The Theory of Stabilization Monoids and Regular Cost Functions

Thomas Colcombet

Monday, July 6th 2009
ICALP

Principle

Developing a quantitative notion of regularity which extends in many ways the standard notion of regularity:

- presentations: automata (det/non-det), algebra, logic, regular expressions
- closure: union, intersection, projection, complementation
- decidability: emptiness

Key idea. Consider mappings $f, g : \mathbb{A}^* \to \omega + 1$ modulo an equivalence which preserves the existence of bounds:

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

Origin and motivations

A model with many “good” properties is worth being studied.

Our automata are extensions of distance automata [Hashiguchi81], desert automata [Kirsten04], and nested distance desert automata [Kirsten05].

Those models were introduced for deciding language theoretic questions: the finite power problem, the finite substitution problem, the star-height problem.

All problems can be solved by a reduction to a problem of existence of bound for the above automata.

In [Bojanczyk, C.06], B-automata (similar) were introduced together with a dual variant, S-automata.

Outline

Cost functions

Definition

Consider mappings $f, g : E \to \omega + 1$ ($E = \mathbb{A}^*$ in this work), define

$f \preccurlyeq g$ if $\forall X \subseteq \mathbb{A}^*.\ g|_X$ is bounded implies $f|_X$ is bounded

$f \approx g$ if $f \preccurlyeq g$ and $g \preccurlyeq f$

A cost function (over E) is an equivalence class for $\approx$.

Cost functions

Definition

Consider mappings $f, g : E \to \omega + 1$ ($E = \mathbb{A}^*$ in this work), define

$f \preccurlyeq g$ if $\forall X \subseteq \mathbb{A}^*.\ g|_X$ is bounded implies $f|_X$ is bounded

$f \approx g$ if $f \preccurlyeq g$ and $g \preccurlyeq f$

A cost function (over E) is an equivalence class for $\approx$.

Example

- $0 \preccurlyeq |\cdot|_a \preccurlyeq |\cdot|$
- $|\cdot|_a$ and $|\cdot|_b$ are incomparable
- $|\cdot|_a + |\cdot|_b \approx \max(|\cdot|_a, |\cdot|b)$ (more generally, $\max \approx +$)

Cost functions

Definition

Consider mappings $f, g : E \to \omega + 1$ ($E = \mathbb{A}^*$ in this work), define

$f \preccurlyeq g$ if $\forall X \subseteq \mathbb{A}^*.\ g|_X$ is bounded implies $f|_X$ is bounded

$f \approx g$ if $f \preccurlyeq g$ and $g \preccurlyeq f$

A cost function (over E) is an equivalence class for $\approx$.

Remark (Cost functions extend languages)

For $L \subseteq E$ (a language), define $\chi_L : E \to \omega + 1$ by:

$$\chi_L(x) = \begin{cases} 0 & \text{if } x \in L \\ \omega & \text{otherwise} \end{cases}$$

Then for all $K, L \subseteq E$, $\chi_K \preccurlyeq \chi L$ iff $L \subseteq K$.

Computing the cost of a sequence

Let Γ be a finite set of counters.

Semantics: counters have initial value 0, and one can perform on them:

- ϵ, which does nothing,
- i which increments the counter by one,
- r which resets the counter (to 0), and;
- c which checks (i.e., observes/tests) the counter value.

Given a sequence $u \in (\{\epsilon, i, r, c\}^{\Gamma})^*$, $C(u)$ is the set of values of counters when checked.

Cost automata

Definition

A cost automaton $\mathcal{A} = (Q, \mathbb{A}, I, F, \Gamma, \Delta)$ has:

- a finite set Q of states, an input alphabet $\mathbb{A}$, a set I of initial states, a set F of final states,
- a finite set Γ of counters,
- a transition relation $\Delta \subseteq Q \times \mathbb{A} \times \{\epsilon, i, r, c\}^{\Gamma} \times Q$.

Runs are defined as usual. One defines:

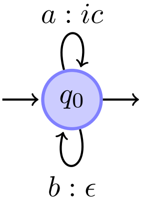
$$\begin{aligned} [\![\mathcal{A}]\!]_B : \quad & \mathbb{A}^* \mapsto \omega + 1 \\ & u \mapsto \inf\{\sup C(\rho) \ : \ \rho \text{ accepting run of } \mathcal{A} \text{ over } u\} \\ [\![\mathcal{A}]\!]_S : \quad & \mathbb{A}^* \mapsto \omega + 1 \\ & u \mapsto \sup\{\inf C(\rho) \ : \ \rho \text{ accepting run of } \mathcal{A} \text{ over } u\} \end{aligned}$$

Cost automata are called B-automata or S-automata accordingly.

Example of a deterministic B-automaton

$$[\![\mathcal{A}]\!]_B(u) = \inf\{\sup C(\sigma) \ : \ \sigma \text{ accepting run of } \mathcal{A} \text{ over } u\}$$

The following deterministic B-automaton counts the number of occurrences of 'a'.

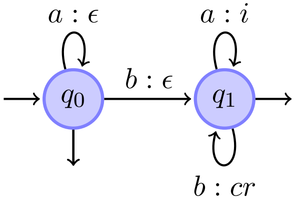


$C(\sigma) = \{1, 2, \ldots, |u|_a\}$

Example of a deterministic S-automaton

$$[\![\mathcal{A}]\!]_S(u) = \sup\{\inf C(\sigma) \ : \ \sigma \text{ accepting run of } \mathcal{A} \text{ over } u\}$$

The following deterministic S-automaton computes the minimal distance between two b's (ω if less than two b's).

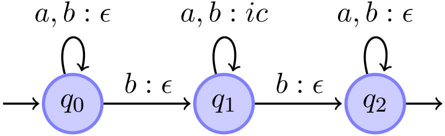


$C(\sigma) = \{$distances between consecutive b's$\}$

Example of a (non-deterministic) B-automaton

$$[\![\mathcal{A}]\!]_B(u) = \inf\{\sup C(\sigma) \ : \ \sigma \text{ accepting run of } \mathcal{A} \text{ over } u\}$$

The following B-automaton computes the minimal distance between two b's (ω if less than two b's).



$C(\sigma) = \{1, 2, \ldots,$ distance between the two guessed b's$\}$

Link with the language case

$$[\![\mathcal{A}]\!]_B(u) = \inf\{\sup C(\sigma) \ : \ \sigma \text{ accepting run of } \mathcal{A} \text{ over } u\}$$
$$[\![\mathcal{A}]\!]_S(u) = \sup\{\inf C(\sigma) \ : \ \sigma \text{ accepting run of } \mathcal{A} \text{ over } u\}$$

Remark

If $\mathcal{A}$ has no counters and accepts L, then $\begin{cases} [\![\mathcal{A}]\!]_B = \chi_L \\ [\![\mathcal{A}]\!]_S = \chi_{\complement L} \end{cases}$

Closure results

Fact

Cost functions accepted by B-automata (resp. S-automata) are closed under $\min$ *and* $\max$*.*
Cost functions accepted by B-automata (resp. S-automata) are closed under $\inf$*-projection (resp.* $\sup$*-projection).*

This corresponds to union, intersection and projection for regular languages.

Closure results

Fact

Cost functions accepted by B-automata (resp. S-automata) are closed under $\min$ *and* $\max$.
Cost functions accepted by B-automata (resp. S-automata) are closed under $\inf$*-projection (resp.* $\sup$*-projection).*

This corresponds to union, intersection and projection for regular languages.

Theorem (duality, [here] and [Bojanczyk&C.06])

A cost function is accepted by a B-automaton iff it is accepted by an S-automaton (modulo $\approx$*). The equivalences are elementary.*
We call such cost functions regular.

This corresponds to the complementation of regular languages.

Closure results

Fact

Cost functions accepted by B-automata (resp. S-automata) are closed under $\min$ *and* $\max$.
Cost functions accepted by B-automata (resp. S-automata) are closed under $\inf$*-projection (resp.* $\sup$*-projection).*

This corresponds to union, intersection and projection for regular languages.

Theorem (duality, [here] and [Bojanczyk&C.06])

A cost function is accepted by a B-automaton iff it is accepted by an S-automaton (modulo $\approx$). The equivalences are elementary.
We call such cost functions regular.

This corresponds to the complementation of regular languages.

Proof: go to the algebraic world...

Key decidability result

Theorem

The relation $\preccurlyeq$ is decidable over regular cost functions.

This corresponds to the decidability of the inclusion for regular langauges.

Corollary (boundedness, limitedness)

One can decide whether a regular cost function is bounded.

Proof. f is bounded iff $f \preccurlyeq 0$. $\square$

Stabilization monoids (following I. Simon)

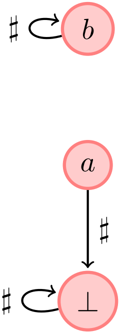
Definition

A stabilization monoid $\langle M, \cdot, \leq, \sharp \rangle$ is an ordered monoid $\langle M, \cdot, \leq \rangle$ together with a stabilization operator $\sharp : E(M) \to E(M)$ ($E(M)$ are the idempotents of M), such that:

- **consistency** if $a \cdot b, b \cdot a \in E(M)$, then $(a \cdot b)^\sharp = a \cdot (b \cdot a)^\sharp \cdot b$,
- **order** for $e, f \in E(M)$, $e \leq f$ implies $e^\sharp \leq f^\sharp \leq f$, and;
- **neutral** $1^\sharp = 1$.

Intuitively, $e^\sharp$ represents the value of e when iterated a lot of time.
If $e^\sharp = e$, one does not care about the number of occurrences of e.
If $e^\sharp < e$, the stabilization monoid 'counts' the iterations of e.

Example of a stabilization semigroup 'counting a's'



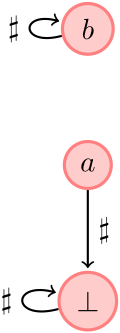
$$b = b \cdot b = b^\sharp \ (= 1)$$

$$a = a \cdot a = b \cdot a = a \cdot b \quad (\neq a^\sharp)$$

$$\perp = \star \cdot \perp = \perp \cdot \star = a^\sharp = \perp^\sharp$$

Example of a stabilization semigroup 'counting a's'

One classifies the words over M^* (with $M = \{a, b, \bot\}$)



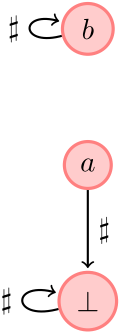
$b = b \cdot b = b^\sharp$ $(= 1)$
words in b^*

$a = a \cdot a = b \cdot a = a \cdot b$ $(\neq a^\sharp)$
words in $(b^*a)^+b^*$ with few a's

$\bot = \star \cdot \bot = \bot \cdot \star = a^\sharp = \bot^\sharp$
words containing $\bot$, or a lot of a's

Example of a stabilization semigroup 'counting a's'

One classifies the words over M^* (with $M = \{a, b, \perp\}$
Formally, $\rho : M^* \to \omega \to M$ (non-decreasing)



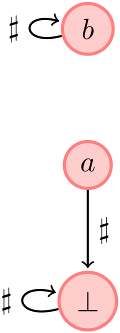
$b = b \cdot b = b^\sharp$ $(= 1)$
words in b^*
$\rho(u)(n) = b$ if $u \in b^*$

$a = a \cdot a = b \cdot a = a \cdot b$ $(\neq a^\sharp)$
words in $(b^*a)^+b^*$ with few a's
$\rho(u)(n) = b$ if $|u|_\perp = 0$ and $1 \leq |u|_a < n$

$\perp = \star \cdot \perp = \perp \cdot \star = a^\sharp = \perp^\sharp$
words containing $\perp$, or a lot of a's
$\rho(u)(n) = \perp$ if $|u|_\perp \geq 1$ or $n \leq |u|_a$

Example of a stabilization semigroup 'counting a's'

One classifies the words over M^* (with $M = \{a, b, \perp\}$
Formally, $\rho : M^* \to \omega \to M$ (non-decreasing)



$b = b \cdot b = b^\sharp$ $(= 1)$
words in b^*
$\rho(u)(n) = b$ if $u \in b^*$

$a = a \cdot a = b \cdot a = a \cdot b$ $(\neq a^\sharp)$
words in $(b^*a)^+b^*$ with few a's
$\rho(u)(n) = b$ if $|u|_\perp = 0$ and $1 \leq |u|_a < n$

$\perp = \star \cdot \perp = \perp \cdot \star = a^\sharp = \perp^\sharp$
words containing $\perp$, or a lot of a's
$\rho(u)(n) = \perp$ if $|u|_\perp \geq 1$ or $n \leq |u|_a$

ρ is an example of a mapping compatible with $\langle M, \leq, \cdot, \sharp \rangle$.

Semantics of stabilization monoids

Theorem (Existence and unicity of semantics)

Every finite stabilization monoid admits a compatible mapping. Furthermore, it is unique up to $\sim$ (a variant of $\approx$).

Recognizability

Let $h : \mathbb{A} \to M$ (extended as a morphism from $\mathbb{A}^*$ to M^*), and I an ideal of $\langle M, \leq \rangle$. Define:

$$\text{for all } u \in \mathbb{A}^*, \qquad f(u) = \sup\{n \ : \ \rho(h(u)) \in I\} \ .$$

Then f is said recognized by M, h, I.

Theorem

A cost function is regular iff it is recognizable by a finite stabilization monoid.

Example: recall that $\rho(u)(n) = \perp$ iff $|u|_\perp \geq 1$ or $n \leq |u|_a$.
Set $f(a) = a$, $h(b) = b$, $I = \{\perp\}$. Then M, h, I recognizes $|\cdot|_a$.

Conclusion

Content of the paper:

- Cost functions extending languages
- Automata (B- and S-), duality, closure properties, decidability
- History-deteterminism
- Stabilization monoid/recognizability/equivalence with regularity

Related results, ongoing work, and possible extensions:

- Extension to infinite words, finite trees (with C. Löding), and infinite trees (open)
- Algebraic characterization of families of regular cost functions (with S. Lombardy and D. Kuperberg)
- Equivalence with a variant of monadic second-order logic