The theory of regular cost functions, from finite words to infinite trees.

Denis Kuperberg

University of Warsaw, Poland

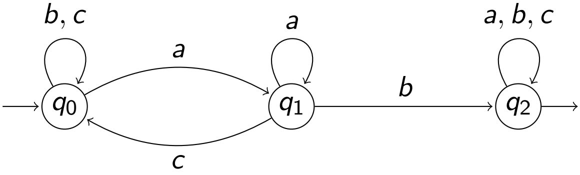
Séminaire de l'ENS Lyon
22-05-2014

Introduction

- Some natural problems are undecidable.
- For some problems, decidability is open.
- **Finite Automata:** Formalism with a lot of decidable properties.
- **Automata theory:**
  Toolbox to decide many problems arising naturally.
  Verification of systems can be done automatically.
  Theoretical and practical advantages.
- **Problem:**
  Decidability is still open for some automata-related problems.

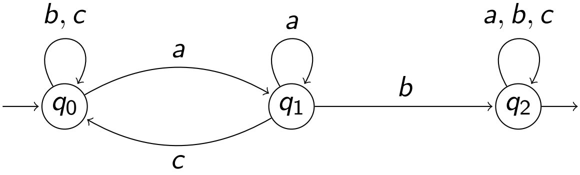
1. Automata theory
2. Regular Cost Functions
3. Formalisms on finite words
4. Cost functions on infinite words and trees

Descriptions of a language



Language recognized : $L_{ab} = \{\text{words containing } ab\}$.

Descriptions of a language



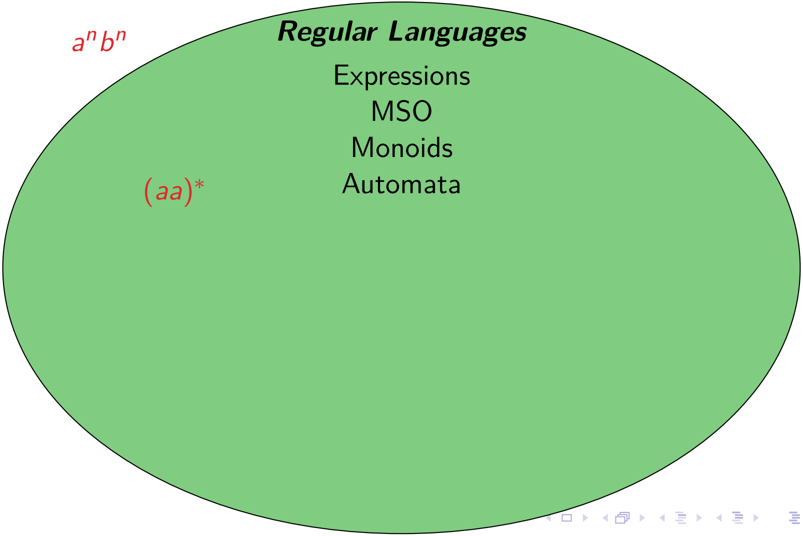
Language recognized : $L_{ab} = \{\text{words containing } ab\}$.

Other ways than automata to specify L_{ab} :

- Regular expression : $\mathbb{A}^* ab \mathbb{A}^*$,
- Logical sentence (MSO) : $\exists x\ \exists y\ a(x) \wedge b(y) \wedge (y = Sx)$.
- Finite monoid : $M = \{1, a, b, c, ba, 0\},\ P = \{0\}$
  $ab = 0,\ aa = ca = a,\ bb = bc = b,\ cc = ac = cb = c$

Regular Languages

All these formalisms are effectively equivalent.



$a^n b^n$

Regular Languages

Expressions

MSO

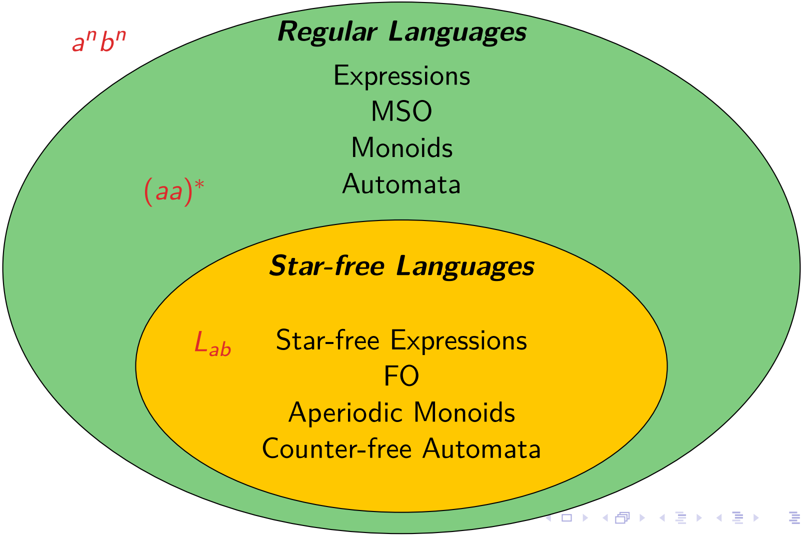
Monoids

Automata

$(aa)^*$

Regular Languages

All these formalisms are effectively equivalent.



$a^n b^n$

Regular Languages

Expressions
MSO
Monoids
Automata

$(aa)^*$

Star-free Languages

L_{ab}

Star-free Expressions
FO
Aperiodic Monoids
Counter-free Automata

Historical motivation

Given a class of languages $\mathcal{C}$, is there an algorithm which given an automaton for L, decides whether $L \in \mathcal{C}$?

Theorem (Schützenberger 1965)

It is decidable whether a regular language is star-free, thanks to the equivalence with aperiodic monoids.

Historical motivation

Given a class of languages $\mathcal{C}$, is there an algorithm which given an automaton for L, decides whether $L \in \mathcal{C}$?

Theorem (Schützenberger 1965)

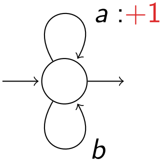
It is decidable whether a regular language is star-free, thanks to the equivalence with aperiodic monoids.

Finite Power Problem: Given L, is there n such that

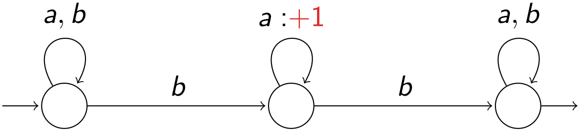
$$(L + \varepsilon)^n = L^* \ ?$$

There is no known algebraic characterization,
other technics are needed to show decidability.

Distance Automata



$\mathcal{A}_1$: number of a



$\mathcal{A}_2$: smallest block of a

Unbounded: There are words with arbitrarily large value.

Deciding **Boundedness** for distance automata ⇒ solving finite power problem.

Theorem (Hashiguchi 82, Kirsten 05)

Boundedness is decidable for distance automata.

Problems solved using counters

- ***Finite Power*** (finite words) [Simon '78, Hashiguchi '79]
  Is there n such that $(L+\varepsilon)^n = L^*$?
- ***Fixed Point Iteration*** (finite words)
  [Blumensath+Otto+Weyer '09]
  Can we bound the number of fixpoint iterations in a MSO formula ?
- ***Star-Height*** (finite words/trees)
  [Hashiguchi '88, Kirsten '05, Colcombet+Löding '08]
  Given n, is there an expression for L, with at most n nesting of Kleene stars?
- ***Parity Rank*** (infinite trees)
  [reduction in Colcombet+Löding '08, decidability open, deterministic input Niwinski+Walukiewicz '05]
  Given $i < j$, is there a parity automaton for L using ranks $\{i, i+1, \ldots, j\}$?

Theory of Regular Cost Functions

Aim: General framework for previous constructions.

- Generalize from languages $L : \mathbb{A}^* \to \{0, 1\}$
  to functions $f : \mathbb{A}^* \to \mathbb{N} \cup \{\infty\}$
- Accordingly generalize automata, logics, semigroups, in order to obtain a theory of regular cost functions, which behaves as well as possible.
- Obtain decidability results thanks to this new theory.

Cost automata over words

Nondeterministic finite-state automaton $\mathcal{A}$
$+$ finite set of counters
(initialized to 0, values range over $\mathbb{N}$)
$+$ counter operations on transitions
(increment `I`, reset `R`, check `C`, no change ε)

Semantics: $[\![\mathcal{A}]\!] : \Sigma^* \to \mathbb{N} \cup \{\infty\}$

Cost automata over words

Nondeterministic finite-state automaton $\mathcal{A}$

$+$ finite set of counters

(initialized to 0, values range over $\mathbb{N}$)

$+$ counter operations on transitions

(increment `I`, reset `R`, check `C`, no change ε)

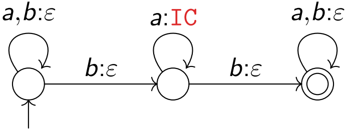
Semantics: $[\![\mathcal{A}]\!] : \Sigma^* \to \mathbb{N} \cup \{\infty\}$

$val_B(\rho) :=$ max checked counter value during run ρ

$[\![\mathcal{A}]\!]_B(u) := \min\{val_B(\rho) : \rho \text{ is an accepting run of } \mathcal{A} \text{ on } u\}$

Example

$[\![\mathcal{A}]\!]_B(u) =$ min length of block of a's surrounded by b's in u



Cost automata over words

Nondeterministic finite-state automaton $\mathcal{A}$

$+$ finite set of counters

(initialized to 0, values range over $\mathbb{N}$)

$+$ counter operations on transitions

(increment $\mathtt{I}$, reset $\mathtt{R}$, check $\mathtt{C}$, no change ε)

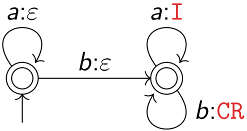
Semantics: $[\![\mathcal{A}]\!] : \Sigma^* \to \mathbb{N} \cup \{\infty\}$

$val_S(\rho) :=$ min checked counter value during run ρ

$[\![\mathcal{A}]\!]_S(u) := \max\{val_S(\rho) : \rho$ is an accepting run of $\mathcal{A}$ on $u\}$

Example

$[\![\mathcal{A}]\!]_S(u) =$ min length of block of a's surrounded by b's in u



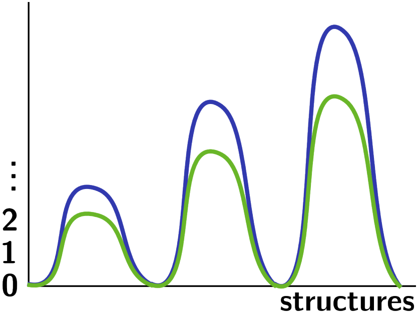
Boundedness relation

“$[\![\mathcal{A}]\!] = [\![\mathcal{B}]\!]$”: undecidable [Krob ’94]

Boundedness relation

"$[\![\mathcal{A}]\!] = [\![\mathcal{B}]\!]$": undecidable [Krob '94]

"$[\![\mathcal{A}]\!] \approx [\![\mathcal{B}]\!]$": decidable on words
[Colcombet '09, following Bojánczyk+Colcombet '06]
for all subsets U, $[\![\mathcal{A}]\!](U)$ bounded iff $[\![\mathcal{B}]\!](U)$ bounded

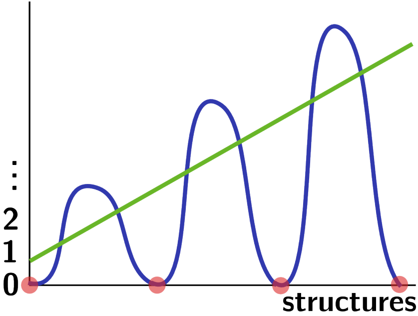


$$[\![\mathcal{A}]\!] \approx [\![\mathcal{B}]\!]$$

Boundedness relation

"$[\![\mathcal{A}]\!] = [\![\mathcal{B}]\!]$": undecidable [Krob '94]

"$[\![\mathcal{A}]\!] \approx [\![\mathcal{B}]\!]$": decidable on words
[Colcombet '09, following Bojánczyk+Colcombet '06]
for all subsets U, $[\![\mathcal{A}]\!](U)$ bounded iff $[\![\mathcal{B}]\!](U)$ bounded



$[\![\mathcal{A}]\!] \not\approx [\![\mathcal{B}]\!]$

Therefore we always identify two functions if they are bounded on the same sets.

Example

For any function f, we have $f \approx 2f \approx \exp(f)$.
But $(u \mapsto |u|_a) \not\approx (u \mapsto |u|_b)$, as witnessed by the set a^*.

Therefore we always identify two functions if they are bounded on the same sets.

Example

For any function f, we have $f \approx 2f \approx \exp(f)$.
But $(u \mapsto |u|_a) \not\approx (u \mapsto |u|_b)$, as witnessed by the set a^*.

Theorem (Colcombet '09, following Hashiguchi, Leung, Simon, Kirsten, Bojańczyk+Colcombet)

Cost automata $\Leftrightarrow$ Cost logics $\Leftrightarrow$ Stabilisation monoids.
For some suitable models of Cost Logics and Stabilisation Monoids, extending the classical ones.
Boundedness decidable (easy in S-automata and stabilisation monoids)

All these equivalences are only valid up to $\approx$.
It provides a toolbox to decide boundedness problems.

Languages as cost functions

A language L is represented by its characteristic function

$$\chi_L(u) = \begin{cases} 0 & \text{if } u \in L \\ \infty & \text{if } u \notin L \end{cases}$$

If $\mathcal{A}$ is a classical automaton for L, then $[\![\mathcal{A}]\!]_B = \chi_L$ and $[\![\mathcal{A}]\!]_S = \chi_{\overline{L}}$. Switching between B and S is the generalization of language complementation.

Cost function theory strictly extends language theory.

All theorems on cost functions are in particular true for languages.

Research program: Studying cost function theory, and generalise known theorems from languages to cost functions.

1. Automata theory
2. Regular Cost Functions
3. **Formalisms on finite words**
4. Cost functions on infinite words and trees

Classical Logics on Finite Words

- Linear Temporal Logic (LTL) over $\mathbb{A}^*$:

$$\varphi := a \mid \Omega \mid \neg\varphi \mid \varphi \vee \psi \mid \mathbf{X}\varphi \mid \varphi\mathbf{U}\psi$$

$$\varphi\mathbf{U}\psi: \quad \begin{matrix} \varphi & \varphi & \varphi & \varphi & \varphi & \varphi & \varphi & \varphi & \psi & & \\ a_0 & a_1 & a_2 & a_3 & a_4 & a_5 & a_6 & a_7 & a_8 & a_9 & a_{10} \end{matrix}$$

Future operators **G** (Always) and **F** (Eventually).
Example: To describe L_{ab}, we can write $\mathbf{F}(a \wedge \mathbf{X}b)$.

Classical Logics on Finite Words

- Linear Temporal Logic (LTL) over $\mathbb{A}^*$:

$$\varphi := a \mid \Omega \mid \neg\varphi \mid \varphi \vee \psi \mid \mathbf{X}\varphi \mid \varphi\mathbf{U}\psi$$

$$\varphi\mathbf{U}\psi: \quad \begin{array}{ccccccccccc} \varphi & \varphi & \varphi & \varphi & \varphi & \varphi & \varphi & \varphi & \psi & & \\ a_0 & a_1 & a_2 & a_3 & a_4 & a_5 & a_6 & a_7 & a_8 & a_9 & a_{10} \end{array}$$

  Future operators **G** (Always) and **F** (Eventually).
  Example: To describe L_{ab}, we can write $\mathbf{F}(a \wedge \mathbf{X}b)$.

- First-Order Logic (FO): we quantify over positions in the word.

$$\varphi := a(x) \mid x \leq y \mid \neg\varphi \mid \varphi \vee \psi \mid \exists x \varphi$$

Classical Logics on Finite Words

- Linear Temporal Logic (LTL) over $\mathbb{A}^*$:

$$\varphi := a \mid \Omega \mid \neg\varphi \mid \varphi \vee \psi \mid \mathbf{X}\varphi \mid \varphi\mathbf{U}\psi$$

$$\varphi\mathbf{U}\psi: \quad \begin{array}{ccccccccccc} \varphi & \varphi & \varphi & \varphi & \varphi & \varphi & \varphi & \varphi & \psi & & \\ a_0 & a_1 & a_2 & a_3 & a_4 & a_5 & a_6 & a_7 & a_8 & a_9 & a_{10} \end{array}$$

  Future operators **G** (Always) and **F** (Eventually).
  Example: To describe L_{ab}, we can write $\mathbf{F}(a \wedge \mathbf{X}b)$.

- First-Order Logic (FO): we quantify over positions in the word.

$$\varphi := a(x) \mid x \leq y \mid \neg\varphi \mid \varphi \vee \psi \mid \exists x\varphi$$

- MSO: FO with quantification on sets, noted X, Y.

Generalisation: cost LTL

- CLTL over $\mathbb{A}^*$:

$$\varphi := a \mid \Omega \mid \varphi \wedge \psi \mid \varphi \vee \psi \mid \mathbf{X}\varphi \mid \varphi \mathbf{U} \psi \mid \varphi \mathbf{U}^{\leq N} \psi$$

Negations pushed to the leaves, to guarantee monotonicity.

Generalisation: cost LTL

- CLTL over $\mathbb{A}^*$:

$$\varphi := a \mid \Omega \mid \varphi \wedge \psi \mid \varphi \vee \psi \mid \mathbf{X}\varphi \mid \varphi\mathbf{U}\psi \mid \varphi\mathbf{U}^{\leq N}\psi$$

  Negations pushed to the leaves, to guarantee monotonicity.

- $\varphi\mathbf{U}^{\leq N}\psi$ means that ψ is true in the future, and φ is false at most N times in the mean time.

$$\varphi\mathbf{U}^{\leq N}\psi: \qquad \begin{array}{ccccccccccc} \varphi & \varphi & \times & \varphi & \varphi & \times & \varphi & \varphi & \psi & & \\ a_0 & a_1 & a_2 & a_3 & a_4 & a_5 & a_6 & a_7 & a_8 & a_9 & a_{10} \end{array}$$

Generalisation: cost LTL

- CLTL over $\mathbb{A}^*$:

$$\varphi := a \mid \Omega \mid \varphi \wedge \psi \mid \varphi \vee \psi \mid \mathbf{X}\varphi \mid \varphi \mathbf{U} \psi \mid \varphi \mathbf{U}^{\leq N} \psi$$

  Negations pushed to the leaves, to guarantee monotonicity.

- $\varphi \mathbf{U}^{\leq N} \psi$ means that ψ is true in the future, and φ is false at most N times in the mean time.

$$\varphi \mathbf{U}^{\leq N} \psi: \qquad \begin{array}{ccccccccccc} \varphi & \varphi & \times & \varphi & \varphi & \times & \varphi & \varphi & \psi & & \\ a_0 & a_1 & a_2 & a_3 & a_4 & a_5 & a_6 & a_7 & a_8 & a_9 & a_{10} \end{array}$$

- "Error variable" N is unique, shared by all occurences of $\mathbf{U}^{\leq N}$.

Generalisation: cost LTL

- CLTL over $\mathbb{A}^*$:

  $$\varphi := a \mid \Omega \mid \varphi \wedge \psi \mid \varphi \vee \psi \mid \mathbf{X}\varphi \mid \varphi\mathbf{U}\psi \mid \varphi\mathbf{U}^{\leq N}\psi$$

  Negations pushed to the leaves, to guarantee monotonicity.

- $\varphi\mathbf{U}^{\leq N}\psi$ means that ψ is true in the future, and φ is false at most N times in the mean time.

  $$\varphi\mathbf{U}^{\leq N}\psi: \qquad \begin{array}{ccccccccccc} \varphi & \varphi & \times & \varphi & \varphi & \times & \varphi & \varphi & \psi & & \\ a_0 & a_1 & a_2 & a_3 & a_4 & a_5 & a_6 & a_7 & a_8 & a_9 & a_{10} \end{array}$$

- "Error variable" N is unique, shared by all occurences of $\mathbf{U}^{\leq N}$.
- $\mathbf{G}^{\leq N}\varphi$: φ is false at most N times in the future ($\varphi\mathbf{U}^{\leq N}\Omega$).

Generalisation : Cost FO and Cost MSO

- CFO over $\mathbb{A}^*$:

  $$\varphi := a(x) \mid x = y \mid x < y \mid \varphi \wedge \varphi \mid \varphi \vee \varphi \mid \exists x \varphi \mid \forall x \varphi \mid \forall^{\leq N} x \varphi$$

  Negations pushed to the leaves, to guarantee monotonicity.
- As before, N unique free variable.
- $\forall^{\leq N} x \varphi(x)$ means φ is false on at most N positions.
- CMSO extends CFO by allowing quantification over sets.

Semantics of Cost Logics

From formula to cost function:

Formula $\varphi \longrightarrow$ cost function $[\![\varphi]\!] : \mathbb{A}^* \to \mathbb{N} \cup \{\infty\}$, defined by

$$[\![\varphi]\!](u) = \inf\{n \in \mathbb{N} : \varphi \text{ is true over } u \text{ with } n \text{ as error value}\}$$

Example with the alphabet $\{a, b\}$

- $\mathrm{number}_a = [\![\mathbf{G}^{\leq N} b]\!] = [\![\forall^{\leq N} x\ b(x)]\!]$.

Semantics of Cost Logics

From formula to cost function:

Formula $\varphi \longrightarrow$ cost function $[\![\varphi]\!] : \mathbb{A}^* \to \mathbb{N} \cup \{\infty\}$, defined by

$$[\![\varphi]\!](u) = \inf\{n \in \mathbb{N} : \varphi \text{ is true over } u \text{ with } n \text{ as error value}\}$$

Example with the alphabet $\{a, b\}$

- $\mathrm{number}_a = [\![\mathbf{G}^{\leq N} b]\!] = [\![\forall^{\leq N} x \ b(x)]\!]$.
- $\mathrm{maxblock}_a = [\![\mathbf{G}(\bot \mathbf{U}^{\leq N}(b \vee \Omega))]\!]$
  $= [\![\forall X \ \mathrm{block}_a(X) \Rightarrow (\forall^{\leq N} x . x \notin X)]\!]$
  where $\mathrm{block}_a(X) = \forall x \in X, y \in X, z . x \leq z \leq y \Rightarrow (z \in X \wedge a(z))$.

Semantics of Cost Logics

From formula to cost function:

Formula $\varphi \longrightarrow$ cost function $[\![\varphi]\!] : \mathbb{A}^* \to \mathbb{N} \cup \{\infty\}$, defined by

$$[\![\varphi]\!](u) = \inf\{n \in \mathbb{N} : \varphi \text{ is true over } u \text{ with } n \text{ as error value}\}$$

Example with the alphabet $\{a, b\}$

- $\mathrm{number}_a = [\![\mathbf{G}^{\leq N} b]\!] = [\![\forall^{\leq N} x \ b(x)]\!]$.
- $\mathrm{maxblock}_a = [\![\mathbf{G}(\bot \mathbf{U}^{\leq N}(b \vee \Omega))]\!]$
  $= [\![\forall X \ \mathrm{block}_a(X) \Rightarrow (\forall^{\leq N} x.x \notin X)]\!]$
  where $\mathrm{block}_a(X) = \forall x \in X, y \in X, z.x \leq z \leq y \Rightarrow (z \in X \wedge a(z))$.
- If φ is a classical formula for L, then $[\![\varphi]\!] = \chi_L$.

Stabilisation monoids

- **Aim:** Generalise monoids to a quantitative setting.

Stabilisation monoids

- **Aim:** Generalise monoids to a quantitative setting.
- Stabilisation $\sharp$ means “repeat many times” the element.

Stabilisation monoids

- **Aim:** Generalise monoids to a quantitative setting.
- Stabilisation $\sharp$ means “repeat many times” the element.
- if we “count” a, then $a^\sharp \neq a$, otherwise $a^\sharp = a$.

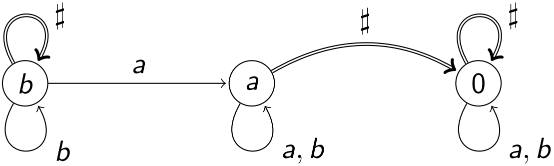
Stabilisation monoids

- **Aim:** Generalise monoids to a quantitative setting.
- Stabilisation $\sharp$ means "repeat many times" the element.
- if we "count" a, then $a^\sharp \neq a$, otherwise $a^\sharp = a$.

Example: Stabilisation Monoid for number_a

$M = \{b, a, 0\}$, $P = \{a, b\}$,

b**:** "no a", a**:** "a little number of a", 0**:** "a lot of a".



Cayley graph

Aperiodic Monoids

Definition: A [stabilisation] monoid M is aperiodic if for all $x \in M$ there is $n \in \mathbb{N}$ such that $x^n = x^{n+1}$.

Aperiodic Monoids

Definition: A [stabilisation] monoid M is aperiodic if for all $x \in M$ there is $n \in \mathbb{N}$ such that $x^n = x^{n+1}$.

Theorem (McNaughton-Papert, Schützenberger, Kamp)

Aperiodic Monoids ⇔ FO ⇔ LTL ⇔ Star-free Expressions.

We want to generalise this theorem to cost functions.
The problems are:

- No complementation ⇒ No Star-free expressions.
- Deterministic automata are strictly weaker.
- Heavy formalisms (semantics of stabilisation monoids).
- New quantitative behaviours.
- Original proofs already hard.

Aperiodic cost functions

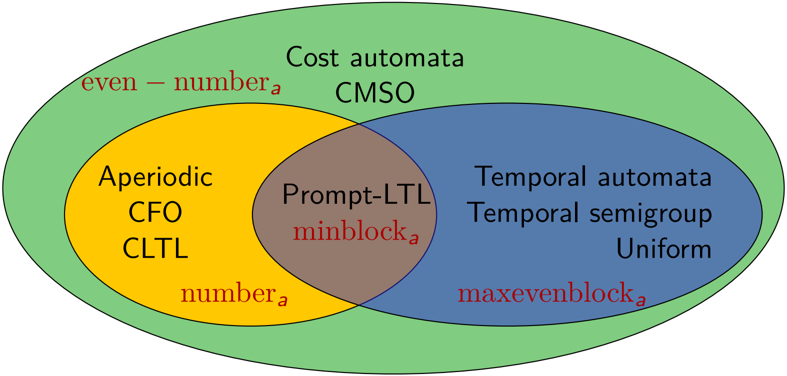
Theorem (K. STACS 2011)

Aperiodic stabilisation monoid $\Leftrightarrow$ *CLTL* $\Leftrightarrow$ *CFO.*

Proof Ideas:

- Generalisation of Myhill-Nerode $\Rightarrow$ Syntactic object.
- Induction on $(|M|, |\mathbb{A}|)$.
- Extend functions to sequences of words.
- Use bounded approximations.
- Extend CLTL with Past operators, show Separability.

Cost Functions on finite words



Cost automata
CMSO

$\mathrm{even} - \mathrm{number}_a$

Aperiodic
CFO
CLTL

number_a

Prompt-LTL

$\mathrm{minblock}_a$

Temporal automata
Temporal semigroup
Uniform

$\mathrm{maxevenblock}_a$

Decidability of membership and effectiveness of translations [K+Colcombet+Lombardy ICALP '10, K. STACS '11].
Generalization of Myhill-Nerode Equivalence [K. STACS '11].
Boundedness of CLTL is PSPACE-complete [K. LMCS].

Generalisation of input structures

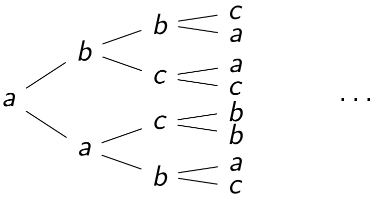
Input structures:

Finite words: *accba*

Infinite words: *abaabaccbaba*...

Finite trees

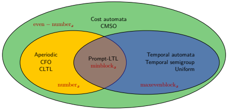
Infinite trees:



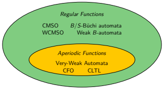
Generalisation of input structures

Input structures:

Finite words: *accba*

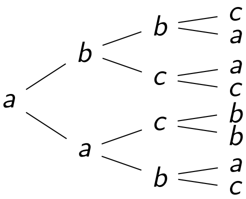


Infinite words: *abaabaccbaba* . . .

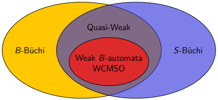


Finite trees

Infinite trees:



· · ·



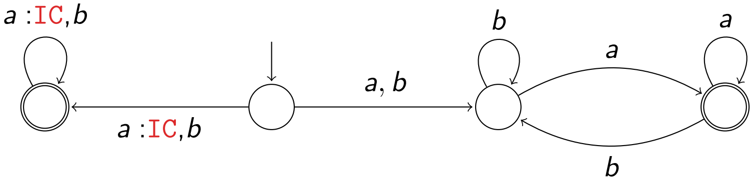
Different kinds of results:

- Generalisation of language notions and theorems,
- Study of classes specific to cost functions,
- Reduction of classical decision problems to boundedness problems.

Formalism on infinite words

Nondeterministic B/S-automata with Büchi condition: infinitely many accepting states must be seen in an accepting run.

Example: B-Büchi automaton:

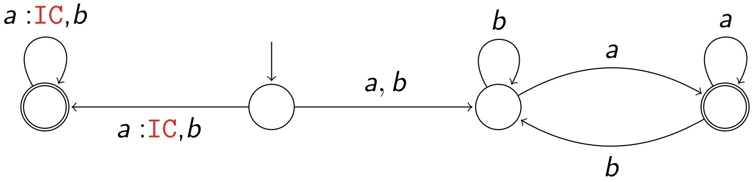


Computes $f(u)=\begin{cases} 0 & \text{if } |u|_a=\infty \\ |u|_a & \text{otherwise}\end{cases}$

Formalism on infinite words

Nondeterministic B/S-automata with Büchi condition: infinitely many accepting states must be seen in an accepting run.

Example: B-Büchi automaton:

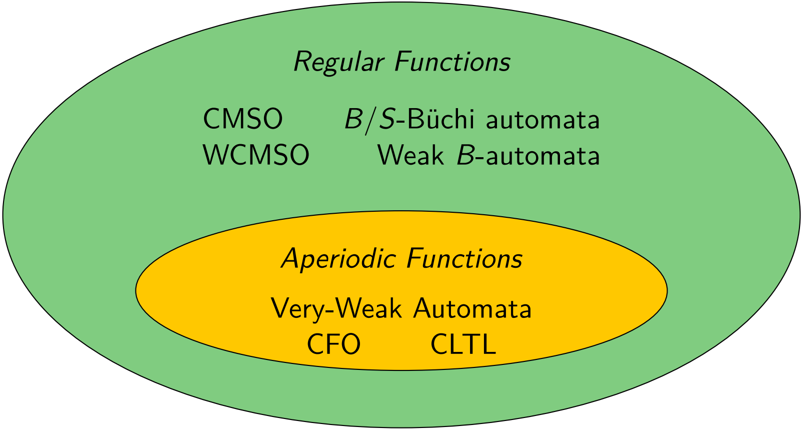


Computes $f(u) = \begin{cases} 0 & \text{if } |u|_a = \infty \\ |u|_a & \text{otherwise} \end{cases}$

- Weak alternating B-automata: semantic is a game between two players, Min and Max.
- Cost MSO/FO/LTL as before.
- Weak Cost MSO: quantification is restricted to finite sets.

Can we lift results from classical theory to cost function theory?

Picture on infinite words



Regular Functions

CMSO B/S-Büchi automata

WCMSO Weak B-automata

Aperiodic Functions

Very-Weak Automata

CFO CLTL

Decidability of membership and effectiveness of translations

Boundedness decidable [K+Vanden Boom ICALP '12].

Cost functions on finite trees

The theory of cost functions on finite trees is developed in [Colcombet+Löding LICS '10].

- cost automata can be defined on finite trees.
- Nondeterministic/alternating B/S variants are all equivalent.
- Equivalent to Cost MSO on finite trees.
- For all above formalisms, boundedness is decidable.
- Application to the star-height problem on finite trees.

Cost functions on infinite trees

Example of cost function on infinite trees:

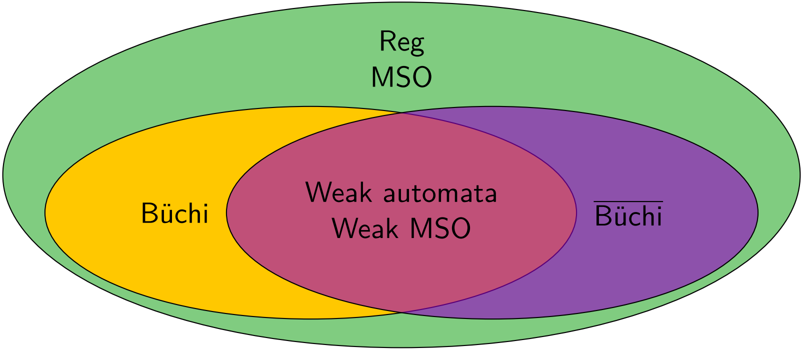
$$f(t) = \begin{cases} \infty & \text{if } |\pi|_a = \infty \text{ on some branch } \pi \text{ of } t \\ \min_{\pi \text{ branch of } t} |\pi|_b & \text{otherwise} \end{cases}$$

- Parity acceptance condition: ranks $[i, j]$, automaton accepts if on every branch, the maximal infinitely occuring rank is even.
- Boundedness of B-coBüchi ($[0, 1]$-parity) is decidable [CKV+Löding CSL '13].
- Decidability of boundedness of B-Büchi ($[1, 2]$-parity) is open...
- Decidability of boundedness for B-Parity $\Rightarrow$ Solution to the classical Mostowski index problem.
- Can we make some progress?

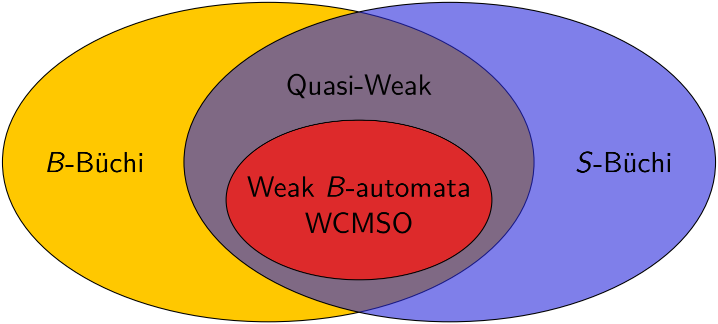
Languages on infinite trees

Theorem (Rabin 1970, Kupferman + Vardi 1999)

L recognizable by an alternating weak automaton $\Leftrightarrow$
L recognizable by WMSO $\Leftrightarrow$ there are Büchi automata $\mathcal{U}$ and $\mathcal{U}'$ such that $L = L(\mathcal{U}) = \overline{L(\mathcal{U}')}$.



Picture on infinite trees



Boundedness decidable for Quasi-Weak automata.
[KV FSTTCS '11].
If $\mathcal{A}$ is a Büchi automaton, it is decidable whether $L(\mathcal{A})$ is weak
[CKVL CSL '13].
Logic and μ-calculus for the Quasi-Weak class
[CKV+Blumensath+Parys CSL-LICS '14].

Conclusion

Achievements:

- Robust quantitative extension of regular language theory.
- Embeds proof using different kind of automata with counters.
- Rich quantitative behaviours occur.
- New proofs on regular languages and reductions obtained.
- In particular: progress on deciding whether a language is weak.

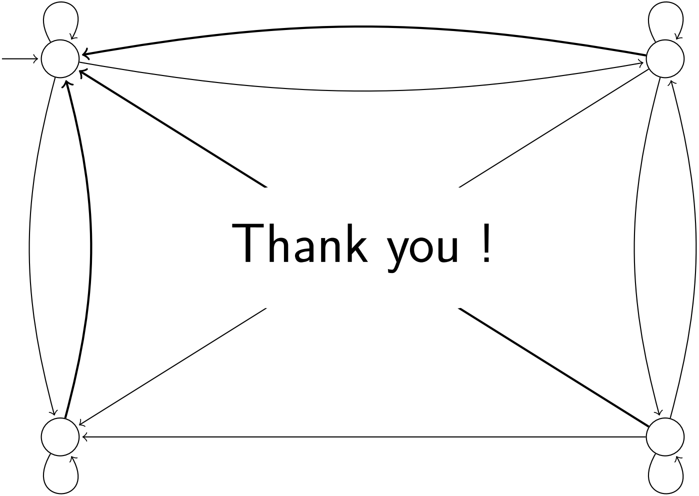
Conclusion

Achievements:

- Robust quantitative extension of regular language theory.
- Embeds proof using different kind of automata with counters.
- Rich quantitative behaviours occur.
- New proofs on regular languages and reductions obtained.
- In particular: progress on deciding whether a language is weak.

Current challenges and related works:

- Main open problem: decide boundedness on infinite trees. Application to language theory.
- Link with other formalisms, as MSO+U of Bojańczyk.
- Decide properties of cost automata, like optimal number of counters.
- Fine study of approximations (Daviaud)
- Alternative formalisms: IST, profinite words



Thank you !