Decidability problems in automata theory

Denis Kuperberg

Onera/DTIM, Toulouse

Séminaire du DTIM
06-10-2014
Some natural problems are **undecidable**.

For some problems, decidability is open.

**Finite Automata**: Abstract machine with a lot of **decidable** properties and **equivalent** formalisms.

**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. Hard decision problems
3. Regular Cost Functions
4. Formalisms on finite words
Descriptions of a language

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

Other ways than automata to specify \( L_{ab} \subseteq A^* \):

- Regular expression: \( A^* ab A^* \),
- Logical sentence (MSO): \( \exists x \; \exists y \; a(x) \land b(y) \land (y = Sx) \).
- Finite monoid: ?
Decision problems

What can we easily decide/compute with automata?

- Emptyness (Hard for Logic !)
- Complementation
- Union/Intersection
- Concatenation
- ...

What about expressibility in smaller formalisms? Like FO or star-free expressions? I.e. given \( L \), is there a FO-formula for \( L \) (resp. star-free expression)?
Decision problems

What can we easily decide/compute with automata?

- Emptyness (Hard for Logic !)
- Complementation
- Union/Intersection
- Concatenation
- ...

What about expressibility in smaller formalisms?
Like FO or star-free expressions?

I.e. given $L$, is there a FO-formula for $L$? (resp. star-free expression)
What is a monoid?

A monoid is a set $M$ equipped with an associative and containing a neutral element $1 \in M$.

Examples: Groups, $(\mathbb{A}^*, \text{concat})$, $(\{0, 1\}, \land)$. 
What is a monoid?

A monoid is a set $M$ equipped with an associative and containing a neutral element $1 \in M$.

Examples: Groups, $(A^*, \text{concat})$, $(\{0, 1\}, \land)$.

A monoid $M$ recognizes a language $L \subseteq A^*$ if there exists $P \subseteq M$ and a morphism $h : A^* \to M$ such that $L = h^{-1}(P)$.

Examples:
- Any language $L \subseteq A^*$ is recognized by $A^*$. 
What is a monoid?

A monoid is a set $M$ equipped with an associative and containing a neutral element $1 \in M$.

Examples: Groups, $(\mathbb{A}^*, \text{concat})$, $(\{0, 1\}, \land)$.

A monoid $M$ recognizes a language $L \subseteq \mathbb{A}^*$ if there exists $P \subseteq M$ and a morphism $h : \mathbb{A}^* \rightarrow M$ such that $L = h^{-1}(P)$.

Examples:

- Any language $L \subseteq \mathbb{A}^*$ is recognized by $\mathbb{A}^*$.
- $(\mathbb{A} \mathbb{A})^*$ is recognized by the group $\mathbb{Z}/2\mathbb{Z}$, with $P = \{0\}$ and $h(a) = 1$ for all $a \in \mathbb{A}$. 
What is a monoid?

A monoid is a set $M$ equipped with an associative and containing a neutral element $1 \in M$.

**Examples:** Groups, $(\mathbb{A}^*, \text{concat})$, $(\{0, 1\}, \wedge)$.

A monoid $M$ recognizes a language $L \subseteq \mathbb{A}^*$ if there exists $P \subseteq M$ and a morphism $h : \mathbb{A}^* \to M$ such that $L = h^{-1}(P)$.

**Examples:**

- Any language $L \subseteq \mathbb{A}^*$ is recognized by $\mathbb{A}^*$.
- $(\mathbb{A}\mathbb{A})^*$ is recognized by the group $\mathbb{Z}/2\mathbb{Z}$, with $P = \{0\}$ and $h(a) = 1$ for all $a \in \mathbb{A}$.
- $\mathbb{A}^* b \mathbb{A}^*$ is recognized by $(\{0, 1\}, \wedge)$, with $P = \{0\}$, $h(b) = 0$ and $h(a) = 1$ for $a \neq b$. 
All these formalisms are effectively equivalent.
All these formalisms are effectively equivalent.

Regular Languages

Expressions
MSO
Finite Monoids
Automata

Star-free Languages
Star-free Expressions
FO
LTL
Aperiodic Monoids
Counter-free Automata

a^n b^n

(aa)^*
1. Automata theory

2. Hard decision problems

3. Regular Cost Functions

4. Formalisms on finite words
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 (i.e. without groups).*
Algebraic approach to membership problems

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 (i.e. without groups).*

**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

\(A_1\): number of \(a\)

\(A_2\): smallest block of \(a\)

**Unbounded**: There are words with arbitrarily large value (on any run).
Distance Automata

$A_1$: number of $a$

$A_2$: smallest block of $a$

Unbounded: There are words with arbitrarily large value (on any run).

Theorem (Hashiguchi 82, Kirsten 05)

Boundedness is decidable for distance automata.

Deciding Boundedness for distance automata $\Rightarrow$ solving finite power problem.
1. Start with an automaton for $L$. 

![Diagram of an automaton with transitions and loops]
1. Start with an automaton for $L$.
2. Add increment $\varepsilon$-transitions from final states to initial.
Reductions from Finite Power to Boundedness

1. Start with an automaton for $L$.
2. Add increment $\varepsilon$-transitions from final states to initial.
3. Decide boundedness
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 \( \{i, i+1, \ldots, j\} \)?
1 Automata theory
2 Hard decision problems
3 Regular Cost Functions
4 Formalisms on finite words
**Theory of Regular Cost Functions**

**Aim:** General framework for previous constructions.

- Generalize from languages $L : A^* \rightarrow \{0, 1\}$
  
  to functions $f : A^* \rightarrow \mathbb{N} \cup \{\infty\}$

- 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 $\varepsilon$)

Semantics: $[[\mathcal{A}]] : \mathbb{A}^* \rightarrow \mathbb{N} \cup \{\infty\}$
Boundedness relation

“$[A] = [B]$”: undecidable [Krob '94]
Boundedness relation

"\[[A] = [B]\]"\n: undecidable [Krob '94]

"\[[A] \approx [B]\]"\n: 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
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 $\iff$ Cost logics $\iff$ Stabilisation monoids.

For some suitable models of Cost Logics and Stabilisation Monoids, extending the classical ones.

Boundedness decidable.

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

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

Cost function theory strictly extends language theory.

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

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

Hard decision problems

Regular Cost Functions

Formalisms on finite words
Linear Temporal Logic (LTL) over $\mathbb{A}^*$:

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

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

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

**Example:** To describe $L_{ab}$, we can write $F(a \land Xb)$. 
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.
  \[
  \varphi := a(x) \mid x \leq y \mid \neg \varphi \mid \varphi \lor \psi \mid \exists x \varphi
  \]
Classical Logics on Finite Words

- **Linear Temporal Logic (LTL)** over $A^*$:
  \[ \varphi := a | \Omega | \neg \varphi | \varphi \lor \psi | X\varphi | \varphi U \psi \]
  
  $\varphi U \psi$: $a_0 a_1 a_2 a_3 a_4 a_5 a_6 a_7 a_8 a_9 a_{10}$

  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.
  \[ \varphi := a(x) | x \leq y | \neg \varphi | \varphi \lor \psi | \exists x \varphi \]

- **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 $A^*$:

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

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.

$$\varphi U\leq N\psi: \quad \varphi \varphi \times \varphi \varphi \times \varphi \varphi \psi \quad a_0 \ a_1 \ a_2 \ a_3 \ a_4 \ a_5 \ a_6 \ a_7 \ a_8 \ a_9 \ a_{10}$$
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 $A^*$:

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

  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 $U\leq N$.

- $G\leq N\varphi$: $\varphi$ 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 on alphabet $\{\epsilon, \text{Request, Grant}\}$:
$G(\text{Request} \implies \bot \cup \leq^N \text{Grant})$.

If $\varphi$ is a classical formula for $L$, then $[[\varphi]] = \chi_L$. 
Aperiodic Monoids
Aperiodic Monoids

Theorem (McNaughton-Papert, Schützenberger, Kamp)

\[ \text{Aperiodic Monoids} \iff \text{FO} \iff \text{LTL} \iff \text{Star-free Expressions}. \]

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 ⇔ CLTL ⇔ CFO.

Proof Ideas:

- Generalisation of Myhill-Nerode ⇒ Syntactic object.
- Induction on $(|M|, |A|)$.
- Extend functions to sequences of words.
- Use bounded approximations.
- Extend CLTL with Past operators, show Separability.
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].
As for regular languages, theory of regular cost functions can be extended to

- Infinite words,
- Finite trees,
- Infinite trees.

Led to rich developments, results for both functions and languages, but also complications and open problems...

**Main open problem**: Decidability of boundedness on infinite trees?
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.
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.

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!