HORS & Weak MSO+U Logic

Paweł Parys, Szymon Toruńczyk

University of Warsaw

NII Shonan Meeting "Higher-Order Model Checking"

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.

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.

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

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

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 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 MSO+U (generalizes satifiability)
- decidable when $\phi\in$ (quasi-weak cost-MSO) and $\mathcal{G}$ safe
  follows from [Blumensath, Colcombet, Kuperberg, P., Vanden Boom 2014]
  (in quasi-weak cost-MSO we can express the diagonal problem)

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 MSO+U (generalizes satifiability)
- decidable when $\phi\in$ (quasi-weak cost-MSO) and $\mathcal{G}$ safe
  follows from [Blumensath, Colcombet, Kuperberg, P., Vanden Boom 2014]
  (in quasi-weak cost-MSO we can express the diagonal problem)
- **Contribution: decidable for $\phi\in$ WMSO+U & all $\mathcal{G}$**

Decision problems

HORS model-checking

input: formula ϕ, HORS $\mathcal{G}$,

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

Contribution: decidable for $\phi \in$ WMSO+U & all $\mathcal{G}$

Moreover: for every $\phi \in$ WMSO+U we construct a "model" of λY-calculus recognizing ϕ

sort α ⟶ finite set $D_\phi[\alpha]$

term K of sort α,
valuation of free variables v ⟶ an element $[\![K,v]\!]_\phi \in D_\phi[\alpha]$

compositional!

(current version: for every n we have a different model that works well for terms of orders $\leqslant$n)

Construction of the model – preparation

(only logic, no automata!)

Step 1: WMSO+U is compositional

$t \longrightarrow [t]_\phi \in$ finite set (of phenotypes)

$[t]_\phi$ determines whether $t \models \phi$

$[a(t_1,...,t_n)]_\phi$ determined by a, $[t_1]_\phi$, ..., $[t_n]_\phi$

Construction of the model – preparation

(only logic, no automata!)

Step 1: WMSO+U is compositional

t,v ⟶ $[t,v]_\phi \in$ finite set (of phenotypes)

v = valuation of free variables of ϕ

$[t,v]_\phi$ determines whether $t,v \models \phi$

$[a(t_1,...,t_n),v]_\phi$ determined by a, $v \cap$ root, $[t_1, v \cap t_1]_\phi$, ..., $[t_n, v \cap t_n]_\phi$

e.g. $[t,v]_{UX.\varphi} = (\{\tau : \exists_{fin}\mathbf{X}.\ [t,v[X \to \mathbf{X}]]_\varphi = \tau\}, \{\tau : U\mathbf{X}.\ [t,v[X \to \mathbf{X}]]_\varphi = \tau\})$

Construction of the model – preparation

(only logic, no automata!)

Step 1: WMSO+U is compositional

$t,v \longrightarrow [t,v]_\phi \in$ finite set (of phenotypes)

v = valuation of free variables of ϕ

$[t,v]_\phi$ determines whether $t,v \models \phi$

$[a(t_1,\ldots,t_n),v]_\phi$ determined by a, $v\cap$root, $[t_1,v\cap t_1]_\phi$, ..., $[t_n,v\cap t_n]_\phi$

e.g. $[t,v]_{UX.\varphi}=(\{\tau : \exists_{fin}\mathbf{X}.\ [t,v[X\to\mathbf{X}]]_\varphi=\tau\}, \{\tau : U\mathbf{X}.\ [t,v[X\to\mathbf{X}]]_\varphi=\tau\})$

Step 2: assume (w.l.o.g.) that all types are homogeneous

i.e. in $\alpha_1\to\ldots\to\alpha_n\to o$ we have $ord(\alpha_1)\geq\ldots\geq ord(\alpha_n)$

Construction of the model – preparation

(only logic, no automata!)

Step 1: WMSO+U is compositional

$t,v \longrightarrow [t,v]_\phi \in$ finite set (of phenotypes)

v = valuation of free variables of ϕ

$[t,v]_\phi$ determines whether $t,v \models \phi$

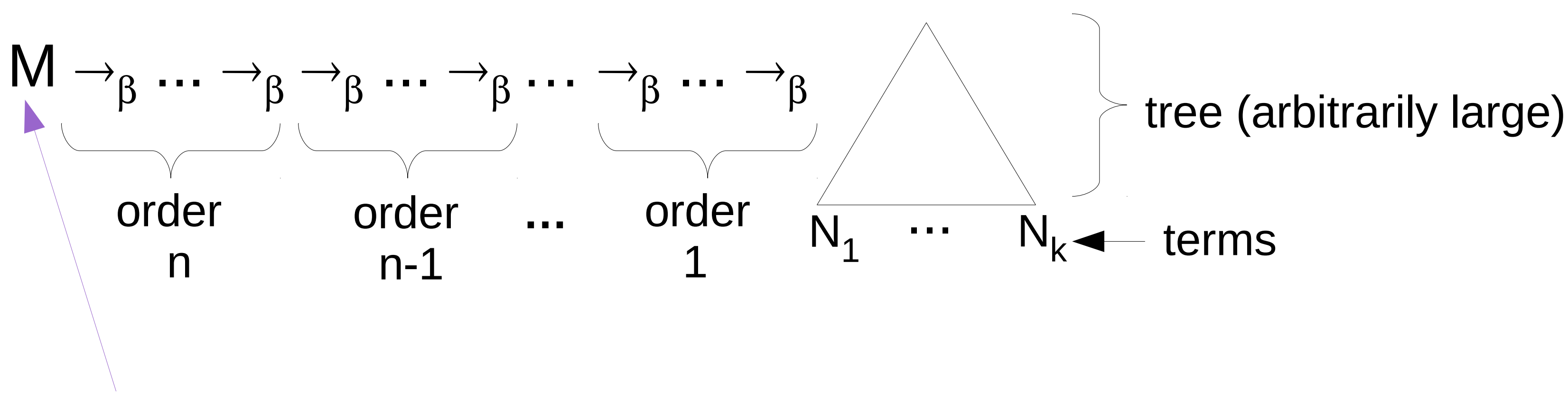
$[a(t_1,\dots,t_n),v]_\phi$ determined by a, $v \cap root$, $[t_1, v\cap t_1]_\phi, \dots, [t_n, v \cap t_n]_\phi$

e.g. $[t,v]_{UX.\varphi} = (\{\tau : \exists_{fin}\mathbf{X}.\ [t,v[X\to\mathbf{X}]]_\varphi = \tau\}, \{\tau : U\mathbf{X}.\ [t,v[X\to\mathbf{X}]]_\varphi = \tau\})$

Step 2: assume (w.l.o.g.) that all types are homogeneous

i.e. in $\alpha_1 \to \dots \to \alpha_n \to o$ we have $ord(\alpha_1) \geq \dots \geq ord(\alpha_n)$

Then we can perform β-reductions starting from variables of the highest order



infinite λ-term (obtained by replacing every nonterminal A by its rule $\lambda x_1.\cdots.\lambda x_m.K$, or by replacing every Y by appropriate infinite term)

Construction of the model

Let ϕ=UX.φ

Goal: construct a model for UX.φ

term K^{α} → value $[\![K]\!]_{\phi} \in$ finite set for each α

$[\![K^{o}]\!]_{\phi}$ determines $[BT(K)]_{\phi}$

for each τ: does there exist arbitrarily large set X s.t. $[BT(K),X]_{\varphi}=\tau$?
(where free variables of ϕ are empty sets)

Construction of the model

Let ϕ=UX.φ

Goal: construct a model for UX.φ

term K^α → value $[\![K]\!]_\phi \in$ finite set for each α

$[\![K^o]\!]_\phi$ determines $[BT(K)]_\phi$

for each τ: does there exist arbitrarily large set X s.t. $[BT(K),X]_\varphi=\tau$?
(where free variables of ϕ are empty sets)

Inductive construction!

We have a model for φ (such that $[\![N^o]\!]_\varphi$ determines $[BT(N),\emptyset]_\varphi$)

We design an intersection type system, where we put flags in derivations.

(we can derive N^o:(F,M,τ) using k flags) $\Leftrightarrow$ ($[BT(N),X]_\varphi=\tau$)
where $|X|\approx k$

Then $[\![N]\!]_\phi$ = ($[\![N]\!]_\varphi$,
types of N,
types of N that can be derived with arb. many flags)

Intersection types

Intersection types refining sort o:

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

φ-type of Böhm tree

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

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

$$\mathcal{T}^{\alpha} = \{\, (\tau_1, T_1) \to \ldots \to (\tau_1, T_k) \to (F, M, \tau) \,\}$$

φ-type of Böhm tree

markers used in the derivation

values in the φ-model

flags used in the derivation

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

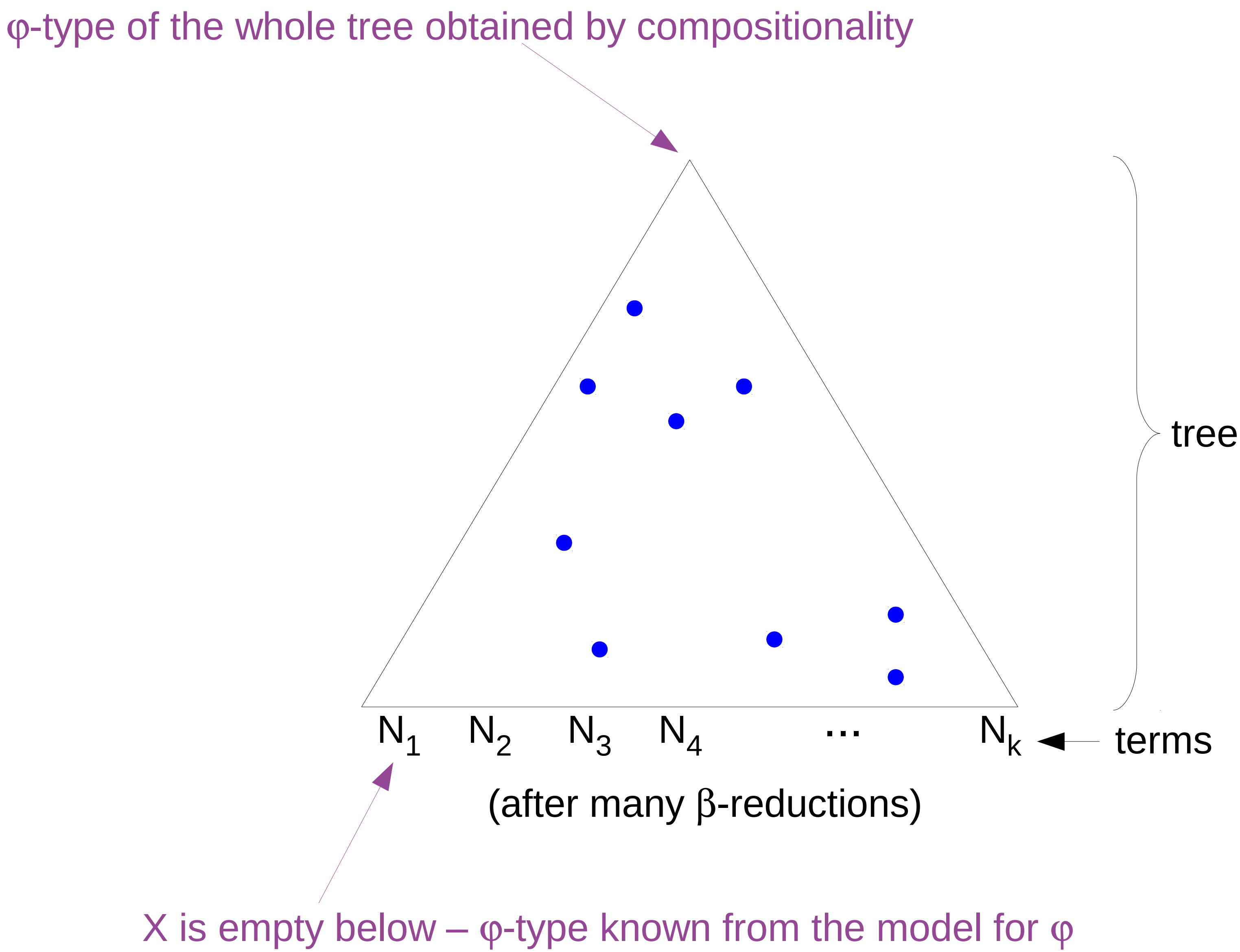
(for each order m we have flags of order m, and a marker of order m)

Only finite derivations!

(after finitely many steps we use a rule that extracts a type from the φ-model)

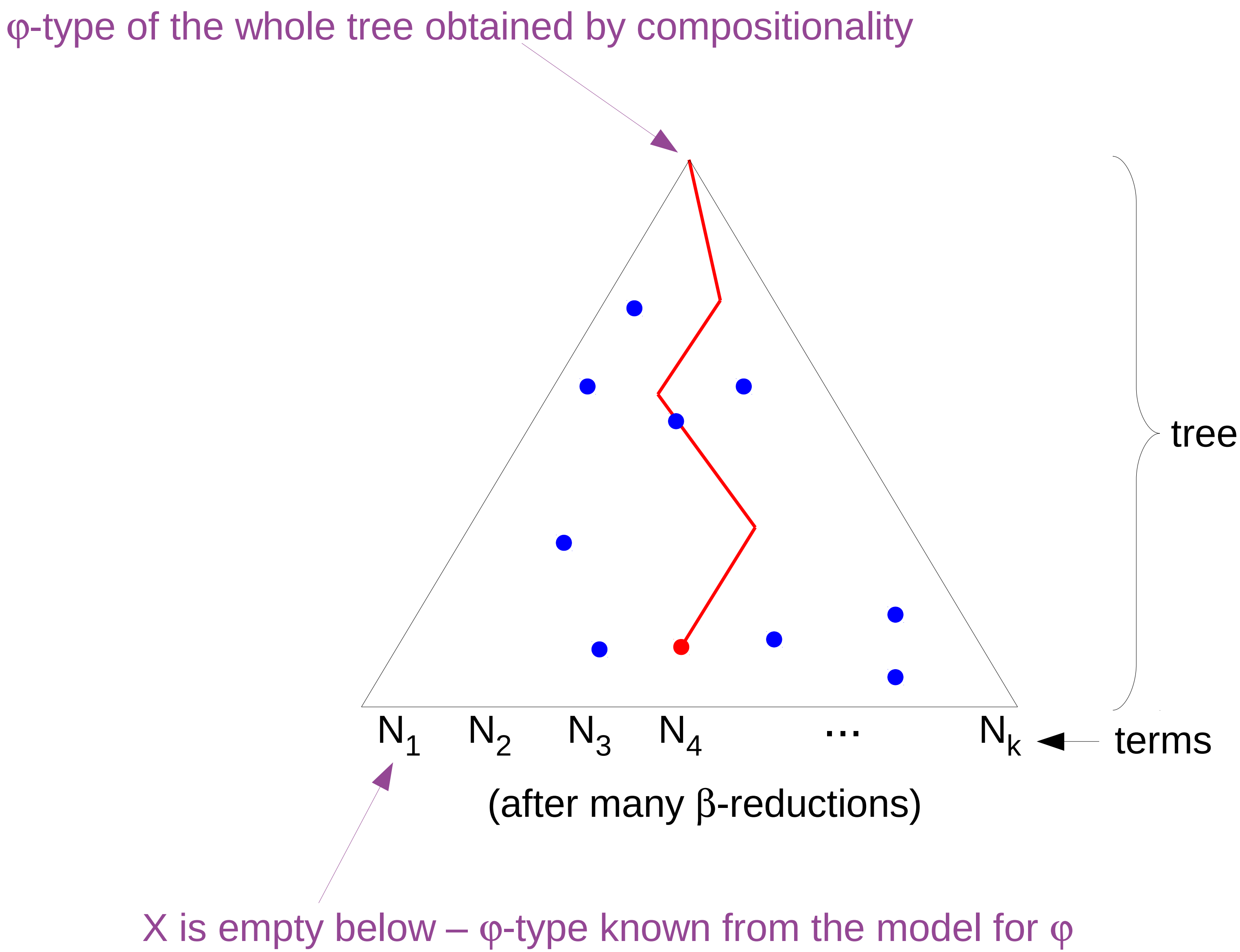
Flags & markers

flags of order 0 = nodes in X



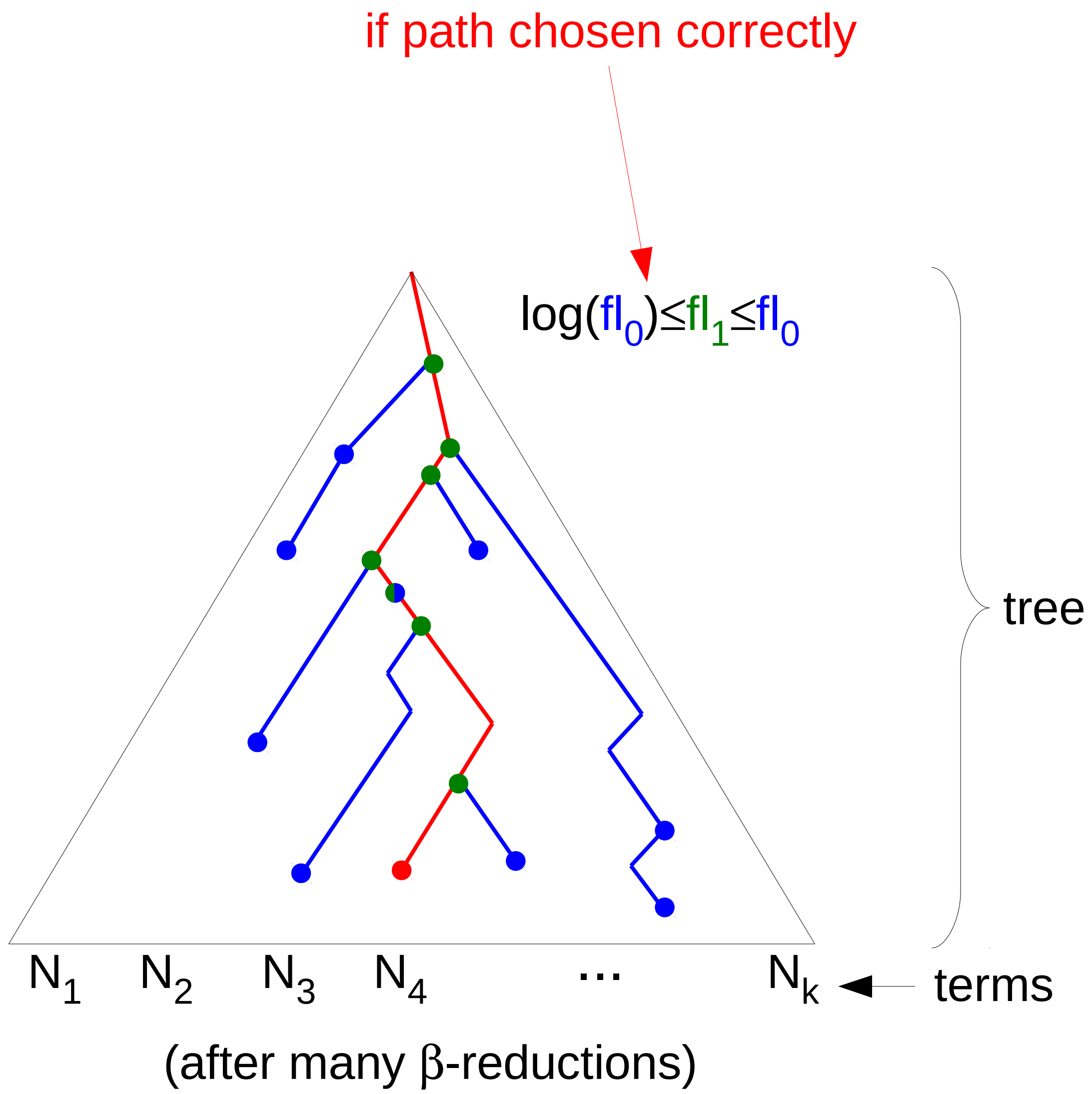
Flags & markers

flags of order 0 = nodes in X
one marker of order 0



Flags & markers

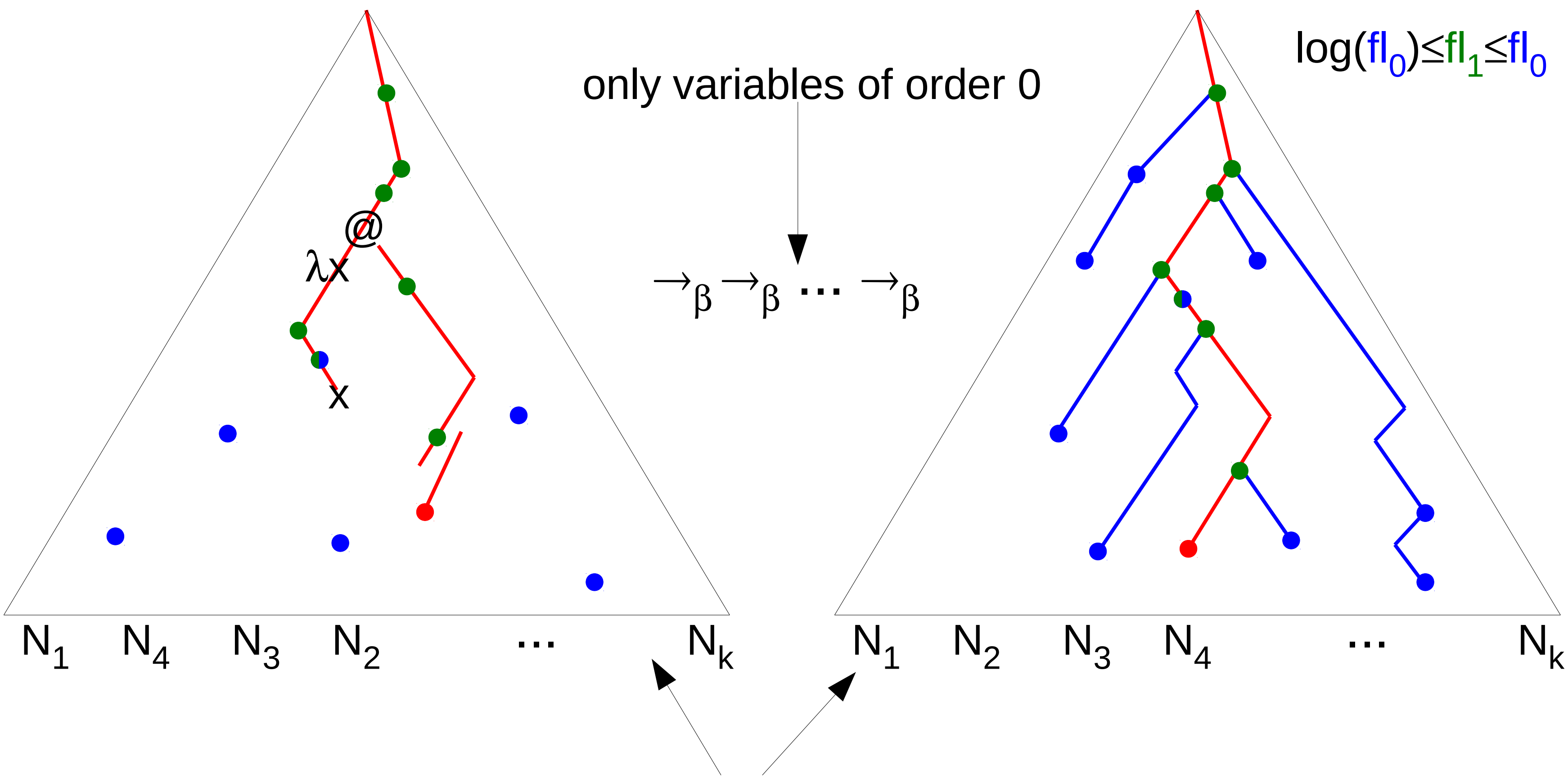
flags of order 0 = nodes in X
one marker of order 0
flags of order 1



Flags & markers

flags of order 0 = nodes in X
one marker of order 0
flags of order 1

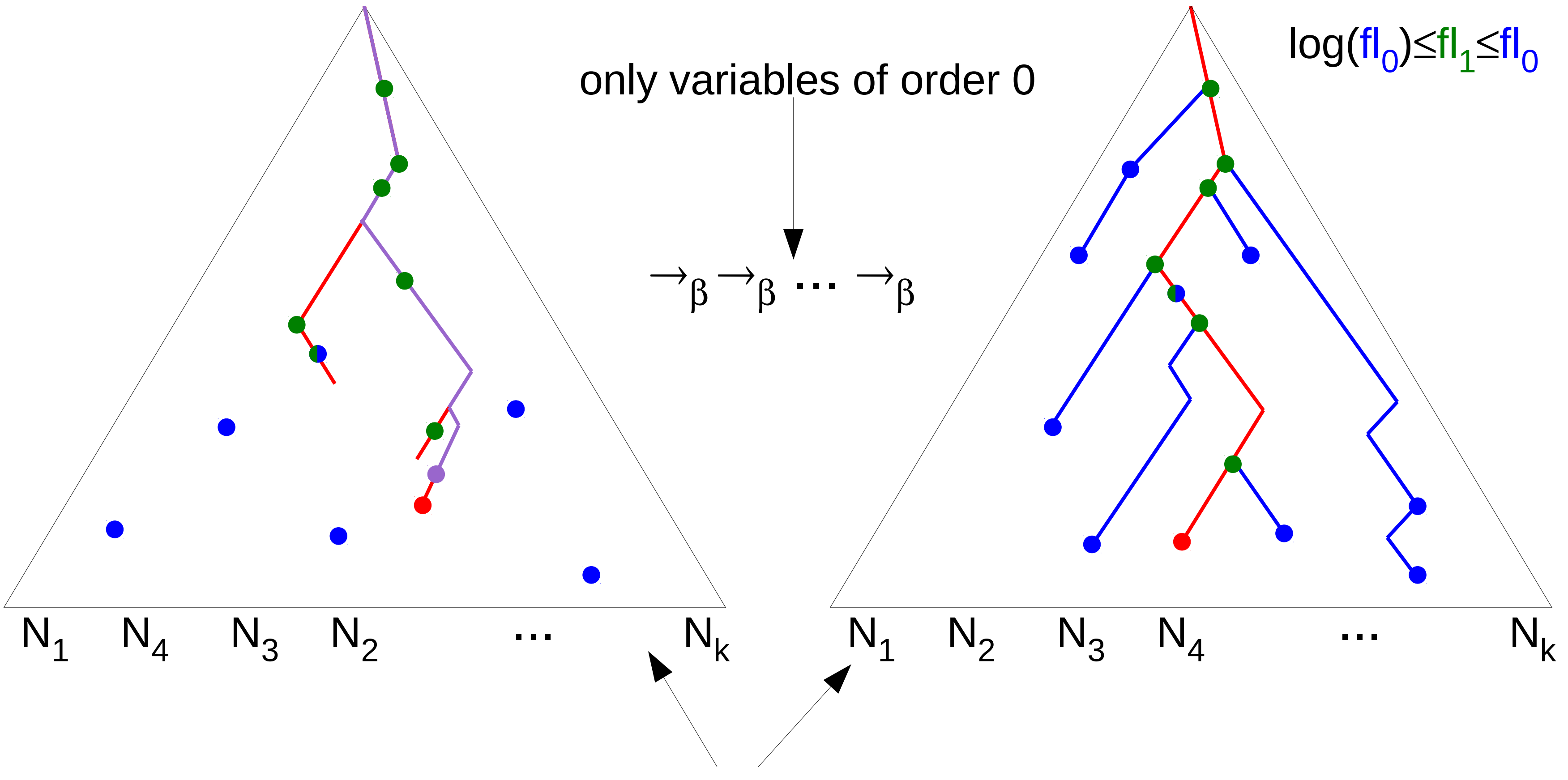
the type system ensures that a variable with marker is used exactly once!



number of order-1 flags unchanged!

Flags & markers

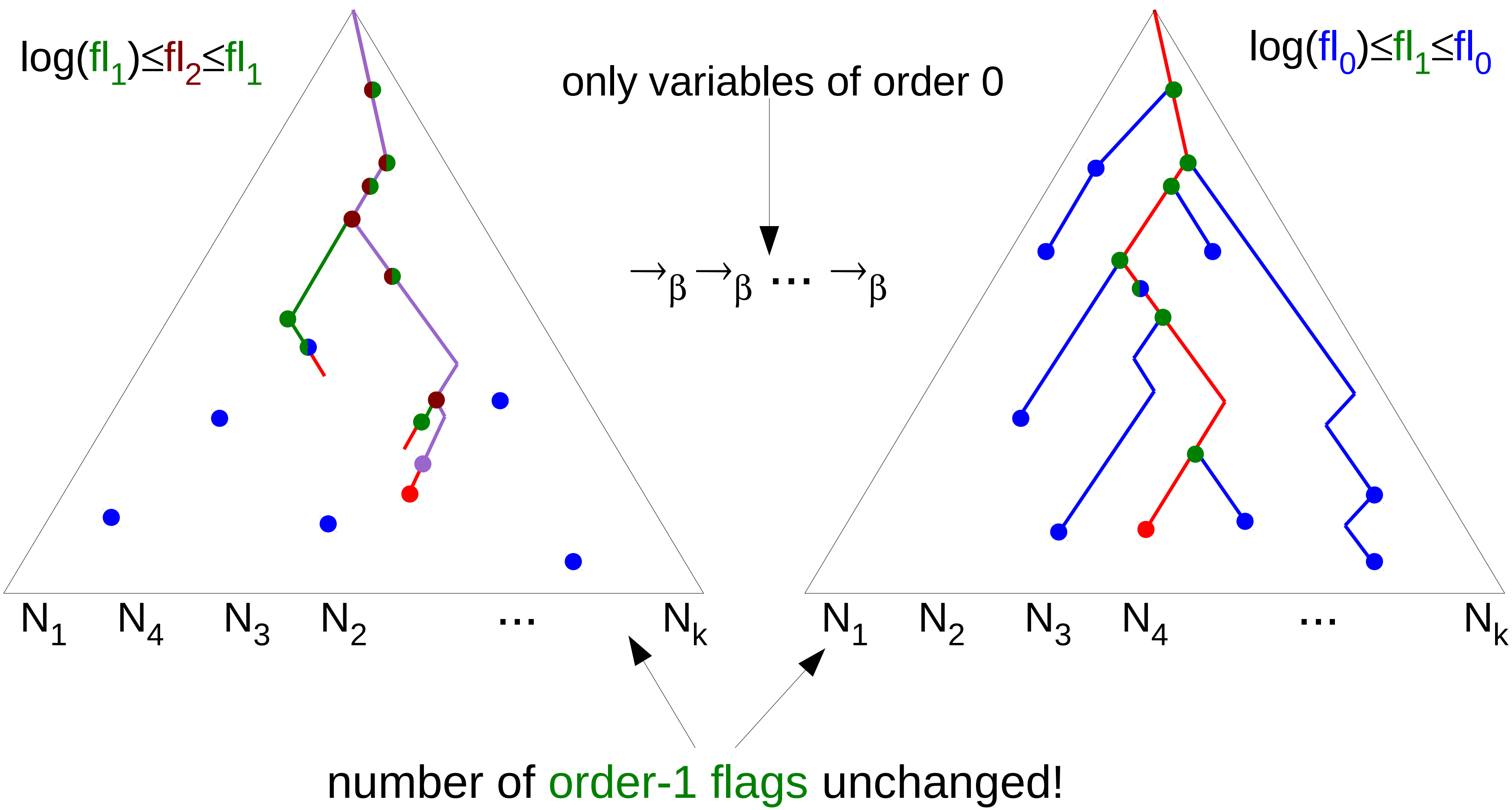
flags of order 0 = nodes in X
one marker of order 0
flags of order 1
one marker of order 1



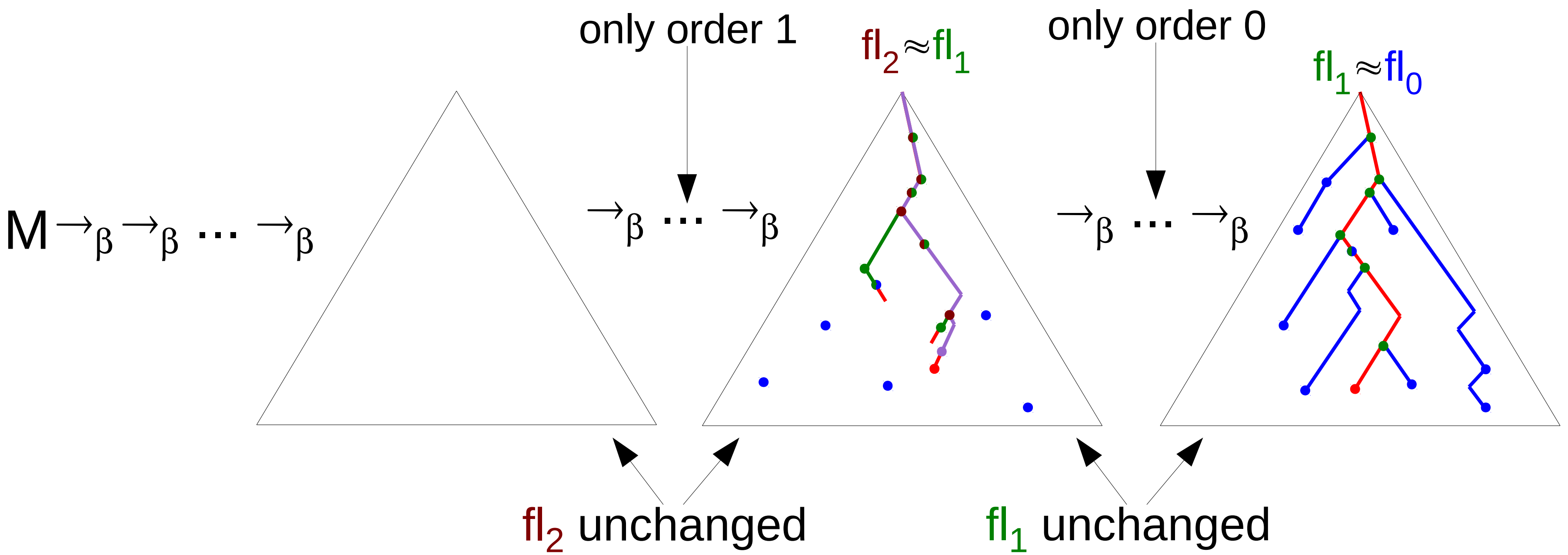
number of order-1 flags unchanged!

Flags & markers

flags of order 0 = nodes in X
one marker of order 0
flags of order 1
one marker of order 1
flags of order 2



Flags & markers



continue like this...

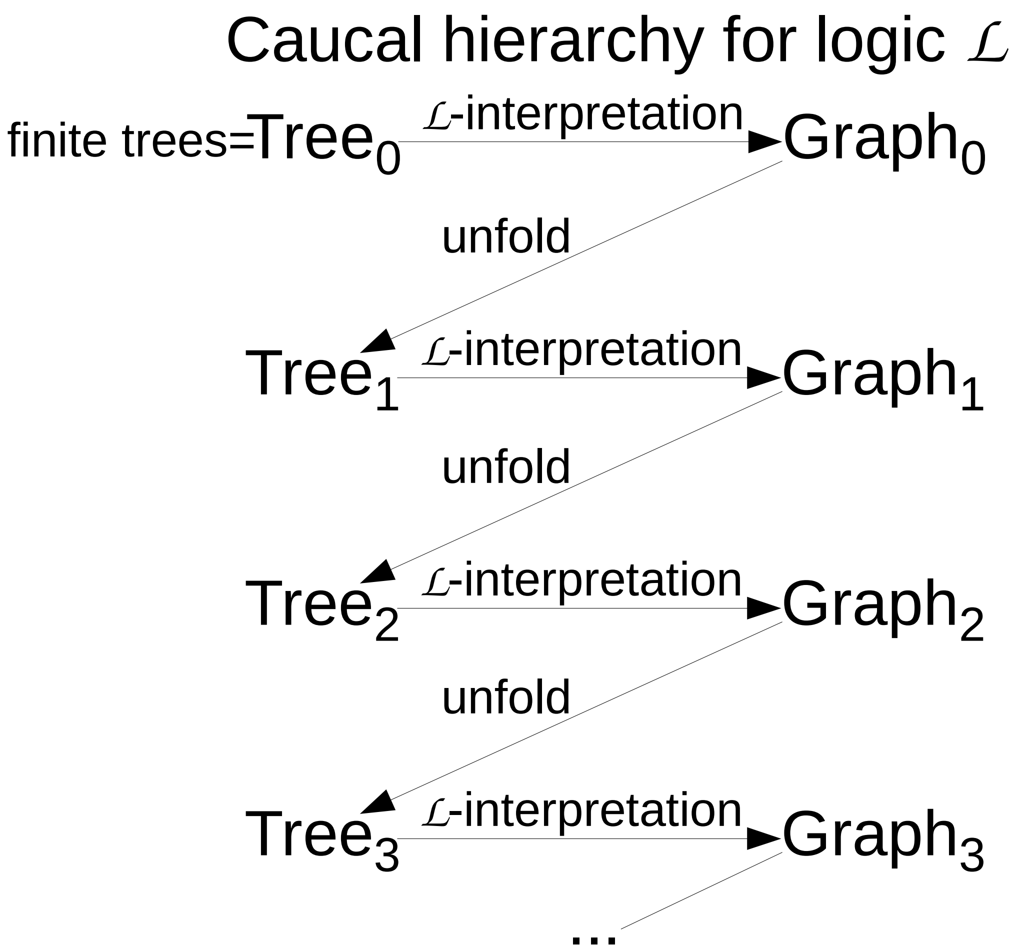
$$fl_n \approx |X|$$

Model vs decidability

1) While considering UX.φ, we need a model for φ (decidability not enough)

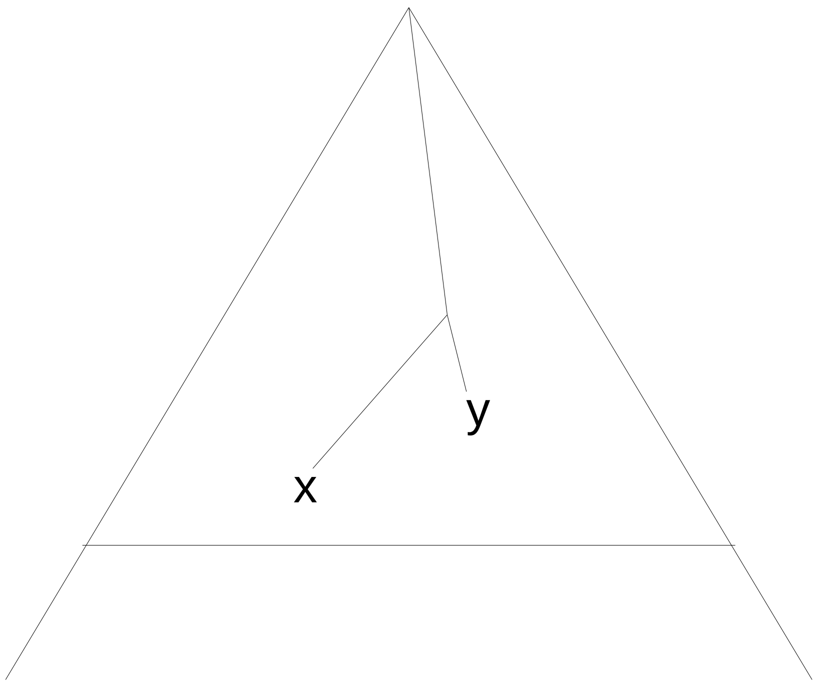
2) Having a model gives some advantages:
   - reflection
   - transfer theorem
   - …
   - WMSO+U gives the same Caucal hierarchy as MSO

Caucal hierarchy for WMSO+U



FO-hierarchy = WMSO-hierarchy = MSO-hierarchy **= WMSO+U-hierarchy**

Caucal hierarchy for WMSO+U

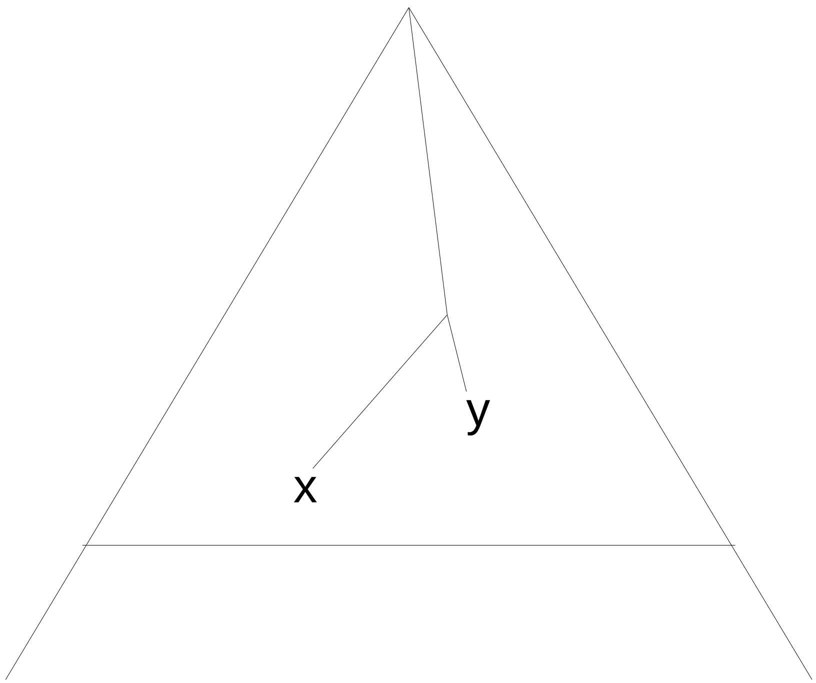


$\phi(x,y) \in$ MSO/WMSO+U

↓

$\phi'(x,y) \in$ (WMSO using as predicates formulae $\varphi(z) \in$ MSO/WMSO+U)

Caucal hierarchy for WMSO+U

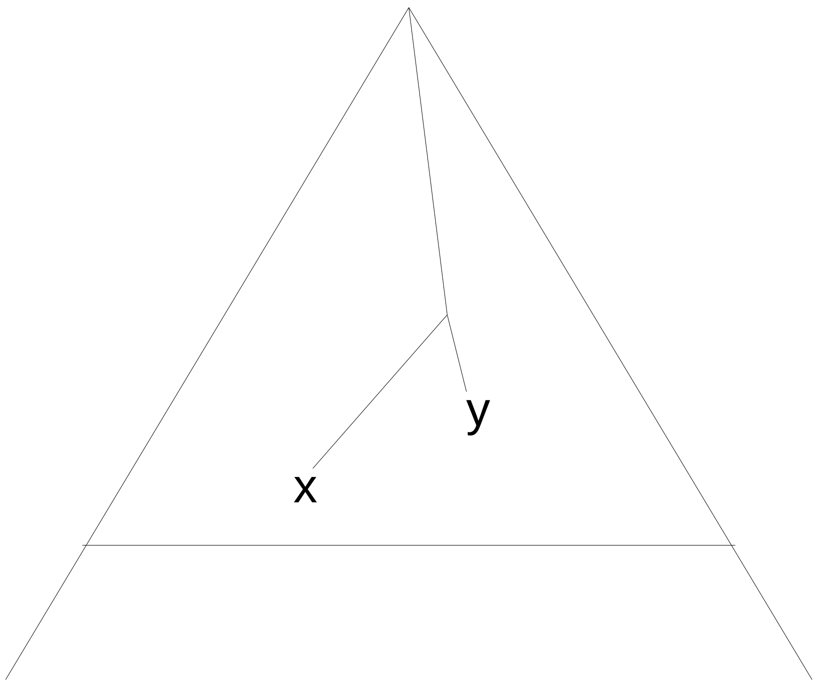


$\phi(x,y) \in$ MSO/WMSO+U

[Colcombet 2007]

$\phi'(x,y) \in$ (FO using as predicates formulae $\varphi(z) \in$ MSO/WMSO+U)

Caucal hierarchy for WMSO+U



$\phi(x,y)\in$ MSO/WMSO+U

↓ [Colcombet 2007]

$\phi'(x,y)\in$ (FO using as predicates formulae $\varphi(z)\in$ MSO/WMSO+U)

↓

$\phi''(x,y)\in$ (FO reading MSO/WMSO+U-types from labels)

Tree_n is closed for labeling by values of $\varphi(z)\in$ MSO/WMSO+U because:

- $\mathrm{Tree}_n \approx$ Böhm trees of safe HORSes
- we can enrich a safe HORS by the labeling, using our model (reflection)

What next? - ideas

- Model independent from the maximal order of terms
- A similar type system, but with separate marker/flag for each pair (order, input letter) allows (?) to solve the diagonal problem in ≈(n-1)-EXPTIME
- Pumping lemma for nondeterministic HORSes (???)
  ⇒ bound on size of ideals ⇒ complexity of computing downward closure

Thank you!