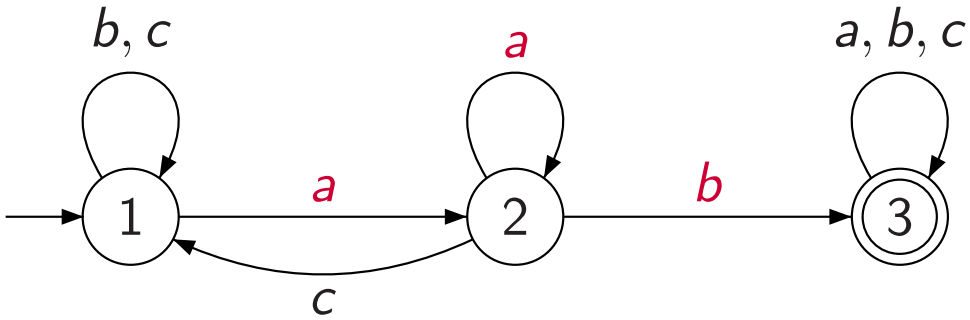
$\{a, b, c\}^* ab \{a, b, c\}^*$



Green's relations

- $u \mathcal{R} v$ if $uM = vM$,
- $u \mathcal{L} v$ if $Mu = Mv$,
- $u \mathcal{J} v$ if $MuM = MvM$,
- $\mathcal{H} = \mathcal{R} \cap \mathcal{L}$.

$\{a, b, c\}^* ab \{a, b, c\}^*$



^{*1}

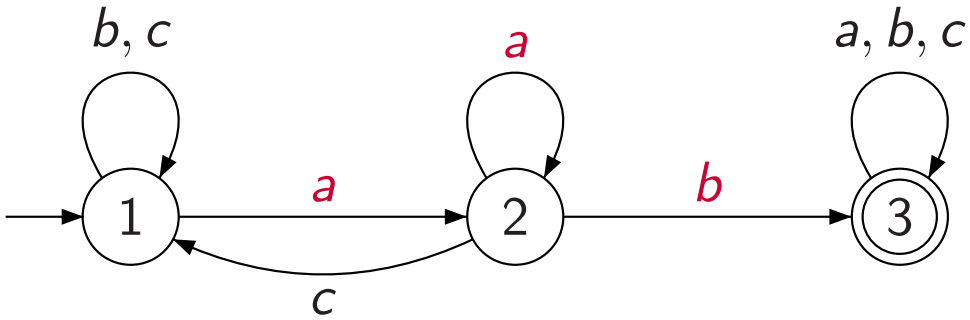
$^*a$	$^*c$
ba	$^*b$

$^*ab$

Green's relations

- $u \mathcal{R} v$ if $uM = vM$,
- $u \mathcal{L} v$ if $Mu = Mv$,
- $u \mathcal{J} v$ if $MuM = MvM$,
- $\mathcal{H} = \mathcal{R} \cap \mathcal{L}$.

$$\{a, b, c\}^* ab \{a, b, c\}^*$$



- $(ca = a) \wedge (ac = c) \Rightarrow a \mathcal{R} c$.

^{*1}

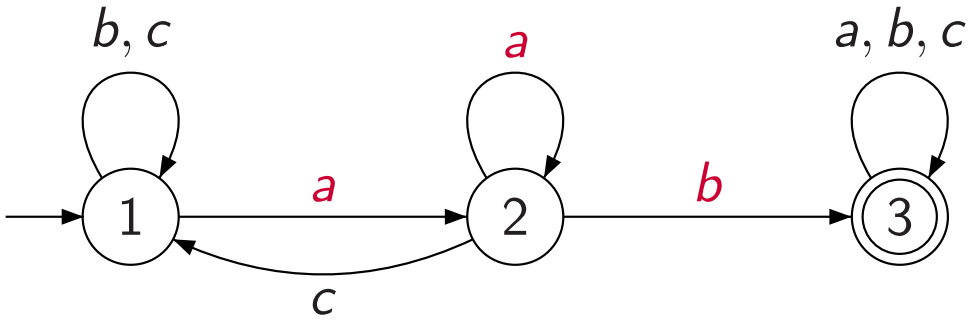
$^*a$	$^*c$
ba	$^*b$

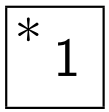
$^*ab$

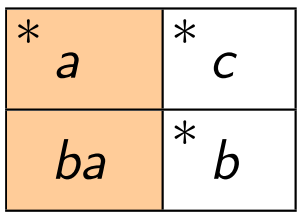
Green's relations

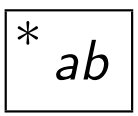
- $u \mathcal{R} v$ if $uM = vM$,
- $u \mathcal{L} v$ if $Mu = Mv$,
- $u \mathcal{J} v$ if $MuM = MvM$,
- $\mathcal{H} = \mathcal{R} \cap \mathcal{L}$.

$$\{a, b, c\}^* ab \{a, b, c\}^*$$







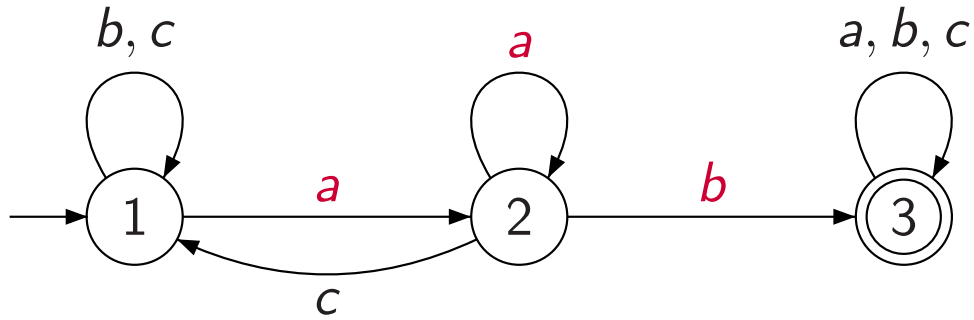


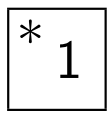
- $(ca = a) \wedge (ac = c) \Rightarrow a \mathcal{R} c$.
- $(ba = ba) \wedge (cba = a) \Rightarrow a \mathcal{L} ba$.

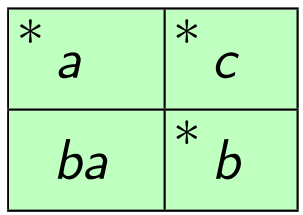
Green's relations

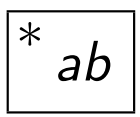
- $u \mathcal{R} v$ if $uM = vM$,
- $u \mathcal{L} v$ if $Mu = Mv$,
- $u \mathcal{J} v$ if $MuM = MvM$,
- $\mathcal{H} = \mathcal{R} \cap \mathcal{L}$.

$$\{a, b, c\}^* ab \{a, b, c\}^*$$







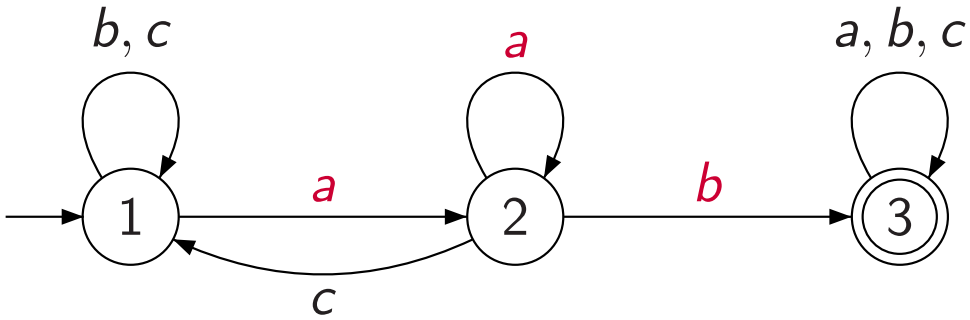


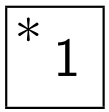
- $(ca = a) \wedge (ac = c) \Rightarrow a \mathcal{R} c$.
- $(ba = ba) \wedge (cba = a) \Rightarrow a \mathcal{L} ba$.
- Hence $b \mathcal{L} c \mathcal{R} a$, so $b \mathcal{J} a$.

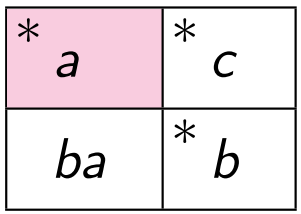
Green's relations

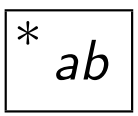
- $u \mathcal{R} v$ if $uM = vM$,
- $u \mathcal{L} v$ if $Mu = Mv$,
- $u \mathcal{J} v$ if $MuM = MvM$,
- $\mathcal{H} = \mathcal{R} \cap \mathcal{L}$.

$$\{a, b, c\}^* ab \{a, b, c\}^*$$







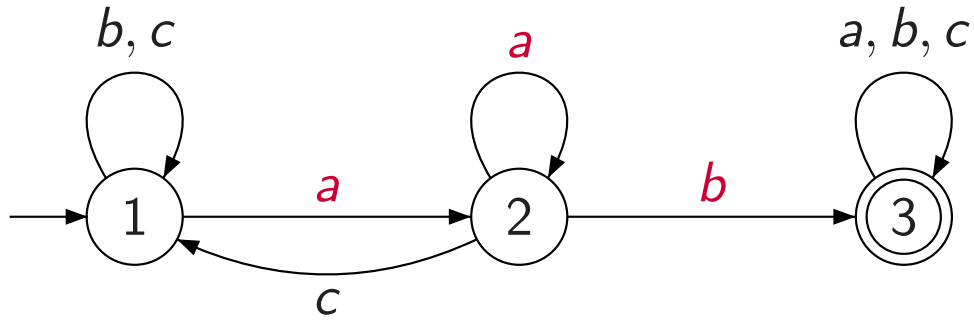


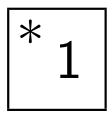
- $(ca = a) \wedge (ac = c) \Rightarrow a \mathcal{R} c$.
- $(ba = ba) \wedge (cba = a) \Rightarrow a \mathcal{L} ba$.
- Hence $b \mathcal{L} c \mathcal{R} a$, so $b \mathcal{J} a$.
- Here, every $\mathcal{H}$-class is trivial.

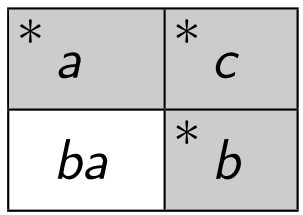
Green's relations

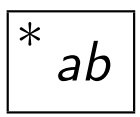
- $u \mathcal{R} v$ if $uM = vM$,
- $u \mathcal{L} v$ if $Mu = Mv$,
- $u \mathcal{J} v$ if $MuM = MvM$,
- $\mathcal{H} = \mathcal{R} \cap \mathcal{L}$.

$\{a, b, c\}^* ab \{a, b, c\}^*$







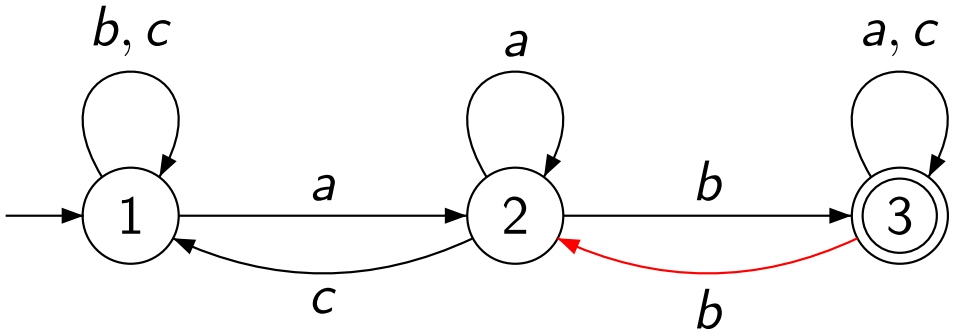


- $(ca = a) \wedge (ac = c) \Rightarrow a \mathcal{R} c$.
- $(ba = ba) \wedge (cba = a) \Rightarrow a \mathcal{L} ba$.
- Hence $b \mathcal{L} c \mathcal{R} a$, so $b \mathcal{J} a$.
- Here, every $\mathcal{H}$-class is trivial.

- In this example, we see that $\mathcal{R}$ and $\mathcal{L}$ commute, and $\mathcal{J} = \mathcal{R} \circ \mathcal{L} = \mathcal{L} \circ \mathcal{R}$.

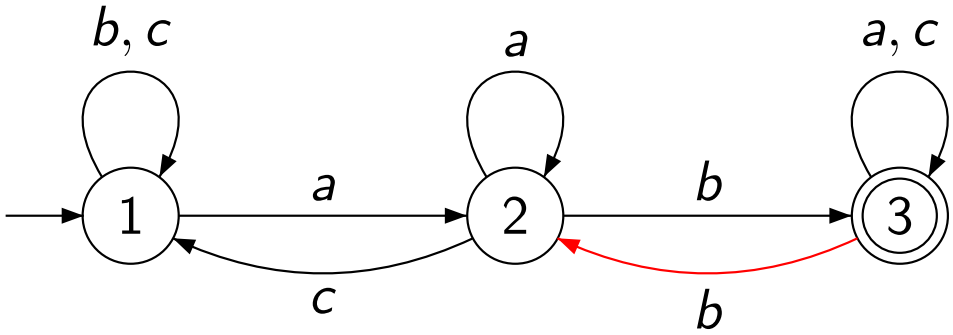
Another example

- $u \mathcal{R} v$ if $uM = vM$,
- $u \mathcal{L} v$ if $Mu = Mv$,
- $u \mathcal{J} v$ if $MuM = MvM$,
- $\mathcal{H} = \mathcal{R} \cap \mathcal{L}$.



Another example

- $u \mathcal{R} v$ if $uM = vM$,
- $u \mathcal{L} v$ if $Mu = Mv$,
- $u \mathcal{J} v$ if $MuM = MvM$,
- $\mathcal{H} = \mathcal{R} \cap \mathcal{L}$.



$*$ 1	b

$*$ a	ab	cb	$abcb$	$*$ c	abc
$*$ bab	ba	$*$ bcb	$babcb$	bc	$babc$

$*$ $cbab$	$*$ cba	$*$ cbc

- Here, nontrivial $\mathcal{H}$-classes.
- An $\mathcal{H}$-class containing an idempotent $*$ is a group.
- H is a group, or $HH \cap H = \varnothing$
- All H-classes from same $\mathcal{J}$-class are isomorphic.

Basic but important properties

- $\mathcal{H} \subseteq \mathcal{R}, \mathcal{L} \subseteq \mathcal{J}$.
- $\mathcal{R}$ is a left congruence, $\mathcal{L}$ is a right congruence.

Basic but important properties

- $\mathcal{H} \subseteq \mathcal{R}, \mathcal{L} \subseteq \mathcal{J}$.
- $\mathcal{R}$ is a left congruence, $\mathcal{L}$ is a right congruence.
- One can also consider the associated preorders. For instance:
  - $s \leqslant_{\mathcal{J}} t$ if $MsM \subseteq MtM$ (we say that s, t are $\mathcal{J}$-comparable).
  - $s <_{\mathcal{J}} t$ if $s \leqslant_{\mathcal{J}} t$ and $s \not\mathcal{J}\ t$.

Basic but important properties

- $\mathcal{H} \subseteq \mathcal{R}, \mathcal{L} \subseteq \mathcal{J}$.
- $\mathcal{R}$ is a left congruence, $\mathcal{L}$ is a right congruence.
- One can also consider the associated preorders. For instance:
  - $s \leqslant_{\mathcal{J}} t$ if $MsM \subseteq MtM$ (we say that s, t are $\mathcal{J}$-comparable).
  - $s <_{\mathcal{J}} t$ if $s \leqslant_{\mathcal{J}} t$ and $s \not\mathcal{J}\ t$.

An important property

In a finite monoid, $\mathcal{J} = \mathcal{R} \circ \mathcal{L} = \mathcal{L} \circ \mathcal{R}$.

Basic but important properties

- $\mathcal{H} \subseteq \mathcal{R}, \mathcal{L} \subseteq \mathcal{J}$.
- $\mathcal{R}$ is a left congruence, $\mathcal{L}$ is a right congruence.
- One can also consider the associated preorders. For instance:
  - $s \leqslant_{\mathcal{J}} t$ if $MsM \subseteq MtM$ (we say that s, t are $\mathcal{J}$-comparable).
  - $s <_{\mathcal{J}} t$ if $s \leqslant_{\mathcal{J}} t$ and $s \not\mathcal{J}\ t$.

An important property

In a finite monoid, $\mathcal{J} = \mathcal{R} \circ \mathcal{L} = \mathcal{L} \circ \mathcal{R}$.

Proof of $\mathcal{J} = \mathcal{R} \circ \mathcal{L}$

$\supseteq$ If $u\ (\mathcal{R} \circ \mathcal{L})\ v$, there is w st. $u\ \mathcal{R}\ w\ \mathcal{L}\ v$ so $u\ \mathcal{J}\ w\ \mathcal{J}\ v$.

Basic but important properties

- $\mathcal{H} \subseteq \mathcal{R}, \mathcal{L} \subseteq \mathcal{J}$.
- $\mathcal{R}$ is a left congruence, $\mathcal{L}$ is a right congruence.
- One can also consider the associated preorders. For instance:
  - $s \leqslant_{\mathcal{J}} t$ if $MsM \subseteq MtM$ (we say that s, t are $\mathcal{J}$-comparable).
  - $s <_{\mathcal{J}} t$ if $s \leqslant_{\mathcal{J}} t$ and $s \not\mathcal{J}\ t$.

An important property

In a finite monoid, $\mathcal{J} = \mathcal{R} \circ \mathcal{L} = \mathcal{L} \circ \mathcal{R}$.

Proof of $\mathcal{J} = \mathcal{R} \circ \mathcal{L}$

$\supseteq$ If $u\ (\mathcal{R} \circ \mathcal{L})\ v$, there is w st. $u\ \mathcal{R}\ w\ \mathcal{L}\ v$ so $u\ \mathcal{J}\ w\ \mathcal{J}\ v$.

$\subseteq$ If $u\ \mathcal{J}\ v$, then $(u = xvt) \wedge (v = yuz)$. Let $w = uz$

$$u = (xy)u(zt) = (xy)^\omega u(zt)^\omega = u(zt)^\omega \leqslant_{\mathcal{R}} uz \leqslant_{\mathcal{R}} u.$$

Hence $u\ \mathcal{R}\ w$ and symmetrically $w\ \mathcal{L}\ v$.

Basic but important properties

- $\mathcal{H} \subseteq \mathcal{R}, \mathcal{L} \subseteq \mathcal{J}$.
- $\mathcal{R}$ is a left congruence, $\mathcal{L}$ is a right congruence.
- One can also consider the associated preorders. For instance:
  - $s \leqslant_{\mathcal{J}} t$ if $MsM \subseteq MtM$ (we say that s, t are $\mathcal{J}$-comparable).
  - $s <_{\mathcal{J}} t$ if $s \leqslant_{\mathcal{J}} t$ and $s \not\mathcal{J}\ t$.

An important property

In a finite monoid, $\mathcal{J} = \mathcal{R} \circ \mathcal{L} = \mathcal{L} \circ \mathcal{R}$.

Proof of $\mathcal{J} = \mathcal{R} \circ \mathcal{L}$

$\supseteq$ If $u\ (\mathcal{R} \circ \mathcal{L})\ v$, there is w st. $u\ \mathcal{R}\ w\ \mathcal{L}\ v$ so $u\ \mathcal{J}\ w\ \mathcal{J}\ v$.

$\subseteq$ If $u\ \mathcal{J}\ v$, then $(u = xvt) \wedge (v = yuz)$. Let $w = uz$

$$u = (xy)u(zt) = (xy)^{\omega}u(zt)^{\omega} = u(zt)^{\omega} \leqslant_{\mathcal{R}} uz \leqslant_{\mathcal{R}} u.$$

Hence $u\ \mathcal{R}\ w$ and symmetrically $w\ \mathcal{L}\ v$.

- With same arguments: $\mathcal{J}$-equivalent and $\mathcal{R}$-comparable implies $\mathcal{R}$-equivalent.

Logics

First Order Logic FO(<)

Syntax: $\mathrm{FO}_A(<)$

$\varphi ::= \bot \mid a(x) \mid \neg\varphi \mid \varphi \vee \varphi \mid x \leqslant y \mid \exists x \varphi$ $\qquad (a \in A,\ x, y \in \mathrm{Var})$

Semantics

- A formula is evaluated on a word $w \in A^*$.
- $\sigma : \mathrm{Var} \to \mathrm{pos}(w) = \{1, 2, \ldots, |w|\}$ is a interpretation of (free) variables.
- $<$ interpreted as the usual ordering between positions.

First Order Logic FO(<)

Syntax: $\mathrm{FO}_A(<)$

$\varphi ::= \bot \mid a(x) \mid \neg\varphi \mid \varphi \vee \varphi \mid x \leqslant y \mid \exists x \varphi$ $\qquad (a \in A,\ x, y \in \mathrm{Var})$

Semantics

- A formula is evaluated on a word $w \in A^*$.
- $\sigma : \mathrm{Var} \to \mathrm{pos}(w) = \{1, 2, \ldots, |w|\}$ is a interpretation of (free) variables.
- $<$ interpreted as the usual ordering between positions.

$$t, \sigma \models a(x) \qquad \text{if} \qquad w_{\sigma(x)} = a$$

First Order Logic FO(<)

Syntax: $\mathrm{FO}_A(<)$

$\varphi ::= \bot \mid a(x) \mid \neg\varphi \mid \varphi \vee \varphi \mid x \leqslant y \mid \exists x \varphi$ $\qquad (a \in A,\ x, y \in \mathrm{Var})$

Semantics

- A formula is evaluated on a word $w \in A^*$.
- $\sigma : \mathrm{Var} \to \mathrm{pos}(w) = \{1, 2, \ldots, |w|\}$ is a interpretation of (free) variables.
- $<$ interpreted as the usual ordering between positions.

$$
\begin{array}{lll}
t, \sigma \models a(x) & \text{if} & w_{\sigma(x)} = a \\
t, \sigma \models \neg\varphi & \text{if} & t, \sigma \not\models \varphi \\
t, \sigma \models \varphi \vee \psi & \text{if} & t, \sigma \models \varphi \ \vee \ t, \sigma \models \psi
\end{array}
$$

First Order Logic FO(<)

Syntax: $\mathrm{FO}_A(<)$

$\varphi ::= \bot \mid a(x) \mid \neg\varphi \mid \varphi \vee \varphi \mid x \leqslant y \mid \exists x \varphi$ $\qquad (a \in A,\ x, y \in \mathrm{Var})$

Semantics

- A formula is evaluated on a word $w \in A^*$.
- $\sigma : \mathrm{Var} \to \mathrm{pos}(w) = \{1, 2, \ldots, |w|\}$ is a interpretation of (free) variables.
- $<$ interpreted as the usual ordering between positions.

$$
\begin{array}{lll}
t, \sigma \models a(x) & \text{if} & w_{\sigma(x)} = a \\
t, \sigma \models \neg\varphi & \text{if} & t, \sigma \not\models \varphi \\
t, \sigma \models \varphi \vee \psi & \text{if} & t, \sigma \models \varphi \ \vee \ t, \sigma \models \psi \\
t, \sigma \models x \leqslant y & \text{if} & \sigma(x) \leqslant \sigma(y)
\end{array}
$$

First Order Logic FO(<)

Syntax: $\mathrm{FO}_A(<)$

$\varphi ::= \bot \mid a(x) \mid \neg\varphi \mid \varphi \vee \varphi \mid x \leqslant y \mid \exists x\varphi$ $\quad (a \in A,\ x, y \in \mathrm{Var})$

Semantics

- A formula is evaluated on a word $w \in A^*$.
- $\sigma : \mathrm{Var} \to \mathrm{pos}(w) = \{1, 2, \ldots, |w|\}$ is a interpretation of (free) variables.
- $<$ interpreted as the usual ordering between positions.

$$
\begin{array}{lll}
t, \sigma \models a(x) & \text{if} & w_{\sigma(x)} = a \\
t, \sigma \models \neg\varphi & \text{if} & t, \sigma \not\models \varphi \\
t, \sigma \models \varphi \vee \psi & \text{if} & t, \sigma \models \varphi \ \vee \ t, \sigma \models \psi \\
t, \sigma \models x \leqslant y & \text{if} & \sigma(x) \leqslant \sigma(y) \\
t, \sigma \models \exists x\varphi & \text{if} & \exists v \in V :\ t, \{\sigma \cup [x \mapsto v]\} \models \varphi
\end{array}
$$

First Order Logic FO(<)

Syntax: $FO_A(<)$

$\varphi ::= \bot \mid a(x) \mid \neg\varphi \mid \varphi \vee \varphi \mid x \leqslant y \mid \exists x\varphi$ $\qquad (a \in A,\ x, y \in \mathrm{Var})$

Semantics

- A formula is evaluated on a word $w \in A^*$.
- $\sigma : \mathrm{Var} \to \mathrm{pos}(w) = \{1, 2, \ldots, |w|\}$ is a interpretation of (free) variables.
- $<$ interpreted as the usual ordering between positions.

$$
\begin{array}{lll}
t, \sigma \models a(x) & \text{if} & w_{\sigma(x)} = a \\
t, \sigma \models \neg\varphi & \text{if} & t, \sigma \not\models \varphi \\
t, \sigma \models \varphi \vee \psi & \text{if} & t, \sigma \models \varphi \ \vee \ t, \sigma \models \psi \\
t, \sigma \models x \leqslant y & \text{if} & \sigma(x) \leqslant \sigma(y) \\
t, \sigma \models \exists x\varphi & \text{if} & \exists v \in V :\ t, \{\sigma \cup [x \mapsto v]\} \models \varphi
\end{array}
$$

Macros

$\forall x\varphi : \neg\exists x\neg\varphi$ $\qquad$ $\varphi \wedge \psi : \neg(\neg\varphi \vee \neg\psi)$ $\qquad$ $\varphi \Rightarrow \psi : \psi \vee \neg\varphi$ $\qquad$ $\varphi \Leftrightarrow \psi \ldots$

First Order Logic FO(<)

Syntax: $\mathrm{FO}_A(<)$

$\varphi ::= \bot \mid a(x) \mid \neg\varphi \mid \varphi \vee \varphi \mid x \leqslant y \mid \exists x\varphi$ $\qquad (a \in A,\ x, y \in \mathrm{Var})$

Semantics

- A formula is evaluated on a word $w \in A^*$.
- $\sigma : \mathrm{Var} \to \mathrm{pos}(w) = \{1, 2, \ldots, |w|\}$ is a interpretation of (free) variables.
- $<$ interpreted as the usual ordering between positions.

$$
\begin{array}{lll}
t, \sigma \models a(x) & \text{if} & w_{\sigma(x)} = a \\
t, \sigma \models \neg\varphi & \text{if} & t, \sigma \not\models \varphi \\
t, \sigma \models \varphi \vee \psi & \text{if} & t, \sigma \models \varphi \ \vee \ t, \sigma \models \psi \\
t, \sigma \models x \leqslant y & \text{if} & \sigma(x) \leqslant \sigma(y) \\
t, \sigma \models \exists x\varphi & \text{if} & \exists v \in V :\ t, \{\sigma \cup [x \mapsto v]\} \models \varphi
\end{array}
$$

Macros

$\forall x\varphi : \neg\exists x\neg\varphi$ $\qquad$ $\varphi \wedge \psi : \neg(\neg\varphi \vee \neg\psi)$ $\qquad$ $\varphi \Rightarrow \psi : \psi \vee \neg\varphi$ $\qquad$ $\varphi \Leftrightarrow \psi \ldots$

A sentence $\varphi \in \mathrm{FO}(<)$ defines the language $\mathcal{L}(\varphi) = \{w \mid w \models \varphi\} \subseteq A^*$.

First Order Logic — Examples

Examples of FO($<$) formulas

- $\varphi = \forall x(a(x) \vee b(x)) \wedge \forall y \forall z\,(z = y + 1) \Rightarrow [a(y) \Leftrightarrow b(z)]$

$$bababababa \models \varphi$$

$$abaabaabaaba \not\models \varphi$$

First Order Logic — Examples

Examples of FO($<$) formulas

- $\varphi = \forall x(a(x) \vee b(x)) \wedge \forall y \forall z\,(z = y + 1) \Rightarrow [a(y) \Leftrightarrow b(z)]$

$$bababababa \models \varphi$$
$$abaabaabaaba \not\models \varphi$$

- $y = x + 1 \stackrel{\text{def}}{=} (x < y) \wedge (\neg\exists z, x < z < y)$.
  Conversely, $<$ is not expressible in FO($+1$) **[Th82]**. How to prove it?
- $\text{first}(x) \stackrel{\text{def}}{=} \neg\exists z, z < x$.
- $(\min \subseteq B) \stackrel{\text{def}}{=} \bigvee_{b \in B} \exists x[b(x) \wedge \text{first}(x)]$.

First Order Logic — Examples

Examples of FO($<$) formulas

- $\varphi = \forall x(a(x) \vee b(x)) \wedge \forall y \forall z\,(z = y + 1) \Rightarrow [a(y) \Leftrightarrow b(z)]$

$$bababababa \models \varphi$$
$$abaabaabaaba \not\models \varphi$$

- $y = x + 1 \stackrel{\text{def}}{=} (x < y) \wedge (\neg \exists z, x < z < y)$.
  Conversely, $<$ is not expressible in FO($+1$) **[Th82]**. How to prove it?
- $\text{first}(x) \stackrel{\text{def}}{=} \neg \exists z, z < x$.
- $(\min \subseteq B) \stackrel{\text{def}}{=} \bigvee_{b \in B} \exists x[b(x) \wedge \text{first}(x)]$.
- $\mathcal{L}(\varphi \wedge \text{first} = \{a\} \wedge \text{last} = \{b\}) = (ab)^+$.
- Hence, $(ab)^+$ is expressible in FO($<$). How to automatically get a formula?
- The language $(aa)^+$ cannot be expressed in FO($<$)! How to prove it?

Linear Temporal Logic LTL (Pnueli 1977)

Syntax: LTL$(A, \mathsf{X}, \mathsf{U})$

$$\varphi ::= \bot \mid a \mid \neg\varphi \mid \varphi \vee \varphi \mid \mathsf{X}\,\varphi \mid \varphi\,\mathsf{U}\,\varphi$$

Linear Temporal Logic LTL (Pnueli 1977)

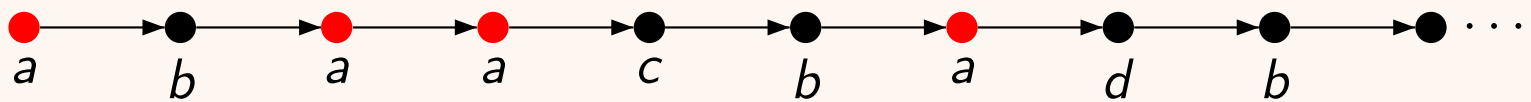
Syntax: LTL$(A, \mathsf{X}, \mathsf{U})$

$$\varphi ::= \bot \mid a \mid \neg\varphi \mid \varphi \vee \varphi \mid \mathsf{X}\,\varphi \mid \varphi\,\mathsf{U}\,\varphi$$

Semantics: $w \in A^*$ and $i \in \mathrm{pos}(w)$

$w, i \models a$ if $w_i = a$

Example



Linear Temporal Logic LTL (Pnueli 1977)

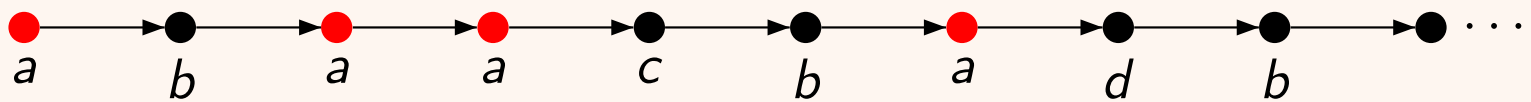
Syntax: LTL$(A, \mathsf{X}, \mathsf{U})$

$$\varphi ::= \bot \mid a \mid \neg\varphi \mid \varphi \vee \varphi \mid \mathsf{X}\,\varphi \mid \varphi\,\mathsf{U}\,\varphi$$

Semantics: $w \in A^*$ and $i \in \mathrm{pos}(w)$

$$\begin{array}{lll} w, i \models a & \text{if} & w_i = a \\ w, i \models \neg\varphi & \text{if} & w, i \not\models \varphi \\ w, i \models \varphi \vee \psi & \text{if} & w, i \models \varphi \ \vee \ w, i \models \psi \end{array}$$

Example



Linear Temporal Logic LTL (Pnueli 1977)

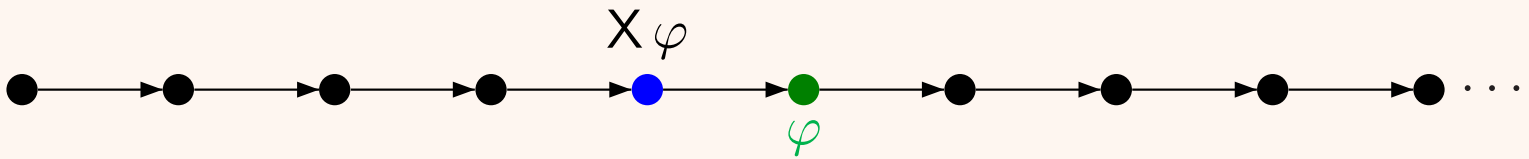
Syntax: LTL$(A, \mathsf{X}, \mathsf{U})$

$$\varphi ::= \bot \mid a \mid \neg\varphi \mid \varphi \vee \varphi \mid \mathsf{X}\varphi \mid \varphi \,\mathsf{U}\, \varphi$$

Semantics: $w \in A^*$ and $i \in \mathrm{pos}(w)$

$$\begin{array}{lll} w, i \models a & \text{if} & w_i = a \\ w, i \models \neg\varphi & \text{if} & w, i \not\models \varphi \\ w, i \models \varphi \vee \psi & \text{if} & w, i \models \varphi \ \vee \ w, i \models \psi \\ w, i \models \mathsf{X}\varphi & \text{if} & w, i+1 \models \varphi \end{array}$$

Example



Linear Temporal Logic LTL (Pnueli 1977)

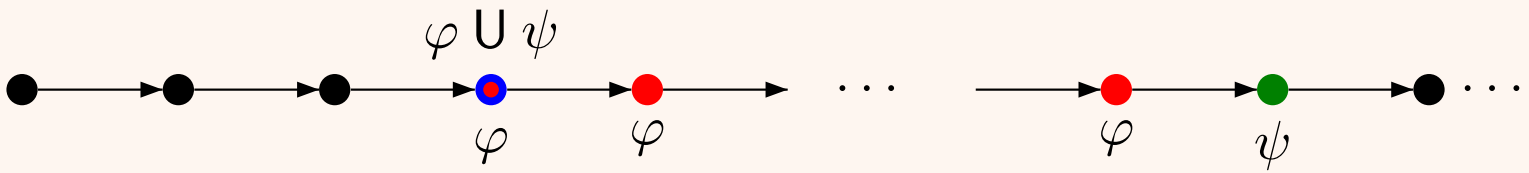
Syntax: LTL$(A, \mathsf{X}, \mathsf{U})$

$$\varphi ::= \bot \mid a \mid \neg\varphi \mid \varphi \vee \varphi \mid \mathsf{X}\,\varphi \mid \varphi\,\mathsf{U}\,\varphi$$

Semantics: $w \in A^*$ and $i \in \mathrm{pos}(w)$

$$\begin{array}{lll}
w, i \models a & \text{if} & w_i = a \\
w, i \models \neg\varphi & \text{if} & w, i \not\models \varphi \\
w, i \models \varphi \vee \psi & \text{if} & w, i \models \varphi \ \vee \ w, i \models \psi \\
w, i \models \mathsf{X}\,\varphi & \text{if} & w, i+1 \models \varphi \\
w, i \models \varphi\,\mathsf{U}\,\psi & \text{if} & \exists k.\ i \leqslant k \ \wedge \ w, k \models \psi \ \wedge \ \forall j.\ (i \leqslant j < k) \Rightarrow w, j \models \varphi
\end{array}$$

Example



Linear Temporal Logic LTL (Pnueli 1977)

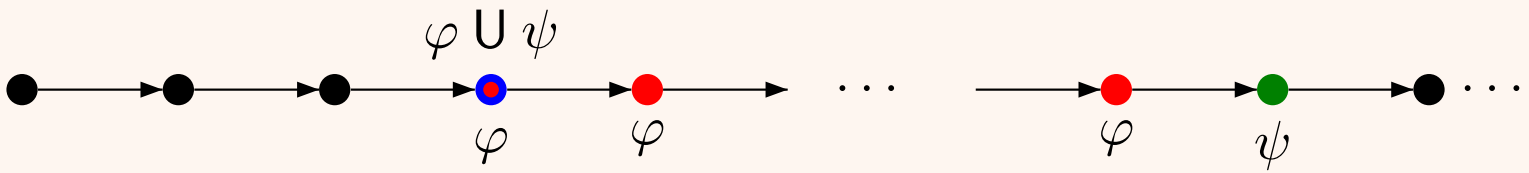
Syntax: LTL$(A, \mathsf{X}, \mathsf{U})$

$$\varphi ::= \bot \mid a \mid \neg\varphi \mid \varphi \vee \varphi \mid \mathsf{X}\,\varphi \mid \varphi\,\mathsf{U}\,\varphi$$

Semantics: $w \in A^*$ and $i \in \mathrm{pos}(w)$

$$\begin{array}{lll}
w, i \models a & \text{if} & w_i = a \\
w, i \models \neg\varphi & \text{if} & w, i \not\models \varphi \\
w, i \models \varphi \vee \psi & \text{if} & w, i \models \varphi \ \vee \ w, i \models \psi \\
w, i \models \mathsf{X}\,\varphi & \text{if} & w, i+1 \models \varphi \\
w, i \models \varphi\,\mathsf{U}\,\psi & \text{if} & \exists k.\ i \leqslant k \ \wedge \ w, k \models \psi \ \wedge \ \forall j.\ (i \leqslant j < k) \Rightarrow w, j \models \varphi
\end{array}$$

Example



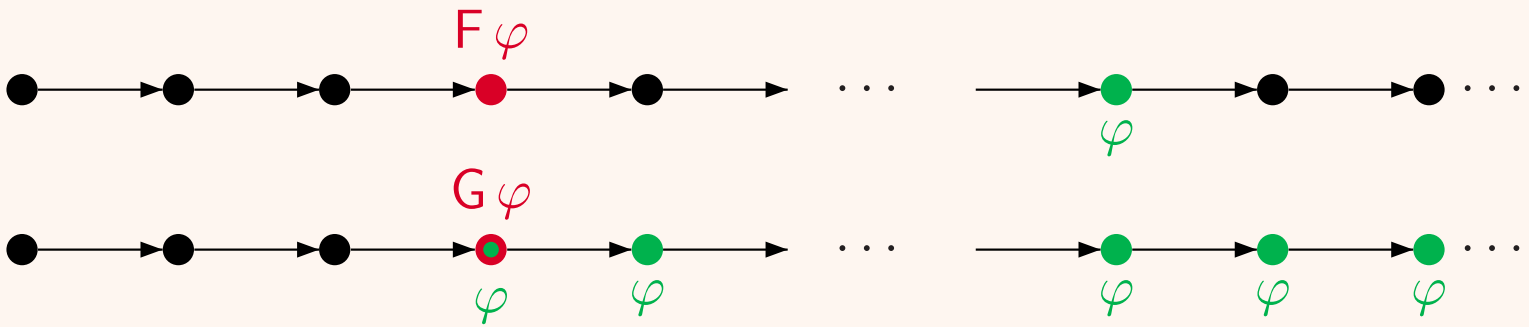
$\varphi \in \mathrm{LTL}(A, \mathsf{X}, \mathsf{U})$ defines the language $\mathcal{L}(\varphi) = \{w \mid w, 1 \models \varphi\}$.

Useful LTL Macros

Macros

- $\mathsf{F}\,\varphi = \top\,\mathsf{U}\,\varphi$ (eventually).
- $\mathsf{G}\,\varphi = \neg\,\mathsf{F}\,\neg\varphi$ (always in the future).

Example



Examples of LTL properties

Some languages definable in LTL

- $\mathsf{F}\, a$ defines the language $A^* a A^*$.
- $\mathsf{G}\, a$ defines the language a^+.
- $(ab)^+$ is also definable: $\mathsf{G}(a \vee b) \wedge a \wedge \mathsf{F}(b \wedge \neg \mathsf{X} \top) \wedge \mathsf{G}(a \leftrightarrow \neg \mathsf{X} b)$

Examples of LTL properties

Some languages definable in LTL

- $\mathsf{F}\, a$ defines the language $A^* a A^*$.
- $\mathsf{G}\, a$ defines the language a^+.
- $(ab)^+$ is also definable: $\mathsf{G}(a \vee b) \wedge a \wedge \mathsf{F}(b \wedge \neg \mathsf{X} \top) \wedge \mathsf{G}(a \leftrightarrow \neg \mathsf{X} b)$

Examples of LTL properties

Some languages definable in LTL

- F a defines the language $A^* a A^*$.
- G a defines the language a^+.
- $(ab)^+$ is also definable: $\mathsf{G}(a \vee b) \wedge a \wedge \mathsf{F}(b \wedge \neg \mathsf{X} \top) \wedge \mathsf{G}(a \leftrightarrow \neg \mathsf{X} b)$

Examples of LTL properties

Some languages definable in LTL

- $\mathsf{F}\, a$ defines the language $A^* a A^*$.
- $\mathsf{G}\, a$ defines the language a^+.
- $(ab)^+$ is also definable: $\mathsf{G}(a \vee b) \wedge a \wedge \mathsf{F}(b \wedge \neg \mathsf{X} \top) \wedge \mathsf{G}(a \leftrightarrow \neg \mathsf{X} b)$

Examples of LTL properties

Some languages definable in LTL

- F a defines the language A^*aA^*.
- G a defines the language a^+.
- $(ab)^+$ is also definable: $\mathsf{G}(a \vee b) \wedge a \wedge \mathsf{F}(b \wedge \neg \mathsf{X} \top) \wedge \mathsf{G}(a \leftrightarrow \neg \mathsf{X} b)$
- Can $(aa)^+$ be defined in LTL? It does not seem so, but how to prove it?
- On A^*, LTL formulas can be easily translated in FO($<$) formulas.
- Does the converse hold? Difficult because LTL only handles one free variable.

Syntactic monoids help!

Expressiveness results

FO with one variable

$\mathrm{FO}^1(<) =$ where only one variable name is allowed.

Example: FO with one variable

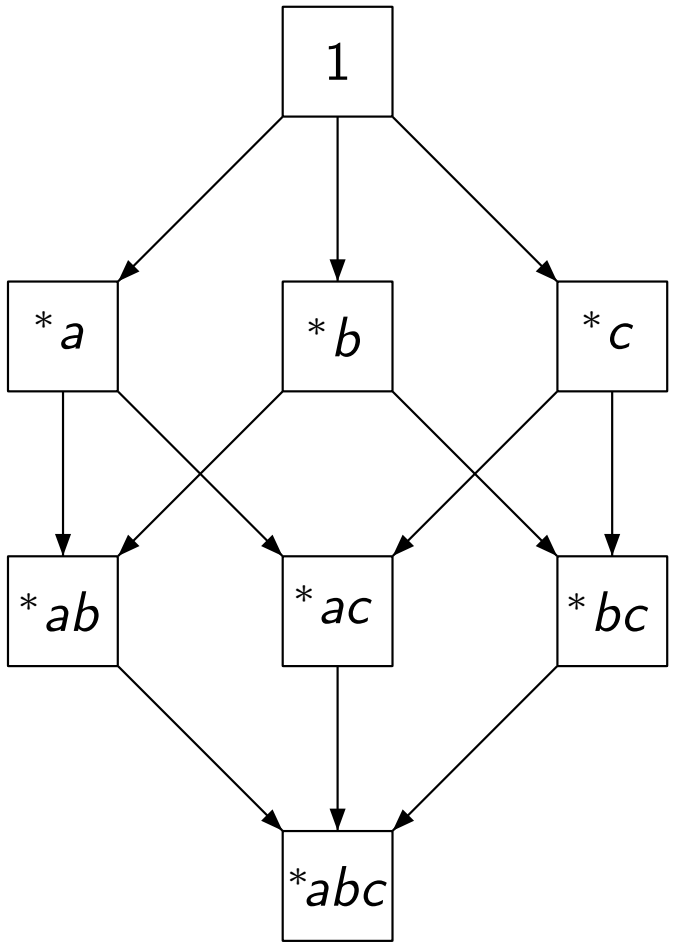
The following conditions are equivalent:

1. L is definable by an $\mathrm{FO}^1(<)$ sentence.
2. L is a boolean combination of languages of the form A^*aA^*.
3. The syntactic monoid of L is idempotent and commutative:

$$\forall s, t \in M(L), \qquad s = s^2 \text{ and } st = ts.$$

Typical idempotent commutative monoid

- $(2^A, \cup, \varnothing)$.
- Hasse diagram of representation in $\mathcal{J}$-classes:



FO with one variable

Example: FO with one variable

The following conditions are equivalent:

1. L is definable by an FO^1 sentence.
2. L is a boolean combination of languages of the form A^*aA^*.
3. The syntactic monoid of L is idempotent and commutative:

$$\forall s, t \in M(L), \qquad s = s^2 \text{ and } st = ts.$$

Proof of $1 \Rightarrow 2$

- In FO^1, binary symbol $<$ is useless: $x \leqslant x \equiv \top$.
- Every formula equivalent to $\exists x \varphi(x)$, with φ quantifier-free.
- Get rid of negation in atoms: $\neg b(x) \equiv \bigvee_{b \neq a} a(x)$. If $a \neq b$: $a(x) \wedge b(x) \equiv \bot$.
- Hence $\bigwedge_{b \in B} b(x) \wedge \bigwedge_{c \in C} \neg c(x)$ reduce to a single predicate $a(x)$ or $\bot$,
- $\exists x.(a(x) \vee b(x)) \equiv \exists x.a(x) \vee \exists x.b(x) \quad \rightsquigarrow$ reduce to boolean combination of formulas $\varphi = \exists x.a(x)$.
- $\mathcal{L}(\exists x.a(x)) = A^*aA^*$.

FO with one variable

Example: FO with one variable

The following conditions are equivalent:

1. L is definable by an FO^1 sentence.
2. L is a boolean combination of languages of the form A^*aA^*.
3. The syntactic monoid of L is idempotent and commutative:

$$\forall s, t \in M(L), \qquad s = s^2 \text{ and } st = ts.$$

Proof of $2 \Rightarrow 1$

- $\mathcal{L}(\exists x.a(x)) = A^*aA^*$.

FO with one variable

Example: FO with one variable

The following conditions are equivalent:

1. L is definable by an FO^1 sentence.
2. L is a boolean combination of languages of the form A^*aA^*.
3. The syntactic monoid of L is idempotent and commutative:

$$\forall s, t \in M(L), \qquad s = s^2 \text{ and } st = ts.$$

Proof of $2 \Rightarrow 3$

- The syntactic monoid $M(A^*aA^*) = \{1, a\}$ is idempotent and commutative.
- Idempotency and commutativity are inherited by boolean combinations.

FO with one variable

Example: FO with one variable

The following conditions are equivalent:

1. L is definable by an FO^1 sentence.
2. L is a boolean combination of languages of the form A^*aA^*.
3. The syntactic monoid of L is idempotent and commutative:

$$\forall s, t \in M(L), \qquad s = s^2 \text{ and } st = ts.$$

Proof of 3 $\Rightarrow$ 2

- Let $\varphi : A^* \to M$ recognizing L, with M idempotent and commutative.
- $L = \bigcup_{s \in \varphi(L)} \varphi^{-1}(s) \rightsquigarrow$ enough to show $\varphi^{-1}(s) \in \mathrm{Bool}(\{A^*aA^* \mid a \in A\})$.
- Since M idempotent commutative: $\mathrm{alph}(u) = \mathrm{alph}(v) \Rightarrow \varphi(u) = \varphi(v)$.
- Let $[\mathrm{alph} = B]$ be the set of words of alphabet exactly B.
- $\varphi^{-1}(s) = \bigcup_{B \in A_s} [\mathrm{alph} = B]$ for $A_s = \{B \subseteq A \mid \exists u \in \varphi^{-1}(s), \mathrm{alph}(u) = B\}$.
- $[\mathrm{alph} = B] = \bigcap_{b \in B} A^*bA^* \setminus \bigcup_{c \notin B} A^*cA^*$.

Piecewise testable languages

- For $u = a_1 \cdots a_n$ with $a_i \in A$, let

$$L(u) = A^* a_1 A^* a_2 \cdots A^* a_n A^*.$$

  $L(u)$ is the set of all words having u as a (scattered) subword.
- Write $u \sqsubseteq v$ if $v \in L(u)$: u is a (scattered) subword of v.
- Piecewise testable language: boolean combination of languages $L(u)$, $u \in A^*$.
- Examples:
  - $B^* = A^* \setminus \bigcup_{c \notin B} A^* c A^*$
  - $(\text{alph} = B) = \bigcap_{b \in B} A^* b A^* \setminus \bigcup_{c \notin B} A^* c A^*$
- Questions: how to decide whether a language is piecewise testable?

$B\Sigma_1(<)$ fragment of $\mathrm{FO}(<)$

- Piecewise testable languages have a logical characterization.
- $B\Sigma_1(<)$: fragment of $\mathrm{FO}(<)$ consisting of Boolean closure of formulas of the form

$$\exists x_1 \ldots \exists x_k \varphi(x_1, \ldots, x_k)$$

with φ quantifier-free.

$B\Sigma_1(<)$ fragment of FO(<)

- Piecewise testable languages have a logical characterization.
- $B\Sigma_1(<)$: fragment of FO(<) consisting of Boolean closure of formulas of the form

$$\exists x_1 \ldots \exists x_k \varphi(x_1, \ldots, x_k)$$

  with φ quantifier-free.
- Clearly $B\Sigma_1(<)$ can express piecewise testability:

$$A^* a_1 A^* a_2 \cdots A^* a_n A^* = \mathcal{L}(\exists x_1 \ldots \exists x_n \wedge \bigwedge (x_i < x_{i+1}) \wedge a_i(x_i)).$$

$B\Sigma_1(<)$ fragment of FO(<)

- Piecewise testable languages have a logical characterization.
- $B\Sigma_1(<)$: fragment of FO(<) consisting of Boolean closure of formulas of the form

$$\exists x_1 \dots \exists x_k \varphi(x_1, \dots, x_k)$$

  with φ quantifier-free.
- Clearly $B\Sigma_1(<)$ can express piecewise testability:

$$A^* a_1 A^* a_2 \cdots A^* a_n A^* = \mathcal{L}(\exists x_1 \dots \exists x_n \wedge \bigwedge (x_i < x_{i+1}) \wedge a_i(x_i)).$$

- Conversely, using disjunctive normal form and $\exists \vec{x} \bigvee_i \varphi_i \equiv \bigvee_i \exists \vec{x} \varphi_i$, one can start from $\exists \vec{x} \varphi$ where φ is a conjunction of atoms.
- One can get rid of negative atoms.
- Therefore, φ fixes conditions on the order of x_i's and their labels.
- This defines a piecewise testable language.

Simon's theorem for piecewise testable languages

Piecewise testable languages, $B\Sigma_1$, $\mathcal{J}$-trivial monoids (Simon-Thomas)

The following conditions are equivalent.

1. L is $B\Sigma_1(<)$ definable.
2. L is piecewise testable.
3. $M(L)$ is finite and $\mathcal{J}$-trivial.

Recall: $\mathcal{J}$-trivial means $u \mathcal{J} v \Rightarrow u = v$.

Corollary

$B\Sigma_1(<)$ definability is decidable.

Example: $(ab)^+$ is not piecewise testable (on any alphabet containing $\{a, b\}$).

Simon's theorem for piecewise testable languages: proof (1)

- Note that $L(u)$ is the set of words having u as a (scattered) subword.
- Define $u \sim_k v$ if u and v have the same (scattered) subwords of length $\leqslant k$.
  Example: $abba \sim_2 baba \not\sim_2 aabb$.
- $\sim_k \subseteq A^* \times A^*$ is a congruence of finite index (at most $|2^{A^{\leqslant k}}|$ classes).

Simon's theorem for piecewise testable languages: proof (1)

- Note that $L(u)$ is the set of words having u as a (scattered) subword.
- Define $u \sim_k v$ if u and v have the same (scattered) subwords of length $\leqslant k$.
  Example: $abba \sim_2 baba \not\sim_2 aabb$.
- $\sim_k \subseteq A^* \times A^*$ is a congruence of finite index (at most $|2^{A^{\leqslant k}}|$ classes).

TFAE:

1. L is piecewise testable.
2. There exists $k \geqslant 0$ such that $\sim_k \subseteq \sim_L$.

Proof of 1 $\Rightarrow$ 2

- $L = \bigcup\Big(\bigcap L(u_i) \cap \bigcap (A^* \setminus L(v_j))\Big)$. *(Finite union and intersections).*
- Let $k = \max\{|u_i|, |v_j|\}$.
- If $u \sim_k v$, then $xuy \sim_k xvy$ for all x, y.
- So by def. of k, xuy, xvy belong to the same languages $L(u_i)$, $L(v_j)$.
- Hence $xuy \in L$ iff $xvy \in L$, therefore $u \sim_L v$.

Simon's theorem for piecewise testable languages: proof (1)

- Note that $L(u)$ is the set of words having u as a (scattered) subword.
- Define $u \sim_k v$ if u and v have the same (scattered) subwords of length $\leqslant k$.
  Example: $abba \sim_2 baba \not\sim_2 aabb$.
- $\sim_k \subseteq A^* \times A^*$ is a congruence of finite index (at most $|2^{A^{\leqslant k}}|$ classes).

TFAE:

1. L is piecewise testable.
2. There exists $k \geqslant 0$ such that $\sim_k \subseteq \sim_L$.

Proof of 2 $\Rightarrow$ 1

- $L = \eta_L^{-1}(\eta_L(L)) = \bigcup_{s \in \eta_L(L)} \eta_L^{-1}(s)$ is a union of $\sim_L$-classes.
- If $\sim_k \subseteq \sim_L$, then L is a (finite) union of $\sim_k$-classes.
- Enough to show that any $\sim_k$-class is piecewise testable.
- The $\sim_k$ class of a word u is $\bigcap_{w \sqsubseteq u, |w| \leqslant k} L(w) \cap \bigcap_{w \not\sqsubseteq u, |w| \leqslant k} A^* \setminus L(w)$.

Simon's theorem for piecewise testable languages: proof (2)

Proof of: L is piecewise testable $\Longrightarrow$ $M(L)$ is $\mathcal{J}$-trivial.

- Write $\underline{u} = \eta_L(u)$ where $\eta_L : A^* \to M(L)$ is the syntactic morphism.
- Assume $\underline{u}\ \mathcal{J}\ \underline{v}$. We want $\underline{u} = \underline{v}$.

Simon's theorem for piecewise testable languages: proof (2)

Proof of: L is piecewise testable $\Longrightarrow$ $M(L)$ is $\mathcal{J}$-trivial.

- Write $\underline{u} = \eta_L(u)$ where $\eta_L : A^* \to M(L)$ is the syntactic morphism.
- Assume $\underline{u}\ \mathcal{J}\ \underline{v}$. We want $\underline{u} = \underline{v}$.
- By definition of $\mathcal{J}$: $\underline{u} = \underline{xvt}$ and $\underline{v} = \underline{yuz}$. Hence $\underline{u} = \underline{xyuzt} = (\underline{xy})^n \underline{u} (\underline{zt})^n$.
- Let k be such that $\sim_k \subseteq \sim_L$.
- Let n be such that $(xy)^n u (zt)^n \sim_k (xy)^{n+1} u (zt)^{n+1}$.
- Therefore, $(xy)^n u (zt)^n \sim_k y (xy)^n u (zt)^n z$
- Since $\sim_k \subseteq \sim_L$, we get $\underline{u} = (\underline{xy})^n \underline{u} (\underline{zt})^n = \underline{y} (\underline{xy})^n \underline{u} (\underline{zt})^n \underline{z} = \underline{v}$.

Simon's theorem for piecewise testable languages: proof (3)

- Recall that $s \leqslant_{\mathcal{J}} t$ if $s = xty$ for some $x, y \in M$.
- Note that if $M(L)$ is $\mathcal{J}$-trivial, then $\leqslant_J$ is a partial order on $M(L)$.
- By definition of $\leqslant_{\mathcal{J}}$, multiplying an element yields an element
  - either equal to the original one,
  - or strictly smaller.

Simon's theorem for piecewise testable languages: proof (4)

$M(L)$ $\mathcal{J}$-trivial $\Longrightarrow$ L piecewise testable: Proof of O. Klima (1)

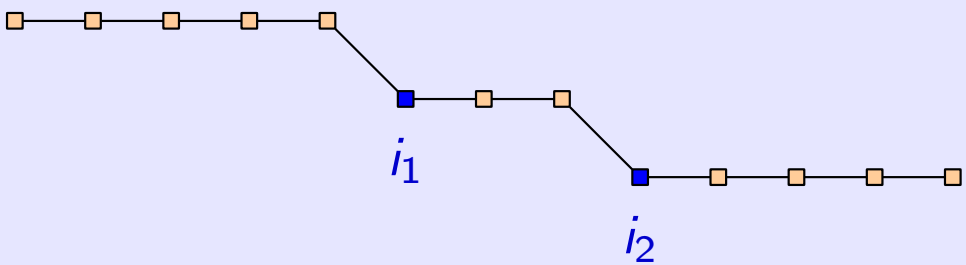
- Let $m =$ size of largest $<_{\mathcal{J}}$-chain, and $k = 2m - 2$. One shows $\sim_k \subseteq \sim_L$.
- Let $u \sim_k v$ with $u = u(1) \cdots u(p)$ and $v = v(1) \cdots v(q)$.
- $u[\ell_1, \ldots, \ell_k] \stackrel{\text{def}}{=}$ word made of letters at positions $\{\ell_1, \ldots, \ell_k\}$ from left to right
- For $i \leqslant j$ $u[i \bullet j] \stackrel{\text{def}}{=} u[i, i+1, \ldots, j]$.

Simon's theorem for piecewise testable languages: proof (4)

$M(L)$ $\mathcal{J}$-trivial $\Longrightarrow$ L piecewise testable: Proof of O. Klima (1)

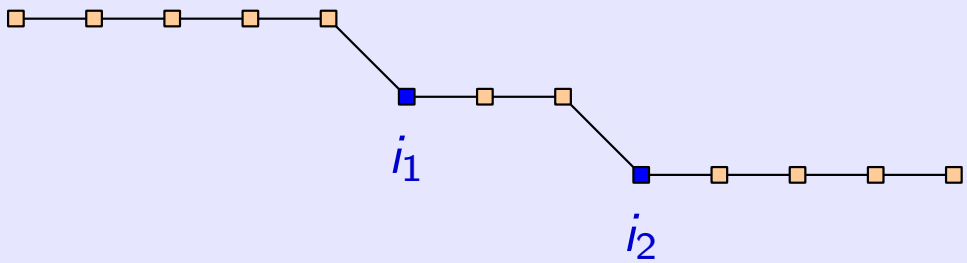
- Let $m =$ size of largest $<_{\mathcal{J}}$-chain, and $k = 2m - 2$. One shows $\sim_k \subseteq \sim_L$.
- Let $u \sim_k v$ with $u = u(1) \cdots u(p)$ and $v = v(1) \cdots v(q)$.
- $u[\ell_1, \ldots, \ell_k] \stackrel{\text{def}}{=}$ word made of letters at positions $\{\ell_1, \ldots, \ell_k\}$ from left to right
- For $i \leqslant j$ $u[i \bullet j] \stackrel{\text{def}}{=} u[i, i+1, \ldots, j]$.

- Position i in u is blue if $\underline{u}[1 \bullet (i-1)] \cdot \underline{u}(i) <_{\mathcal{J}} \underline{u}[1 \bullet (i-1)]$.
- At most $m - 1$ blue positions $i_1 < \cdots < i_r$ in u.



Simon's theorem for piecewise testable languages: proof (4)

$M(L)$ $\mathcal{J}$-trivial $\Longrightarrow$ L piecewise testable: Proof of O. Klima (2)



- Claim1: any subword of u containing all blue positions is $\sim_L$-equvalent to u.
  Indeed for instance, by definition of blue indices, for any $i_\ell < i < i_{\ell+1}$

$$\underline{u}[1 \bullet i_\ell] \;=\; \underline{u}[1 \bullet (i-1)] \;=\; \underline{u}[1 \bullet i]$$
$$\Longrightarrow\; \underline{u}[1 \bullet i_\ell]\underline{u}(i) = \underline{u}[1 \bullet i_\ell]$$

  This is where the hypothesis $M(L)$ is $\mathcal{J}$-trivial is used.
- For same reason: $i_1 < \cdots < i_r$ carries leftmost occurrence of $u[i_1 \bullet i_r]$.

Simon's theorem for piecewise testable languages: proof (4)

$M(L)$ $\mathcal{J}$-trivial $\Longrightarrow$ L piecewise testable: Proof of O. Klima (2)

- $u \sim_k v \Rightarrow u[i_1, \ldots, i_r] \sqsubseteq v = b_1 \cdots b_q$.
- Let $v(\tilde{i}_i) \cdots v(\tilde{i}_r)$ be its leftmost occurrence in v. Call $\tilde{i}_1, \ldots, \tilde{i}_r$ blue in v.

Simon's theorem for piecewise testable languages: proof (4)

$M(L)$ $\mathcal{J}$-trivial $\Longrightarrow$ L piecewise testable: Proof of O. Klima (2)

- $u \sim_k v \Rightarrow u[i_1, \ldots, i_r] \sqsubseteq v = b_1 \cdots b_q$.
- Let $v(\tilde{i}_i) \cdots v(\tilde{i}_r)$ be its leftmost occurrence in v. Call $\tilde{i}_1, \ldots, \tilde{i}_r$ blue in v.
- Dually, position j in v is red if $\underline{v}[j-1, q] >_{\mathcal{J}} \underline{v}(j)\underline{v}[j-1, q]$.
- At most $m-1$ red positions $j_1 < \cdots < j_s$ in v.
- Corresponding red rightmost positions in u: $\tilde{j}_1, \ldots, \tilde{j}_s$.

Simon's theorem for piecewise testable languages: proof (4)

$M(L)$ $\mathcal{J}$-trivial $\Longrightarrow$ L piecewise testable: Proof of O. Klima (2)

- $u \sim_k v \Rightarrow u[i_1, \ldots, i_r] \sqsubseteq v = b_1 \cdots b_q$.
- Let $v(\tilde{i}_i) \cdots v(\tilde{i}_r)$ be its leftmost occurrence in v. Call $\tilde{i}_1, \ldots, \tilde{i}_r$ blue in v.
- Dually, position j in v is red if $\underline{v}[j-1, q] >_{\mathcal{J}} \underline{v}(j)\underline{v}[j-1, q]$.
- At most $m-1$ red positions $j_1 < \cdots < j_s$ in v.
- Corresponding red rightmost positions in u: $\tilde{j}_1, \ldots, \tilde{j}_s$.
- Claim2: $u[i_1, \ldots, i_r, \tilde{j}_1, \ldots, \tilde{j}_s] = v[\tilde{i}_1, \ldots, \tilde{i}_r, j_1, \ldots, j_s]$.
  - The blue (red) positions carry the same word in u and v.
  - The way they are shuffled in u and v only depends on (small) subwords.

Simon's theorem for piecewise testable languages: proof (4)

$M(L)$ $\mathcal{J}$-trivial $\Longrightarrow$ L piecewise testable: Proof of O. Klima (2)

- $u \sim_k v \Rightarrow u[i_1, \ldots, i_r] \sqsubseteq v = b_1 \cdots b_q$.
- Let $v(\tilde{i}_i) \cdots v(\tilde{i}_r)$ be its leftmost occurrence in v. Call $\tilde{i}_1, \ldots, \tilde{i}_r$ blue in v.
- Dually, position j in v is red if $\underline{v}[j-1, q] >_{\mathcal{J}} \underline{v}(j)\underline{v}[j-1, q]$.
- At most $m-1$ red positions $j_1 < \cdots < j_s$ in v.
- Corresponding red rightmost positions in u: $\tilde{j}_1, \ldots, \tilde{j}_s$.
- Claim2: $u[i_1, \ldots, i_r, \tilde{j}_1, \ldots, \tilde{j}_s] = v[\tilde{i}_1, \ldots, \tilde{i}_r, j_1, \ldots, j_s]$.
  - The blue (red) positions carry the same word in u and v.
  - The way they are shuffled in u and v only depends on (small) subwords.
- Consider in u a blue index i_h and a red one $\tilde{j}_\ell$, with for simplicity $u(i_h) \neq u(\tilde{j}_\ell)$.

$$
\begin{aligned}
i_h < \tilde{j}_\ell &\Leftrightarrow u(i_1) \cdots u(i_h) u(\tilde{j}_\ell) \cdots u(\tilde{j}_s) \sqsubseteq u \\
&\Leftrightarrow v(\tilde{i}_1) \cdots v(\tilde{i}_h) v(j_\ell) \cdots v(j_s) \sqsubseteq v \\
&\Leftrightarrow \tilde{i}_h < j_\ell
\end{aligned}
$$

Simon's theorem for piecewise testable languages: proof (4)

$M(L)$ $\mathcal{J}$-trivial $\Longrightarrow$ L piecewise testable: Proof of O. Klima (3)

- Claim1: any subword of u containing all blue positions is $\sim_L$-equvalent to u.
- Claim2: $u[i_1, \ldots, i_r, \tilde{j}_1, \ldots, \tilde{j}_s] = v[\tilde{i}_1, \ldots, \tilde{i}_r, j_1, \ldots, j_s]$
- Hence

$$u \sim_L u[i_1, \ldots, i_r, \tilde{j}_1, \ldots, \tilde{j}_s] = v[\tilde{i}_1, \ldots, \tilde{i}_r, j_1, \ldots, j_s] \sim_L \underline{v}$$

that is

$$u \sim_L v.$$

Star-free languages

- Star-free languages: built from $\varnothing$ and letters using finitely many times boolean operations and product.
  1. $\varnothing \in \mathrm{SF}(A^*)$ and $\{a\} \in \mathrm{SF}(A^*)$,
  2. $K, L \in \mathrm{SF}(A^*) \Rightarrow K \cup L \in \mathrm{SF}(A^*)$,
  3. $K, L \in \mathrm{SF}(A^*) \Rightarrow A^* \setminus K \in \mathrm{SF}(A^*)$,
  4. $K, L \in \mathrm{SF}(A^*) \Rightarrow KL \in \mathrm{SF}(A^*)$.

Star-free languages

- Star-free languages: built from $\varnothing$ and letters using finitely many times boolean operations and product.
  1. $\varnothing \in \mathrm{SF}(A^*)$ and $\{a\} \in \mathrm{SF}(A^*)$,
  2. $K, L \in \mathrm{SF}(A^*) \Rightarrow K \cup L \in \mathrm{SF}(A^*)$,
  3. $K, L \in \mathrm{SF}(A^*) \Rightarrow A^* \setminus K \in \mathrm{SF}(A^*)$,
  4. $K, L \in \mathrm{SF}(A^*) \Rightarrow KL \in \mathrm{SF}(A^*)$.

- Examples of star-free languages:
  - $A^* = A^* \setminus \varnothing$,
  - Finite languages,
  - Piecewise testable languages,
  - $B^* = A^* \setminus \bigcup_{c \notin B} A^* c A^*$,
  - $(ab)^+ = aA^* \cap A^*b \cap (A^* \setminus A^*(aa \cup bb)A^*)$

    This is not piecewise testable, because the syntactic monoid is not $\mathcal{J}$-trivial.
  - Star-free languages are regular, but $(aa)^*$ is not star-free. How to prove it?

Aperiodic monoids

- A finite monoid is aperiodic if it has only trivial subgroups, or equivalently:

$$\exists n, \forall s \in M, s^n = s^{n+1}$$

  or equivalently again

$$\forall s \in M, s^\omega = s^{\omega+1}$$

- A language is aperiodic if it is recognized by a finite aperiodic monoid.

The Schützenberger-Kamp-McNaughton-Papert theorem

Theorem Schützenberger-Kamp-McNaughton-Papert

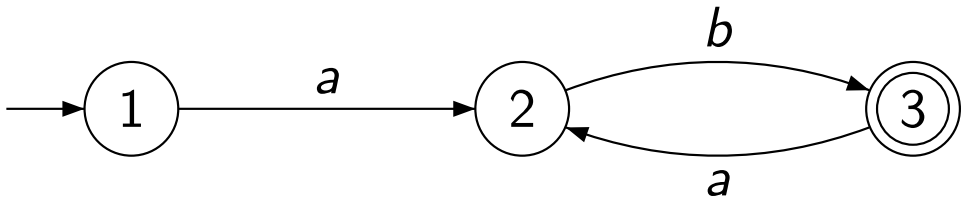
For a word language $L \subseteq A^*$, TFAE

1. L can be expressed in $\mathrm{FO}(<)$.
2. L can be expressed in $\mathrm{FO}^3(<)$.
3. L can be expressed in LTL with past operators.
4. L can be expressed in pure future LTL.
5. L is star-free.
6. L is aperiodic.
7. L is recognizable and its minimal automaton is counter-free: no loop $q \xrightarrow{u} p \xrightarrow{u} \cdots \xrightarrow{u} q$ with $u \in A^+$ and $p \neq q$.

- A pool of difficult results.
- Elegant proof of Th. Wilke for $4 \iff 6$, refined by Diekert, Gastin, Kufleitner.
- $6 \implies$ decidable! Complexity of 7 : PSPACE-complete (J. Stern).
- Some implications are trivial. Eg. $4 \Rightarrow 3 \Rightarrow 2 \Rightarrow 1$, or $6 \Leftrightarrow 7$.

Example: $L = (ab)^+$

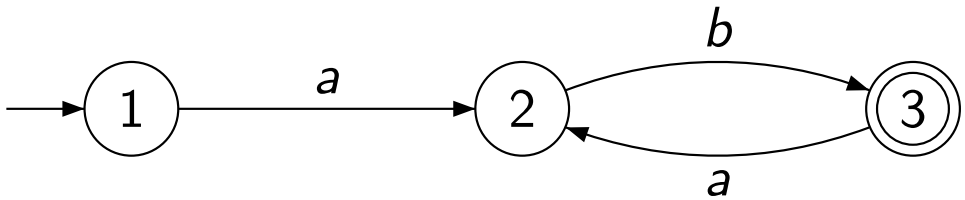
- Counter-free *(sink ommitted)*:



- $M(L)$ has 6 elements $\{1, a, b, a^2, ab, ba\}$ and $s^2 = s^3$ for all $s \in M$.
- Star-free $(ab)^+ = aA^* \cap A^*b \cap (A^* \setminus A^*(aa \cup bb)A^*)$.
- FO$(<)$ and FO$^3(<)$ $\forall x\,(a(x) \vee b(x)) \wedge$
  $\forall y \forall z\,(z = y + 1) \Rightarrow [a(y) \Leftrightarrow b(z)] \wedge$
  $\text{first} = \{a\} \wedge \text{last} = \{b\}$.
- LTL: $\mathsf{G}(a \vee b) \wedge a \wedge \mathsf{F}(b \wedge \neg \mathsf{X} \top) \wedge \mathsf{G}(a \leftrightarrow \neg \mathsf{X} b)$

Example: $L = (ab)^+$

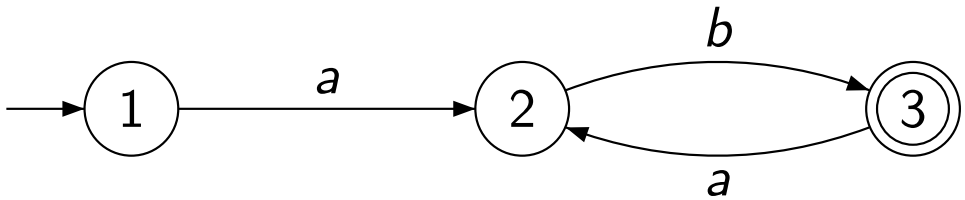
- Counter-free *(sink ommitted)*:



- $M(L)$ has 6 elements $\{1, a, b, a^2, ab, ba\}$ and $s^2 = s^3$ for all $s \in M$.
- Star-free $(ab)^+ = aA^* \cap A^*b \cap (A^* \setminus A^*(aa \cup bb)A^*)$.
- FO$(<)$ and FO$^3(<)$ $\forall x\,(a(x) \vee b(x)) \wedge$
  $\forall y \forall z\,(z = y + 1) \Rightarrow [a(y) \Leftrightarrow b(z)] \wedge$
  $\text{first} = \{a\} \wedge \text{last} = \{b\}$.
- LTL: $\mathsf{G}(a \vee b) \wedge a \wedge \mathsf{F}(b \wedge \neg \mathsf{X} \top) \wedge \mathsf{G}(a \leftrightarrow \neg \mathsf{X} b)$

Example: $L = (ab)^+$

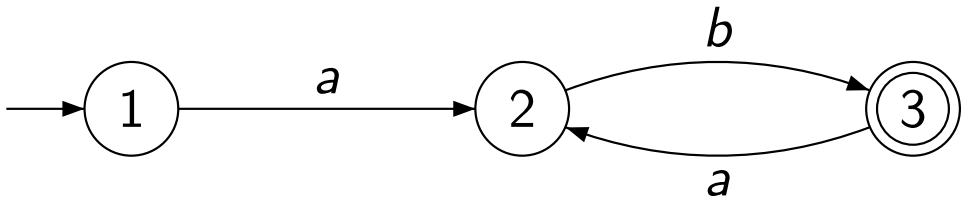
- Counter-free *(sink ommitted)*:



- $M(L)$ has 6 elements $\{1, a, b, a^2, ab, ba\}$ and $s^2 = s^3$ for all $s \in M$.
- Star-free $(ab)^+ = aA^* \cap A^*b \cap (A^* \setminus A^*(aa \cup bb)A^*)$.
- FO$(<)$ and FO$^3(<)$ $\forall x\,(a(x) \vee b(x)) \wedge$
  $\forall y \forall z\,(z = y + 1) \Rightarrow [a(y) \Leftrightarrow b(z)] \wedge$
  $\text{first} = \{a\} \wedge \text{last} = \{b\}$.
- LTL: $\mathsf{G}(a \vee b) \wedge a \wedge \mathsf{F}(b \wedge \neg \mathsf{X} \top) \wedge \mathsf{G}(a \leftrightarrow \neg \mathsf{X} b)$

Example: $L = (ab)^+$

- Counter-free *(sink ommitted)*:



- $M(L)$ has 6 elements $\{1, a, b, a^2, ab, ba\}$ and $s^2 = s^3$ for all $s \in M$.
- Star-free $(ab)^+ = aA^* \cap A^*b \cap (A^* \setminus A^*(aa \cup bb)A^*)$.
- FO$(<)$ and FO$^3(<)$ $\forall x\,(a(x) \vee b(x)) \wedge$
  $\forall y \forall z\,(z = y + 1) \Rightarrow [a(y) \Leftrightarrow b(z)] \wedge$
  $\text{first} = \{a\} \wedge \text{last} = \{b\}$.
- LTL: $\mathsf{G}(a \vee b) \wedge a \wedge \mathsf{F}(b \wedge \neg \mathsf{X} \top) \wedge \mathsf{G}(a \leftrightarrow \neg \mathsf{X} b)$

From Star-free to Aperiodic (easy)

Lemma: Star-free are aperiodic

Any star-free language is aperiodic.

In particular, $(aa)^+$ is not aperiodic.

From Star-free to Aperiodic (easy)

Lemma: Star-free are aperiodic

Any star-free language is aperiodic.

In particular, $(aa)^+$ is not aperiodic.

Proof Induction on $L \in \mathrm{SF}(A^*)$: find $i(L)$ such that

$$\forall u \in A^*, \quad u^{i(L)} \sim_L u^{i(L)+1}.$$

- $i(\varnothing) = 0$ and $i(a) = 2$.
- $i(A^* \setminus L) = i(L)$,
- $i(K \cup L) = \max(i(K), i(L))$,
- $i(KL) = i(K) + i(L)$: if $w = xu^{i(K)+i(L)}y \in KL$, then
  - either $xu^{i(K)}y' \in K$ for some prefix of $xu^{i(K)}y'$ of w, whence $xu^{i(K)+1}y' \in K$. Therefore $xu^{i(K)+i(L)+1}y \in KL$,
  - or symmetric case.

From Aperiodic to Star-free (difficult)

Original intuition of the proof from Th. Wilke.
Also works from aperiodic to LTL.

- See M as a transformation monoid, acting on set $Q = M$ of states.
- Induction on $(|M|, |A|)$ ordered lexicographically.
- Either all letters induce identity on Q: easy.
- Or some letter a does not act surjectively on Q. [Aperiodicity used here].
- In this case, decompose element on M as $u_1 a u_2 a \cdots a u_n$.
  - The initial and final segments u_1, u_n are on a smaller alphabet.
  - The intermediate segments use less states.

Tool for Induction (Aperiodic to Star-free)

- Suitable construction for the induction [Diekert, Gastin]
- For $m \in M$, define a new internal composition on the set $mM \cap Mm$:

$$xm \circ my = xmy.$$

- One can check that this is a well-defined product.
- We have $mx \circ my = mxy$, hence
  - $(mM \cap Mm, \circ, m)$ is a monoid.
  - If M is aperiodic, then so is $(mM \cap Mm, \circ, m)$.
  - If M is aperiodic and $m \neq 1$, then $|mM \cap Mm| < |M|$.
    Since in this case $1 \notin mM$: $ms = 1 \Rightarrow m^\omega s^\omega = 1 \Rightarrow m.(m^\omega s^\omega) = 1 \Rightarrow m = 1$.

From Aperiodic to Star-free (1)

Proof of V. Diekert, M. Kufleitner (adapted from Th. Wilke)

- Fix $\alpha : A^* \to M$ aperiodic and let $L = \alpha^{-1}(\alpha(L)) = \bigcup_{s \in \alpha(L)} \alpha^{-1}(s)$.
- Induction on $(|M|, |A|)$ ordered lexicographically.
- Enough to show that $\alpha^{-1}(s)$ is star-free.
- Assume first $s = 1$. Then: $\alpha^{-1}(1) = \{a \in A \mid \alpha(a) = 1\}^*$, hence star-free.

  Indeed, $uv = 1 \Rightarrow u = v = 1$.

From Aperiodic to Star-free (2)

Proof of V. Diekert, M. Kufleitner (adapted from Th. Wilke)

- Assume now $s \neq 1$.
- If $\alpha(u) = s$, then u contains a letter a such that $\alpha(a) \neq 1$. Let $\underline{a} = \alpha(a)$.
- Let $B = A \setminus \{a\}$, $\underline{B} = \alpha(B^*)$ and $\beta : B^* \to \underline{B}$ be the restriction of α to B^*.
- $\underline{B}$ is a submonoid of M (and will be considered as an alphabet, too).

From Aperiodic to Star-free (2)

Proof of V. Diekert, M. Kufleitner (adapted from Th. Wilke)

- Assume now $s \neq 1$.
- If $\alpha(u) = s$, then u contains a letter a such that $\alpha(a) \neq 1$. Let $\underline{a} = \alpha(a)$.
- Let $B = A \setminus \{a\}$, $\underline{B} = \alpha(B^*)$ and $\beta : B^* \to \underline{B}$ be the restriction of α to B^*.
- $\underline{B}$ is a submonoid of M (and will be considered as an alphabet, too).

- $u = u_1 a u_2 a u_3$ with $a \notin \mathrm{alph}(u_1 u_3)$ yields

$$\alpha^{-1}(s) = \bigcup_{a \in A, \alpha(a) \neq 1} \bigcup_{s = s_1 s_2 s_3} \beta^{-1}(s_1) \cdot (\alpha^{-1}(s_2) \cap aA^* \cap A^* a) \cdot \beta^{-1}(s_3)$$

- By induction hypothesis, we have $\beta^{-1}(s_i)$ star-free ($|\underline{B}| \leqslant |M|$ and $|B| < |A|$).
- Therefore we are left to show $\alpha^{-1}(s) \cap aA^* \cap A^* a$ is star-free.

From Aperiodic to Star-free (3)

Proof of V. Diekert, M. Kufleitner (adapted from Th. Wilke)

- Let us show that $\alpha^{-1}(s) \cap aA^* \cap A^*a$ is star-free.
- Let $\underline{B}^*$ be the free monoid over alphabet $\underline{B} = \alpha(B^*)$ (could have $|\underline{B}| > |A|$).
- "Decompose" α as $aA^* \xrightarrow{\sigma} \underline{B}^* \xrightarrow{\gamma} (\underline{a}M \cap M\underline{a}, \circ, \underline{a})$, as follows.
  - $\sigma(au_1au_2 \cdots au_n) = \beta(u_1) \cdot \beta(u_2) \cdots \beta(u_n)$. Erase a
  - γ morphism defined, for $\underline{b} \in \underline{B}$, by $\gamma(\underline{b}) = \underline{aba}$. Reintroduce it morphically!

From Aperiodic to Star-free (3)

Proof of V. Diekert, M. Kufleitner (adapted from Th. Wilke)

- Let us show that $\alpha^{-1}(s) \cap aA^* \cap A^*a$ is star-free.
- Let $\underline{B}^*$ be the free monoid over alphabet $\underline{B} = \alpha(B^*)$ (could have $|\underline{B}| > |A|$).
- "Decompose" α as $aA^* \xrightarrow{\sigma} \underline{B}^* \xrightarrow{\gamma} (\underline{a}M \cap M\underline{a}, \circ, \underline{a})$, as follows.
  - $\sigma(au_1au_2 \cdots au_n) = \beta(u_1) \cdot \beta(u_2) \cdots \beta(u_n)$. Erase a
  - γ morphism defined, for $\underline{b} \in \underline{B}$, by $\gamma(\underline{b}) = \underline{aba}$. Reintroduce it morphically!
- Almost a factorization of α.

$$\alpha^{-1}(s) \cap aA^* \cap A^*a \;=\; \sigma^{-1}(\gamma^{-1}(s)).a$$

From Aperiodic to Star-free (3)

Proof of V. Diekert, M. Kufleitner (adapted from Th. Wilke)

- Let us show that $\alpha^{-1}(s) \cap aA^* \cap A^*a$ is star-free.
- Let $\underline{B}^*$ be the free monoid over alphabet $\underline{B} = \alpha(B^*)$ (could have $|\underline{B}| > |A|$).
- "Decompose" α as $aA^* \xrightarrow{\sigma} \underline{B}^* \xrightarrow{\gamma} (\underline{a}M \cap M\underline{a}, \circ, \underline{a})$, as follows.
  - $\sigma(au_1au_2 \cdots au_n) = \beta(u_1) \cdot \beta(u_2) \cdots \beta(u_n)$. Erase a
  - γ morphism defined, for $\underline{b} \in \underline{B}$, by $\gamma(\underline{b}) = \underline{aba}$. Reintroduce it morphically!
- Almost a factorization of α.

$$\alpha^{-1}(s) \cap aA^* \cap A^*a \;=\; \sigma^{-1}(\gamma^{-1}(s)).a$$

- Since $|\underline{a}M \cap M\underline{a}| < |M|$, we get $\gamma^{-1}(s) \in \mathrm{SF}(\underline{B}^*)$.
- It remains to show that σ^{-1} preserves star-freeness.
  - For a one-letter word $\underline{b} \in \underline{B}$, it holds $\sigma^{-1}(\underline{b}) = a\beta^{-1}(\underline{b})$: use $|B| < |A|$.
  - SF operators commute with σ^{-1}, eg., $\sigma^{-1}(K \cup L) = \sigma^{-1}(K) \cup \sigma^{-1}(L)$.

Summary

- Many classes of regular languages have an algebraic characterization.
- General framework given in Eilenberg-Reiterman theorem.
- Variety of regular languages $\mathcal{V} : A \mapsto A^*\mathcal{V}$, closed under
  - boolean combinations,
  - inverse image by homomorphisms $\varphi : A^* \to B^*$,
  - quotients $L \mapsto a^{-1}L$, $L \mapsto La^{-1}$.
- Variety of finite monoids: class closed under $(M, N) \mapsto M \times N$ and division.
- Eilenberg's theorem: Bijective correspondence between varieties of regular languages and varieties of monoids.
- Reiterman's theorem: Equational definition of varieties.

Extensions

- To other classes than varieties
  - Drop complement: Positive varieties (Pin)
  - Drop closure under quotients (Pippenberg)
  - Drop both (Polák)
  - General framework: duality (Gehrke, Grigorieff, Pin): definition by (in)equations/identities.
- To other structures, like infinite words, Mazurkiewicz traces, trees.

Some related references

- V. Diekert and P. Gastin. First-order definable languages. In J. Flum, E. Grädel, and T. Wilke, editors, *Logic and Automata: History and Perspectives*, volume 2 of *Texts in Logic and Games*, pages 261–306. Amsterdam University Press, 2008.
- V. Diekert, P. Gastin, and M. Kufleitner. A survey on small fragments of first-order logic over finite words. *Int. J. Found. Comp. Sci. (IJFCS)*, 19(3):513–548, 2008.
- O. Klíma. Piecewise testable languages via combinatorics on words. In *Words'09*, 2009.
- J.-E. Pin. Syntactic semigroups. In G. Rozenberg and A. Salomaa, editors, *Handbook of formal languages, vol. 1*, pages 679–746. Springer, New York, NY, USA, 1997.
- Th. Schwentick, D. Thérien, and H. Vollmer. Partially-ordered two-way automata: A new characterization of DA. In *DLT'02*, pages 1–28. Springer, 2002.
- P. Tesson and D. Therien. Diamonds are forever: The variety DA. In *Semigroups, Algorithms, Automata and Languages, Coimbra (Portugal) 2001*, pages 475–500. World Scientific, 2002.
- P. Tesson and D. Thérien. Logic meets algebra: the case of regular languages. *Logical Methods in Computer Science*, 3(1), 2007.
- W. Thomas. Languages, automata, and logic. In G. Rozenberg and A. Salomaa, editors, *Handbook of formal languages, vol. 3: Beyond words*, pages 389–455. Springer, 1997.
- Th. Wilke. Classifying discrete temporal properties. In *STACS'99*, LNCS, pages 32–46. Springer, 1999.

Questions?