

Linear temporal logic for regular cost functions

Denis Kuperberg

Liafa/CNRS/Université Paris 7, Denis Diderot, France

STACS 2011

Introduction

- ▶ Regular cost functions : counting extension of regular languages

Introduction

- ▶ Regular cost functions : counting extension of regular languages
- ▶ LTL^{\leq} : new simple way to define regular cost functions

Introduction

- ▶ Regular cost functions : counting extension of regular languages
- ▶ LTL^{\leq} : new simple way to define regular cost functions
- ▶ Translation from LTL^{\leq} to automata

Introduction

- ▶ Regular cost functions : counting extension of regular languages
- ▶ LTL^{\leq} : new simple way to define regular cost functions
- ▶ Translation from LTL^{\leq} to automata
- ▶ Algebraic characterization of LTL^{\leq} -definable cost functions

Outline

Introduction

Counting events in words

B -automata

Cost functions

Quantitative Linear temporal logic

Definition

Semantics

Algebraic characterization

B-automata

Aim : To represent functions $\mathbb{A}^* \rightarrow \mathbb{N} \cup \{\infty\}$ with automata.

B-automaton :

- ▶ nondeterministic finite-state
- ▶ finite set of counters, ranging over \mathbb{N} , initial value 0
- ▶ each transition performs actions on each counter

Atomic actions : increment (i), reset (r), do nothing (ε).

Semantics of B -automata

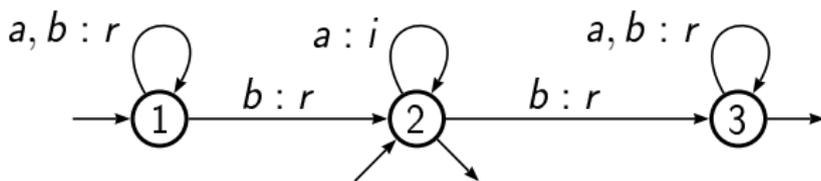
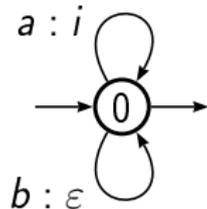
Function $\mathbb{A}^* \rightarrow \mathbb{N} \cup \{\infty\}$ associated with a B -automaton \mathcal{A} :

$[\mathcal{A}]_B(u) = \inf\{n / \text{there is a run where all counter values stay below } n\}$

with $\inf \emptyset = \infty$.

Example

$[\mathcal{A}]_B = |\cdot|_a$ and $[\mathcal{A}']_B : a^{n_1} b a^{n_2} \dots b a^{n_k} \mapsto \min(n_1, n_2, \dots, n_k)$



More on B -automata

Remark : A standard automaton \mathcal{A} computing L can be viewed as a B -automaton without any counter.

Then $\llbracket \mathcal{A} \rrbracket_B = \chi_L$ with $\chi_L(u) = \begin{cases} 0 & \text{if } u \in L \\ \infty & \text{if } u \notin L \end{cases}$

Theorem ([Krob '94])

The equivalence of two distance automata (particular case of B -automata) is undecidable.

How to get a decidable quantitative extension of regular languages?

Solution : Loosing some precision on the counting, but keeping information about bounds.

Cost functions

If $f, g : \mathbb{A}^* \rightarrow \mathbb{N} \cup \{\infty\}$, then

$$f \approx g \text{ if } \forall X \subseteq \mathbb{A}^*, f|_X \text{ bounded} \Leftrightarrow g|_X \text{ bounded.}$$

Cost function : equivalence class for \approx relation.

Example

For $\mathbb{A} = \{a, b, c\}$,

- ▶ $\max(|\cdot|_a, |\cdot|_b) \approx |\cdot|_a + |\cdot|_b$,
- ▶ $|\cdot|_a \not\approx \text{maxblock}_a$: on $X = (ab)^*$, only maxblock_a is bounded.

Known results on B -automata

Extension of the notion of language via $\chi_L : L = L' \Leftrightarrow \chi_L \approx \chi_{L'}$.

Theorem (Colcombet '09)

It is decidable whether two B -automata compute the same cost function (modulo \approx).

B -automata-computable cost functions are called **regular**.

Example

χ_L is a regular cost function iff L is a regular language.

Introduction

Counting events in words

B-automata

Cost functions

Quantitative Linear temporal logic

Definition

Semantics

Algebraic characterization

- ▶ LTL on \mathbb{A} describes regular languages :

$\varphi := a \mid \Omega \mid \varphi \wedge \psi \mid \varphi \vee \psi \mid X\varphi \mid \varphi U\psi$

where Ω marks the end of the word.

$\varphi U\psi :$

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

- ▶ LTL on \mathbb{A} describes regular languages :

$$\varphi := a \mid \Omega \mid \varphi \wedge \psi \mid \varphi \vee \psi \mid X\varphi \mid \varphi U\psi$$

where Ω marks the end of the word.

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

- ▶ LTL^{\leq} on \mathbb{A} describes regular cost functions :

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

- ▶ LTL on \mathbb{A} describes regular languages :

$$\varphi := a \mid \Omega \mid \varphi \wedge \psi \mid \varphi \vee \psi \mid X\varphi \mid \varphi U\psi$$

where Ω marks the end of the word.

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

- ▶ LTL^{\leq} on \mathbb{A} describes regular cost functions :

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

- ▶ $\varphi U^{\leq N}\psi$ means that ψ is true somewhere in the future, and φ is false at most N times until then.

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

- ▶ LTL on \mathbb{A} describes regular languages :

$$\varphi := a \mid \Omega \mid \varphi \wedge \psi \mid \varphi \vee \psi \mid X\varphi \mid \varphi U\psi$$

where Ω marks the end of the word.

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

- ▶ LTL^{\leq} on \mathbb{A} describes regular cost functions :

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

- ▶ $\varphi U^{\leq N}\psi$ means that ψ is true somewhere in the future, and φ is false at most N times until then.

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

- ▶ The "error value" variable N is unique, and is shared by all occurrences of $U^{\leq N}$ operator.

Semantics of LTL^{\leq}

From formula to cost function :

$\llbracket \varphi \rrbracket$ is the cost function associated to φ , defined by

$$\llbracket \varphi \rrbracket(u) = \inf\{N \in \mathbb{N}, \varphi \text{ is true on } u \text{ with } N \text{ as error value}\}$$

Example

For all $u \in \{a, b\}^*$, we have

- ▶ $\llbracket bU^{\leq N}\Omega \rrbracket(u) = |u|_a$.
- ▶ $\llbracket G(\perp U^{\leq N}(b \vee \Omega)) \rrbracket(a^{n_0} b a^{n_1} b \dots b a^{n_k}) = \max(n_0, n_1, \dots, n_k)$
- ▶ $\llbracket F(b \wedge X(\perp U^{\leq N}(b \vee \Omega))) \rrbracket(a^{n_0} b a^{n_1} b \dots b a^{n_k}) = \min(n_1, \dots, n_k)$

Semantics of LTL^{\leq}

From formula to cost function :

$\llbracket \varphi \rrbracket$ is the cost function associated to φ , defined by

$$\llbracket \varphi \rrbracket(u) = \inf\{N \in \mathbb{N}, \varphi \text{ is true on } u \text{ with } N \text{ as error value}\}$$

Example

For all $u \in \{a, b\}^*$, we have

- ▶ $\llbracket bU^{\leq N}\Omega \rrbracket(u) = |u|_a$.
- ▶ $\llbracket G(\perp U^{\leq N}(b \vee \Omega)) \rrbracket(a^{n_0} b a^{n_1} b \dots b a^{n_k}) = \max(n_0, n_1, \dots, n_k)$
- ▶ $\llbracket F(b \wedge X(\perp U^{\leq N}(b \vee \Omega))) \rrbracket(a^{n_0} b a^{n_1} b \dots b a^{n_k}) = \min(n_1, \dots, n_k)$

Theorem

We can effectively translate an LTL^{\leq} -formula φ into a B -automaton with $2^{|\varphi|}$ states.

Introduction

Counting events in words

B -automata

Cost functions

Quantitative Linear temporal logic

Definition

Semantics

Algebraic characterization

Algebraic characterization of regular cost functions

Complete framework with new contributions in red :

Regular languages	Regular cost functions
Classic automaton	B -automaton [Colcombet '09]
LTL	LTL^{\leq}
Finite monoid	Finite stabilization monoid [Col09]
Minimal monoid	Minimal stabilization monoid [Colcombet, K., Lombardy '10]
Syntactic congruence	Cost functions syntactic congruence
LTL \Leftrightarrow aperiodic monoid [Kamp,McNaughton&Papert, Schützenberger]	$LTL^{\leq} \Leftrightarrow$ aperiodic stabilization monoid

Algebraic characterization of regular cost functions

Complete framework with new contributions in red :

Regular languages	Regular cost functions
Classic automaton	B -automaton [Colcombet '09]
LTL	LTL^{\leq}
Finite monoid	Finite stabilization monoid [Col09]
Minimal monoid	Minimal stabilization monoid [Colcombet, K., Lombardy '10]
Syntactic congruence	Cost functions syntactic congruence
LTL \Leftrightarrow aperiodic monoid [Kamp,McNaughton&Papert, Schützenberger]	$LTL^{\leq} \Leftrightarrow$ aperiodic stabilization monoid

Corollary

The class of LTL^{\leq} -definable cost functions is decidable.

Conclusion

Summary

- ▶ Definition of LTL^{\leq} to easily describe cost functions
- ▶ Translation from LTL^{\leq} to B -automata
- ▶ Syntactic congruence for cost functions
- ▶ Algebraic characterization and decidability of the class of LTL^{\leq} -definable cost functions.

Future work

- ▶ Extension to infinite words
- ▶ Other characterizations of this class by first-order logic, star-free expressions, . . .