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

$M = \{1, a, b, c, ba\}, P = \{0\}$
Descriptions of a language

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

Other ways than automata to specify \( L_{ab} \):

- Regular expression: \( A^* ab A^* \),
- Logical sentence (MSO): \( \exists x \ \exists y \ a(x) \land b(y) \land (y = Sx) \).
- Finite monoid: \( M = \{ 1, a, b, c, ba, 0 \} \), \( P = \{ 0 \} \)
  \( ab = 0, \ aa = ca = a, \ bb = bc = b, \ cc = ac = cb = c \)
All these formalisms are effectively equivalent.
Regular Languages

All these formalisms are effectively equivalent.
Historical motivation

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

**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 $C$, is there an algorithm which given an automaton for $L$, decides whether $L \in C$?

**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

\[ A_1: \text{number of } a \]

\[ a, b \]

\[ a : +1 \]

\[ b \]

\[ A_2: \text{smallest block of } a \]

Unbounded: There are words with arbitrarily large value.

Deciding **Boundedness** for distance automata \( \Rightarrow \) 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 $\{i, i + 1, \ldots, j\}$?
Automata theory

Regular Cost Functions

Formalisms on finite words

Cost functions on infinite words and trees
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 $A$

- finite set of counters
  (initialized to 0, values range over $\mathbb{N}$)

- counter operations on transitions
  (increment $I$, reset $R$, check $C$, no change $\varepsilon$)

**Semantics:** $[A] : \Sigma^* \rightarrow \mathbb{N} \cup \{\infty\}$
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 $\varepsilon$)

Semantics: $\llbracket \mathcal{A} \rrbracket : \Sigma^* \rightarrow \mathbb{N} \cup \{\infty\}$

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

$\llbracket \mathcal{A} \rrbracket_B(u) := \min\{val_B(\rho) : \rho$ is an accepting run of $\mathcal{A}$ on $u\}$

Example

$\llbracket \mathcal{A} \rrbracket_B(u) = \min$ length of block of $a$’s surrounded by $b$’s in $u$

$a,b: \varepsilon \quad a: IC \quad a,b: \varepsilon$

\begin{tikzpicture}
  \node (start) at (0,0) [circle, draw] {};
  \node (b1) at (1,0) [circle, draw] {};
  \node (b2) at (2,0) [circle, draw] {};
  \node (f1) at (3,0) [circle, draw] {};

  \draw[->] (start) edge [loop above] node {$a,b: \varepsilon$} (start);
  \draw[->] (start) edge [loop below] node {$b: \varepsilon$} (start);
  \draw[->] (b1) edge [loop above] node {$a: IC$} (b1);
  \draw[->] (b1) edge [loop below] node {$b: \varepsilon$} (b1);
  \draw[->] (b2) edge [loop below] node {$b: \varepsilon$} (b2);
  \draw[->] (f1) edge [loop above] node {$a,b: \varepsilon$} (f1);
\end{tikzpicture}
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 $\varepsilon$)

Semantics: $\llbracket \mathcal{A} \rrbracket : \Sigma^* \rightarrow \mathbb{N} \cup \{\infty\}$

$\text{val}_S(\rho) := \min \text{ checked counter value during run } \rho$

$\llbracket \mathcal{A} \rrbracket_S(u) := \max \{ \text{val}_S(\rho) : \rho \text{ is an accepting run of } \mathcal{A} \text{ on } u \}$

Example

$\llbracket \mathcal{A} \rrbracket_S(u) = \min \text{ length of block of } a \text{'s surrounded by } b \text{'s in } u$

\[
\begin{align*}
  &a: \varepsilon & a: I \\
  &b: \varepsilon & b: CR
\end{align*}
\]
Boundedness relation

“\([A] = [B]\)”: undecidable [Krob ’94]
Boundedness relation

“$[A] = [B]$”: undecidable [Krob ’94]

“$[A] \approx [B]$”: decidable on words

[Colcombet ’09, following Bojánczyk+Colcombet ’06] for all subsets $U$, $[A](U)$ bounded iff $[B](U)$ bounded
Boundedness relation

“$[A] = [B]$”: undecidable [Krob '94]

“$[A] \approx [B]$”: decidable on words

[Colcombet '09, following Bojánczyk+Colcombet '06] for all subsets $U$, $[A](U)$ bounded iff $[B](U)$ bounded

\[ \begin{bmatrix} 0 \\ 1 \\ 2 \end{bmatrix} \]
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 $A$ is a classical automaton for $L$, then $[A]_B = \chi_L$ and $[A]_S = \chi_L$. 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 $\mathbb{A}^*$:

  $\varphi := a \mid \Omega \mid \neg \varphi \mid \varphi \lor \psi \mid X \varphi \mid \varphi U \psi$

  $\varphi U \psi$:  
  $a_0 a_1 a_2 a_3 a_4 a_5 a_6 a_7 a_8 a_9 a_{10}$

  Future operators $G$ (Always) and $F$ (Eventually).

  **Example**: To describe $L_{ab}$, we can write $F(a \land X b)$. 
Classical Logics on Finite Words

- **Linear Temporal Logic (LTL)** over $A^*$:

  $$\varphi := a \mid \Omega \mid \neg \varphi \mid \varphi \lor \psi \mid X \varphi \mid \varphi U \psi$$

  $$\varphi U \psi:$$

  $$a_0 \ a_1 \ a_2 \ a_3 \ a_4 \ a_5 \ a_6 \ a_7 \ a_8 \ a_9 \ a_{10}$$

  Future operators $G$ (Always) and $F$ (Eventually).

  **Example**: To describe $L_{ab}$, we can write $F(a \land Xb)$.

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

  $$\varphi := a(x) \mid x \leq y \mid \neg \varphi \mid \varphi \lor \psi \mid \exists x \varphi$$
Classical Logics on Finite Words

- **Linear Temporal Logic (LTL) over $A^*$:**
  \[
  \varphi := a \mid \Omega \mid \neg\varphi \mid \varphi \lor \psi \mid X\varphi \mid \varphi U \psi
  \]
  \[
  \varphi U \psi: \quad a_0 a_1 a_2 a_3 a_4 a_5 a_6 a_7 a_8 a_9 a_{10}
  \]
  Future operators $G$ (Always) and $F$ (Eventually).
  **Example:** To describe $L_{ab}$, we can write $F(a \land Xb)$.

- **First-Order Logic (FO):** we quantify over positions in the word.
  \[
  \varphi := a(x) \mid x \leq y \mid \neg\varphi \mid \varphi \lor \psi \mid \exists x \varphi
  \]

- **MSO:** FO with quantification on sets, noted $X$, $Y$. 
Generalisation: cost LTL

- **CLTL** over $A^*$:

\[
\varphi := a \mid \Omega \mid \varphi \land \psi \mid \varphi \lor \psi \mid X\varphi \mid \varphi U\psi \mid \varphi U^{\leq N} \psi
\]

Negations pushed to the leaves, to guarantee monotonicity.
Generalisation: cost LTL

- **CLTL** over $A^*$:

  $$\varphi := a | \Omega | \varphi \land \psi | \varphi \lor \psi | X\varphi | \varphi U \psi | \varphi U \leq N \psi$$

  Negations pushed to the leaves, to guarantee monotonicity.

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

  $$\varphi U \leq N \psi:: \varphi \varphi \times \varphi \varphi \times \varphi \varphi \psi$$

  $$a_0 a_1 a_2 a_3 a_4 a_5 a_6 a_7 a_8 a_9 a_{10}$$
Generalisation: cost LTL

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

$$\varphi := a \mid \Omega \mid \varphi \land \psi \mid \varphi \lor \psi \mid X\varphi \mid \varphi U\psi \mid \varphi U^{\leq N}\psi$$

Negations pushed to the leaves, to guarantee monotonicity.

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

$$\varphi U^{\leq N}\psi: \quad \varphi \varphi \times \varphi \varphi \times \varphi \varphi \psi$$

$$a_0 a_1 a_2 a_3 a_4 a_5 a_6 a_7 a_8 a_9 a_{10}$$

- “Error variable” $N$ is unique, shared by all occurrences of $U^{\leq N}$. 

Generalisation: cost LTL

- **CLTL** over \( A^* \):

\[
\varphi := a | \Omega | \varphi \land \psi | \varphi \lor \psi | X\varphi | \varphi U \psi | \varphi U \leq N \psi
\]

Negations pushed to the leaves, to guarantee monotonicity.

- \( \varphi U \leq N \psi \) means that \( \psi \) is true in the future, and \( \varphi \) is false at most \( N \) times in the mean time.

\[
\varphi U \leq N \psi: \quad \varphi \varphi \times \varphi \varphi \times \varphi \varphi \psi
\]

\[
a_0 a_1 a_2 a_3 a_4 a_5 a_6 a_7 a_8 a_9 a_{10}
\]

- “Error variable” \( N \) is unique, shared by all occurrences of \( U \leq N \).
- \( G \leq N \varphi: \) \( \varphi \) is false at most \( N \) times in the future (\( \varphi U \leq N \Omega \)).
Generalisation: Cost FO and Cost MSO

- **CFO** over $A^*$:

$$\varphi ::= a(x) \mid x = y \mid x < y \mid \varphi \land \varphi \mid \varphi \lor \varphi \mid \exists x \varphi \mid \forall x \varphi \mid \forall \leq N x \varphi$$

Negations pushed to the leaves, to guarantee monotonicity.

- As before, $N$ unique free variable.
- $\forall \leq N x \varphi(x)$ means $\varphi$ 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 $\varphi \rightarrow$ cost function $[\varphi] : A^* \rightarrow \mathbb{N} \cup \{\infty\}$, 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 $\{a, b\}$

- number$_a = [G^\leq N b] = [\forall^\leq N x \ b(x)]$. 
From formula to cost function:

Formula $\varphi \rightarrow$ cost function $[\varphi] : \mathbb{A}^* \rightarrow \mathbb{N} \cup \{\infty\}$, 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 $\{a, b\}$

- **number$_a = [G^{\leq_N} b] = [\forall^{\leq_N} x \ b(x)].**
- **maxblock$_a = [G(\perp \bigcup^{\leq_N} (b \lor \Omega))] = [\forall X \ \text{block}_a(X) \Rightarrow (\forall^{\leq_N} x. x \notin X)]$$
  
  where $\text{block}_a(X) = \forall x \in X, y \in X, z.x \leq z \leq y \Rightarrow (z \in X \land a(z))$. 
Semantics of Cost Logics

From formula to cost function:
Formula $\varphi \rightarrow$ cost function $[\varphi] : \mathbb{A}^* \rightarrow \mathbb{N} \cup \{\infty\}$, 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 $\{a, b\}$

- $\text{number}_a = [G_{\leq N} b] = [\forall x b(x)]$.
- $\text{maxblock}_a = [G(\bot U_{\leq N} (b \lor \Omega))]$
  $$= [\forall X \text{ block}_a(X) \Rightarrow (\forall x \leq N x \notin X)]$$
  where $\text{block}_a(X) = \forall x \in X, y \in X, z. x \leq z \leq y \Rightarrow (z \in X \land a(z))$.
- If $\varphi$ 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 $\# \equiv$ means “repeat many times” the element.
Stabilisation monoids

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

Example: Stabilisation Monoid for $M = \{b, a, 0\}$, $P = \{a, b\}$,

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

Cayley graph

$0$ $\# b$ $a$

$b^\# b^\# a$. 

$20/33$
**Aim:** Generalise monoids to a quantitative setting.

Stabilisation $\#$ means “repeat many times” the element.

if we “count” $a$, then $a^\# \neq a$, otherwise $a^\# = a$.

**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

![Cayley graph diagram]
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 $\Leftrightarrow$ FO $\Leftrightarrow$ LTL $\Leftrightarrow$ Star-free Expressions.

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

- No complementation $\Rightarrow$ 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 $\iff$ CLTL $\iff$ CFO.

**Proof Ideas:**
- Generalisation of Myhill-Nerode $\Rightarrow$ Syntactic object.
- Induction on $(|M|, |A|)$.
- Extend functions to sequences of words.
- Use bounded approximations.
- Extend CLTL with Past operators, show Separability.
Cost Functions on finite words

Decidability of membership and effectiveness of translations
[K+Colcombet+Lombardy ICALP ‘10, K. STACS ‘11].


Boundedness of CLTL is PSPACE-complete [K. LMCS].
1 Automata theory

2 Regular Cost Functions

3 Formalisms on finite words

4 Cost functions on infinite words and trees
Generalisation of input structures

Input structures:

Finite words: \textit{accba}

Infinite words: \textit{abaabacccba} \ldots

Finite trees

Infinite trees: \begin{tikzpicture}[grow=up,sibling distance=25pt,level distance=25pt]
  \node {a}
  child {node {b} child {node {a}}}
  child {node {c} child {node {a}}}
  child {node {b} child {node {a}}}
  child {node {c} child {node {b}}}
  child {node {b} child {node {c}}}
\end{tikzpicture} \ldots
Generalisation of input structures

Input structures:

Finite words: \textit{accba}

Infinite words: \textit{abaabaccbaba} \ldots

Finite trees

\begin{tikzpicture}
  \node (a) at (0,0) {a};
  \node (b) at (1,1) {b};
  \node (c) at (1,-1) {c};
  \node (d) at (2,0) {a};
  \node (e) at (3,1) {b};
  \node (f) at (3,-1) {c};
  \draw (a) -- (b);
  \draw (a) -- (c);
  \draw (b) -- (d);
  \draw (b) -- (e);
  \draw (c) -- (f);
\end{tikzpicture}

Infinite trees: \begin{tikzpicture}
  \node (a) at (0,0) {a};
  \node (b) at (1,1) {b};
  \node (c) at (1,-1) {c};
  \node (d) at (2,0) {a};
  \node (e) at (3,1) {b};
  \node (f) at (3,-1) {c};
  \draw (a) -- (b);
  \draw (a) -- (c);
  \draw (b) -- (d);
  \draw (b) -- (e);
  \draw (c) -- (f);
\end{tikzpicture} \ldots

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

Example: $B$-Büchi automaton:

\[
\begin{align*}
  a : \text{IC}, b \\
  a : \text{IC}, b \\
  b \\
  a \\
  b \\
\end{align*}
\]

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:

\[
\begin{array}{c}
\text{a : IC, b} \\
\text{a : IC, b} \\
\text{a, b} \\
\text{b} \\
\text{b}
\end{array}
\]

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?
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:

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

- **Parity** acceptance condition: ranks \([i, j]\), automaton accepts if on every branch, the maximal infinitely occurring 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 $\iff$

$L$ recognizable by WMSO $\iff$ there are Büchi automata $U$ and $U'$ such that $L = L(U) = \overline{L(U')}$. 

![Diagram showing the relationships between Regular (Reg), MSO, Büchi, Weak automata, and Weak MSO formalisms.](diagram.png)
Boundedness decidable for Quasi-Weak automata.

[KV FSTTCS ’11].

If $A$ is a Büchi automaton, it is decidable whether $L(A)$ is weak

[CKVL CSL ’13].

Logic and $\mu$-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!