Expressive power of Cost Logics over Infinite Words (and Trees)

Denis Kuperberg, joint work with Michael Vanden Boom

FREC Days
05-06-2012
Introduction

- Regular cost functions: counting extension of regular languages.
Introduction

- Regular cost functions: counting extension of regular languages.
- Motivation: solving bound-related problems on regular languages (e.g. star-height).
Introduction

- Regular cost functions: counting extension of regular languages.
- Motivation: solving bound-related problems on regular languages (e.g. star-height).
- Definable over finite or infinite structures, like words or trees.
Introduction

- Regular cost functions: counting extension of regular languages.
- Motivation: solving bound-related problems on regular languages (e.g. star-height).
- Definable over finite or infinite structures, like words or trees.
- Definable via automata, logics, algebraic structures,...
Cost automata over infinite 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
\[
[\mathcal{A}] : \mathcal{A}^\omega \rightarrow \mathbb{N} \cup \{\infty\}
\]
Cost automata over infinite 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

\[ \text{val}_B(\rho) := \text{max checked counter value during run } \rho \]
\[ [\mathcal{A}]_B(u) := \min \{ \text{val}_B(\rho) : \rho \text{ is an accepting run of } \mathcal{A} \text{ on } u \} \]

Example

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

\[ a, b : \varepsilon \]

\[ a : \text{IC} \]

\[ a, b : \varepsilon \]
Cost automata over infinite 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

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

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

Example

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

![Diagram](attachment:image.png)
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

$[A] \not\approx [B]$
Applications

Many problems for a regular language $L$ can be reduced to deciding $\approx$ for some class of automata with counting features:

- **Finite power property** (finite words)
  [Simon '78, Hashiguchi '79]
  is there some $n$ such that $(L + \epsilon)^n = L^*$?

- **Star-height problem** (finite words/trees)
  [Hashiguchi '88, Kirsten '05, Colcombet+Löding '08]
  given $n$, is there a regular expression for $L$ with at most $n$ nestings of Kleene star?

- **Parity-index problem** (infinite trees)
  [reduction in Colcombet+Löding '08, decidability open]
  given $i < j$, is there a parity automaton for $L$ which uses only priorities $\{i, i+1, \ldots, j\}$?
Applications

Many problems for a regular language $L$ can be reduced to deciding $\approx$ for some class of automata with counting features:

- **Finite power property** (finite words)
  [Simon ’78, Hashiguchi ’79]
  is there some $n$ such that $(L + \epsilon)^n = L^*$?

- **Star-height problem** (finite words/trees)
  [Hashiguchi ’88, Kirsten ’05, Colcombet+Löding ’08]
  given $n$, is there a regular expression for $L$ with at most $n$ nestings of Kleene star?

- **Parity-index problem** (infinite trees)
  [reduction in Colcombet+Löding ’08, decidability open]
  given $i < j$, is there a parity automaton for $L$ which uses only priorities $\{i, i + 1, \ldots, j\}$?
Languages as cost functions

A standard automaton $A$ computing a language $L$ can be viewed as a $B$- or $S$-automaton without any counters. Then $[A]_B = \chi_L$ and $[A]_S = \chi_L$, with

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

- A standard automaton $A$ computing a language $L$ can be viewed as a $B$- or $S$-automaton without any counters. Then $\lceil A \rceil_B = \chi_L$ and $\lceil A \rceil_S = \chi_L$, with

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

- Switching between $B$ and $S$ semantics corresponds to a complementation.

- Aim: Extend classic theorems from languages to cost functions.
Languages as cost functions

- A standard automaton $A$ computing a language $L$ can be viewed as a $B$- or $S$-automaton without any counters. Then $[A]_B = \chi_L$ and $[A]_S = \chi_{\overline{L}}$, with

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

- Switching between $B$ and $S$ semantics corresponds to a complementation.

- If $L$ and $L'$ are languages, $\chi_L \approx \chi_{L'}$ iff $L = L'$, so cost function theory, even up to $\approx$, strictly extends language theory.
Languages as cost functions

- A standard automaton $A$ computing a language $L$ can be viewed as a $B$- or $S$-automaton without any counters. Then $\llbracket A \rrbracket_B = \chi_L$ and $\llbracket A \rrbracket_S = \chi_L$, with

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

- Switching between $B$ and $S$ semantics corresponds to a complementation.

- If $L$ and $L'$ are languages, $\chi_L \approx \chi_{L'}$ iff $L = L'$, so cost function theory, even up to $\approx$, strictly extends language theory.

- **Aim:** Extend classic theorems from languages to cost functions.
Cost functions on infinite words

- In the following, input structures = $\mathbb{A}$-labelled infinite words.
Cost functions on infinite words

- In the following, input structures = $A$-labelled infinite words.
- Dual $B$- and $S$- semantics as before, defining functions:
  $A^\omega \rightarrow \mathbb{N} \cup \{\infty\}$.

Example:
- $a, c$: $\varepsilon$
- $b, c$: $\varepsilon$

This automaton computes
- $\|u\|_a$ if $\|u\|_b < \infty$
- $\infty$ if $\|u\|_b = \infty$
Cost functions on infinite words

- In the following, input structures = \( \mathbb{A} \)-labelled infinite words.
- Dual \( B \)- and \( S \)- semantics as before, defining functions:
  \( \mathbb{A}^\omega \to \mathbb{N} \cup \{\infty\} \).
- Acceptance condition : Büchi.
Cost functions on infinite words

- In the following, input structures = \( \mathbb{A} \)-labelled infinite words.
- Dual \( B \)- and \( S \)- semantics as before, defining functions:
  \( \mathbb{A}^\omega \rightarrow \mathbb{N} \cup \{\infty\} \).
- Acceptance condition : Büchi.
- Example :

\[
\begin{align*}
  a &: \text{IC} \\
  b, c &: \varepsilon \\
  \text{Acceptance condition: Büchi.}
\end{align*}
\]
Cost functions on infinite words

- In the following, input structures = A-labelled infinite words.
- Dual B- and S- semantics as before, defining functions: \( A^\omega \rightarrow \mathbb{N} \cup \{\infty\} \).
- Acceptance condition: Büchi.
- Example:

This automaton computes

\[ \begin{cases} |u|_a \text{ if } |u|_b < \infty \\ \infty \text{ if } |u|_b = \infty \end{cases} \]
Logics on infinite words

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

$$\varphi := a | \varphi \land \varphi | \varphi \lor \varphi | \varphi R \varphi | \varphi U \varphi$$

where the negations have been pushed to the leaves, and the $U$ corresponds to “Next Until”.

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

We can define $X$ (Next), $G$ (Always) and $F$ (Eventually) in terms of these operators.
Logics on infinite words

- LTL on $A$ describes regular languages:
  $$\varphi := a \mid \varphi \land \varphi \mid \varphi \lor \varphi \mid \varphi R \varphi \mid \varphi U \varphi$$
  where the negations have been pushed to the leaves, and the $U$ corresponds to “Next Until”.

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

  We can define $X$ (Next), $G$ (Always) and $F$ (Eventually) in terms of these operators.

- First-Order Logic (FO):

  $$\varphi := a(x) \mid x = y \mid x < y \mid \varphi \land \varphi \mid \varphi \lor \varphi \mid \exists x \varphi \mid \forall x \varphi$$
Logics on infinite words

- LTL on \( \mathbb{A} \) describes regular languages:
  \[ \varphi := a \mid \varphi \land \varphi \mid \varphi \lor \varphi \mid \varphi R \varphi \mid \varphi U \varphi \]
  where the negations have been pushed to the leaves, and the \( U \) corresponds to “Next Until”.

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

We can define \( X \) (Next), \( G \) (Always) and \( F \) (Eventually) in terms of these operators.

- First-Order Logic (FO):
  \[ \varphi := a(x) \mid x = y \mid x < y \mid \varphi \land \varphi \mid \varphi \lor \varphi \mid \exists x. \varphi \mid \forall x. \varphi \]

- (Weak) MSO: FO with quantification over (finite) sets, set variables noted \( X, Y \).
Cost LTL

- CLTL on Λ describes regular cost functions:

\[ \psi := a | \psi \land \psi | \psi \lor \psi | \psi R \psi | \psi U \psi | \psi U^{\leq N} \psi \]
Cost LTL

- CLTL on $\mathbb{A}$ describes regular cost functions:
  $\varphi := a \mid \varphi \land \varphi \mid \varphi \lor \varphi \mid \varphi R \varphi \mid \varphi U \varphi \mid \varphi U \leq N \varphi$

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

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

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

- CLTL on $\mathbb{A}$ describes regular cost functions:
  \[
  \varphi := a \mid \varphi \land \varphi \mid \varphi \lor \varphi \mid \varphi R \varphi \mid \varphi U \varphi \mid \varphi U \leq N \varphi
  \]

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

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

- The “error value” variable $N$ is unique, and is shared by all occurrences of $U \leq N$ operator.
Cost LTL

- **CLTL** on $\mathbb{A}$ describes regular cost functions:
  \[ \varphi := a \mid \varphi \wedge \varphi \mid \varphi \vee \varphi \mid \varphi R \varphi \mid \varphi U \varphi \mid \varphi U^\leq N \varphi \]

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

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

- The “error value” variable $N$ is unique, and is shared by all occurrences of $U^\leq N$ operator.

- $G^\leq N$ and $R^\leq N$ can be defined in terms of the previous operators.
CFO and CMSO

- CFO on $\mathbb{A}$ describes regular cost functions:

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

- As before, $N$ is a unique free variable and counts the number of mistakes.

- (Weak) CMSO extends CFO with quantification over (finite) sets.
Semantics of Cost Logics

From formula to cost function:

\(\llbracket \varphi \rrbracket\) is the cost function associated to \(\varphi\), 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 \(u \in \{a, b\}^\omega\), we have

- \(|u|_a = \llbracket bU^{\leq N}(Gb) \rrbracket (u) = \llbracket \forall x \leq N x. b(x) \rrbracket (u)\).

- \(\maxblock_a(u) = \llbracket G(\bot U^{\leq N} b) \rrbracket (u) = \llbracket \forall X, \block_a(X) \Rightarrow (\forall x \leq N x, x \notin X) \rrbracket (u)\).
Alternating $B$-automata

- **Alternating $B$-automaton**: Game between Eve and Adam, with counter actions on transitions. Eve must satisfy acceptance condition AND low counter value.
Alternating $B$-automata

- **Alternating $B$-automaton:** Game between Eve and Adam, with counter actions on transitions. Eve must satisfy acceptance condition AND low counter value.

- **Weak $B$-automaton:** Büchi condition, no cycle with both accepting and rejecting states.
Alternating $B$-automata

- **Alternating $B$-automaton**: Game between Eve and Adam, with counter actions on transitions. Eve must satisfy acceptance condition AND low counter value.

- **Weak $B$-automaton**: Büchi condition, no cycle with both accepting and rejecting states.

- **Very-weak $B$-automaton**: Büchi condition, no non-trivial cycle.
Classical picture

Regular Languages

- MSO
- Weak MSO
- Büchi automata
- Weak automata

Star-Free Languages

- FO
- LTL
- Very-Weak automata
- aperiodic semigroups
Cost Functions

Regular Cost Functions

CMSO  $B/S$-Büchi automata
WCMSO  Weak $B$-automata

First-Order Fragment

CFO
CLTL
Very-Weak $B$-automata
VWBA with one counter
(aperiodic stab. semigroups)
Proof ideas for WCMSO to CMSO

- By Colcombet, CMSO $\Leftrightarrow$ nondeterministic $B/S$-Büchi automata.
- By [Vanden Boom 11], WCMSO $\Leftrightarrow$ weak alternating $B$-automata.

We just need to show a translation from nondeterministic $B$-Büchi automata to weak alternating $B$-automata.
Fix a word $u$, and analyze the run-DAG of the Büchi-automaton on $u$ (here for $u = baab^\omega$):

**Ranks**: No more Büchi or finite path on the remaining DAG. Initial node gets a rank $\Rightarrow u$ is rejected.
Fix a word $u$, and analyze the run-DAG of the Büchi-automaton on $u$ (here for $u = baab^\omega$):

**Ranks**: No more Büchi or finite path on the remaining DAG. Initial node gets a rank $\Rightarrow u$ is rejected.
Fix a word $u$, and analyze the run-DAG of the Büchi-automaton on $u$ (here for $u = baab^\omega$):

**Ranks** : No more Büchi or finite path on the remaining DAG. Initial node gets a rank $\Rightarrow u$ is rejected.
Extending to cost functions

Run-DAG for $u = a^\omega$:

Problem to assign ranks: how to prove that this run has value $\infty$?
Solution:
Normal form for nondeterministic $B$-Büchi automata: must do a reset on every counter after each Büchi state.
The modified automaton guesses whether there is

- a finite number of increments $\Rightarrow$ ignore early Büchi states:

```
| IC IC IC IC | R R R |
| B B B B B B |
```

- infinitely many resets $\Rightarrow$ delay Büchi states locally:

```
| R R R R R |
| B B B B B |
```

On these Büchi automata in normal form, we can define ranks in a sound way, for each value $n$. 
Description of the weak alternating automaton

The weak $B$-automaton $W$ describes a game between two players:

- Eve wants to prove that $A$ accepts with low value
- Adam wants to prove that this is not the case

It allows Eve to play a run of $A$, and Adam to guess ranks. It is designed in such a way that for all $n \in \mathbb{N}$:

- playing a $n$-run (if exists) is a strategy of value $\leq n$ for Eve.
- playing the $n$-ranks (if possible) is a strategy of value $> n$ for Adam.

From this we get $[W] = [A]$. 
Summary

**Regular Cost Functions**
- CMSO: $B/S$-Büchi automata
- WCMSO: Weak $B$-automata

**First-Order Fragment**
- CFO
- CLTL
  - Very-Weak $B$-automata
  - VWBA with one counter
    - (aperiodic stab. semigroups)

What are the limits of this correspondence?
On Infinite trees

Theorem (Rabin 1970, Kupferman + Vardi 1999)
A language $L$ of infinite trees is recognizable by an alternating weak automaton iff there are nondeterministic Büchi automata $U$ and $U'$ such that

$$L = L(U) = \overline{L(U')}.$$
Extension to Cost Functions

Complementation becomes switching between $B$- and $S$-semantic:

Inclusions are strict, and intersection is effective.
Application

Going back to the initial purpose of cost functions: deciding problems on languages.

**Theorem**

*Boundedness of Quasi-Weak automata is decidable.*

**Theorem (Colcombet+Löding ’08)**

*Given a regular language $L$ and a parity rank $[i,j]$, we can build a $B-[i,j]$-parity automaton, which computes $\chi_L$ iff $L$ is recognizable by a nondeterministic $[i,j]$-parity automaton.*

**Corollary**

*Given a nondeterministic Büchi automaton for a language $L$, we can decide whether $L$ is weak.*
Proof scheme of the corollary

Input: Büchi automaton $A$ for $L$

- Build a parity automaton $A'$ for $\overline{L}$, the complement of $L$. 

Why does it work?

- If $L$ is weak then $[\mathcal{W}] \approx \chi_L$ by correctness of the construction.
- If $[\mathcal{W}] \approx \chi_L$, then $L$ is weak, because it is recognized by an unfolding of $\mathcal{W}$. 

Proof scheme of the corollary

Input: Büchi automaton $A$ for $L$

- Build a parity automaton $A'$ for $\bar{L}$, the complement of $L$.
- Use [CL08] to get a $B$-Büchi automaton $B$, which recognizes $\chi_{\bar{L}}$ iff $\bar{L}$ is Büchi-recognizable.

Why does it work?

- If $L$ is weak then $[\bar{W}] \approx \chi_{\bar{L}}$ by correctness of the construction.
- If $[\bar{W}] \approx \chi_{\bar{L}}$, then $L$ is weak, because it is recognized by an unfolding of $W$. 
Proof scheme of the corollary
Input: Büchi automaton $A$ for $L$

- Build a parity automaton $A'$ for $\overline{L}$, the complement of $L$.
- Use [CL08] to get a $B$-Büchi automaton $B$, which recognizes $\chi_{\overline{L}}$ iff $\overline{L}$ is Büchi-recognizable.
- Consider $A$ as an $S$-Büchi automaton for $\chi_{\overline{L}}$. 

Why does it work?

- If $L$ is weak then $[\overline{W}] \approx \chi_{\overline{L}}$ by correctness of the construction.
- If $[\overline{W}] \approx \chi_{\overline{L}}$, then $L$ is weak, because it is recognized by an unfolding of $W$. 


Proof scheme of the corollary
Input: Büchi automaton $A$ for $L$

- Build a parity automaton $A'$ for $\overline{L}$, the complement of $L$.
- Use [CL08] to get a $B$-Büchi automaton $B$, which recognizes $\chi_{\overline{L}}$ iff $\overline{L}$ is Büchi-recognizable.
- Consider $A$ as an $S$-Büchi automaton for $\chi_{\overline{L}}$.
- Build the Quasi-Weak automaton $W$ from $A$ and $B$. 

Why does it work?

- If $L$ is weak then $[W] \approx \chi_{\overline{L}}$ by correctness of the construction.
- If $[W] \approx \chi_{\overline{L}}$, then $L$ is weak, because it is recognized by an unfolding of $W$. 
Proof scheme of the corollary
Input: Büchi automaton $\mathcal{A}$ for $L$

- Build a parity automaton $\mathcal{A}'$ for $\overline{L}$, the complement of $L$.
- Use [CL08] to get a $B$-Büchi automaton $\mathcal{B}$, which recognizes $\chi_{\overline{L}}$ iff $\overline{L}$ is Büchi-recognizable.
- Consider $\mathcal{A}$ as an $S$-Büchi automaton for $\chi_{\overline{L}}$.
- Build the Quasi-Weak automaton $\mathcal{W}$ from $\mathcal{A}$ and $\mathcal{B}$.
- Decide whether $[\mathcal{W}] \approx \chi_{\overline{L}}$.

Why does it work?
- If $L$ is weak then $[\mathcal{W}] \approx \chi_{\overline{L}}$ by correctness of the construction.
- If $[\mathcal{W}] \approx \chi_{\overline{L}}$, then $L$ is weak, because it is recognized by an unfolding of $\mathcal{W}$. 
Proof scheme of the corollary

Input: Büchi automaton $A$ for $L$

- Build a parity automaton $A'$ for $\bar{L}$, the complement of $L$.
- Use [CL08] to get a $B$-Büchi automaton $B$, which recognizes $\chi_{\bar{L}}$ iff $\bar{L}$ is Büchi-recognizable.
- Consider $A$ as an $S$-Büchi automaton for $\chi_{\bar{L}}$.
- Build the Quasi-Weak automaton $W$ from $A$ and $B$.
- Decide whether $[W] \approx \chi_{\bar{L}}$.

Why does it work?
Proof scheme of the corollary

Input: Büchi automaton $A$ for $L$

- Build a parity automaton $A'$ for $\overline{L}$, the complement of $L$.
- Use [CL08] to get a $B$-Büchi automaton $B$, which recognizes $\chi_{\overline{L}}$ iff $\overline{L}$ is Büchi-recognizable.
- Consider $A$ as an $S$-Büchi automaton for $\chi_{\overline{L}}$.
- Build the Quasi-Weak automaton $W$ from $A$ and $B$.
- Decide whether $[W] \approx \chi_{\overline{L}}$.

Why does it work?

- If $L$ is weak then $[W] \approx \chi_{\overline{L}}$ by correctness of the construction.
Proof scheme of the corollary

Input: Büchi automaton $A$ for $L$

- Build a parity automaton $A'$ for $\overline{L}$, the complement of $L$.
- Use [CL08] to get a $B$-Büchi automaton $B$, which recognizes $\chi_{\overline{L}}$ iff $\overline{L}$ is Büchi-recognizable.
- Consider $A$ as an $S$-Büchi automaton for $\chi_{\overline{L}}$.
- Build the Quasi-Weak automaton $W$ from $A$ and $B$.
- Decide whether $\llbracket W \rrbracket \approx \chi_{\overline{L}}$.

Why does it work?

- If $L$ is weak then $\llbracket W \rrbracket \approx \chi_{\overline{L}}$ by correctness of the construction.
- If $\llbracket W \rrbracket \approx \chi_{\overline{L}}$, then $L$ is weak, because it is recognized by an unfolding of $W$. 
Thank you!