Regular cost functions

Denis Kuperberg
(joint work with Thomas Colcombet and Sylvain Lombardy)
Liafa, équipe Automates

06/11/2009
Outline

Introduction

Cost functions

Cost automata
  Definition
  Semantics
  Examples
  Temporal automata

Stabilization semigroups
  Definition
  Semantic
  Temporal class

Conclusion
Cost functions

Notations: $\mathbb{A}$ finite alphabet, $\omega + 1 = \omega \cup \{\omega\}$
If $f, g : \mathbb{A}^* \rightarrow \omega + 1$, then

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

iff $\exists \alpha : \omega \rightarrow \omega$ such as $f \leq \alpha \circ g$ and $g \leq \alpha \circ f$

Cost function: element of $(\omega + 1)^{\mathbb{A}^*/\approx}$.
Extension of the notion of language.

Example
For $\mathbb{A} = \{a, b, c\}$,
max$(| \cdot |_a, | \cdot |_b) \approx | \cdot |_a + | \cdot |_b$,
but $| \cdot |_a \not\approx u \leftrightarrow \max\{n/a^n \text{ factor of } u\}$
Introduction

Cost functions

Cost automata
  Definition
  Semantics
  Examples
  Temporal automata

Stabilization semigroups
  Definition
  Semantic
  Temporal class

Conclusion
Cost automata

**Aim**: Calculating cost functions with (nondeterministic) automata.

**Idea**: Counters which can be modified when a transition is used.

Formally, \( \mathcal{A} = \langle Q, A, In, Fin, \Gamma, \Delta \rangle \) with

\[
\Delta \subseteq Q \times A \times (\{i, r, c\}^*)^\Gamma \times Q
\]

**Actions**: increment \((i)\), reset \((r)\), check \((c)\).

**Example**

\( a : ic \)  

\( b : \varepsilon \)
Semantics of cost automata

Let $C(e) = \{\text{values of counters checked during the execution } e\}$

$$\llbracket A \rrbracket_B(u) = \inf \{ \sup C(e) : e \text{ run over } u \},$$

and $$\llbracket A \rrbracket_S(u) = \sup \{ \inf C(e) : e \text{ run over } u \}.$$  

are two semantics defining respectively $B$-automata and $S$-automata.
Examples of cost automata

With one counter, \( A \) (\( B \)-automaton) and \( A' \) (\( S \)-automaton):

\[
\begin{align*}
A & : \text{ic} \\
b & : \text{r}
\end{align*}
\]

\[
\begin{align*}
a, b & : \text{r} \\
a & : \text{i}
\end{align*}
\]

\[
\begin{align*}
a, b & : \text{r} \\
a & : \text{i} \\
b & : \text{cr}
\end{align*}
\]

\[
\begin{align*}
a, b & : \text{r} \\
a & : \text{i} \\
a & : \text{cr}
\end{align*}
\]

\( \llbracket A \rrbracket_B \approx \llbracket A' \rrbracket_S \approx \) block-size with
block-size\((u) = \max\{n \in \omega \mid a^n \text{ factor of } u\} \).

**Theorem (Colcombet 09)**

\( B \)-automata and \( S \)-automata are equivalent (modulo \( \approx \)) from the point of view of recognized cost functions.
Temporal automata

Simple automata: Only actions: $\{\varepsilon, ic, r\}$ for $B$-automata and $\{\varepsilon, i, r, cr\}$ for $S$-automata.

Temporal automata: Intuitive idea: measuring the time. Only actions: $\{ic, r\}$ for $B$-automata (Kirsten and Bala’s desert automata) and $\{i, r, cr\}$ for $S$-automata.

Example

For $A = \{a, b\}$, block-size is temporal, but $| \cdot |_a$ is not.

Theorem

For a cost function, it is equivalent (modulo $\approx$) to be recognized by

- temporal $B$-automaton
- temporal $B$-automaton with 1 counter
- temporal $S$-automaton
- temporal $S$-automaton with 1 counter

We say then that the cost function is temporal.
A little proof

Proposition

$| \cdot |_a$ is not temporal.

Proof

Let $\mathcal{A} = \langle Q, \Delta, \mathit{In}, \mathit{Fin}, \{\gamma\}, \Delta \rangle$ temporal $B$-automaton computing $g \approx \alpha | \cdot |_a$ for some $\alpha$.

Let $K > 2|Q| + 1$ and $N > \alpha(K)$.

Let $e$ minimal run of $\mathcal{A}$ over $u = (b^Na)^K$.

We took $K > 2|Q| + 1$, so we can write $u = xvy$, with $|v|_a \geq 2$ and in $e$, $p \xrightarrow{v} p$.

- Path $a \xrightarrow{i\ldots i} a$ in $v$, then $g(u) \geq N > \alpha(|u|_a)$, absurd.

- Each path $a \rightarrow a$ in $v$ contains a reset.
  Then $\forall m, g(xv^my) \leq 2g(u)$, absurd because $|v|_a > 0$. 
Introduction

Cost functions

Cost automata
   Definition
   Semantics
   Examples
   Temporal automata

Stabilization semigroups
   Definition
   Semantic
   Temporal class

Conclusion
Stabilization semigroups

Remind : regular language $\Leftrightarrow$ finite semigroup (Myhill) and star-free language $\Leftrightarrow$ group-trivial semigroup (Schützenberger)

Stabilization semigroup : $S = \langle S, \cdot, \leq, \# \rangle$, ordered semigroup with a $\#$-operator : stabilization over idempotents ($e = e \cdot e$)

We note $S^1 = S \cup \{1\}$ the stabilization monoid associated to $S$. Extension of the standard semigroups and monoids.
Semantic through an example

Example: Stabilization semigroups for $| \cdot |_a$ and block-size:

\[
\begin{align*}
\tau \cdot x &= x \cdot \tau = \tau
\end{align*}
\]
Temporal stabilization semigroups

Definition
\( e \in E(S) \) is **stable** if \( e^\# = e \)

Definition
\( S \) **Temporal stabilization semigroup** :

\[ \forall e \in E(S) \text{ stable}, \forall x, y \in S^1, \ x \cdot e \cdot y \in E(S) \Rightarrow x \cdot e \cdot y \text{ stable.} \]

block-size :

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>a</td>
<td>b</td>
</tr>
<tr>
<td>b</td>
<td>b</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

\[ | \cdot | a : \]

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>a</td>
<td>a</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>b</td>
<td>b</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

\[ \# \cdot \# \]
Results

Theorem

A cost function is temporal iff it is recognized by a temporal stabilization semigroup.
Results

Theorem
A cost function is temporal iff it is recognized by a temporal stabilization semigroup.

Theorem
Let $f$ be a regular cost function,
Results

**Theorem**
A cost function is temporal iff it is recognized by a temporal stabilization semigroup.

**Theorem**
Let $f$ be a regular cost function,
- There exists a (quotient-wise) minimal stabilization semigroup $S$ recognizing $f$
Results

Theorem
A cost function is temporal iff it is recognized by a temporal stabilization semigroup.

Theorem
Let $f$ be a regular cost function,

- There exists a (quotient-wise) minimal stabilization semigroup $S$ recognizing $f$
- $S$ is computable effectively
Results

Theorem
A cost function is temporal iff it is recognized by a temporal stabilization semigroup.

Theorem
Let $f$ be a regular cost function,

- There exists a (quotient-wise) minimal stabilization semigroup $S$ recognizing $f$
- $S$ is computable effectively
- $f$ is temporal iff $S$ is temporal
Results

Theorem
A cost function is temporal iff it is recognized by a temporal stabilization semigroup.

Theorem
Let \( f \) be a regular cost function,

- There exists a (quotient-wise) minimal stabilization semigroup \( S \) recognizing \( f \)
- \( S \) is computable effectively
- \( f \) is temporal iff \( S \) is temporal

Corollary
It is decidable whether a regular cost function is temporal.
Conclusion

Summary:

- Temporal class defined via cost automata
- Simplifications of constructions in this class
- Characterization by stabilization semigroups
- Minimization of stabilization semigroups
- Decidability of the temporal class