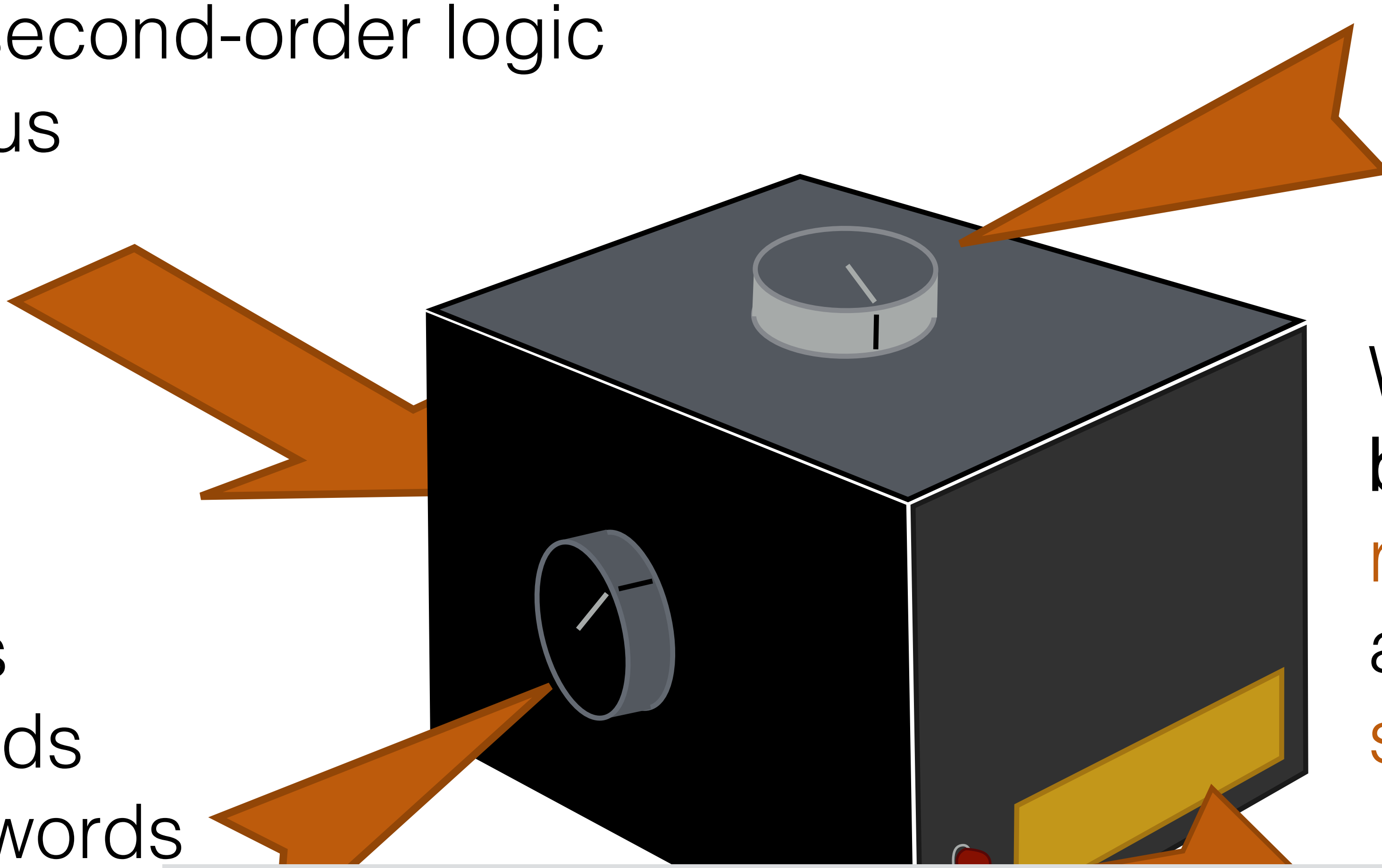
Logic and regular cost functions

LICS 2017
June 23

Thomas Colcombet

Regular languages as a blackbox for logic



Input specification

monadic second-order logic
mu-calculus
…

Problem type

universality
satisfiability
implication

We have another **blackbox** called the regular model-checker, and another one called synthesizer.

Domain

finite words
omega words
countable words
finite trees
infinite trees
tree-like structur
bounded tree-width graphs

Büchi, Elgot, Kleene, Kupferman, Gurevitch, Harrington, McNaughton, Muller, Rabin, Schützenberger, Safra, Scott, Schupp, Shelah, Trakhtenbrot, Vardi, Walukiewicz

ter example)

Monadic second-order logic

Monadic second order logic (MSO) expresses properties using

- the predicates of the structure, and membership
- Boolean connectives
- existential and universal quantifiers over elements (x,y,z…)
- existential and universal quantifiers over sets of elements (X,Y,Z,…)
- membership predicates

Example: over directed graphs,

$\texttt{Reach}(x, y)$ = « all sets that contain x and are closed under the edge relation also contain y »

$$\forall Z \quad ((x \in Z) \wedge (\forall z \forall z' (z \in Z \wedge \texttt{Edge}(z, z')) \to z' \in Z)) \quad \to y \in Z$$

Words can be seen as structures:

$$u = ([n], \leq, a_1, \ldots, a_n)$$

positions in the word

$a_i(m)$ holds if the letter at position m is a_i .

Over words MSO is equivalent to regular languages:

Non-determinsitic automata

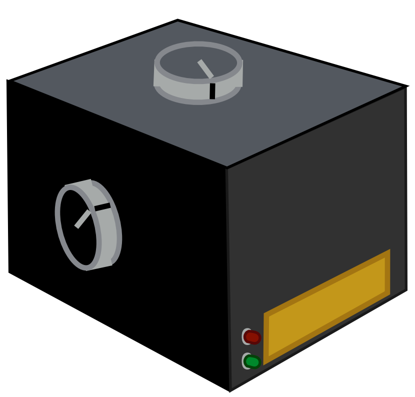
« there exists a coloring of the input word that encodes a run… »

The strength of rational regular languages

Regular languages are effectively equivalently described by
- logically: definability in monadic second-order logic (MSO).
- machines: non-deterministic, alternating automata (…)
- expressions: rational expressions, generalized versions, …
- algebra: recognizability by monoid, deterministic automata, …

Regular languages are closed under **union**, **intersection**, **complement**, **projection**, **concatenation**, etc…

Regular languages have decidable **emptiness**, **inclusion**, **universality**, **equivalence**.



Essentially works in two steps:
- translate the inputs in a suitable form
- solve the problem

Blackbox B

(B for boundedness)

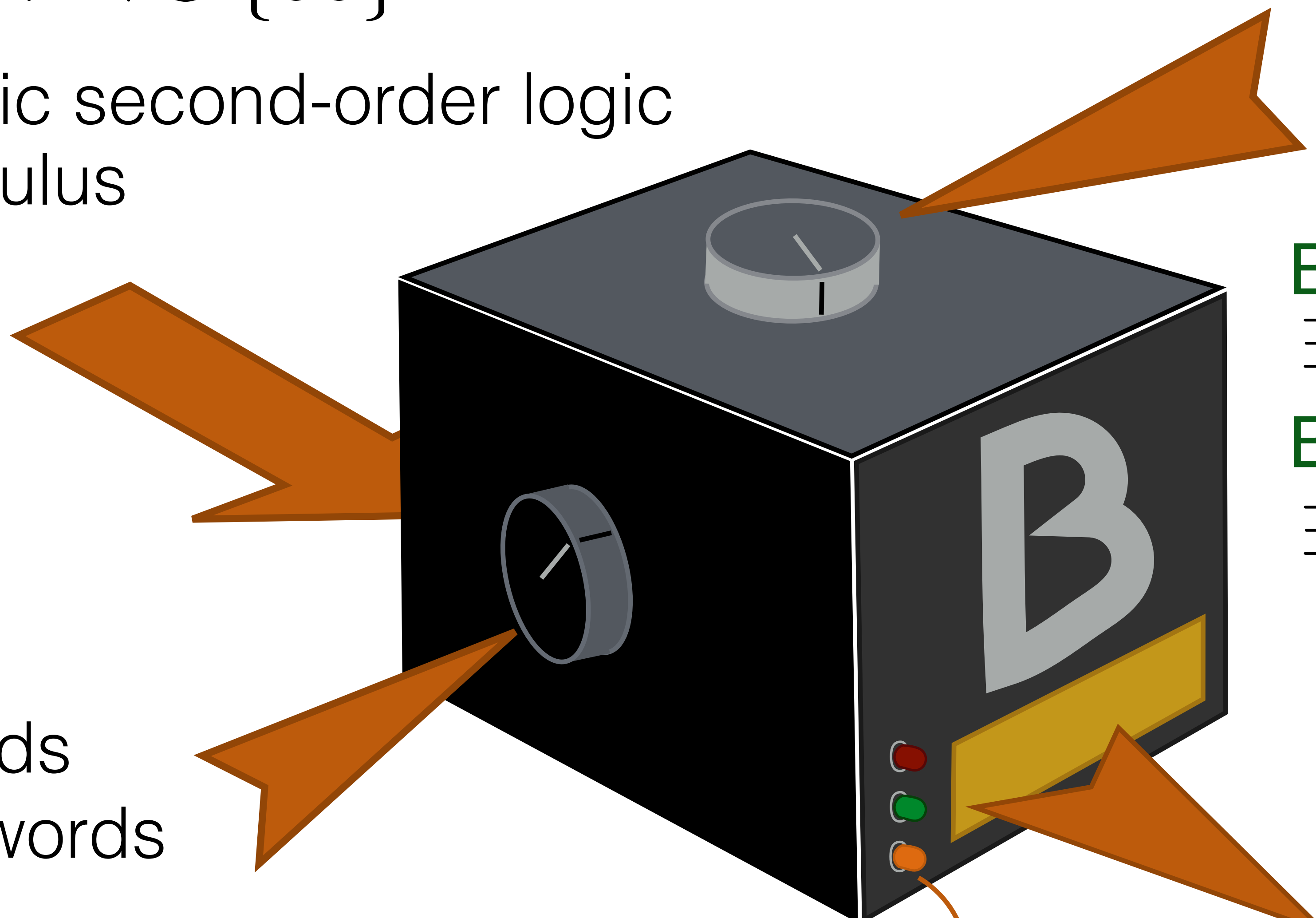
Functions (quantitive specification)

$\texttt{inputs} \to \mathbb{N} \cup \{\infty\}$

cost monadic second-order logic
mu-B—calculus
…

Domain

finite words
omega words
countable words
finite trees
infinite trees
tree-like structures
bounded tree-width graphs



Problem type

boundedness
divergence
domination

Boundedness

$\exists n \in \mathbb{N}\ \forall u\ \ f(u) \leq n$

Boundedness over K

$\exists n\ \forall u \in K\ \ f(u) \leq n$

yes
no
(counter example description)

ERROR (some problems are open)

Cost monadic second-order logic

Cost MSO = MSO + $|X| \leqslant n$ appearing positively.

Unique new variable ranging over naturals

A formula computes for each structure a value in $\mathbb{N} \cup \{\infty\}$:

$$[\![\varphi]\!](\mathfrak{A}) = \inf\{n \mid \mathfrak{A} \models \varphi(n)\}$$

Example: over directed graphs,

$$\texttt{Diameter}() = \forall x \forall y \exists Z\, \texttt{Reach}(x, y, Z) \wedge |Z| \leqslant n$$

It is possible to reach y from x while remaining in Z.

Over words

$|\ |_a$: number of occurrences of letter « a »

$\mathrm{maxblock}$: maximum length of a block of consecutive « a » letters

$\mathrm{minblock}$: minimum length of a block of consecutive « a » letters

distance (min-plus), max-plus and nested distance desert automata

Cost monadic second-order logic

Cost MSO = MSO + $|X| \leqslant n$ appearing positively.

Unique new variable ranging over naturals

A formula computes for each structure a value in $\mathbb{N} \cup \{\infty\}$:

$$[\![\varphi]\!](\mathfrak{A}) = \inf\{n \mid \mathfrak{A} \models \varphi(n)\}$$

disjunction = min

$$[\![\varphi \vee \psi]\!] = \min([\![\varphi]\!], [\![\psi]\!])$$

conjunction = max

$$[\![\varphi \wedge \psi]\!] = \max([\![\varphi]\!], [\![\psi]\!])$$

existential quantifier = infimum

$$[\![\exists X \ \phi(X)]\!] = \inf\{[\![\phi(X)]\!] \mid X \subseteq \mathcal{U}\}$$

universal quantifier = supremum

$$[\![\forall X \ \phi(X)]\!] = \sup\{[\![\phi(X)]\!] \mid X \subseteq \mathcal{U}\}$$

Cost monadic second-order logic

Cost MSO = MSO + $|X| \leqslant n$ appearing positively.

Unique new variable ranging over naturals

A formula computes for each structure a value in $\mathbb{N} \cup \{\infty\}$:

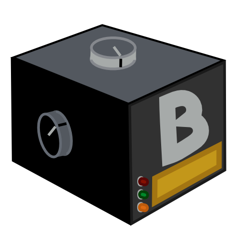
$$[\![\varphi]\!](\mathfrak{A}) = \inf\{n \mid \mathfrak{A} \models \varphi(n)\}$$

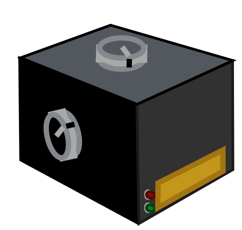
Remark

All MSO formula can be seen as a cost MSO formula (that would not use the new construct). Then:

$$[\![\varphi]\!](u) = \chi_{\mathcal{L}(\varphi)} \;\; = \begin{cases} 0 & \text{if } u \in \mathcal{L}(\varphi) \\ \infty & \text{otherwise.} \end{cases}$$

Hence, boundedness for cost MSO generalizes universality for MSO.

More generally, 

extends 

Blackbox B

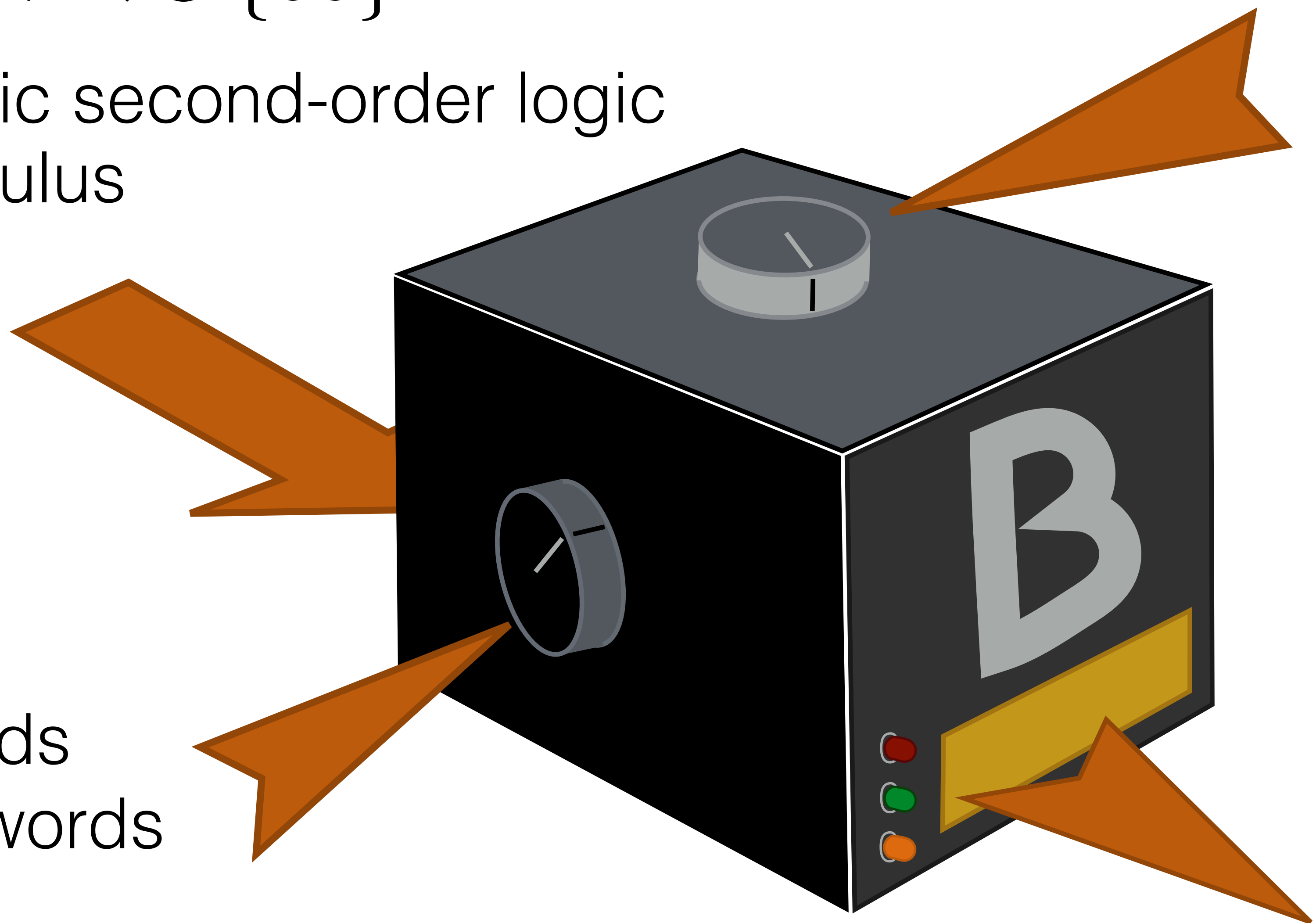
Functions (quantitive specification)

$\texttt{inputs} \to \mathbb{N} \cup \{\infty\}$

cost monadic second-order logic
mu-B—calculus
…

Problem type

boundedness
divergence
domination

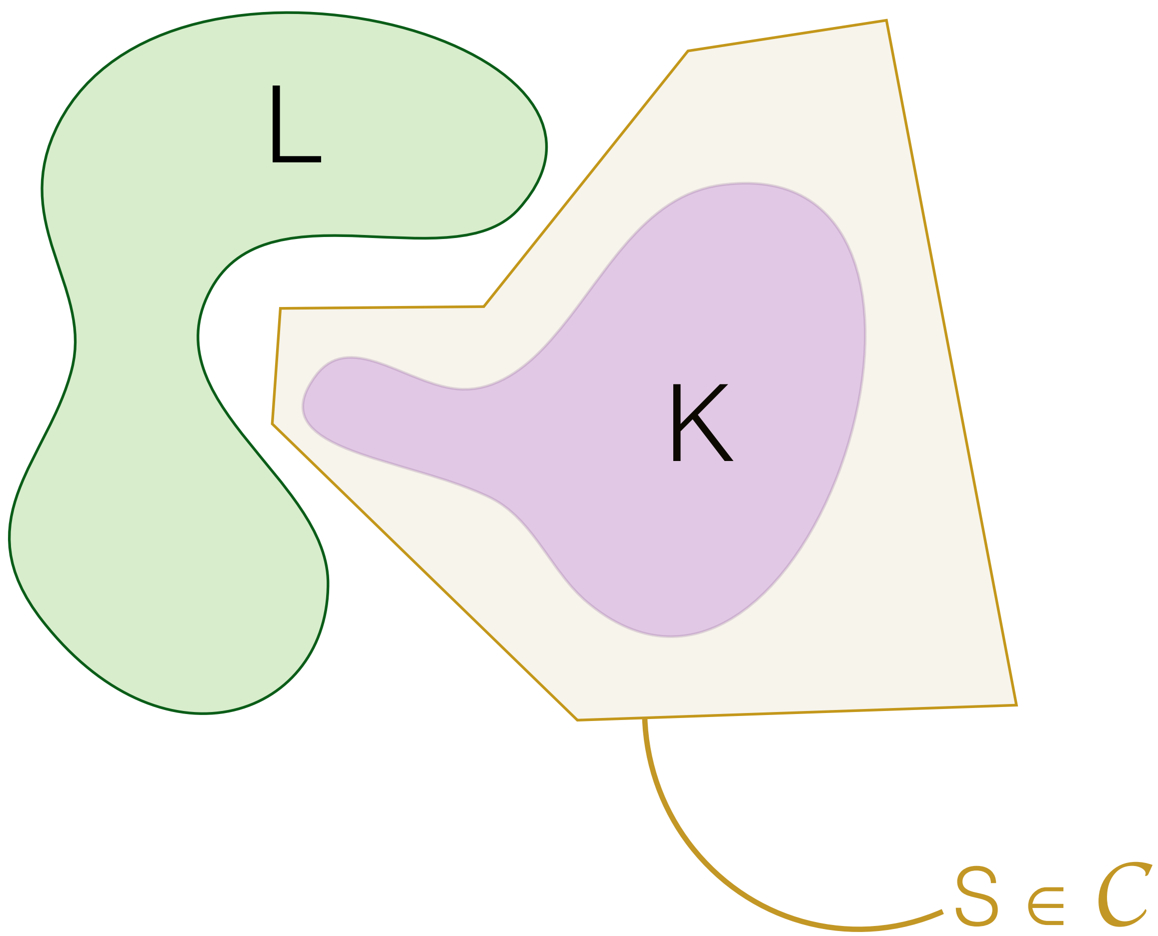


Domain

finite words
omega words
countable words
finite trees
infinite trees
tree-like structures
bounded tree-width graphs

yes
no
(counter example description)

Separation problems



« K is $\mathcal{C}$-separable from L »

MSO definable

Given two languages K,L, is it possible to separate them by a language from a given **class $\mathcal{C}$** ?

logical fragment
or some automata class

Remark: K is $\mathcal{C}$-separable from ᶜK if and only if K belongs to $\mathcal{C}$.

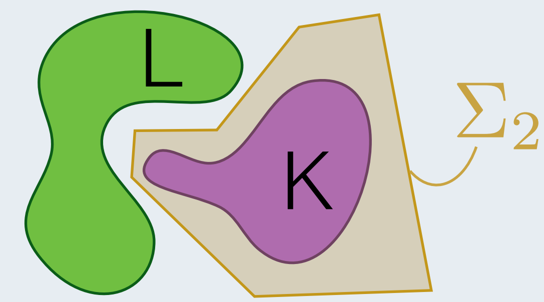
Separation by Σ_2

[Place&Zeitoun, Hashiguchi]

A Σ_2 -language is a language definable by a $\exists^*\forall^*$ first-order formula.

[Place&Zeitoun 14]
The Σ_2-separation problem is decidable.

Given two regular languages K,L, is it possible to separate them by a Σ_2 language?

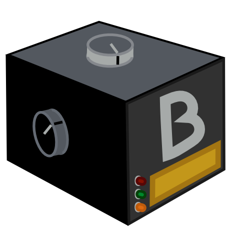


[Pin&Straubing] A language is Σ_2 iff it is a polynomial, i.e. a finite union of languages of the form $A_0^*b_1A_1^*b_2\dots b_kA_k^*$ (monomial of length k).

Define the property ψ of a word u (given an integer n):
it is accepted by some monomial of length k≤n that does not intersect L.

1) ψ is definable in cost MSO:
  a) guess a coloring of the form $A_0A_0\dots A_0b_1A_1\dots A_1b_2\dots b_kA_k\dots A_k$
  b) check that it accepts the input word and c) that k≤n
  d) verify that $A_0^*b_1A_1^*b_2\dots b_kA_k^*$ does not intersect L.

2) [ψ] is bounded over K iff K and L are Σ_2 -separable (next slide)

3) 

Generic recipe for separation

Consider a **class** $\mathcal{C}$ of machines/formulae that:

- define sets/languages (of words, trees, …) — polynomials
- is closed under union — obvious
- has a notion of **size** (of the acceptor) such that there are finitely many machines of a given size. — max dimension of a monomial

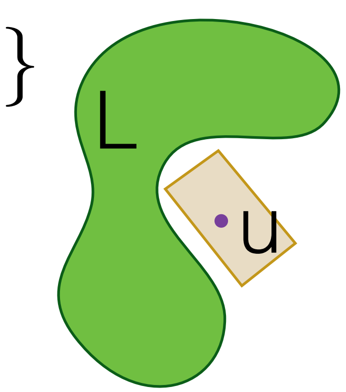
Fix a language L, and define — Problem in general: is it expressible?

$$\mathtt{sep}_L : \mathit{inputs} \to \mathbb{N} \cup \{\infty\}$$

$$u \mapsto \inf\{\mathit{size}(A) \mid A \in \mathcal{C},\ u \in \mathcal{L}(A),\ \mathcal{L}(A) \cap L = \emptyset\}$$

(this is the semantic of the ψ property)

measures how difficult it is to separate a single input u from L using a machine from the class $\mathcal{C}$.



Lemma: $\mathtt{sep}_L$ is bounded over K if and only if K is $\mathcal{C}$-separable from L.

⇐ it is bounded by $\mathit{size}(\mathtt{separator})$

⇒ Let n be the bound. Consider:

$$S = \bigcup_{\substack{\mathcal{L}(A) \cap L = \emptyset \\ \mathit{size}(A) \leq n}} \mathcal{L}(A)$$

Rabin-Mostowski hierarchy

Automata world

An infinite tree language as **RM-index (i,j)** if is is accepted by an alternating automaton using priorities in [i,j].

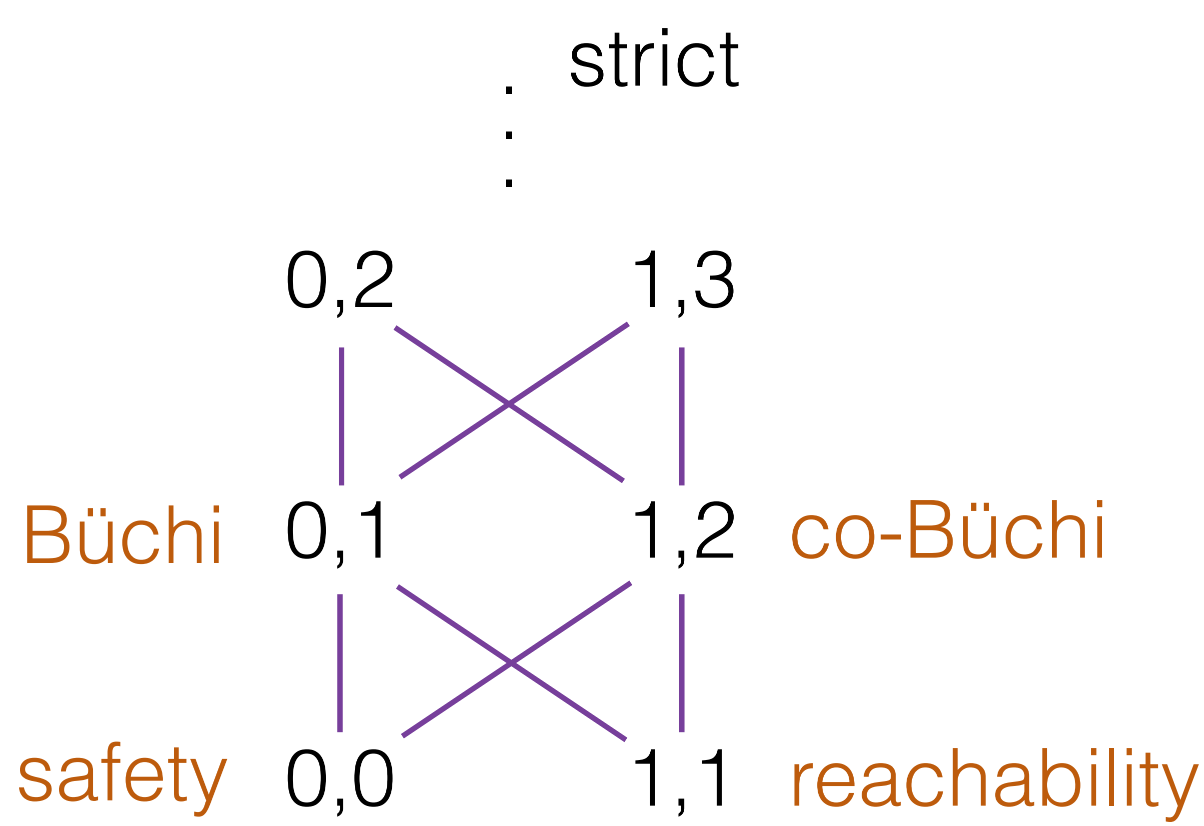
non-deterministic RM-index …

not very exact ⇔

Logic world

A μ-calculus definable statement as **RM index (i,j)** if it is defined by a formula of alternation j-i-1 (starting with μ or ν depending on the parity of j).

disjunctive RM-index …



[C.&Löding 08] (Reykjavic)
The problem of separation of regular language of infinite trees by NDRM (i,j)-languages
reduces to
the boundedness of cost MSO over infinite trees.

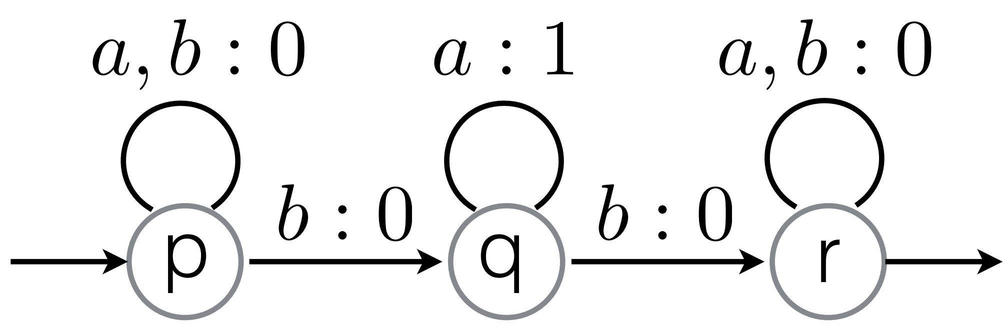
OPEN

Fixpoint boundedness

Is it possible to bound the number of iterations of a fixpoint ?

Distance automata

correspond to the semiring $(\mathbb{N} \cup \{\infty\}, 0, \infty, \min, +)$.



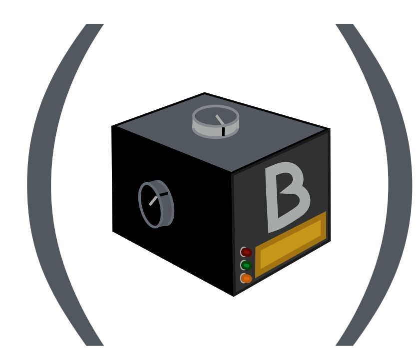
A distance automaton computes:

$$[\![\mathcal{A}]\!] : A^* \to \mathbb{N} \cup \{\infty\}$$

$u \mapsto$ the infimum over all accepting runs of the sum of weights.

minblock = The minimum length of a block of consecutive a-letters surrounded by b's.

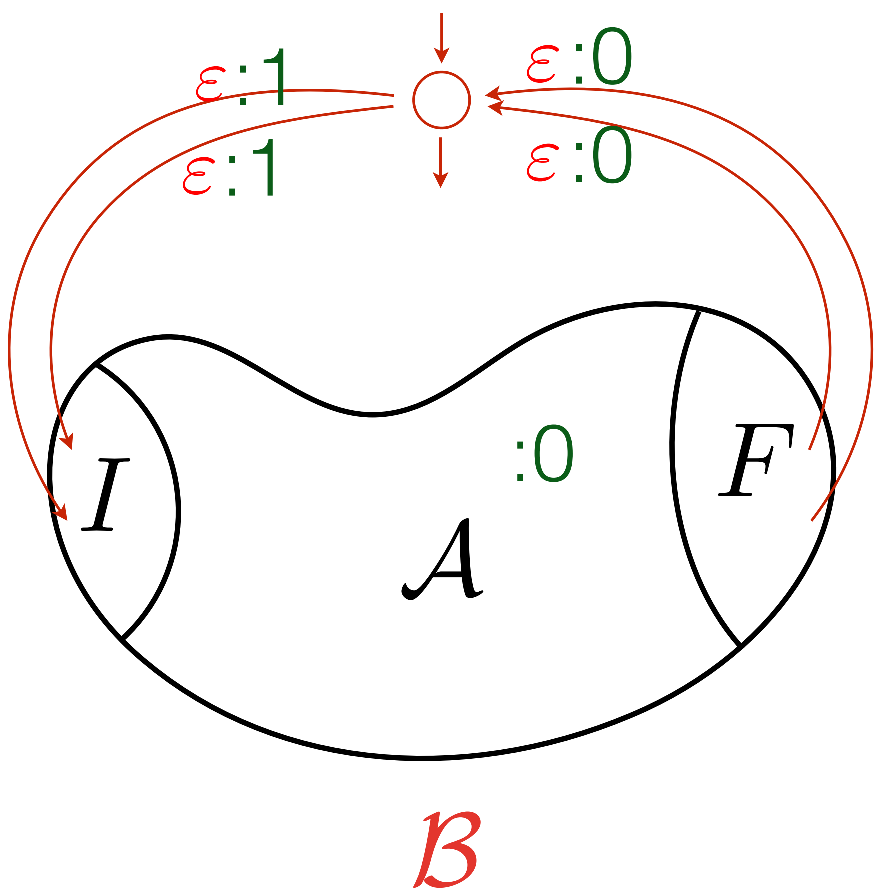
[Hashiguchi 81] The boundedness of distance automata is decidable.
[Leung88] [Simon78,94] [Kirsten05]
[C. & Bojanczyk 06] [C. 09] [Bojanczyk15]



Finite power property

Finite power property: $L^* = L^{\leqslant n}$ for some n ?

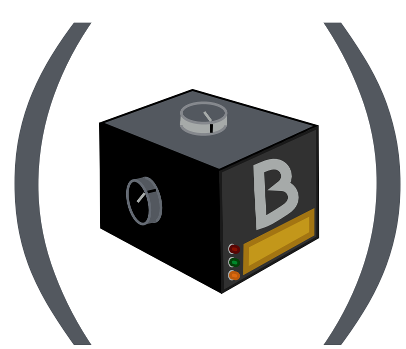
[Simon78] The finite power property is decidable.



Accepts u with weight $\leqslant n$
if and only if
$u \in L^{\leqslant n}$

Hence $[\![\mathcal{B}]\!](u) = \inf\{n \mid u \in L^{\leq n}\}$

Lemma: $[\![\mathcal{B}]\!]$ is bounded over L^* if and only if $L^* = L^{\leqslant n}$.



Counting Kleene stars

B-regular expressions

$$E ::= \emptyset \quad | \quad a \quad | \quad E+E \quad | \quad EE \quad | \quad E^* \quad | \quad E^{\leqslant n}$$

counting variant of Kleene star

$$[\![E]\!](u) = \inf\{n \mid u \in \mathcal{L}(E,n)\}$$

Examples

$$[\![a^{\leqslant n}]\!](u) = \begin{cases} |u| & \text{if } u \in a^* \\ \infty & \text{otherwise} \end{cases} \qquad [\![a^{\leqslant n}a^{\leqslant n}]\!](u) = \begin{cases} |u| & \text{if } u \in a^* \\ \infty & \text{otherwise} \end{cases}$$

$$[\![(b^*ab^*)^{\leqslant n}]\!](u) = |u|_a$$

$[\![a^{\leqslant n}(ba^{\leqslant n})^*]\!](u) =$ maxblock

$$a^{k_0}ba^{k_1}b\dots ba^{k_\ell} \mapsto \max(k_0,\dots,k_\ell)$$

$[\![(a^*b)^*\,a^{\leqslant n}\,(ba^*)^*]\!](u) =$ minblock

$$a^{k_0}ba^{k_1}b\dots ba^{k_\ell} \mapsto \min(k_0,\dots,k_\ell)$$

Finite power property becomes:
Is $[\![L^{\leqslant n}]\!]$ bounded over L^* ?

Bounding fixpoints

Fixpoints: Let $\varphi(x,Y)$ be a monadic second-order formula positive in Y. It has a fixpoint which is the least set Y s.t. $\{x \mid \mathfrak{A} \models \varphi(x,Y)\} \subseteq Y$.

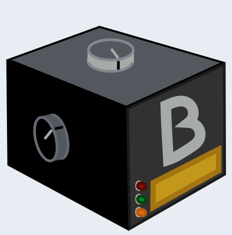
It is the least fixpoint of $F_{\mathfrak{A}} : Z \mapsto \{x \mid \mathfrak{A} \models \varphi(x,Z)\}$.

Fixpoint boundedness problem (generalization of finite power property):
Does there exist $n \in \mathbb{N}$ such that $F^n_{\mathfrak{A}}(\emptyset) = \mathit{fixpoint}(F_{\mathfrak{A}})$ for all input $\mathfrak{A}$?

> [Blumensath&Otto&Weyer12] The boundedness of a fixpoint of a monadic second-order logic is decidable over the class of infinite trees.

The map $(t,x) \mapsto \inf\{n \mid x \in F^n_t(\emptyset)\}$ for x a node of an infinite tree t is computed by an alternating two-way distance parity automaton running over t starting from x.

Boundedness of such automata is decidable.



> cost MSO extended with $\mathtt{Fix}^{\leqslant n}_{\varphi}$ has decidable boundedness, divergence and domination over finite words, infinite words, countable words, finite trees (over infinite trees for cost WMSO).

Church synthesis with bounded error

[Church synthesis problem] Two players (controller and environment) alternatively choose a letter, producing in the end an infinite word. Controller wins if he can guarantee this word to be in a given ω-regular language.
Questions are:
- decide the winner,
- if controller wins, provide a finite automaton implementing the strategy.

[Büchi&Landweber69] This is doable.

[Rabinowich&Velner11] is it possible to decide if there exists n such that controller can guarantee being in L up to a change of up to n of his letters.

[Regular cost functions] immediately yes (the idea is that the number letters that have to be modified for reaching the language is cost MSO definable).

Applications

Separation: Sigma2, star height for words [Hiaschiguchi88,Kirsten05], for trees [C.&Löding08], for the Rabin-Mostowski index [C.&Löding08]

Fixpoints for MSO [Blumensath&Otto&Weyer12], for guarded negation datalog [Barany&tenCate&Otto12,Benedikt&tenCate&C.&vandenBoom], […]

Quantitative version of Church Synthesis problem (open question of [Rabinovich and Velner 2011])

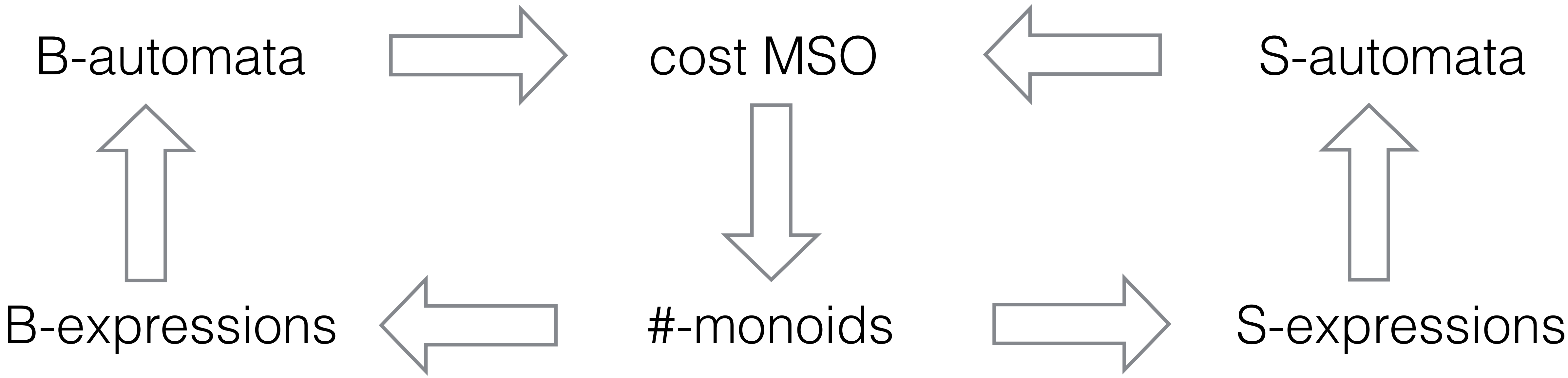
Finite satisfiability of modal mu-calculus with backward modalities [Bojanczyk 02]

Weak MSO+U over infinite trees [Bojanczyk and Torunczyk 2011]

Games with promises [C. and Göller 2015]

Non-standard models

Reasoning up to boundedness



example of conversions for words functions

When performing such translation it is in general not possible to preserve exactly the function computed (morally because of [Krob 94]).

For instance $\max(|u|_a, |u|_b)$ is definable in cost MSO but not using a B-regular expression.

However, we only care about boundedness queries, and hence we do not care about the exact values of functions.

For instance $[\![c^*((a+b)c^*)^{\leqslant n}]\!]$ computes $|u|_a + |u|_b$ and this function is as good as $\max(|u|_a, |u|_b)$ as far as boundedness queries are concerned.

Boundedness equivalence

$f \preccurlyeq g$ (f **is dominated** by g) if for all sets X⊆A* $g|_X$ is bounded implies $f|_X$ is bounded

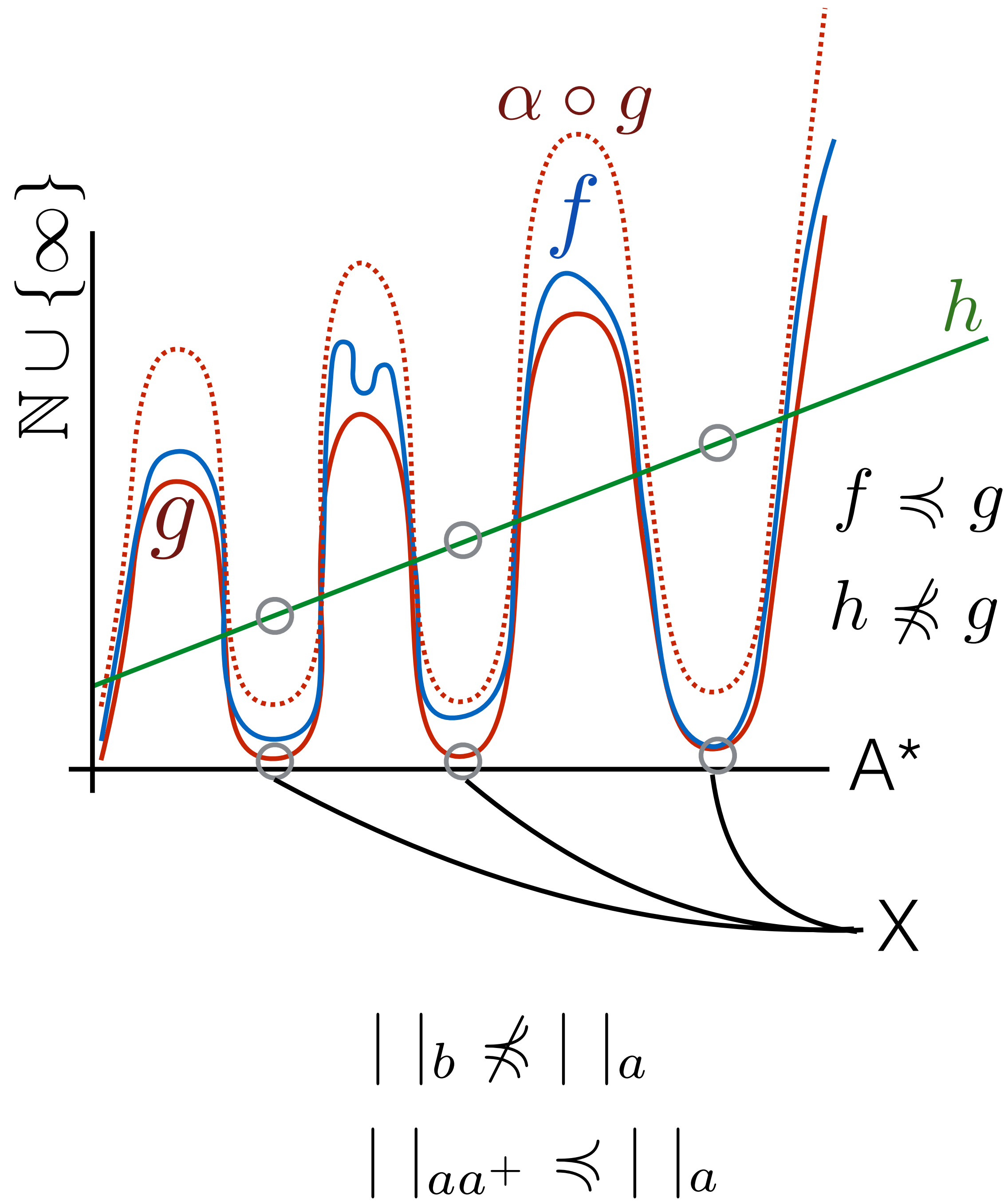
f≈g if f≼g and g≼f

cost function = ≈-class

Lemma: $f \preccurlyeq g$ if and only if

$$f \leqslant \alpha \circ g$$

for some non-decreasing map α from naturals to naturals extended with α(∞)=∞.



Let $F(f) = \{X \subseteq A^* : f|_X \text{is bounded}\}$

- $F(f)$ is a filter of countable basis.
- F is a bijection between cost functions and filters of countable basis.

Cost functions = filters of countable basis.

Example of equivalence

$$[\![a^{\leqslant n}]\!](u) = |u| \qquad \approx \qquad [\![(a^{\leqslant n})^{\leqslant n}]\!](u) = \lceil\sqrt{|u|}\rceil$$

Precise proof: $\left\lceil\sqrt{|u|}\right\rceil \leqslant |u|$ and $|u| \leq \left(\left\lceil\sqrt{|u|}\right\rceil\right)^2$

Informal proof:

$[\![a^{\leqslant n}]\!](u)$ is small if and only it has small length

$[\![(a^{\leqslant n})^{\leqslant n}]\!](u)$ is small if and only if it can be decomposed into a small number of words of small length

since a small sum of small numbers is small, and
all small numbers can be written as a small sum of small numbers,

$[\![a^{\leqslant n}]\!](u)$ is small if and only if $[\![(a^{\leqslant n})^{\leqslant n}]\!](u)$ is small

This kind of proofs avoid to be precise in the computations!
But, what does it mean mathematically to be small?
(small + 1 is small and 0 is small should imply that every integer is small)

This is similar to infinitesimals, which have been made formal using model theory, yielding non-standard analysis [Robinson66].

Non-standard analysis (internal set theory)

[Robinson 66] ...

[Nelson 77] Internal set theory (**IST**) a conservative extension of ZFC.

This is an extension of set theory (ZFC) in which one can prove exactly the same statements of ZFC.

All the language of ZFC is available.

There is a new unary symbol $St(x)$ to be read 'x is standard'.

All the proof arguments usable in ZFC are available.

Three new axioms (schema) are available:

- transfer: $\forall x \varphi$ iff $\forall^{St} x \varphi$
  ($\exists x \varphi$ iff $\exists^{St} x \varphi$)

  φ has to be an internal formula, i.e., no symbol $St(x)$.

- standardization: given a standard set X and a formula φ, there exists a standard set Y such that for all standard elements x ,

  $x \in X$ and $x \models \varphi$ iff $x \in Y$

- idealization: (simplified) there exists a non-standard integer.

More on IST

- **transfer**: $\forall x \varphi$ iff $\forall^{St} x \varphi$
($\exists x \varphi$ iff $\exists^{St} x \varphi$)

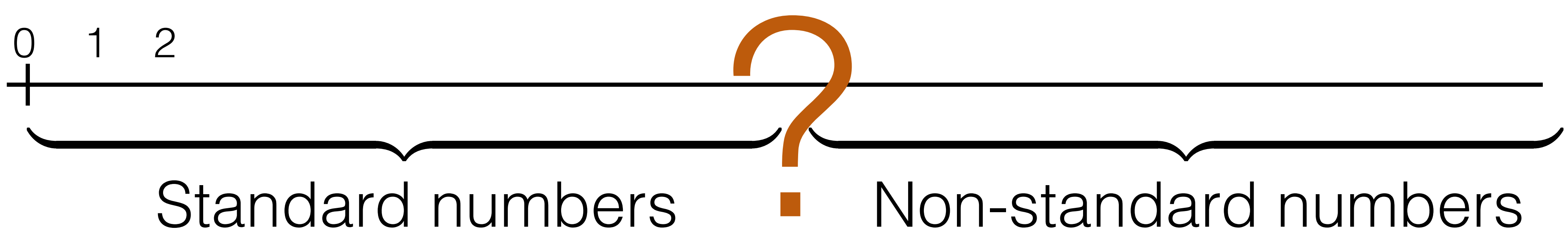
φ has be an internal formula,
i.e., no symbol $St(x)$.

All objects definable in standard mathematics are standard.

This is the case for $0, 1, \pi, \mathbb{N}, \mathbb{R}, A^* \ldots$

Things definable by internal formulae with standard parameters are standard.
→ Successor, predecessor, square of standard integers are integers.

Lemma: If m<n for n standard, then m is standard.



The set of standard integers does not exist!

Indeed,

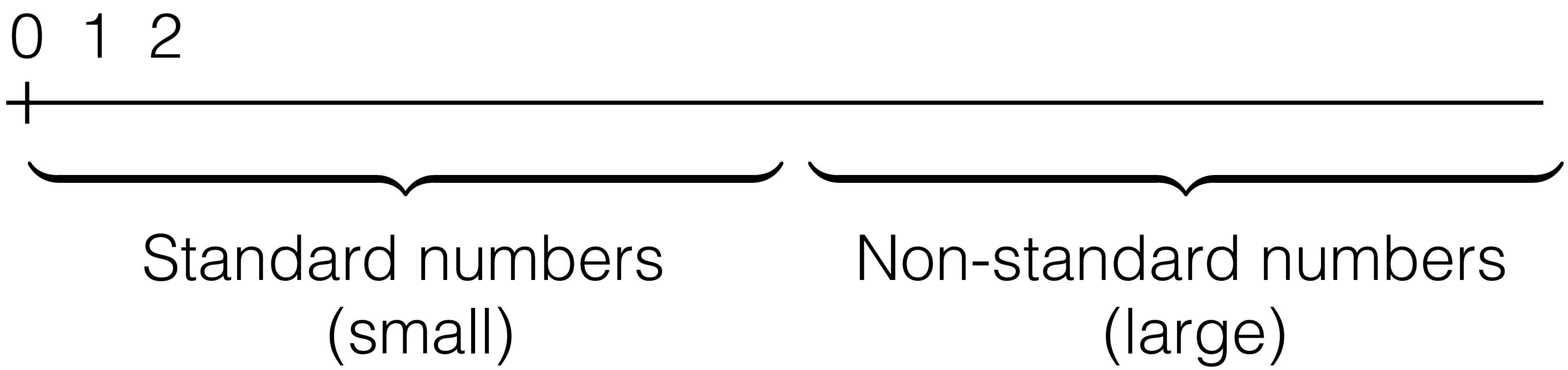
$$\{x \in A \mid \varphi\}$$

is only valid for internal formulae!

For this reason, it is not possible to prove that all integers are standard.

Regular cost functions in IST

[C.13]



Given a standard function $f : A^* \to \mathbb{N} \cup \{\infty\}$, define:

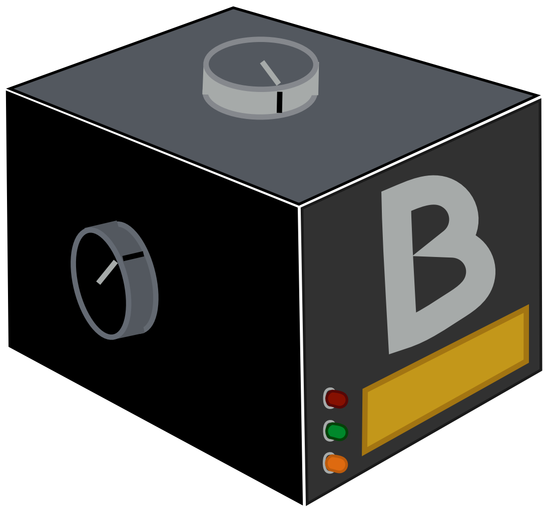
$$L_f = \{u \in A^* \mid \underbrace{\mathtt{St}(f(u)) \wedge f(u) \neq \infty}_{f(u) \text{ is small}}\}^{external}$$

> Then $f \preccurlyeq g$ if and only if $L_g \subseteq L_f$.

Hence, cost functions (functions up to boundedness equivalence) can be seen as (external) sets in IST.

-> the informal small/large arguments can now be made formal.

Conclusion



Achim Blumensath
Mikolaj Bojanczyck
Michael vanden Boom
Nathanaël Fijalkow
Kosaburo Hashiguchi
Stefan Göller
Daniel Kirsten
Denis Kuperberg
Hing Leung
Christof Löding
Amaldev Manuel
Pawel Parys
Imre Simon
Szymon Torunczyck

Regular cost functions offer a rich toolbox for solving boundedness problems in a way similar to the theory of regular languages.

This is tightly related to logic:

- in the formalisms (cost MSO, muB-calculus) and the results
- for the logic problems it helps to solve
  - boundedness of fixpoints
  - synthesis
  - separation of logical classes
- in its theory that makes use of non-standard analysis