Recursion Schemes and the WMSO+U Logic

Paweł Parys

University of Warsaw

STACS 2018

Higher-order recursion schemes – what is this?

Definition

Recursion schemes = simply-typed lambda-calculus + recursion

In other words:

- programs with recursion
- higher-order functions (i.e., functions taking other functions as parameters)
- every function/parameter has a fixed type
- no data values, only functions

Higher-order recursion schemes – example

fun f(x) {
    a(x);
    *if * then f(x);*
    b(x);
}
f(x)

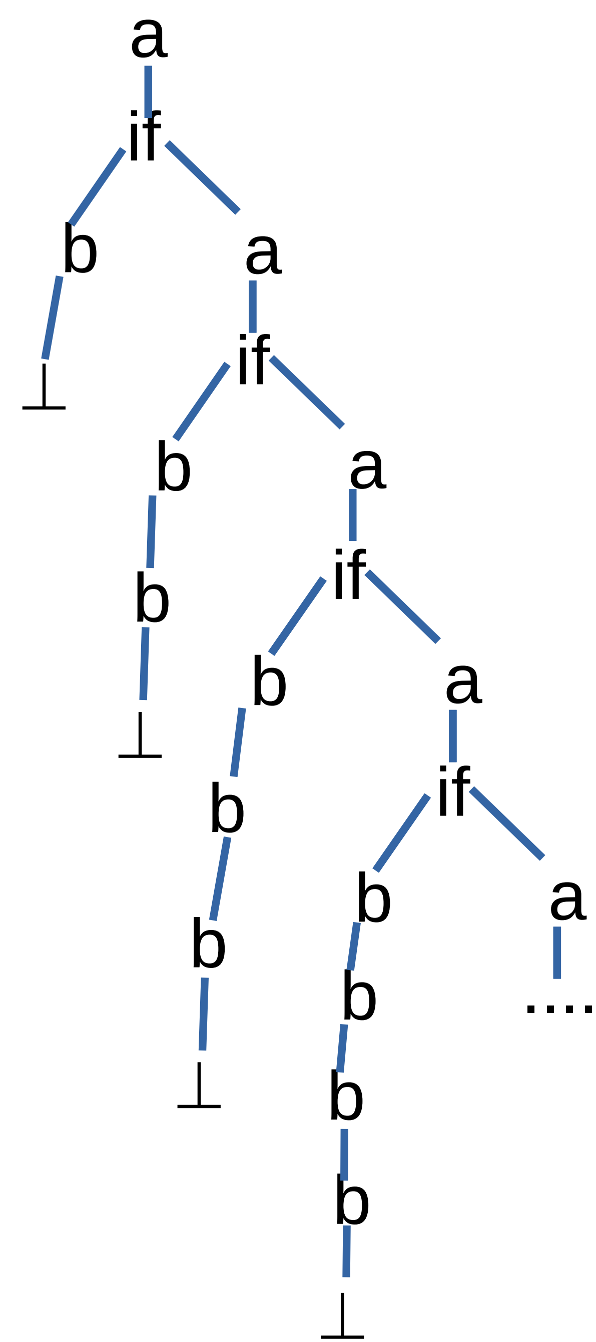
branching (we are not sure what will be chosen)

recursion

uniterpreted constants (unknown functions)

Higher-order recursion schemes – example

```
fun f(x) {
      a(x);
      if * then f(x);
      b(x);
}
f(x)
```

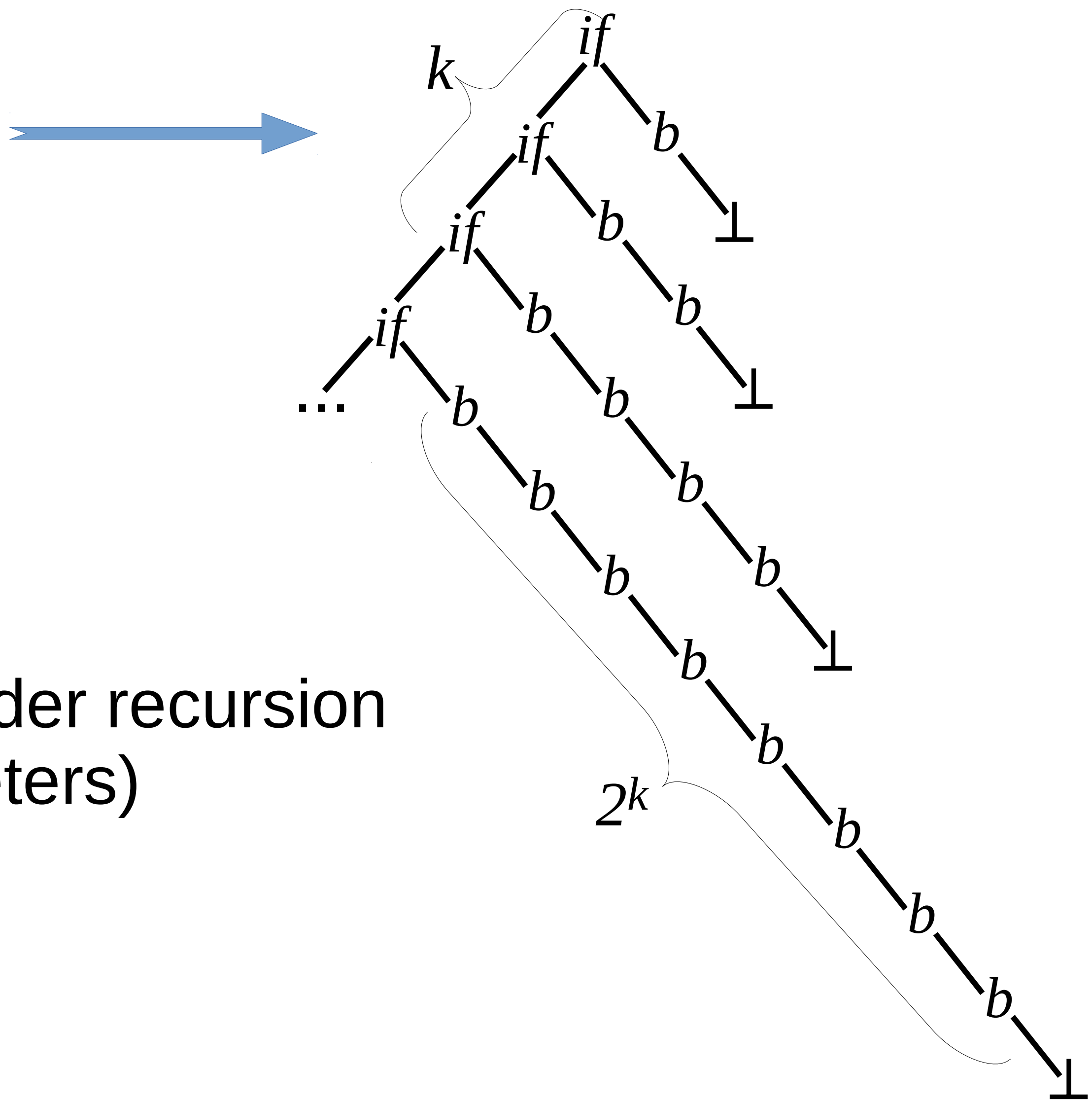


We are interested in trees representing the control flow of such programs.

Observation: these trees need not to be regular

Higher-order recursion schemes – example

```
fun A(f,x) {
   if * then A(D(f),x) else f(x);
}
fun D(f)(x) {
   f(x); f(x);
}
fun P(x) {
   b(x);
}
A(P,x)
```



This program uses higher-order recursion (passes functions as parameters)

Model-checking

Theorem [Ong 2006]
MSO model-checking on trees generated by recursion schemes is decidable.

Input: MSO formula ϕ, recursion scheme $\mathcal{G}$

Question: is ϕ true in the (infinite) tree generated by $\mathcal{G}$?

Model-checking

- a program in a functional programming language (e.g. OCAML)
- a property ψ

does the program satisfy ψ?

Model-checking

- a program in a functional programming language (e.g. OCAML)
- a property ψ

does the program satisfy ψ?

Approximation

ignore some details,

simulate some details using functions

- a recursion scheme $\mathcal{G}$
- a formula ϕ

is ϕ true in the tree generated by $\mathcal{G}$?

decidable

Model-checking

- a program in a functional programming language (e.g. OCAML)
- a property ψ

Approximation

ignore some details,

simulate some details using functions

- a recursion scheme $\mathcal{G}$
- a formula ϕ

does the program satisfy ψ?

- yes
- ?

is ϕ true in the tree generated by $\mathcal{G}$?

- yes
- no

decidable

There exist tools that take (short) programs in Ocaml and can verify some useful properties.

This work – can we go beyond MSO?

What about checking properties not expressible in MSO, e.g., talking about boundedness?

This work – can we go beyond MSO?

What about checking properties not expressible in MSO, e.g., talking about boundedness?

We consider the WMSO+U logic.

“+U” = we add a new quantifier „U” [Bojańczyk, 2004]

$$UX.\phi(X)$$

$\phi(X)$ holds for finite sets of arbitrarily large size

$$\forall n \in \mathbb{N}\ \exists X\ (\ n < |X| < \infty\ \wedge\ \phi(X)\ )$$

This work – can we go beyond MSO?

What about checking properties not expressible in MSO, e.g., talking about boundedness?

We consider the WMSO+U logic.

“+U” = we add a new quantifier „U” [Bojańczyk, 2004]

$$UX.\phi(X)$$

$\phi(X)$ holds for finite sets of arbitrarily large size

$$\forall n \in \mathbb{N}\ \exists X\ (\ n<|X|<\infty\ \wedge\ \phi(X)\ )$$

“W” = weak – we can quantify only over finite sets
($\exists X$ / $\forall X$ means: exists a finite set X / for all finite sets X)

Decision problems for MSO+U

Satisfiability (the problem usually considered for MSO+U):
input: formula ϕ, question: is ϕ true in some tree?

- undecidable for MSO+U, even for words [Bojańczyk, P., Toruńczyk 2016]
  some fragments of MSO+U decidable for words [Bojańczyk, Colcombet 2006]
- decidable for WMSO+U [Bojańczyk, Toruńczyk 2012]
  also extended by the quantifier „exists path” [Bojańczyk 2014]

Decision problems for MSO+U

Satisfiability (the problem usually considered for MSO+U):
input: formula ϕ, question: is ϕ true in some tree?

- undecidable for MSO+U, even for words [Bojańczyk, P., Toruńczyk 2016]
  some fragments of MSO+U decidable for words [Bojańczyk, Colcombet 2006]
- decidable for WMSO+U [Bojańczyk, Toruńczyk 2012]
  also extended by the quantifier „exists path" [Bojańczyk 2014]

HORS model-checking
input: formula ϕ, HORS $\mathcal{G}$,
question: is ϕ true in the tree generated by $\mathcal{G}$

- decidable for $\phi \in$MSO [Ong 2006]
- undecidable for $\phi \in$MSO+U (generalizes satifiability)
- **Contribution: decidable for $\phi \in$WMSO+U**

About the proof

Theorem – the following problem is decidable:

input: formula ϕ, HORS $\mathcal{G}$,

question: is ϕ true in the tree generated by $\mathcal{G}$?

Key ingredients:

- decidability of the "diagonal problem" for HORSes:

  input: HORS $\mathcal{G}$, letter a

  question: are there paths with arbitrarily many letters a in the tree generated by $\mathcal{G}$?

  [Hague, Kochems, Ong 2016, Clemente, P., Salvati, Walukiewicz 2016]

  Remark 1: this property is not regular

  Remark 2: this is a „universal" property that can be expressed by a single „U" quantifier

About the proof

Theorem – the following problem is decidable:
input: formula ϕ, HORS $\mathcal{G}$,
question: is ϕ true in the tree generated by $\mathcal{G}$?

Key ingredients:

- decidability of the "diagonal problem" for HORSes:
  input: HORS $\mathcal{G}$, letter a
  question: are there paths with arbitrarily many letters a in the tree generated by $\mathcal{G}$?
  [Hague, Kochems, Ong 2016, Clemente, P., Salvati, Walukiewicz 2016]
  Remark 1: this property is not regular
  Remark 2: this is a „universal" property that can be expressed by a single „U" quantifier
- „reflection" for the diagonal problem: [P. 2017]
  input: HORS $\mathcal{G}$, letter a
  output: HORS $\mathcal{H}$, generating the same tree as $\mathcal{G}$, but with additional labels – in each node it is written whether there are paths starting in this node with arbitrarily many letters a

(step 3)

About the proof

Theorem – the following problem is decidable:

input: formula ϕ, HORS $\mathcal{G}$,

question: is ϕ true in the tree generated by $\mathcal{G}$?

Key ingredients:

- „reflection" for the diagonal problem: [P. 2017] (step 3)
  input: HORS $\mathcal{G}$, letter a
  output: HORS $\mathcal{H}$, generating the same tree as $\mathcal{G}$, but with additional labels – in each node it is written whether there are paths starting in this node with arbitrarily many letters a
- „reflection" for (W)MSO: [Broadbent, Carayol, Ong, Serre 2010] (step 4)
  input: HORS $\mathcal{G}$, formula $\psi(x)\in$WMSO
  output: HORS $\mathcal{H}$, generating the same tree as $\mathcal{G}$, but with additional labels – in each node it is written whether ψ holds in this node

About the proof

Theorem – the following problem is decidable:

input: formula ϕ, HORS $\mathcal{G}$,

question: is ϕ true in the tree generated by $\mathcal{G}$?

Key ingredients:

- translation: formulas ⇒ automata (step 1)
  We define a new model of automata: nested U-prefix automata.

About the proof

Theorem – the following problem is decidable:

input: formula ϕ, HORS $\mathcal{G}$,

question: is ϕ true in the tree generated by $\mathcal{G}$?

Key ingredients:

- translation: formulas ⇒ automata (step 1)

  We define a new model of automata: nested U-prefix automata.
- This is a sequence of automata – $A_1, A_2, \ldots, A_k$

  Every A_i is a nondeterministic automaton, where
  - there is special state $\perp$ meaning "end of run" – only a finite prefix of a run can use other states, from some moment there are only $\perp$ states
  - some states are marked as "important"

About the proof

Theorem – the following problem is decidable:

input: formula ϕ, HORS $\mathcal{G}$,

question: is ϕ true in the tree generated by $\mathcal{G}$?

Key ingredients:

- translation: formulas ⇒ automata (step 1)
  We define a new model of automata: nested U-prefix automata.
- This is a sequence of automata – $A_1, A_2, \ldots, A_k$
  Every A_i is a nondeterministic automaton, where
  - ➔ there is special state $\perp$ meaning “end of run” – only a finite prefix of a run can use other states, from some moment there are only $\perp$ states
  - ➔ some states are marked as “important”
- Effect of running A_i on a tree t: we mark every node v such that in the subtree of t starting in v there are runs of A_i with arbitrarily many important states (alphabet changes from Σ to $\Sigma\times\{0,1\}$)

About the proof

Theorem – the following problem is decidable:

input: formula ϕ, HORS $\mathcal{G}$,

question: is ϕ true in the tree generated by $\mathcal{G}$?

Key ingredients:

- translation: formulas ⇒ automata (step 1)
  We define a new model of automata: nested U-prefix automata.
- This is a sequence of automata – $A_1, A_2, ..., A_k$
  Every A_i is a nondeterministic automaton, where
  - there is special state $\perp$ meaning "end of run" – only a finite prefix of a run can use other states, from some moment there are only $\perp$ states
  - some states are marked as "important"
- Effect of running A_i on a tree t: we mark every node v such that in the subtree of t starting in v there are runs of A_i with arbitrarily many important states (alphabet changes from Σ to $\Sigma\times\{0,1\}$)
- The translation (formula ⇒ nested automaton) is not difficult
  Every quantifier corresponds to one A_i

About the proof

Theorem – the following problem is decidable:
input: formula ϕ, HORS $\mathcal{G}$,
question: is ϕ true in the tree generated by $\mathcal{G}$?

Key ingredients:

Step 1: formula ⇒ nested automaton $A_1, A_2, ..., A_k$

For every A_i and HORS $\mathcal{G}_i$ generating a tree t_i we want to create a HORS $\mathcal{G}_{i+1}$ generating $t_{i+1}=A_i(t_i)$ (i.e., the effect of running A_i on t_i):

- Step 2: Create $\mathcal{H}_i$ that generates t_i enriched with all possible runs of A_i (on additional new branches below every node of A_i)
  - This tree is an effect of running a finite-state transducer on t_i
  - HORSes can be composed with transducers
- Step 3: Use diagonal reflection to see whether there are runs having arbitrarily many "important" states
- Step 4: Move the new information to the original tree, and remove the additional branches (MSO reflection is useful here)

About the proof

Theorem – the following problem is decidable:

input: formula ϕ, HORS $\mathcal{G}$,

question: is ϕ true in the tree generated by $\mathcal{G}$?

Conclusion of the proof:

- The proof consists of a few (clearly separated) steps
- The technical difficulty is hidden in the "diagonal reflection" theorem

Future work

The diagonal problem for HORS is decidable in a more general version:

input: HORS $\mathcal{G}$, letters $a_1,...,a_k$

question: are there paths with arbitrarily many appearances of every letter $a_1,...,a_k$ in the tree generated by $\mathcal{G}$?

[Hague, Kochems, Ong 2016, Clemente, P., Salvati, Walukiewicz 2016]

(and we have the reflection property for this problem)

Future work

The diagonal problem for HORS is decidable in a more general version:

input: HORS $\mathcal{G}$, letters $a_1,\dots,a_k$

question: are there paths with arbitrarily many appearances of every letter $a_1,\dots,a_k$ in the tree generated by $\mathcal{G}$?

[Hague, Kochems, Ong 2016, Clemente, P., Salvati, Walukiewicz 2016]

(and we have the reflection property for this problem)

1) In our proof we use only the one-letter case.
2) It seems that the multi-letter case cannot be expressed in WMSO+U (U says that one quality is unbounded, not that many qualities are unbounded simultaneously)

Future work

The diagonal problem for HORS is decidable in a more general version:
input: HORS $\mathcal{G}$, letters $a_1,\ldots,a_k$
question: are there paths with arbitrarily many appearances of every letter $a_1,\ldots,a_k$ in the tree generated by $\mathcal{G}$?
[Hague, Kochems, Ong 2016, Clemente, P., Salvati, Walukiewicz 2016]
(and we have the reflection property for this problem)

1) In our proof we use only the one-letter case.
2) It seems that the multi-letter case cannot be expressed in WMSO+U (U says that one quality is unbounded, not that many qualities are unbounded simultaneously)

Question: Design a more general logic, capable to express the multi-letter diagonal problem (and prove its decidability for trees generated by HORSes, via a reduction to this version of the diagonal problem)

Thank you!