Intersection Types and Counting

Paweł Parys

University of Warsaw

ITRS 2016

Our setting

We consider infinitary, simply typed λ-calculus.

Our setting

We consider infinitary, simply typed λ-calculus.

Simple types (sorts): o, $o\to(o\to o)$, $(o\to o)\to o$, $(o\to o)\to(((o\to o)\to o)\to o)$

Our setting

We consider infinitary, simply typed λ-calculus.

Simple types (sorts): $\underset{0}{o}$, $\underbrace{o\to o\to o}_{1}$, $\underbrace{(o\to o)\to o}_{2}$, $\underbrace{(o\to o)\to((o\to o)\to o)\to o}_{3}$

Order: $\mathrm{ord}(o)=0$, $\mathrm{ord}(\alpha\to\beta)=\max(\mathrm{ord}(\alpha)+1, \mathrm{ord}(\beta))$

Our setting

We consider infinitary, simply typed λ-calculus.

Simple types (sorts): $\underbrace{o}_{0}$, $\underbrace{o\to o\to o}_{1}$, $\underbrace{(o\to o)\to o}_{2}$, $\underbrace{(o\to o)\to((o\to o)\to o)\to o}_{3}$

Order: $ord(o)=0$, $ord(\alpha\to\beta)=\max(ord(\alpha)+1, ord(\beta))$

λ-terms:

- variables: x^α, y^β, ...
- constants: a^α, b^β, … – only for sorts of order ≤ 1
- λ-abstraction: $(\lambda x^\alpha.K^\beta)^{\alpha\to\beta}$
- application: $(K^{\alpha\to\beta} L^\alpha)^\beta$

  \+ coinduction

Every term has a particular sort.
We allow infinite terms, but the set of types of subterms should be finite.

Our setting – λY-calculus

λY-term is a finite representation of an infinite λ-term:

- In a λ-term we may use a binder "Y"
- Meaning:

  $(Yx^{\alpha}.M^{\alpha})^{\alpha}$ - this is the unique (infinite) λ-term such that

  $Yx.M = M[Yx.M/x]$

Example:

the λY-term: Yx.((λy.ay) x)

represents the λ-term: ((λy.ay) ((λy.ay) ((λy.ay) ((λy.ay) ...))))

Our setting – Böhm trees

- Every finite λ-term K reduces to a term in β-normal form.

Our setting – Böhm trees

- Every finite λ-term K reduces to a term in β-normal form.
- Every (infinite) λ-term K reduces to term in head-β-normal form, i.e.:

$\lambda x_1.\cdots.\lambda x_n.y\, M_1 \ldots M_k$ or $\lambda x_1.\cdots.\lambda x_n.a\, M_1 \ldots M_k$

Our setting – Böhm trees

- Every finite λ-term K reduces to a term in β-normal form.
- Every (infinite) λ-term K reduces to term in head-β-normal form, i.e.:

  $\lambda x_1.\cdots.\lambda x_n.y\, M_1 \ldots M_k$ or $\lambda x_1.\cdots.\lambda x_n.a\, M_1 \ldots M_k$
- We may reduce each $M_1, \ldots, M_k$ to head-β-normal form, etc.
- The limit is called the *Böhm tree* of K.

Our setting – Böhm trees

- Every finite λ-term K reduces to a term in β-normal form.
- Every (infinite) λ-term K reduces to term in head-β-normal form, i.e.:

  $\lambda x_1.\cdots.\lambda x_n.y\, M_1 \ldots M_k$ or $\lambda x_1.\cdots.\lambda x_n.a\, M_1 \ldots M_k$
- We may reduce each $M_1, \ldots, M_k$ to head-β-normal form, etc.
- The limit is called the *Böhm tree* of K.

Suppose that:

→ K is of sort o

→ K has no free variables

→ we only use constants of order ≤ 1.

Then the Böhm tree is a tree built out of constants.

Our setting – Böhm trees

- Every finite λ-term K reduces to a term in β-normal form.
- Every (infinite) λ-term K reduces to term in head-β-normal form, i.e.:

  $\lambda x_1.\cdots.\lambda x_n.y\, M_1 \ldots M_k$ or $\lambda x_1.\cdots.\lambda x_n.a\, M_1 \ldots M_k$
- We may reduce each $M_1, \ldots, M_k$ to head-β-normal form, etc.
- The limit is called the *Böhm tree* of K.

Suppose that:
- K is of sort o
- K has no free variables
- we only use constants of order ≤ 1.

Then the Böhm tree is a tree built out of constants.

Example:

$$Yx.((\lambda y.ay)\ x) = ((\lambda y.ay)\ ((\lambda y.ay)\ ((\lambda y.ay)\ ((\lambda y.ay)\ \ldots))))$$

$$\downarrow$$

$$(a\ ((\lambda y.ay)\ ((\lambda y.ay)\ ((\lambda y.ay)\ \ldots))))$$

Our setting – Böhm trees

- Every finite λ-term K reduces to a term in β-normal form.
- Every (infinite) λ-term K reduces to term in head-β-normal form, i.e.:

  $\lambda x_1.\cdots.\lambda x_n.y\, M_1 \ldots M_k$ or $\lambda x_1.\cdots.\lambda x_n.a\, M_1 \ldots M_k$
- We may reduce each $M_1, \ldots, M_k$ to head-β-normal form, etc.
- The limit is called the *Böhm tree* of K.

Suppose that:
- K is of sort o
- K has no free variables
- we only use constants of order ≤ 1.

Then the Böhm tree is a tree built out of constants.

Example:

$$Yx.((\lambda y.ay)\ x) = ((\lambda y.ay)\ ((\lambda y.ay)\ ((\lambda y.ay)\ ((\lambda y.ay)\ \ldots))))$$

$$\downarrow$$

$$(a\ (a\ ((\lambda y.ay)\ ((\lambda y.ay)\ \ldots))))$$

Our setting – Böhm trees

- Every finite λ-term K reduces to a term in β-normal form.
- Every (infinite) λ-term K reduces to term in head-β-normal form, i.e.:

  $\lambda x_1.\cdots.\lambda x_n.y\, M_1 \ldots M_k$ or $\lambda x_1.\cdots.\lambda x_n.a\, M_1 \ldots M_k$
- We may reduce each $M_1, \ldots, M_k$ to head-β-normal form, etc.
- The limit is called the *Böhm tree* of K.

Suppose that:
- K is of sort o
- K has no free variables
- we only use constants of order ≤ 1.

Then the Böhm tree is a tree built out of constants.

Example:

$$Yx.((\lambda y.ay)\ x) = ((\lambda y.ay)\ ((\lambda y.ay)\ ((\lambda y.ay)\ ((\lambda y.ay)\ \ldots))))$$

$$\downarrow$$

$$(a\ (a\ (a\ ((\lambda y.ay)\ \ldots))))$$

Our setting – Böhm trees

- Every finite λ-term K reduces to a term in β-normal form.
- Every (infinite) λ-term K reduces to term in head-β-normal form, i.e.:

  $\lambda x_1.\cdots.\lambda x_n.y\, M_1 \ldots M_k$ or $\lambda x_1.\cdots.\lambda x_n.a\, M_1 \ldots M_k$
- We may reduce each $M_1, \ldots, M_k$ to head-β-normal form, etc.
- The limit is called the *Böhm tree* of K.

Suppose that:

- K is of sort o
- K has no free variables
- we only use constants of order ≤ 1.

Then the Böhm tree is a tree built out of constants.

Example:

$Yx.((\lambda y.ay)\ x) = ((\lambda y.ay)\ ((\lambda y.ay)\ ((\lambda y.ay)\ ((\lambda y.ay)\ \ldots))))$

$(a\ (a\ (a\ (a\ \ldots))))$

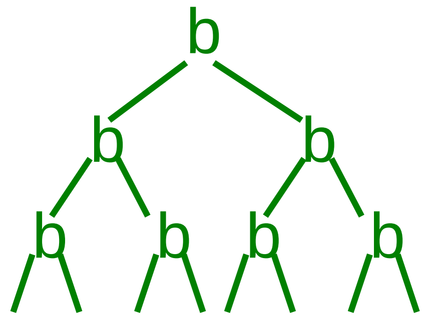
a
|
a
|
a
|
a
|
...

Our setting – Böhm trees

Example:

$Yx.((\lambda y.byy)\ x) = ((\lambda y.byy)\ ((\lambda y.byy)\ ((\lambda y.byy)\ ((\lambda y.byy)\ \ldots))))$

$(b\,(b\ (b\ \ldots)\ (b\ \ldots))\,(b\ (b\ \ldots)\ (b\ \ldots)))$

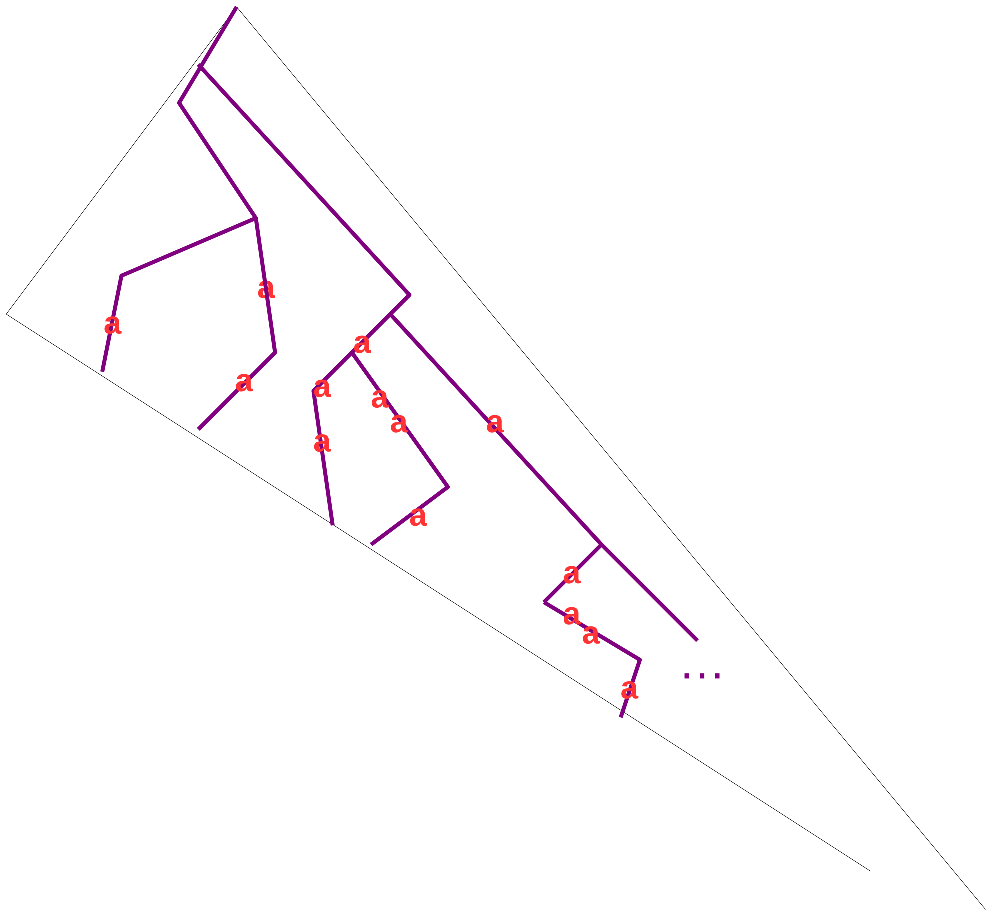


Equivalent formalism:
trees generated by Higher Order Recursion Schemes (HORSes)

Considered problem

Input: closed λY-term K of sort o (i.e. infinite λ-term represented in a finite way)

Question: In the Böhm tree of K, are there finite paths with arbitrarily many symbols "*a*"?

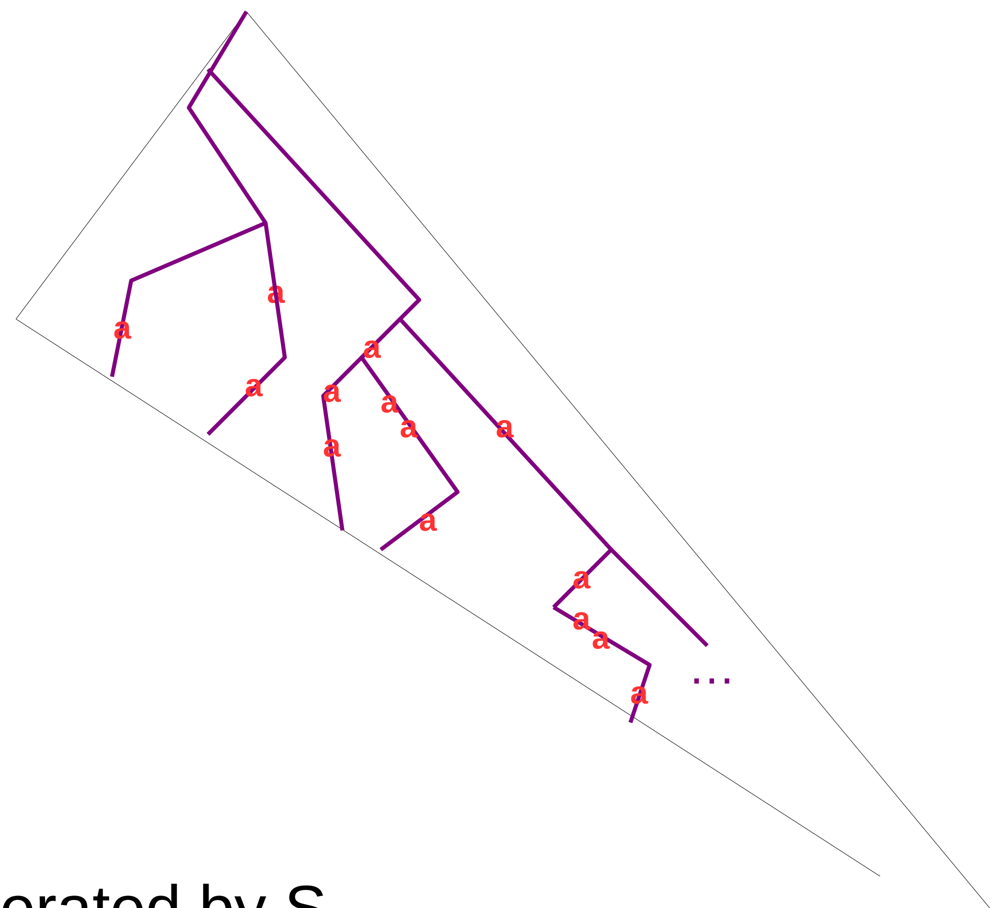


Considered problem

=deterministic HORS

Input: closed λY-term K of sort o (i.e. infinite λ-term represented in a finite way)

Question: In the Böhm tree of K, are there finite paths with arbitrarily many symbols “*a*”?



Equivalent problem:

Input: nondeterministic HORS *S*

Question: is L(S) finite?

L(S) = the set of finite trees generated by S

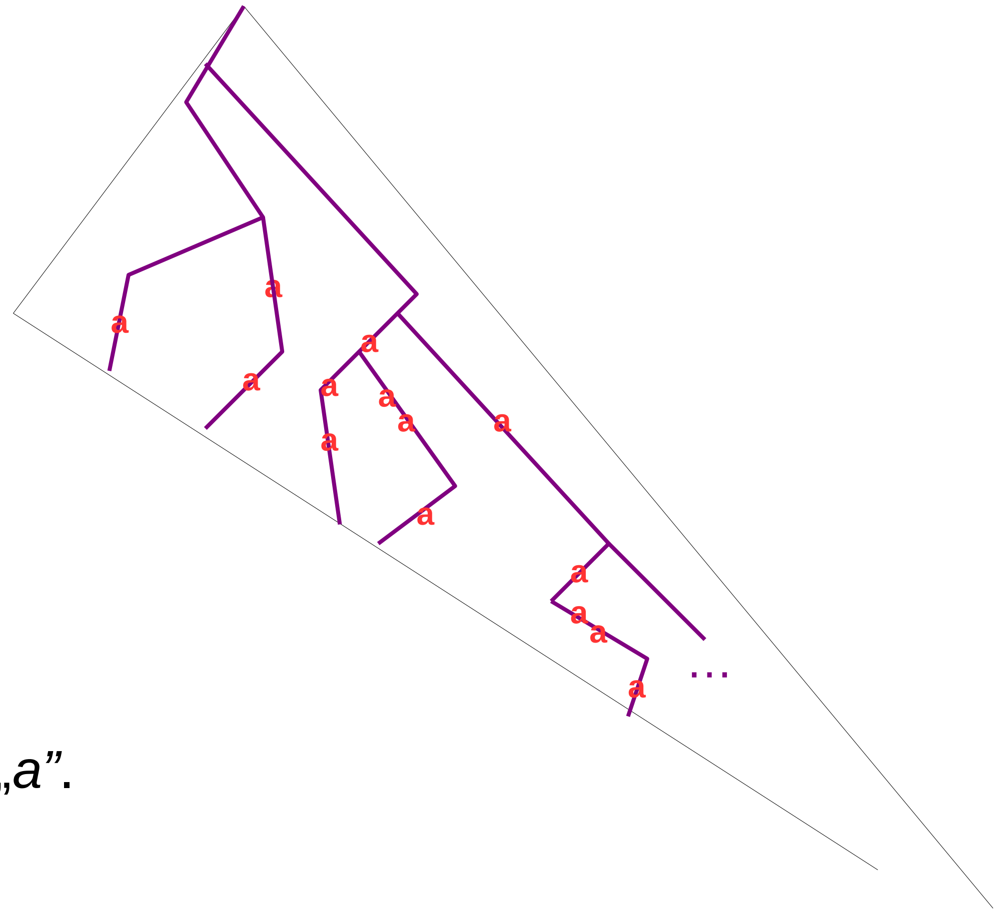
History

Thm [Ong 2006].
The following problem is decidable (MSO model-checking):
Input: closed λY-term K of sort o, regular property ϕ
Question: Is ϕ true in the Böhm tree of K?

Considered problem

Input: closed λY-term K of sort o

Question: In the Böhm tree of K, are there finite paths with arbitrarily many symbols "*a*"?



Notice:

There may be no path with infinitely many „*a*".

Our property **is not regular!!!**

History

Thm [Ong 2006].
The MSO model-checking problem for HORS is decidable.

Our problem is a special case of the ***diagonal problem***:
Input: closed λY-term K of sort o, set Σ of symbols
Question: In the Böhm tree of K, are there finite paths with arbitrarily many appearances of every symbol from Σ?
(i.e. for every N there exists a path P such that every symbol from Σ appears on P at least N times)

History

Thm [Ong 2006].
The MSO model-checking problem for HORS is decidable.

Our problem is a special case of the ***diagonal problem***:
Input: closed λY-term K of sort o, set Σ of symbols
Question: In the Böhm tree of K, are there finite paths with arbitrarily many appearances of every symbol from Σ?
(i.e. for every N there exists a path P such that every symbol from Σ appears on P at least N times)

Thm [Hague, Kochems, Ong 2016], [Clemente, P., Salvati, Walukiewicz 2016].
The diagonal problem is decidable.

Proof: perform a sequence of transformations of the input HORS, reducing its order.

We present a new solution, using intersection types.

History

Thm [Ong 2006].
The MSO model-checking problem for HORS is decidable.

Thm [Hague, Kochems, Ong 2016], [Clemente, P., Salvati, Walukiewicz 2016].
The diagonal problem is decidable.

[P. 2014]
An intersection type system for (finite) λ-terms s.t.
the "size" of the (unique) derivation for K ≈ the number of symbols "a" in the normal form of K

number of flags

History

Thm [Ong 2006].
The MSO model-checking problem for HORS is decidable.

Thm [Hague, Kochems, Ong 2016], [Clemente, P., Salvati, Walukiewicz 2016].
The diagonal problem is decidable.

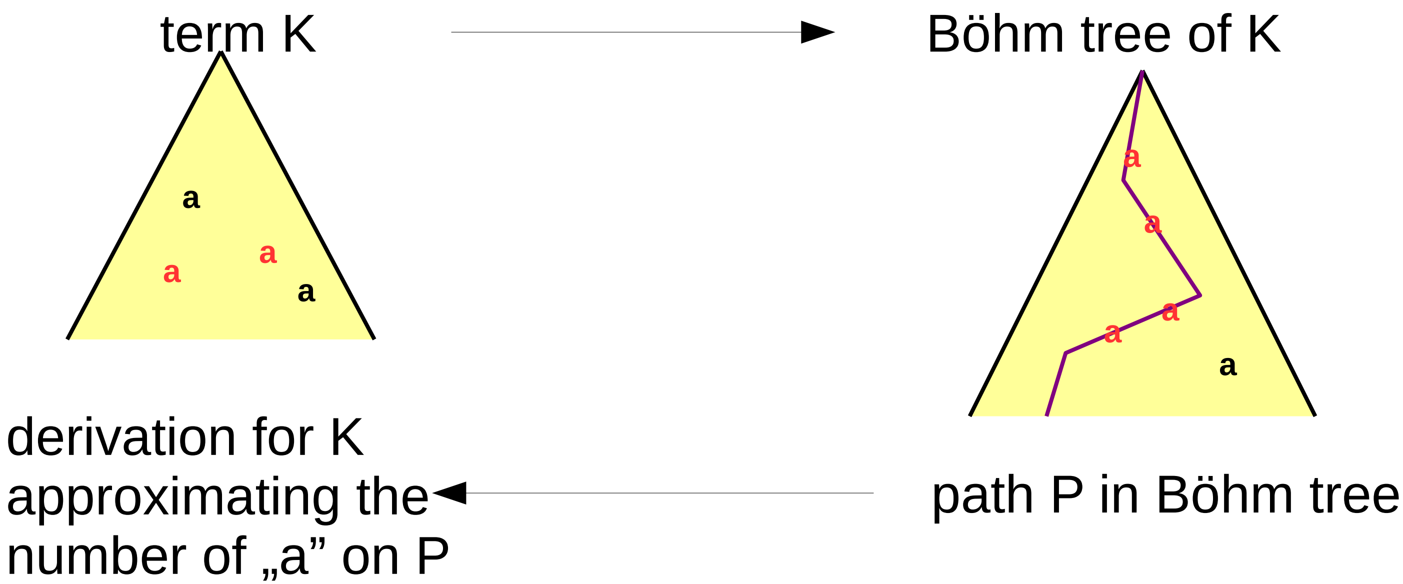
[P. 2014]
An intersection type system for (finite) λ-terms s.t.
the "size" of the (unique) derivation for K ≈ the number of symbols "a" in the normal form of K

number of flags

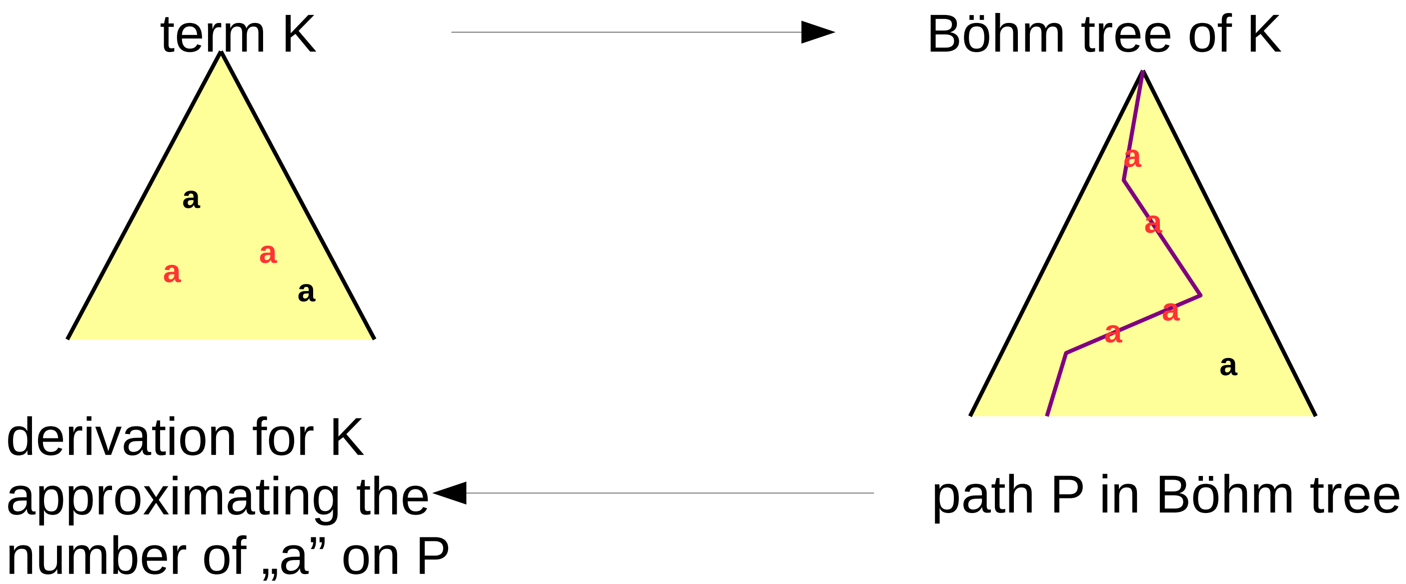
Here we need an additional existential quantifier in the front:

there exist "big" derivations for K ⟺ in the Böhm tree of K there exist paths with arbitrarily many „a"

Intersection types - idea



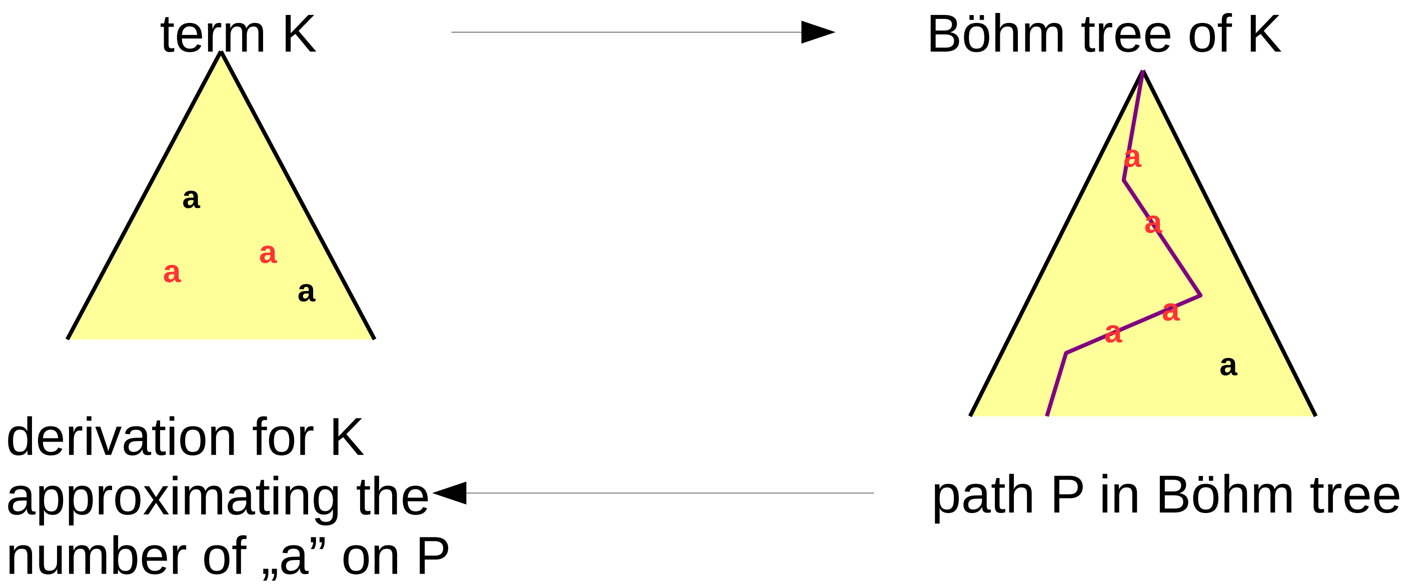
Intersection types - idea



Standard use of intersection types:

- which „a” of K will appear in the Böhm tree

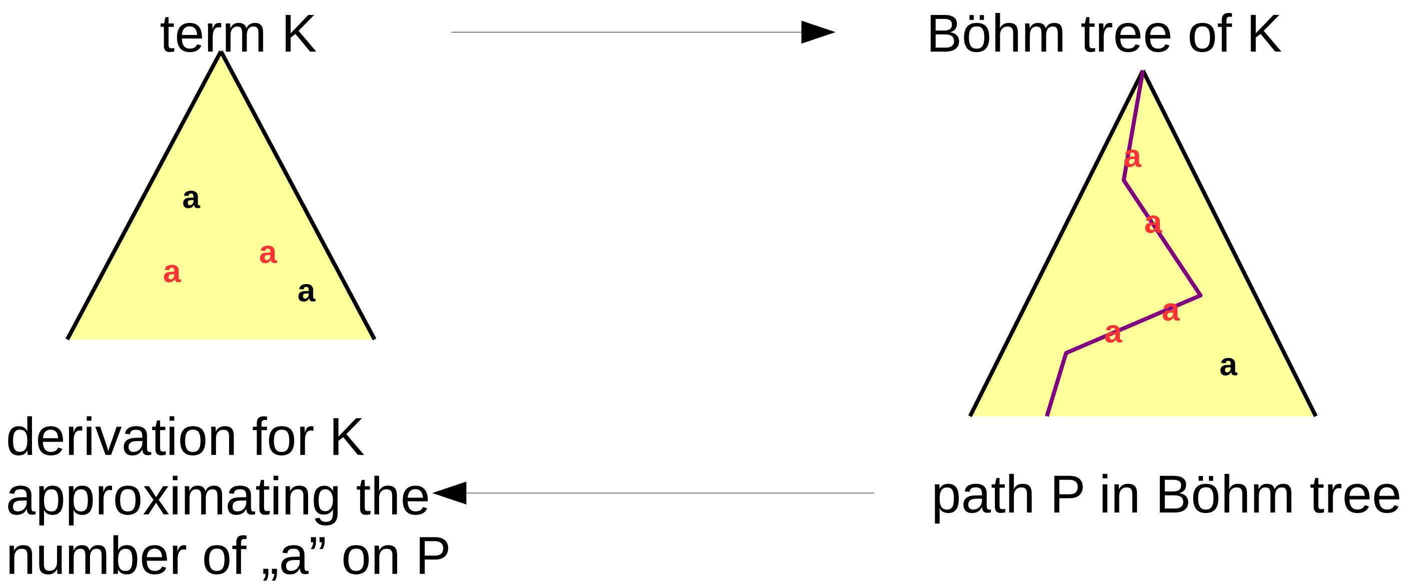
Intersection types - idea



Almost standard use of intersection types:

- which „a” of K will appear on P in the Böhm tree

Intersection types - idea



Almost standard use of intersection types:
- which „a” of K will appear on P in the Böhm tree

Difficulty:
- single „a” of K may result in many „a” on P

$(\lambda\, y.\ y\ (y\ b^{o})).a^{o \to o}$

Idea of solution:
- detect (and count) places where variable containing „a” is duplicated

Intersection types

Solution: type derivations are labeled by flags and markers.

Intersection types refining sort o:

$$\mathcal{T}^o = \{\ (F,M,o)\ \}$$

markers used in the derivation

flags used in the derivation

(for each order m we have flags of order m,
and a marker of order m)

Intersection types

Solution: type derivations are labeled by flags and markers.

Intersection types refining sort $\alpha = \alpha_1 \to \ldots \to \alpha_k \to o$:

$$\mathcal{T}^{\alpha} = \{\, T_1 \to \ldots \to T_k \to (F, M, o) \,\}$$

markers used in the derivation

flags used in the derivation

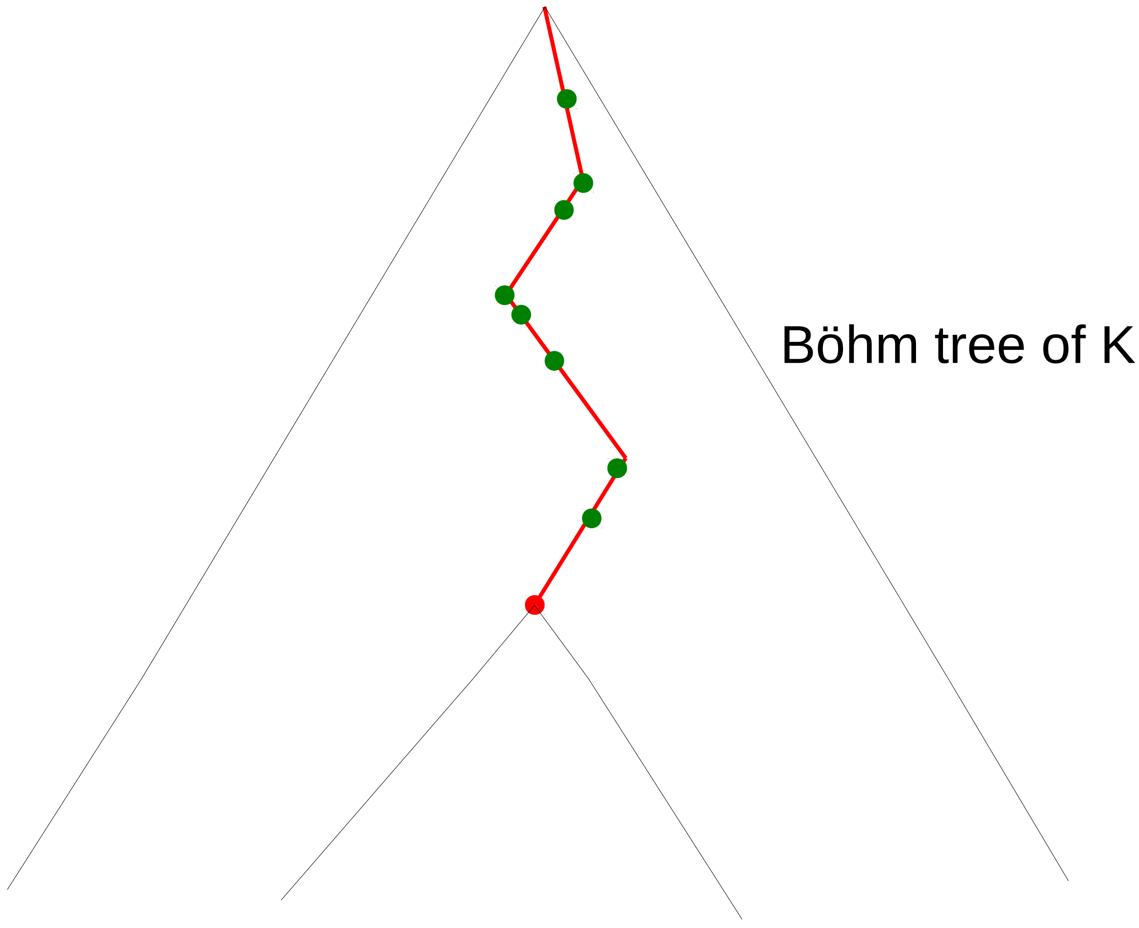
sets of types refining $\alpha_1, \ldots, \alpha_k$

(for each order m we have flags of order m,
and a marker of order m)

Only finite derivations!

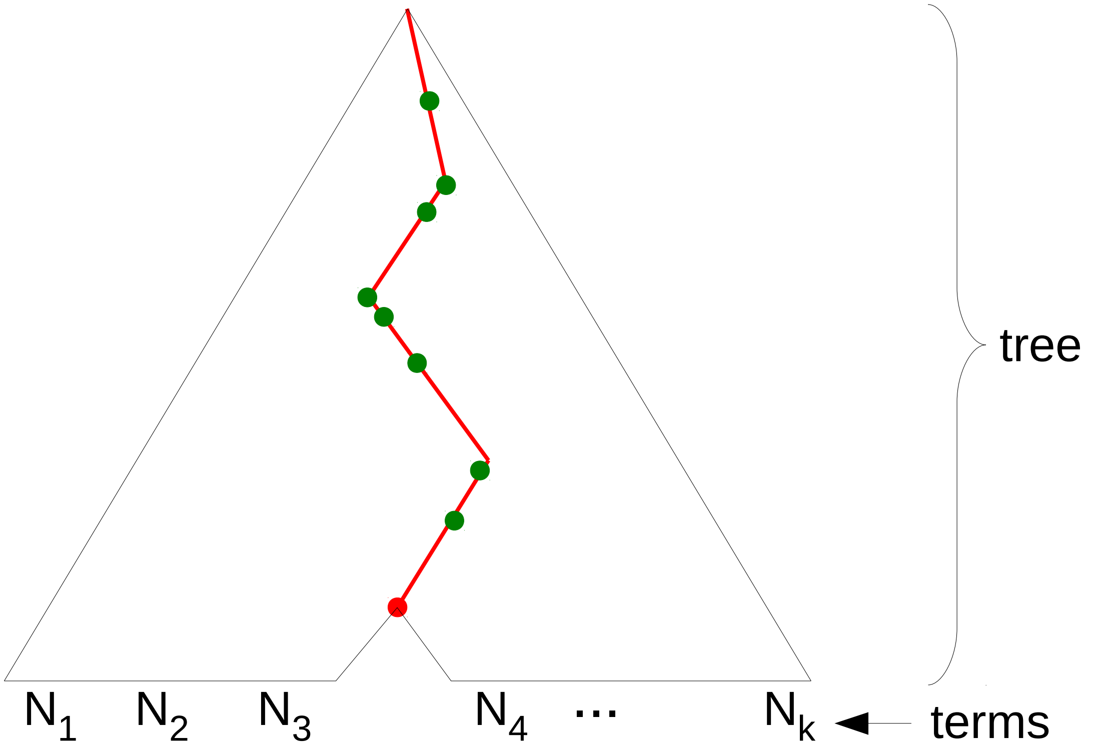
Flags & markers

one marker of order 0 (= end of path)
flags of order 1 (= „a” on the path)



Flags & markers

one marker of order 0 (= end of path)
flags of order 1 (= „a” on the path)

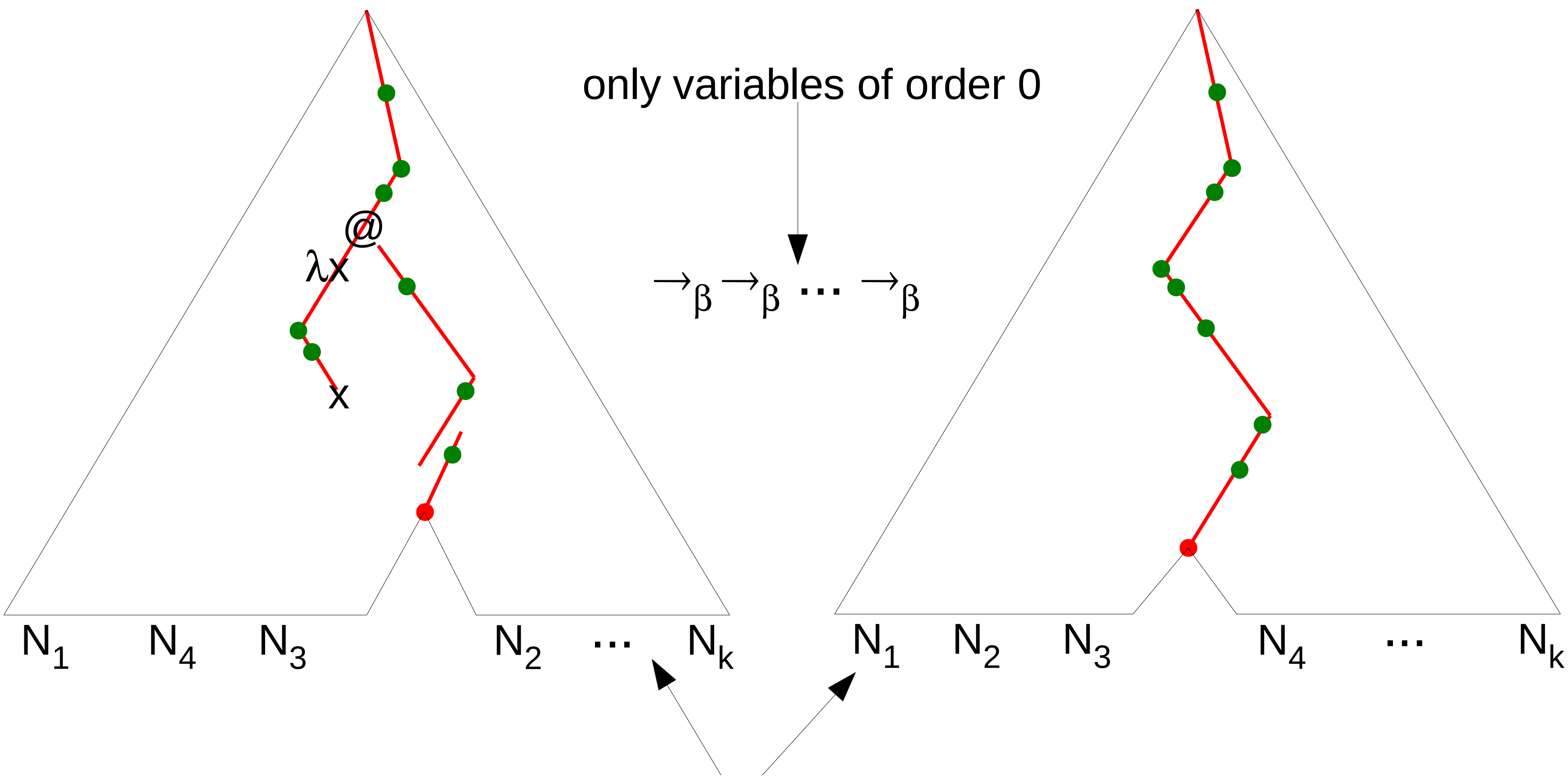


(after many β-reductions)

Flags & markers

one marker of order 0
flags of order 1

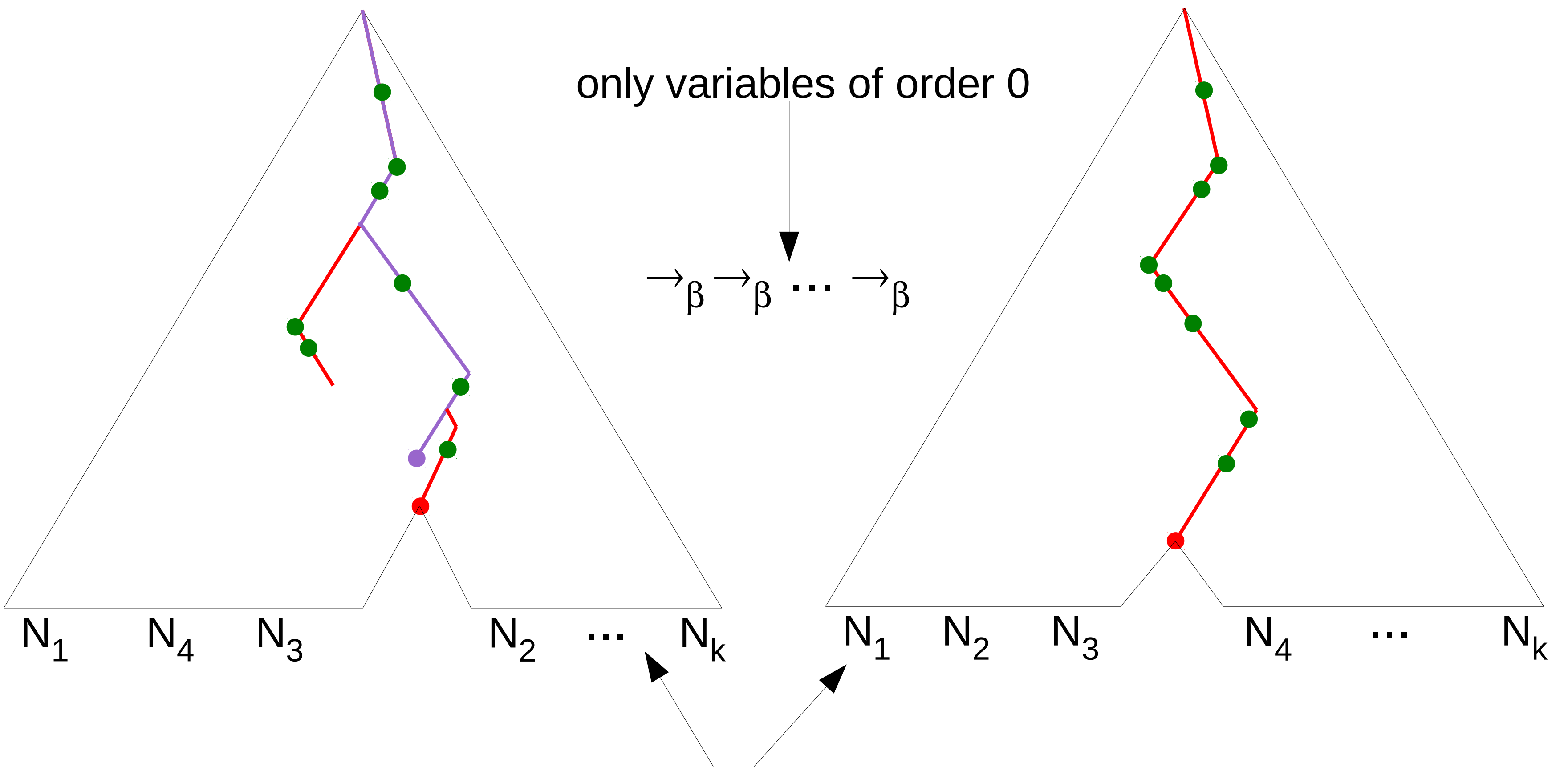
the type system ensures that a variable with marker is used exactly once!



number of order-1 flags unchanged!

Flags & markers

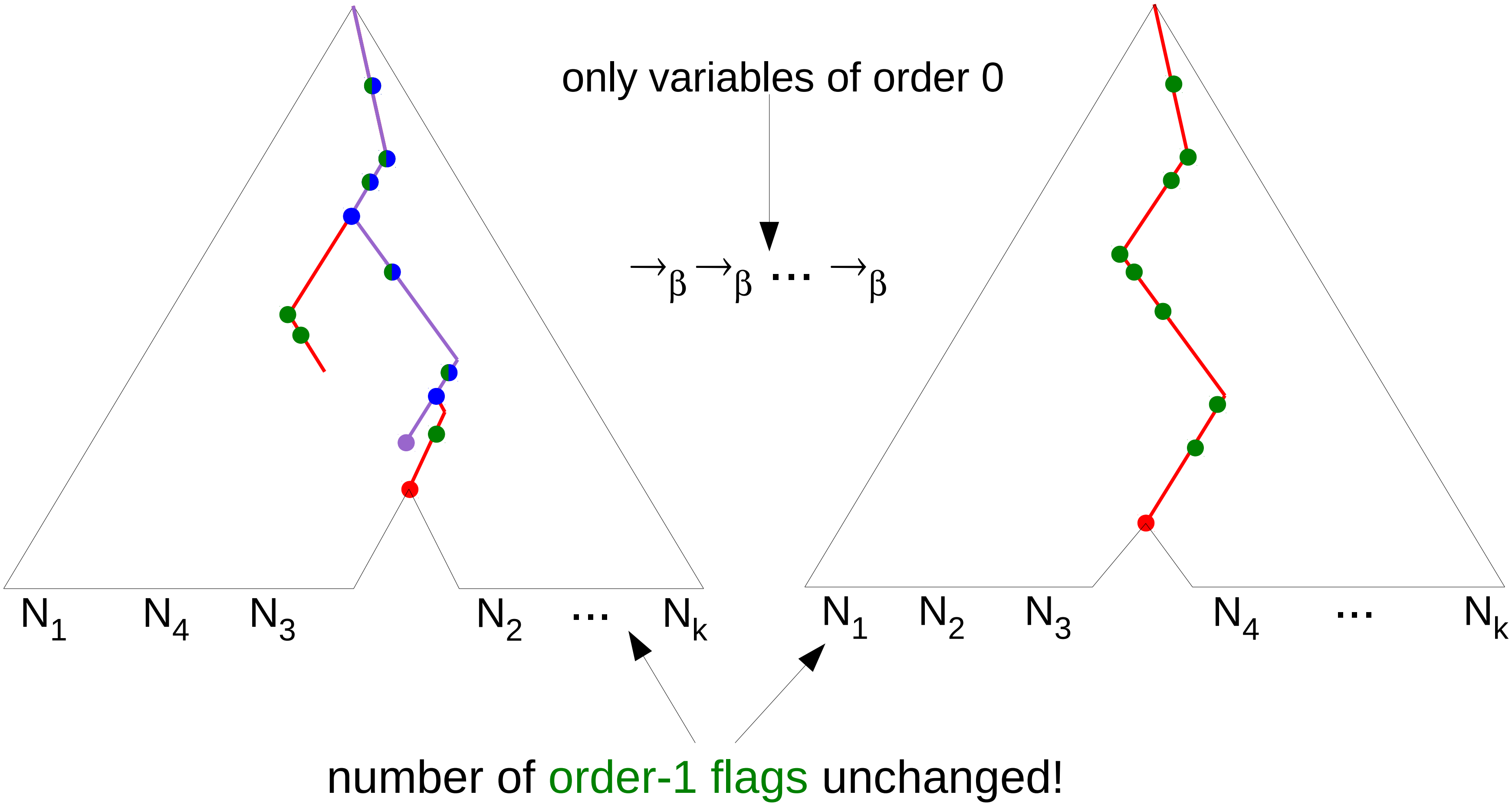
one marker of order 0
flags of order 1
one marker of order 1



number of order-1 flags unchanged!

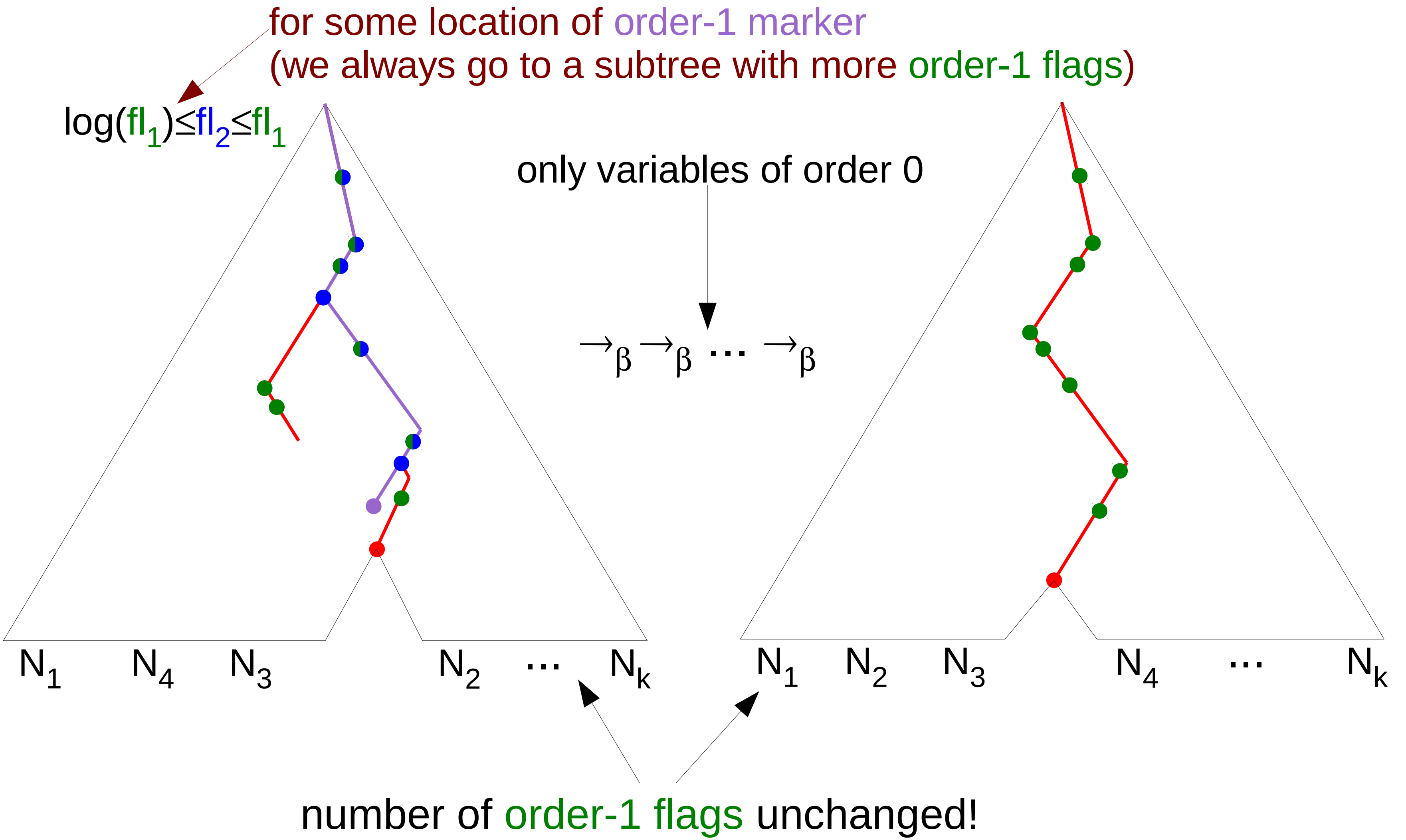
Flags & markers

one marker of order 0
flags of order 1
one marker of order 1
flags of order 2 – places on the path to order-1 marker having a descendant with order-1 flag

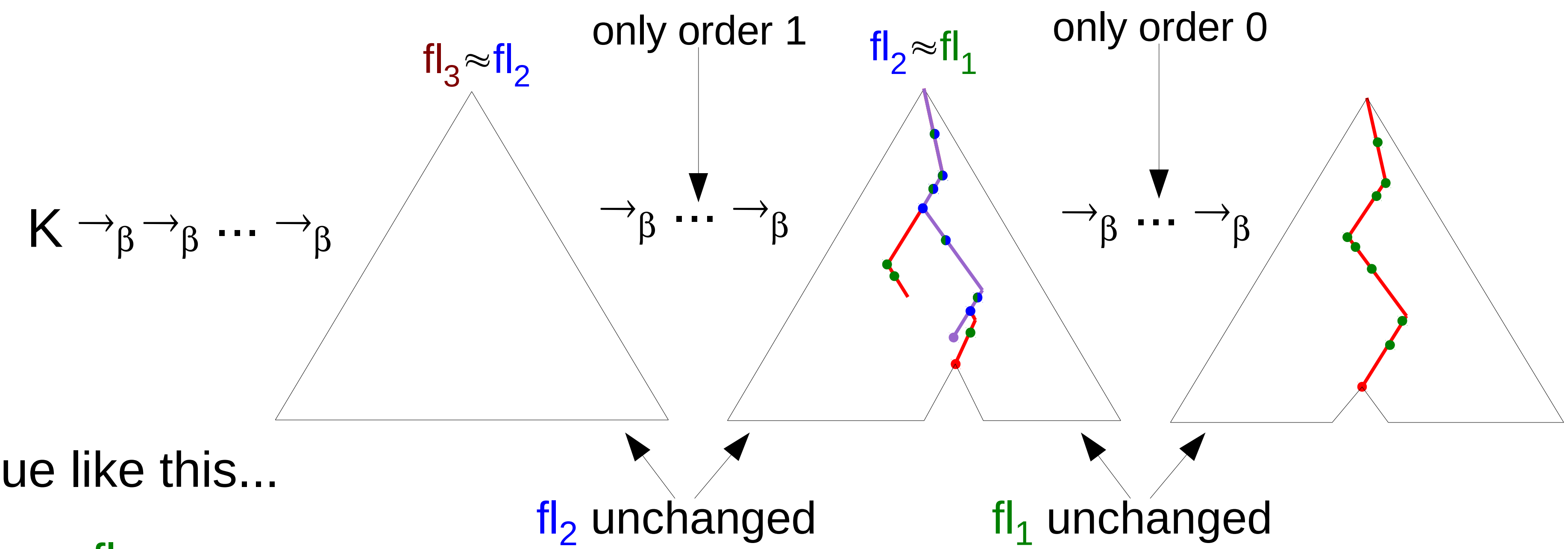


Flags & markers

one marker of order 0
flags of order 1
one marker of order 1
flags of order 2 – places on the path to order-1 marker having a descendant with order-1 flag



Flags & markers

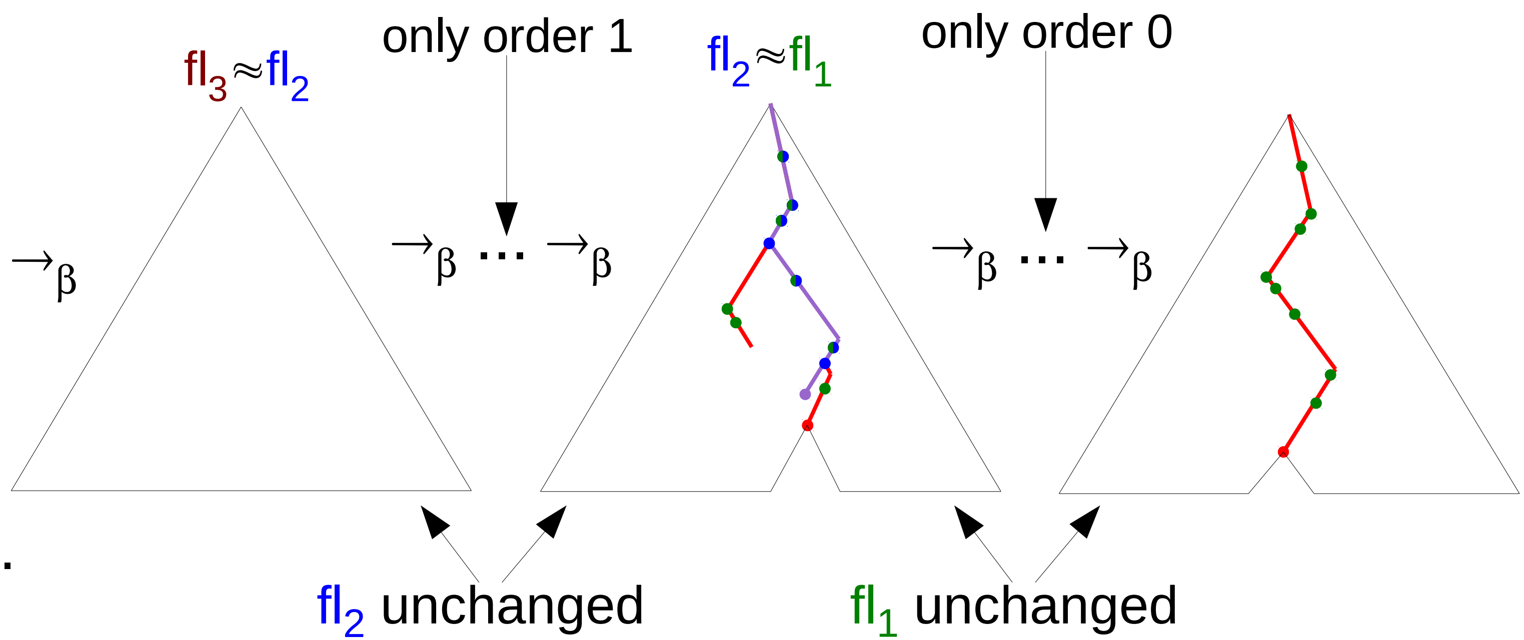


continue like this...

$fl_n \approx fl_1$

Flags & markers

$K \to_\beta \to_\beta \cdots \to_\beta$



continue like this...

$fl_n \approx fl_1$

We put all the flags & markers in derivations for K.
The number of order-n flags approximates the number of „a” on some path in the Böhm tree of K.

there exist derivations for K with arbitrarily many order-n flags

in the Böhm tree of K there exist paths with arbitrarily many „a”

easy to decide

Details in the paper...

Extensions (work in progress)

Diagonal problem:

Input: HORS K, set Σ of symbols

Question: In the tree generated by K, are there finite paths with arbitrarily many appearances of every symbol from Σ?

(i.e. for every N there exists a path P such that every symbol from Σ appears on P at least N times)

Thm [Hague, Kochems, Ong 2016], [Clemente, P., Salvati, Walukiewicz 2016].
The diagonal problem is decidable.

Our type system works for $|\Sigma|=1$.

Can be extended to $|\Sigma|>1$:

- $|\Sigma|$ markers of every order
- different flags for every $a\in\Sigma$

Extensions (work in progress)

Diagonal problem:
Input: HORS K, set Σ of symbols
Question: In the tree generated by K, are there finite paths with arbitrarily many appearances of every symbol from Σ?
(i.e. for every N there exists a path P such that every symbol from Σ appears on P at least N times)

Thm [Hague, Kochems, Ong 2016], [Clemente, P., Salvati, Walukiewicz 2016].
The diagonal problem is decidable.

Our type system works for |Σ|=1.
Can be extended to |Σ|>1:
- |Σ| markers of every order
- different flags for every a∈Σ

algorithm of high complexity:
f(n)-EXPTIME
for some $f(n)=O(n^2)$,
where n = order of the HORS

Thm.(Conjecture)
The diagonal problem for order-n HORSes is (n-1)-EXPTIME-complete.

Carefull optimization (reduction of number of types) required.

Extensions (work in progress)

MSO+U logic (introduced by Bojańczyk in 2004)

MSO+U extends MSO by the following „U” quantifier:

$$UX.\phi(X)$$

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

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

This construction may be nested inside other quantifiers,
and ϕ may have free variables other than X.

Extensions (work in progress)

WMSO+U logic (introduced by Bojańczyk in 2004)

MSO+U extends MSO by the following „U” quantifier:

$$UX.\phi(X)$$

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

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

This construction may be nested inside other quantifiers,
and ϕ may have free variables other than X.

We consider Weak MSO+U (quantification over finite sets only):

$$\exists X \rightarrow \exists_{fin} X$$

e.g. we can express that there exist paths with arbitrarily many „a”

Decision problems

Satisfiability

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

Satisfiability

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 $\mathcal{G}$

- undecidable for $\phi \in$ MSO+U (generalizes satifiability)

Decision problems

Satisfiability

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 $\mathcal{G}$

- undecidable for $\phi \in$ MSO+U (generalizes satifiability)
- **Thm** (conjecture)**: decidable for $\phi \in$ WMSO+U**

Solution: this work + a model of λ-calculus recognizing WMSO properties

Thank you!