Size-based termination for higher-order rewriting

Frédéric Blanqui

Deduc⊢eam

1st part

Outline

Higher-order rewriting

- terms: $t = \mathsf{c} \mid \mathsf{f} \mid x \mid \lambda x^U.t \mid tt$
- types: $U = \mathsf{B} \mid U \Rightarrow U$

Higher-order rewriting

- terms: $t = \mathsf{c} \mid \mathsf{f} \mid x \mid \lambda x^U.t \mid tt$
- types: $U = \mathsf{B} \mid U \Rightarrow U$
- operational semantics: $\begin{cases} (\lambda x^U.t)u & \to_\beta \quad t_x^u \\ \mathsf{f}\, l_1 \ldots l_n & \to_{\mathcal{R}} \quad r \end{cases}$

Higher-order rewriting

- terms: $t = \mathsf{c} \mid \mathsf{f} \mid x \mid \lambda x^U.t \mid tt$
- types: $U = \mathsf{B} \mid U \Rightarrow U$
- operational semantics: $\begin{cases} (\lambda x^U.t)u & \to_\beta & t_x^u \\ \mathsf{f}\, l_1 \ldots l_n & \to_{\mathcal{R}} & r \end{cases}$

example 1: $\mathsf{nil} : \mathsf{L}$, $\mathsf{cons} : \mathsf{N} \Rightarrow \mathsf{L} \Rightarrow \mathsf{L}$, $\mathsf{map} : (\mathsf{N} \Rightarrow \mathsf{N}) \Rightarrow \mathsf{L} \Rightarrow \mathsf{L}$

$$\begin{array}{rcl} \mathsf{map}\ f\ \mathsf{nil} & \to_{\mathcal{R}} & \mathsf{nil} \\ \mathsf{map}\ f\ (\mathsf{cons}\ x\ l) & \to_{\mathcal{R}} & \mathsf{cons}\ (f\ x)\ (\mathsf{map}\ f\ l) \end{array}$$

$\mathsf{map}\ (\lambda x.2 * x)\ (\mathsf{cons}\ 5\ l)$
$\to_{\mathcal{R}} \mathsf{cons}\ ((\lambda x.2 * x)\ 5)\ (\mathsf{map}\ (\lambda x.2 * x)\ l)$
$\to_\beta \mathsf{cons}\ (2 * 5)\ (\mathsf{map}\ (\lambda x.2 * x)\ l)$
$\ldots$

Higher-order rewriting

example 2: recursor on Howard's constructive ordinals

- $0 : \mathsf{O}$
- $\mathsf{s} : \mathsf{O} \Rightarrow \mathsf{O}$
- $\mathsf{lim} : (\mathsf{N} \Rightarrow \mathsf{O}) \Rightarrow \mathsf{O}$
- ordrec :
  $T \Rightarrow (\mathsf{O} \Rightarrow T \Rightarrow T) \Rightarrow ((\mathsf{N} \Rightarrow \mathsf{O}) \Rightarrow (\mathsf{N} \Rightarrow T) \Rightarrow T) \Rightarrow \mathsf{O} \Rightarrow T$

$$
\begin{array}{rcl}
\mathsf{ordrec}\ u\ v\ w\ 0 & \to_{\mathcal{R}} & U \\
\mathsf{ordrec}\ u\ v\ w\ (\mathsf{s}\ x) & \to_{\mathcal{R}} & v\ x\ (\mathsf{ordrec}\ u\ v\ w\ x) \\
\mathsf{ordrec}\ u\ v\ w\ (\mathsf{lim}\ z) & \to_{\mathcal{R}} & w\ z\ (\lambda n.\mathsf{ordrec}\ u\ v\ w\ (z\ n))
\end{array}
$$

Higher-order rewriting

example 3: recursor on continuations

- $\mathsf{D} : \mathsf{C}$
- $\mathsf{C} : ((\mathsf{C} \Rightarrow \mathsf{L}) \Rightarrow \mathsf{L}) \Rightarrow \mathsf{C}$
- contrec :
  $T \Rightarrow (((\mathsf{C} \Rightarrow \mathsf{L}) \Rightarrow \mathsf{L}) \Rightarrow ((T \Rightarrow \mathsf{L}) \Rightarrow \mathsf{L}) \Rightarrow T) \Rightarrow \mathsf{C} \Rightarrow T$
- $\mathsf{ex} : \mathsf{C} \Rightarrow \mathsf{L}$

$$
\begin{array}{rcl}
\mathsf{contrec}\ u\ v\ \mathsf{D} & \to_{\mathcal{R}} & u \\
\mathsf{contrec}\ u\ v\ (\mathsf{C}\, z) & \to_{\mathcal{R}} & v\ z\ (\lambda x.z\,(\lambda y.x\,(\mathsf{contrec}\ u\ v\ y))) \\
\mathsf{ex}\ (\mathsf{C}\, z) & \to_{\mathcal{R}} & z\ \mathsf{ex}
\end{array}
$$

Outline

Hughes-Pareto-Sabry (POPL'96) remark:

1. a *closed* term t of base type terminates if its denotational semantics $[\![t]\!] \neq \bot$ [Plotkin 1977]

Hughes-Pareto-Sabry (POPL'96) remark:

1. a *closed* term t of base type terminates if its denotational semantics $[\![t]\!] \neq \bot$ [Plotkin 1977]
2. the denotational semantics of an inductive type is the least fixpoint of a monotone function [Scott 1969]

   example: for lists, $[\![\mathsf{L}]\!] = \mathrm{lfp}(F^{\mathsf{L}})$ where $F^{\mathsf{L}}(X) = 1 + [\![\mathsf{N}]\!] \times X$

Hughes-Pareto-Sabry (POPL'96) remark:

1. a *closed* term t of base type terminates if its denotational semantics $[\![t]\!] \neq \bot$ [Plotkin 1977]
2. the denotational semantics of an inductive type is the least fixpoint of a monotone function [Scott 1969]

   example: for lists, $[\![\mathsf{L}]\!] = \mathrm{lfp}(F^{\mathsf{L}})$ where $F^{\mathsf{L}}(X) = 1 + [\![\mathsf{N}]\!] \times X$
3. $\mathrm{lfp}(F^{\mathsf{L}})$ can be reached by transfinite iteration of F^{L}
   i.e. there is an ordinal $\mathfrak{h}$ such that $\mathrm{lfp}(F^{\mathsf{L}}) = [\![\mathsf{L}]\!]^{\mathrm{default}}_{\mathfrak{h}}$ where:

$$[\![\mathsf{L}]\!]^{\mathrm{default}}_{\mathfrak{a}} = \begin{cases} \bot & \text{if } \mathfrak{a} = 0 \\ F^{\mathsf{L}}([\![\mathsf{L}]\!]_{\mathfrak{b}}) & \text{if } \mathfrak{a} = \mathfrak{b} + 1 \\ \bigcup_{\mathfrak{b}<\mathfrak{a}} [\![\mathsf{L}]\!]_{\mathfrak{b}} & \text{if } \mathfrak{a} \text{ limit ordinal} > 0 \end{cases}$$

Hughes-Pareto-Sabry (POPL'96) remark:

4. $[\![\mathsf{A}]\!]^{\mathrm{default}}$ gives a notion of size for every $t : \mathsf{A}$:

   $\|t\|^{\mathrm{default}} =$ smallest ordinal $\mathfrak{a}$ such that $t \in [\![\mathsf{A}]\!]^{\mathrm{default}}_{\mathfrak{a}}$

examples:

- for lists $F^{\mathsf{L}}(X) = 1 + \mathsf{N} \times X$, $\|l\|^{\mathrm{default}}$ is the length of l
- for trees $F^{\mathsf{T}}(X) = 1 + X \times X$, $\|t\|^{\mathrm{default}}$ is the height of t

Terms of size $> \omega$ (excluded by Hughes-Pareto-Sabry)

1. higher-order inductive types:

   with $\mathsf{zero} : \mathsf{O}$, $\mathsf{succ} : \mathsf{O} \Rightarrow \mathsf{O}$, $\mathsf{lim} : (\mathsf{N} \Rightarrow \mathsf{O}) \Rightarrow \mathsf{O}$

   $\mathsf{inj} : \mathsf{N} \Rightarrow \mathsf{O}$ and $\left\{ \begin{array}{rcl} \mathsf{inj}\,0 & \to_{\mathcal{R}} & \mathsf{zero} \\ \mathsf{inj}\,(\mathsf{s}\,x) & \to_{\mathcal{R}} & \mathsf{succ}\,(\mathsf{inj}\,x) \end{array} \right.$

   we have $\|\mathsf{lim}\ \mathsf{inj}\| = \omega + 1$

Terms of size $> \omega$ (excluded by Hughes-Pareto-Sabry)

1. higher-order inductive types:

   with $\mathsf{zero} : \mathsf{O}$, $\mathsf{succ} : \mathsf{O} \Rightarrow \mathsf{O}$, $\mathsf{lim} : (\mathsf{N} \Rightarrow \mathsf{O}) \Rightarrow \mathsf{O}$

   $\mathsf{inj} : \mathsf{N} \Rightarrow \mathsf{O}$ and $\begin{cases} \mathsf{inj}\,0 & \to_{\mathcal{R}} & \mathsf{zero} \\ \mathsf{inj}\,(\mathsf{s}\,x) & \to_{\mathcal{R}} & \mathsf{succ}\,(\mathsf{inj}\,x) \end{cases}$

   we have $\|\mathsf{lim}\ \mathsf{inj}\| = \omega + 1$

2. non-confluent infinitely branching rewrite systems:

   with $\mathsf{f} \to \mathsf{s}^i\,0 \quad (i \in \mathbb{N})$

   we have $\|\mathsf{f}\| = \omega + 1$

Hughes-Pareto-Sabry (POPL'96)

idea: abstract interpretation of semantic types

1. consider abstract size expressions: $a \in \mathtt{A} \cup \{\infty\}$
   where $a \in \mathtt{A} = \alpha \mid 0 \mid \mathsf{s}a \mid a + a$ (Presburger arithmetic)

Hughes-Pareto-Sabry (POPL'96)

idea: abstract interpretation of semantic types

1. consider abstract size expressions: $a \in \mathtt{A} \cup \{\infty\}$
   where $a \in \mathtt{A} = \alpha \mid 0 \mid \mathsf{s}a \mid a + a$ (Presburger arithmetic)
2. add types L_a (terms of size $\leq a$) with $a \in \mathtt{A} \cup \{\infty\}$:
   $[\![\mathsf{L}_\infty]\!] = [\![\mathsf{L}]\!]$ and $[\![\mathsf{L}_a]\!]\mu = [\![\mathsf{L}]\!]_{[\![a]\!]\mu}$ if $\mu : \mathtt{X} \to \mathfrak{h}$

Hughes-Pareto-Sabry (POPL'96)

3. typing rules = deduction rules wrt types and size annotations:

$$\text{(nil)}\ \frac{}{\vdash \mathsf{nil} : \mathsf{L}_{\alpha+1}} \qquad \text{(cons)}\ \frac{}{\vdash \mathsf{cons} : \mathsf{N} \Rightarrow \mathsf{L}_{\alpha} \Rightarrow \mathsf{L}_{\alpha+1}}$$

$$\text{(match)}\ \frac{\vdash l : \mathsf{L}_{\alpha+1} \quad \vdash t : T(\alpha) \quad x : \mathsf{N}, l' : \mathsf{L}_{\alpha} \vdash u : T(\alpha)}{\vdash \mathsf{match}\ l\ \mathsf{with}\ \mathsf{nil} \mapsto t \mid \mathsf{cons}\, x\, l' \mapsto u : T(\alpha)}$$

$$\text{(fixpoint)}\ \frac{F : \mathsf{L}_{\alpha} \Rightarrow T(\alpha), l : \mathsf{L}_{\alpha+1} \vdash t : T(\alpha+1)}{\mu F.\lambda l.t : \mathsf{L}_{\alpha} \Rightarrow T(\alpha)}$$

Hughes-Pareto-Sabry (POPL'96)

$$\text{(sub)}\ \frac{\vdash t : T \quad T \leq U}{\vdash t : U}$$

$$\text{(sub-refl)}\ \frac{}{T \leq T} \qquad \text{(sub-trans)}\ \frac{T \leq U \quad U \leq V}{T \leq V}$$

$$\text{(sub-base)}\ \frac{a \leq_{\mathtt{A}}^{\infty} b}{L_a \leq L_b} \quad \text{where } a \leq_{\mathtt{A}}^{\infty} b \text{ iff } a \leq_{\mathtt{A}} b \text{ or } b = \infty$$

$$\text{(sub-arrow)}\ \frac{T' \leq T \quad U \leq U'}{T \Rightarrow U \leq T' \Rightarrow U'}$$

Hughes-Pareto-Sabry (POPL'96)

example: $\mathsf{map} : (\mathsf{N} \Rightarrow \mathsf{N}) \Rightarrow \mathsf{L}_\alpha \Rightarrow \mathsf{L}_\alpha$?

$\mathsf{map} = \lambda f.\mu F.\lambda l.t$

$t = \mathsf{match}\ l\ \mathsf{with}\ \mathsf{nil} \mapsto \mathsf{nil} \mid \mathsf{cons}\, x\, l' \mapsto \mathsf{cons}\,(f\ x)\,(F\ l')$

$\Gamma = f : \mathsf{N} \Rightarrow \mathsf{N}, F : \mathsf{L}_\alpha \Rightarrow \mathsf{L}_\alpha, l : \mathsf{L}_{\alpha+1}$

$\Gamma \vdash t : \mathsf{L}_{\alpha+1}$?

Hughes-Pareto-Sabry (POPL'96)

example: $\mathsf{map} : (\mathsf{N} \Rightarrow \mathsf{N}) \Rightarrow \mathsf{L}_\alpha \Rightarrow \mathsf{L}_\alpha$?

$\mathsf{map} = \lambda f.\mu F.\lambda l.t$

$t = \mathsf{match}\ l\ \mathsf{with}\ \mathsf{nil} \mapsto \mathsf{nil} \mid \mathsf{cons}\, x\, l' \mapsto \mathsf{cons}\,(f\ x)\,(F\ l')$

$\Gamma = f : \mathsf{N} \Rightarrow \mathsf{N}, F : \mathsf{L}_\alpha \Rightarrow \mathsf{L}_\alpha, l : \mathsf{L}_{\alpha+1}$

$\Gamma \vdash t : \mathsf{L}_{\alpha+1}$?

1. $\Gamma \vdash \mathsf{nil} : \mathsf{L}_{\alpha+1}$?

Hughes-Pareto-Sabry (POPL'96)

example: $\mathsf{map} : (\mathsf{N} \Rightarrow \mathsf{N}) \Rightarrow \mathsf{L}_\alpha \Rightarrow \mathsf{L}_\alpha$?

$\mathsf{map} = \lambda f.\mu F.\lambda l.t$

$t = \mathsf{match}\ l\ \mathsf{with}\ \mathsf{nil} \mapsto \mathsf{nil} \mid \mathsf{cons}\, x\, l' \mapsto \mathsf{cons}\,(f\ x)\,(F\ l')$

$\Gamma = f : \mathsf{N} \Rightarrow \mathsf{N}, F : \mathsf{L}_\alpha \Rightarrow \mathsf{L}_\alpha, l : \mathsf{L}_{\alpha+1}$

$\Gamma \vdash t : \mathsf{L}_{\alpha+1}$?

1. $\Gamma \vdash \mathsf{nil} : \mathsf{L}_{\alpha+1}$?
2. $\Gamma, x : \mathsf{N}, l' : \mathsf{L}_\alpha \vdash \mathsf{cons}\,(f\ x)\,(F\ l') : \mathsf{L}_{\alpha+1}$?

Hughes-Pareto-Sabry (POPL'96)

example: $\mathsf{map} : (\mathsf{N} \Rightarrow \mathsf{N}) \Rightarrow \mathsf{L}_\alpha \Rightarrow \mathsf{L}_\alpha$?

$\mathsf{map} = \lambda f.\mu F.\lambda l.t$

$t = \mathsf{match}\ l\ \mathsf{with}\ \mathsf{nil} \mapsto \mathsf{nil} \mid \mathsf{cons}\, x\, l' \mapsto \mathsf{cons}\,(f\ x)\,(F\ l')$

$\Gamma = f : \mathsf{N} \Rightarrow \mathsf{N}, F : \mathsf{L}_\alpha \Rightarrow \mathsf{L}_\alpha, l : \mathsf{L}_{\alpha+1}$

$\Gamma \vdash t : \mathsf{L}_{\alpha+1}$?

1. $\Gamma \vdash \mathsf{nil} : \mathsf{L}_{\alpha+1}$?
2. $\Gamma, x : \mathsf{N}, l' : \mathsf{L}_\alpha \vdash \mathsf{cons}\,(f\ x)\,(F\ l') : \mathsf{L}_{\alpha+1}$?
   $\Gamma, x : \mathsf{N}, l' : \mathsf{L}_\alpha \vdash F\ l' : \mathsf{L}_\alpha$?

Extensions of Hugues-Pareto-Sabry

- termination of arbitrary terms [Amadio-Coupet 1997]
  using an interpretation in reducibility candidates
- higher-order data types [Barthe-Frade-Giménez-Pinto 2004]
- system Fω [Abel 2004]
- dependent types and rewriting [B. 2004]
- calculus of constructions [Barthe-Grégoire-Pastawski 2006]
- ...

Extensions of Hugues-Pareto-Sabry

- termination of arbitrary terms [Amadio-Coupet 1997]
  using an interpretation in reducibility candidates
- higher-order data types [Barthe-Frade-Giménez-Pinto 2004]
- system Fω [Abel 2004]
- dependent types and rewriting [B. 2004]
- calculus of constructions [Barthe-Grégoire-Pastawski 2006]
- ...

But:

- size algebra always the same ($\mathtt{s}$ and sometimes $+$)
  [$\mathtt{s}$ is already enough to subsume Coq termination checker]

Extensions of Hugues-Pareto-Sabry

- termination of arbitrary terms [Amadio-Coupet 1997]
  using an interpretation in reducibility candidates
- higher-order data types [Barthe-Frade-Giménez-Pinto 2004]
- system Fω [Abel 2004]
- dependent types and rewriting [B. 2004]
- calculus of constructions [Barthe-Grégoire-Pastawski 2006]
- ...

But:

- size algebra always the same ($\mathtt{s}$ and sometimes $+$)
  [$\mathtt{s}$ is already enough to subsume Coq termination checker]
- notion of size always the same (height)

Extensions of Hugues-Pareto-Sabry

- termination of arbitrary terms [Amadio-Coupet 1997]
  using an interpretation in reducibility candidates
- higher-order data types [Barthe-Frade-Giménez-Pinto 2004]
- system Fω [Abel 2004]
- dependent types and rewriting [B. 2004]
- calculus of constructions [Barthe-Grégoire-Pastawski 2006]
- ...

But:

- size algebra always the same ($\mathtt{s}$ and sometimes $+$)
  [$\mathtt{s}$ is already enough to subsume Coq termination checker]
- notion of size always the same (height)
- for fixpoint-based function definitions only

In the following

1. extension to user-defined notions of size

In the following

1. extension to user-defined notions of size
2. extension to rewriting-based function definitions

In the following

1. extension to user-defined notions of size
2. extension to rewriting-based function definitions
3. extension to arbitrary size algebra

In the following

1. extension to user-defined notions of size
2. extension to rewriting-based function definitions
3. extension to arbitrary size algebra
4. general type-checking algorithm

In the following

1. extension to user-defined notions of size
2. extension to rewriting-based function definitions
3. extension to arbitrary size algebra
4. general type-checking algorithm
5. completeness in the successor algebra

Outline

The notion of size

$$\|t\|^{\mathrm{default}} = \text{smallest ordinal } \mathfrak{a} \text{ such that } t \in [\![\mathsf{A}]\!]^{\mathrm{default}}_{\mathfrak{a}}$$

$$\text{where } [\![\mathsf{A}]\!]^{\mathrm{default}}_{\mathfrak{a}} = \begin{cases} \bot & \text{if } \mathfrak{a} = 0 \\ F^{\mathsf{A}}([\![\mathsf{A}]\!]_{\mathfrak{b}}) & \text{if } \mathfrak{a} = \mathfrak{b} + 1 \\ \bigcup_{\mathfrak{b}<\mathfrak{a}} [\![\mathsf{A}]\!]_{\mathfrak{b}} & \text{if } \mathfrak{a} \text{ limit ordinal} > 0 \end{cases}$$

The notion of size

$$\|t\|^{\text{default}} = \text{smallest ordinal } \mathfrak{a} \text{ such that } t \in [\![A]\!]^{\text{default}}_{\mathfrak{a}}$$

$$\text{where } [\![A]\!]^{\text{default}}_{\mathfrak{a}} = \begin{cases} \bot & \text{if } \mathfrak{a} = 0 \\ F^A([\![A]\!]_{\mathfrak{b}}) & \text{if } \mathfrak{a} = \mathfrak{b} + 1 \\ \bigcup_{\mathfrak{b}<\mathfrak{a}} [\![A]\!]_{\mathfrak{b}} & \text{if } \mathfrak{a} \text{ limit ordinal} > 0 \end{cases}$$

remark: $[\![A]\!]^{\text{default}}$ is a particular *stratification* of $[\![A]\!]$

i.e. an increasing sequence $[\![A]\!]_0 \subseteq [\![A]\!]_1 \subseteq \ldots \subseteq [\![A]\!]_{\mathfrak{h}} = [\![A]\!]$

The notion of size

$$\|t\|^{\text{default}} = \text{smallest ordinal } \mathfrak{a} \text{ such that } t \in [\![A]\!]^{\text{default}}_{\mathfrak{a}}$$

$$\text{where } [\![A]\!]^{\text{default}}_{\mathfrak{a}} = \begin{cases} \bot & \text{if } \mathfrak{a} = 0 \\ F^A([\![A]\!]_{\mathfrak{b}}) & \text{if } \mathfrak{a} = \mathfrak{b} + 1 \\ \bigcup_{\mathfrak{b}<\mathfrak{a}} [\![A]\!]_{\mathfrak{b}} & \text{if } \mathfrak{a} \text{ limit ordinal} > 0 \end{cases}$$

remark: $[\![A]\!]^{\text{default}}$ is a particular *stratification* of $[\![A]\!]$

i.e. an increasing sequence $[\![A]\!]_0 \subseteq [\![A]\!]_1 \subseteq \ldots \subseteq [\![A]\!]_{\mathfrak{h}} = [\![A]\!]$

any other stratification is fine!

Defining stratifications from constructor-size functions

we will restrict our attention to stratifications given by a function $\Sigma^{\mathsf{c}} : \mathfrak{h}^n \to \mathfrak{h}$ for each constructor c of arity n so that:

$$\|\mathsf{c}t_1 \dots t_n\|^{\Sigma} = \Sigma^{\mathsf{c}}(\|t_1\|^{\Sigma}, \dots \|t_n\|^{\Sigma})$$

examples:

- for lists, $\Sigma^{\mathsf{cons}}_{\mathrm{default}}(\mathfrak{a}, \mathfrak{b}) = \mathfrak{b} + 1$ gives the length
- for trees, $\Sigma^{\mathsf{node}}_{\mathrm{default}}(\mathfrak{a}, \mathfrak{b}) = \max(\mathfrak{a}, \mathfrak{b}) + 1$ gives the height

Other notions of size

example 1: for trees, the number of nodes is given by

$$\Sigma^{\mathsf{node}}(\mathfrak{a}, \mathfrak{b}) = \mathfrak{a} + \mathfrak{b} + 1$$

Other notions of size

example 1: for trees, the number of nodes is given by

$$\Sigma^{\mathsf{node}}(\mathfrak{a}, \mathfrak{b}) = \mathfrak{a} + \mathfrak{b} + 1$$

example 2: normalization of if-expressions

$$\begin{array}{rcl} \mathsf{nm}\ \mathsf{at} & \to & \mathsf{at} \\ \mathsf{nm}\ (\mathsf{if}\ \mathsf{at}\ y\ z) & \to & \mathsf{if}\ \mathsf{at}\ (\mathsf{nm}\ y)\ (\mathsf{nm}\ z) \\ \mathsf{nm}\ (\mathsf{if}\ (\mathsf{if}\ u\ v\ w)\ y\ z) & \to & \mathsf{nm}\ (\mathsf{if}\ u\ (\mathsf{nm}\ (\mathsf{if}\ v\ y\ z))\ (\mathsf{nm}\ (\mathsf{if}\ w\ y\ z))) \end{array}$$

Other notions of size

example 1: for trees, the number of nodes is given by

$$\Sigma^{\mathsf{node}}(\mathfrak{a}, \mathfrak{b}) = \mathfrak{a} + \mathfrak{b} + 1$$

example 2: normalization of if-expressions

$$\begin{array}{rcl} \mathsf{nm}\ \mathsf{at} & \to & \mathsf{at} \\ \mathsf{nm}\ (\mathsf{if}\ \mathsf{at}\ y\ z) & \to & \mathsf{if}\ \mathsf{at}\ (\mathsf{nm}\ y)\ (\mathsf{nm}\ z) \\ \mathsf{nm}\ (\mathsf{if}\ (\mathsf{if}\ u\ v\ w)\ y\ z) & \to & \mathsf{nm}\ (\mathsf{if}\ u\ (\mathsf{nm}\ (\mathsf{if}\ v\ y\ z))\ (\mathsf{nm}\ (\mathsf{if}\ w\ y\ z))) \end{array}$$

can be proved terminating by taking:

$$\Sigma^{\mathsf{if}}(\mathfrak{a}, \mathfrak{b}, \mathfrak{c}) = (\mathfrak{a} + 1)(\mathfrak{b} + \mathfrak{c} + 3)$$

User-defined notions of size

question: for which functions Σ^c can we define a stratification in reducibility candidates $[\![A]\!]^{\Sigma}_{a}$ so that $\|c\vec{t}\|^{\Sigma} = \Sigma^c(\|\vec{t}\|^{\Sigma})$?

User-defined notions of size

question: for which functions Σ^c can we define a stratification in reducibility candidates $[\![A]\!]^{\Sigma}_{a}$ so that $\|c\vec{t}\|^{\Sigma} = \Sigma^c(\|\vec{t}\|^{\Sigma})$?

answer: if Σ^c is both

- *monotone* wrt every argument:

$$\Sigma^c(\vec{a}) \leq \Sigma^c(\vec{b}) \text{ if } \vec{a} \leq_{\text{prod}} \vec{b}$$

User-defined notions of size

question: for which functions Σ^c can we define a stratification in reducibility candidates $[\![A]\!]^{\Sigma}_{\mathfrak{a}}$ so that $\|c\vec{t}\|^{\Sigma} = \Sigma^c(\|\vec{t}\|^{\Sigma})$?

answer: if Σ^c is both

- *monotone* wrt every argument:

$$\Sigma^c(\vec{\mathfrak{a}}) \leq \Sigma^c(\vec{\mathfrak{b}}) \text{ if } \vec{\mathfrak{a}} \leq_{\text{prod}} \vec{\mathfrak{b}}$$

- *strictly extensive* wrt recursive arguments:

$$\mathfrak{a}_i < \Sigma^c(\vec{\mathfrak{a}}) \text{ if the } i\text{-th argument of c is recursive}$$

User-defined notions of size

question: for which functions Σ^{c} can we define a stratification in reducibility candidates $[\![\mathsf{A}]\!]^{\Sigma}_{\mathfrak{a}}$ so that $\|\mathsf{c}\vec{t}\|^{\Sigma} = \Sigma^{\mathsf{c}}(\|\vec{t}\|^{\Sigma})$?

answer: if Σ^{c} is both

- *monotone* wrt every argument:
$$\Sigma^{\mathsf{c}}(\vec{\mathfrak{a}}) \leq \Sigma^{\mathsf{c}}(\vec{\mathfrak{b}}) \text{ if } \vec{\mathfrak{a}} \leq_{\mathrm{prod}} \vec{\mathfrak{b}}$$
- *strictly extensive* wrt recursive arguments:
$$\mathfrak{a}_i < \Sigma^{\mathsf{c}}(\vec{\mathfrak{a}}) \text{ if the } i\text{-th argument of c is recursive}$$

$$[\![\mathsf{A}]\!]^{\Sigma}_{\mathfrak{a}+1} = \{t \in \mathrm{SN} \mid t \to^* \mathsf{c}\vec{t} \Rightarrow t_i \in [\![\tau_i^{\mathsf{c}}]\!]^{[\![\mathsf{A}]\!]^{\Sigma}_{\mathfrak{a}}}_{\mathsf{A}} \wedge \Sigma^{\mathsf{c}}(\|\vec{t}\|^{\Sigma}) \leq \mathfrak{a}+1\}$$

Outline

A more general framework

size algebra:

- a first-order term algebra $\mathtt{A}$
- a quasi-ordering $\leq_{\mathtt{A}}$ stable by substitution
- an interpretation $[\![\ ]\!]$ in ordinals

A more general framework

size algebra:

- a first-order term algebra $\mathtt{A}$
- a quasi-ordering $\leq_{\mathtt{A}}$ stable by substitution
- an interpretation $[\![\ ]\!]$ in ordinals

examples:

- successor algebra: $\mathtt{s}$ (already enough to subsume Coq)
- max-successor algebra: $\mathtt{max}$, $\mathtt{s}$
- max-plus algebra: max, $\mathtt{s}$, $+$

Outline

Annotation of constructors

$$\mathsf{c} : T_1 \Rightarrow \ldots \Rightarrow T_n \Rightarrow \mathsf{A}$$
$$\rightsquigarrow$$
$$\mathsf{c} : \mathrm{Annot}(T_1, \mathsf{A}, \alpha_1) \Rightarrow \ldots \Rightarrow \mathrm{Annot}(T_n, \mathsf{A}, \alpha_n) \Rightarrow \mathsf{A}_{\sigma(\alpha_1,\ldots,\alpha_n)}$$

- $\sigma(\alpha_1, \ldots, \alpha_n)$ *monotone* wrt every argument
- $\sigma(\alpha_1, \ldots, \alpha_n)$ *strictly extensive* wrt recursive arguments

Annotation of constructors

$$\mathsf{c} : T_1 \Rightarrow \ldots \Rightarrow T_n \Rightarrow \mathsf{A}$$
$$\rightsquigarrow$$
$$\mathsf{c} : \mathrm{Annot}(T_1, \mathsf{A}, \alpha_1) \Rightarrow \ldots \Rightarrow \mathrm{Annot}(T_n, \mathsf{A}, \alpha_n) \Rightarrow \mathsf{A}_{\sigma(\alpha_1,\ldots,\alpha_n)}$$

- $\sigma(\alpha_1, \ldots, \alpha_n)$ *monotone* wrt every argument
- $\sigma(\alpha_1, \ldots, \alpha_n)$ *strictly extensive* wrt recursive arguments

$$\Sigma^{\mathsf{c}}(\mathfrak{a}_1, \ldots, \mathfrak{a}_n) = [\![\sigma(\alpha_1, \ldots, \alpha_n)]\!]\mu$$
$$\text{where } \mu(\alpha_i) = \max\{\mathfrak{a}_j \mid \alpha_j = \alpha_i\}$$

Annotation of constructors

$\mathsf{c} : T_1 \Rightarrow \ldots \Rightarrow T_n \Rightarrow \mathsf{A}$

$\rightsquigarrow$

$\mathsf{c} : \mathrm{Annot}(T_1, \mathsf{A}, \alpha_1) \Rightarrow \ldots \Rightarrow \mathrm{Annot}(T_n, \mathsf{A}, \alpha_n) \Rightarrow \mathsf{A}_{\sigma(\alpha_1,\ldots,\alpha_n)}$

- $\sigma(\alpha_1, \ldots, \alpha_n)$ *monotone* wrt every argument
- $\sigma(\alpha_1, \ldots, \alpha_n)$ *strictly extensive* wrt recursive arguments

$$\Sigma^{\mathsf{c}}(\mathfrak{a}_1, \ldots, \mathfrak{a}_n) = [\![\sigma(\alpha_1, \ldots, \alpha_n)]\!]\mu$$
$$\text{where } \mu(\alpha_i) = \max\{\mathfrak{a}_j \mid \alpha_j = \alpha_i\}$$

examples:

- $\mathsf{node} : \mathsf{T}_\alpha \Rightarrow \mathsf{T}_\alpha \Rightarrow \mathsf{T}_{\mathtt{s}\,\alpha}$ in the successor algebra
- $\mathsf{node} : \mathsf{T}_\alpha \Rightarrow \mathsf{T}_\beta \Rightarrow \mathsf{T}_{\mathtt{s}\,(\mathtt{max}\,\alpha\,\beta)}$ in the max-successor algebra
- $\mathsf{lim} : (\mathsf{N} \Rightarrow \mathsf{O}_\alpha) \Rightarrow \mathsf{O}_{\mathtt{s}\,\alpha}$ in the successor algebra

Termination conditions 1-3

$$\mathsf{f} : \underbrace{\mathsf{A}_1 \Rightarrow \mathsf{A}_2 \Rightarrow \ldots \Rightarrow \mathsf{A}_m}_{\text{termination arguments}} \Rightarrow \underbrace{T_{m+1} \Rightarrow \ldots \Rightarrow T_n}_{\text{parameters}} \Rightarrow \mathsf{A}_{\sigma(\alpha_1,\ldots,\alpha_m)}$$

$$\rightsquigarrow$$

$$\mathsf{f} : (\mathsf{A}_1)_{\alpha_1} \Rightarrow \ldots \Rightarrow (\mathsf{A}_m)_{\alpha_m} \Rightarrow T_{m+1} \Rightarrow \ldots \Rightarrow T_n \Rightarrow \mathsf{A}_{\sigma(\alpha_1,\ldots,\alpha_m)}$$

Termination conditions 1-3

$$\mathsf{f} : \underbrace{\mathsf{A}_1 \Rightarrow \mathsf{A}_2 \Rightarrow \ldots \Rightarrow \mathsf{A}_m}_{\text{termination arguments}} \Rightarrow \underbrace{T_{m+1} \Rightarrow \ldots \Rightarrow T_n}_{\text{parameters}} \Rightarrow \mathsf{A}_{\sigma(\alpha_1,\ldots,\alpha_m)}$$

$$\rightsquigarrow$$

$$\mathsf{f} : (\mathsf{A}_1)_{\alpha_1} \Rightarrow \ldots \Rightarrow (\mathsf{A}_m)_{\alpha_m} \Rightarrow T_{m+1} \Rightarrow \ldots \Rightarrow T_n \Rightarrow \mathsf{A}_{\sigma(\alpha_1,\ldots,\alpha_m)}$$

1. Monotony of $\sigma(\alpha_1, \ldots, \alpha_m)$ wrt $\alpha_1, \ldots, \alpha_m$

Termination conditions 1-3

$$\mathsf{f} : \underbrace{\mathsf{A}_1 \Rightarrow \mathsf{A}_2 \Rightarrow \ldots \Rightarrow \mathsf{A}_m}_{\text{termination arguments}} \Rightarrow \underbrace{T_{m+1} \Rightarrow \ldots \Rightarrow T_n}_{\text{parameters}} \Rightarrow \mathsf{A}_{\sigma(\alpha_1,\ldots,\alpha_m)}$$

$\rightsquigarrow$

$$\mathsf{f} : (\mathsf{A}_1)_{\alpha_1} \Rightarrow \ldots \Rightarrow (\mathsf{A}_m)_{\alpha_m} \Rightarrow T_{m+1} \Rightarrow \ldots \Rightarrow T_n \Rightarrow \mathsf{A}_{\sigma(\alpha_1,\ldots,\alpha_m)}$$

1. Monotony of $\sigma(\alpha_1, \ldots, \alpha_m)$ wrt $\alpha_1, \ldots, \alpha_m$

for each rule $\mathsf{f}\, l_1 \ldots l_n \to r$, there are Γ and $\varphi = \{(\vec{\alpha}, \vec{a})\}$ such that:

2. Accessibility: for every $x \in \mathrm{Var}(r)$, either:

Termination conditions 1-3

$$\mathsf{f} : \underbrace{\mathsf{A}_1 \Rightarrow \mathsf{A}_2 \Rightarrow \ldots \Rightarrow \mathsf{A}_m}_{\text{termination arguments}} \Rightarrow \underbrace{T_{m+1} \Rightarrow \ldots \Rightarrow T_n}_{\text{parameters}} \Rightarrow \mathsf{A}_{\sigma(\alpha_1,\ldots,\alpha_m)}$$

$\rightsquigarrow$

$$\mathsf{f} : (\mathsf{A}_1)_{\alpha_1} \Rightarrow \ldots \Rightarrow (\mathsf{A}_m)_{\alpha_m} \Rightarrow T_{m+1} \Rightarrow \ldots \Rightarrow T_n \Rightarrow \mathsf{A}_{\sigma(\alpha_1,\ldots,\alpha_m)}$$

1. Monotony of $\sigma(\alpha_1, \ldots, \alpha_m)$ wrt $\alpha_1, \ldots, \alpha_m$

for each rule $\mathsf{f}\, l_1 \ldots l_n \to r$, there are Γ and $\varphi = \{(\vec{\alpha}, \vec{a})\}$ such that:

2. Accessibility: for every $x \in \mathrm{Var}(r)$, either:
   - x is a parameter: $x = l_i$, $x\Gamma = T_i$ and $i > m$

Termination conditions 1-3

$$\mathsf{f} : \underbrace{\mathsf{A}_1 \Rightarrow \mathsf{A}_2 \Rightarrow \ldots \Rightarrow \mathsf{A}_m}_{\text{termination arguments}} \Rightarrow \underbrace{T_{m+1} \Rightarrow \ldots \Rightarrow T_n}_{\text{parameters}} \Rightarrow \mathsf{A}_{\sigma(\alpha_1,\ldots,\alpha_m)}$$

$\rightsquigarrow$

$$\mathsf{f} : (\mathsf{A}_1)_{\alpha_1} \Rightarrow \ldots \Rightarrow (\mathsf{A}_m)_{\alpha_m} \Rightarrow T_{m+1} \Rightarrow \ldots \Rightarrow T_n \Rightarrow \mathsf{A}_{\sigma(\alpha_1,\ldots,\alpha_m)}$$

1. Monotony of $\sigma(\alpha_1, \ldots, \alpha_m)$ wrt $\alpha_1, \ldots, \alpha_m$

for each rule $\mathsf{f} l_1 \ldots l_n \to r$, there are Γ and $\varphi = \{(\vec{\alpha}, \vec{a})\}$ such that:

2. Accessibility: for every $x \in \mathrm{Var}(r)$, either:
   - x is a parameter: $x = l_i$, $x\Gamma = T_i$ and $i > m$
   - x is a *positive* subterm of a termination argument: $x \trianglelefteq l_i$, $x\Gamma = \mathrm{Annot}(|x\Gamma|, \mathsf{A}^i, \beta_x)$ and $i \leq m$
     because not every subterm of a computable term is computable

Termination conditions 1-3

$$\mathsf{f} : \underbrace{\mathsf{A}_1 \Rightarrow \mathsf{A}_2 \Rightarrow \ldots \Rightarrow \mathsf{A}_m}_{\text{termination arguments}} \Rightarrow \underbrace{T_{m+1} \Rightarrow \ldots \Rightarrow T_n}_{\text{parameters}} \Rightarrow \mathsf{A}_{\sigma(\alpha_1,\ldots,\alpha_m)}$$

$\rightsquigarrow$

$$\mathsf{f} : (\mathsf{A}_1)_{\alpha_1} \Rightarrow \ldots \Rightarrow (\mathsf{A}_m)_{\alpha_m} \Rightarrow T_{m+1} \Rightarrow \ldots \Rightarrow T_n \Rightarrow \mathsf{A}_{\sigma(\alpha_1,\ldots,\alpha_m)}$$

1. Monotony of $\sigma(\alpha_1, \ldots, \alpha_m)$ wrt $\alpha_1, \ldots, \alpha_m$

for each rule $\mathsf{f} l_1 \ldots l_n \to r$, there are Γ and $\varphi = \{(\vec{\alpha}, \vec{a})\}$ such that:

2. Accessibility: for every $x \in \mathrm{Var}(r)$, either:
   - x is a parameter: $x = l_i$, $x\Gamma = T_i$ and $i > m$
   - x is a *positive* subterm of a termination argument: $x \trianglelefteq l_i$, $x\Gamma = \mathrm{Annot}(|x\Gamma|, \mathsf{A}^i, \beta_x)$ and $i \leq m$
     because not every subterm of a computable term is computable
3. Minimality of φ: $(\forall\theta)(\exists\mu)[\|\vec{l}\theta\| = [\![\vec{\alpha}\varphi]\!]\mu$ and $(\forall x)\, \|x\theta\| \leq \beta_x\mu]$

Termination condition 4

4. Subject-reduction and decreasingness: $\Gamma \vdash^{\mathsf{f}}_{\varphi} r : \mathsf{A}_{\sigma(\alpha_1,\ldots,\alpha_m)}\varphi$

Θ: annotated types of constructor and function symbols
$\leq_{\mathcal{F}}$: precedence on function symbols

$$\text{(app-decr)}\ \frac{\begin{array}{c}(\forall i)\,\Gamma \vdash^{\mathsf{f}}_{\varphi} t_i : T_i \\ (h, \vec{T} \Rightarrow T) \in \Gamma \cup \Theta\psi \\ h <_{\mathcal{F}} \mathsf{f} \vee (h \simeq_{\mathcal{F}} \mathsf{f} \wedge \psi <_{\mathtt{A}} \varphi)\end{array}}{\Gamma \vdash^{\mathsf{f}}_{\varphi} h\vec{t} : T}$$

$$\text{(lam)}\ \frac{\Gamma, x : U \vdash^{\mathsf{f}}_{\varphi} v : V}{\Gamma \vdash^{\mathsf{f}}_{\varphi} \lambda x^{U}.v : U \Rightarrow V} \qquad \text{(sub)}\ \frac{\Gamma \vdash^{\mathsf{f}}_{\varphi} t : T \quad T \leq U}{\Gamma \vdash^{\mathsf{f}}_{\varphi} t : U}$$

Example in the successor algebra

$$\begin{array}{rcl} \mathsf{map}\ f\ \mathsf{nil} & \to_{\mathcal{R}} & \mathsf{nil} \\ \mathsf{map}\ f\ (\mathsf{cons}\ x\ l) & \to_{\mathcal{R}} & \mathsf{cons}\ (f\ x)\ (\mathsf{map}\ f\ l) \end{array}$$

Example in the successor algebra

$$\begin{array}{rcl} \mathsf{map}\ f\ \mathsf{nil} & \to_{\mathcal{R}} & \mathsf{nil} \\ \mathsf{map}\ f\ (\mathsf{cons}\ x\ l) & \to_{\mathcal{R}} & \mathsf{cons}\ (f\ x)\ (\mathsf{map}\ f\ l) \end{array}$$

- $\mathsf{nil} : \mathsf{L}$, $\mathsf{cons} : \mathsf{N} \Rightarrow \mathsf{L}_\alpha \Rightarrow \mathsf{L}_{\mathbf{s}\,\alpha}$

Example in the successor algebra

$$\begin{array}{rcl} \mathsf{map}\ f\ \mathsf{nil} & \to_{\mathcal{R}} & \mathsf{nil} \\ \mathsf{map}\ f\ (\mathsf{cons}\ x\ l) & \to_{\mathcal{R}} & \mathsf{cons}\ (f\ x)\ (\mathsf{map}\ f\ l) \end{array}$$

- $\mathsf{nil} : \mathsf{L}$, $\mathsf{cons} : \mathsf{N} \Rightarrow \mathsf{L}_\alpha \Rightarrow \mathsf{L}_{\mathbf{s}\,\alpha}$
  - monotony? ok

Example in the successor algebra

$$\begin{array}{rcl} \mathsf{map}\ f\ \mathsf{nil} & \to_{\mathcal{R}} & \mathsf{nil} \\ \mathsf{map}\ f\ (\mathsf{cons}\ x\ l) & \to_{\mathcal{R}} & \mathsf{cons}\ (f\ x)\ (\mathsf{map}\ f\ l) \end{array}$$

- $\mathsf{nil} : \mathsf{L}$, $\mathsf{cons} : \mathsf{N} \Rightarrow \mathsf{L}_\alpha \Rightarrow \mathsf{L}_{\mathbf{s}\,\alpha}$
  - monotony? ok
  - strict extensivity? ok

Example in the successor algebra

$$\begin{array}{rcl} \mathsf{map}\ f\ \mathsf{nil} & \to_{\mathcal{R}} & \mathsf{nil} \\ \mathsf{map}\ f\ (\mathsf{cons}\ x\ l) & \to_{\mathcal{R}} & \mathsf{cons}\ (f\ x)\ (\mathsf{map}\ f\ l) \end{array}$$

- $\mathsf{nil} : \mathsf{L},\ \mathsf{cons} : \mathsf{N} \Rightarrow \mathsf{L}_\alpha \Rightarrow \mathsf{L}_{\mathbf{s}\,\alpha}$
  - monotony? ok
  - strict extensivity? ok
- $\mathsf{map} : (\mathsf{N} \Rightarrow \mathsf{N}) \Rightarrow \mathsf{L}_\alpha \Rightarrow \mathsf{L}_\alpha$

Example in the successor algebra

$$\begin{array}{rcl} \mathsf{map}\ f\ \mathsf{nil} & \to_{\mathcal{R}} & \mathsf{nil} \\ \mathsf{map}\ f\ (\mathsf{cons}\ x\ l) & \to_{\mathcal{R}} & \mathsf{cons}\ (f\ x)\ (\mathsf{map}\ f\ l) \end{array}$$

- $\mathsf{nil} : \mathsf{L},\ \mathsf{cons} : \mathsf{N} \Rightarrow \mathsf{L}_\alpha \Rightarrow \mathsf{L}_{\mathbf{s}\,\alpha}$
  - monotony? ok
  - strict extensivity? ok
- $\mathsf{map} : (\mathsf{N} \Rightarrow \mathsf{N}) \Rightarrow \mathsf{L}_\alpha \Rightarrow \mathsf{L}_\alpha$
  - monotony? ok

Example in the successor algebra

$$\begin{array}{rcl} \mathsf{map}\ f\ \mathsf{nil} & \to_{\mathcal{R}} & \mathsf{nil} \\ \mathsf{map}\ f\ (\mathsf{cons}\ x\ l) & \to_{\mathcal{R}} & \mathsf{cons}\ (f\ x)\ (\mathsf{map}\ f\ l) \end{array}$$

- $\mathsf{nil} : \mathsf{L},\ \mathsf{cons} : \mathsf{N} \Rightarrow \mathsf{L}_\alpha \Rightarrow \mathsf{L}_{\mathbf{s}\,\alpha}$
  - monotony? ok
  - strict extensivity? ok
- $\mathsf{map} : (\mathsf{N} \Rightarrow \mathsf{N}) \Rightarrow \mathsf{L}_\alpha \Rightarrow \mathsf{L}_\alpha$
  - monotony? ok

rule 2: $\Gamma = f : \mathsf{N} \Rightarrow \mathsf{N}, x : \mathsf{N}, l : \mathsf{L}_{\beta_l}$ and $\varphi = \{(\alpha, \mathbf{s}\,\beta_l)\}$

- accessibility? ok

Example in the successor algebra

$$\begin{array}{rcl} \mathsf{map}\ f\ \mathsf{nil} & \to_{\mathcal{R}} & \mathsf{nil} \\ \mathsf{map}\ f\ (\mathsf{cons}\ x\ l) & \to_{\mathcal{R}} & \mathsf{cons}\ (f\ x)\ (\mathsf{map}\ f\ l) \end{array}$$

- $\mathsf{nil} : \mathsf{L}$, $\mathsf{cons} : \mathsf{N} \Rightarrow \mathsf{L}_\alpha \Rightarrow \mathsf{L}_{\mathbf{s}\,\alpha}$
  - monotony? ok
  - strict extensivity? ok
- $\mathsf{map} : (\mathsf{N} \Rightarrow \mathsf{N}) \Rightarrow \mathsf{L}_\alpha \Rightarrow \mathsf{L}_\alpha$
  - monotony? ok

rule 2: $\Gamma = f : \mathsf{N} \Rightarrow \mathsf{N}, x : \mathsf{N}, l : \mathsf{L}_{\beta_l}$ and $\varphi = \{(\alpha, \mathbf{s}\,\beta_l)\}$

- accessibility? ok
- minimality? ok

Example in the successor algebra

$$\begin{array}{rcl} \mathsf{map}\ f\ \mathsf{nil} & \to_{\mathcal{R}} & \mathsf{nil} \\ \mathsf{map}\ f\ (\mathsf{cons}\ x\ l) & \to_{\mathcal{R}} & \mathsf{cons}\ (f\ x)\ (\mathsf{map}\ f\ l) \end{array}$$

- $\mathsf{nil} : \mathsf{L},\ \mathsf{cons} : \mathsf{N} \Rightarrow \mathsf{L}_\alpha \Rightarrow \mathsf{L}_{\mathbf{s}\,\alpha}$
  - monotony? ok
  - strict extensivity? ok
- $\mathsf{map} : (\mathsf{N} \Rightarrow \mathsf{N}) \Rightarrow \mathsf{L}_\alpha \Rightarrow \mathsf{L}_\alpha$
  - monotony? ok

rule 2: $\Gamma = f : \mathsf{N} \Rightarrow \mathsf{N}, x : \mathsf{N}, l : \mathsf{L}_{\beta_l}$ and $\varphi = \{(\alpha, \mathbf{s}\,\beta_l)\}$

- accessibility? ok
- minimality? ok
- subject-reduction and decreasingness?
  $\Gamma \vdash^{\mathsf{map}}_{\varphi} \mathsf{cons}\,(f\,x)\,(\mathsf{map}\,f\,l) : \mathsf{L}_\alpha\varphi = \mathsf{L}_{\mathbf{s}\,\beta_l}$? $\mathsf{cons} <_{\mathcal{F}} \mathsf{map}$

Example in the successor algebra

$$\begin{array}{rcl} \mathsf{map}\ f\ \mathsf{nil} & \to_{\mathcal{R}} & \mathsf{nil} \\ \mathsf{map}\ f\ (\mathsf{cons}\ x\ l) & \to_{\mathcal{R}} & \mathsf{cons}\ (f\ x)\ (\mathsf{map}\ f\ l) \end{array}$$

- $\mathsf{nil} : \mathsf{L},\ \mathsf{cons} : \mathsf{N} \Rightarrow \mathsf{L}_\alpha \Rightarrow \mathsf{L}_{\mathbf{s}\,\alpha}$
  - monotony? ok
  - strict extensivity? ok
- $\mathsf{map} : (\mathsf{N} \Rightarrow \mathsf{N}) \Rightarrow \mathsf{L}_\alpha \Rightarrow \mathsf{L}_\alpha$
  - monotony? ok

rule 2: $\Gamma = f : \mathsf{N} \Rightarrow \mathsf{N}, x : \mathsf{N}, l : \mathsf{L}_{\beta_l}$ and $\varphi = \{(\alpha, \mathbf{s}\,\beta_l)\}$

- accessibility? ok
- minimality? ok
- subject-reduction and decreasingness?
  $\Gamma \vdash^{\mathsf{map}}_{\varphi} \mathsf{cons}\,(f\,x)\,(\mathsf{map}\,f\,l) : \mathsf{L}_\alpha\varphi = \mathsf{L}_{\mathbf{s}\,\beta_l}$? $\mathsf{cons} <_{\mathcal{F}} \mathsf{map}$
  $\Gamma \vdash^{\mathsf{map}}_{\varphi} f\,x : \mathsf{N}$

Example in the successor algebra

$$\begin{array}{rcl} \mathsf{map}\ f\ \mathsf{nil} & \to_{\mathcal{R}} & \mathsf{nil} \\ \mathsf{map}\ f\ (\mathsf{cons}\ x\ l) & \to_{\mathcal{R}} & \mathsf{cons}\ (f\ x)\ (\mathsf{map}\ f\ l) \end{array}$$

- $\mathsf{nil} : \mathsf{L},\ \mathsf{cons} : \mathsf{N} \Rightarrow \mathsf{L}_\alpha \Rightarrow \mathsf{L}_{\mathtt{s}\,\alpha}$
  - monotony? ok
  - strict extensivity? ok
- $\mathsf{map} : (\mathsf{N} \Rightarrow \mathsf{N}) \Rightarrow \mathsf{L}_\alpha \Rightarrow \mathsf{L}_\alpha$
  - monotony? ok

rule 2: $\Gamma = f : \mathsf{N} \Rightarrow \mathsf{N}, x : \mathsf{N}, l : \mathsf{L}_{\beta_l}$ and $\varphi = \{(\alpha, \mathtt{s}\,\beta_l)\}$

- accessibility? ok
- minimality? ok
- subject-reduction and decreasingness?
  $\Gamma \vdash^{\mathsf{map}}_{\varphi} \mathsf{cons}\,(f\,x)\,(\mathsf{map}\,f\,l) : \mathsf{L}_\alpha\varphi = \mathsf{L}_{\mathtt{s}\,\beta_l}$? $\mathsf{cons} <_{\mathcal{F}} \mathsf{map}$
  $\Gamma \vdash^{\mathsf{map}}_{\varphi} f\,x : \mathsf{N}$
  $\Gamma \vdash^{\mathsf{map}}_{\varphi} \mathsf{map}\,f\,l : \mathsf{L}_{\beta_l}$ since $\beta_l <_{\mathtt{A}} \mathtt{s}\,\beta_l$

Accessibility of variables

$\mathsf{c} : (\mathsf{A} \Rightarrow \mathsf{B}) \Rightarrow \mathsf{A}$, $\mathsf{f} : \mathsf{A} \Rightarrow (\mathsf{A} \Rightarrow \mathsf{B})$

$$\mathsf{f}\ (\mathsf{c}\ x) \quad \to_{\mathcal{R}} \quad x$$

$\mathsf{f}\ (\mathsf{c}(\lambda x.\mathsf{f}xx))\ (\mathsf{c}(\lambda x.\mathsf{f}xx))$
$\to_{\mathcal{R}}\ (\lambda x.\mathsf{f}xx)(\mathsf{c}(\lambda x.\mathsf{f}xx))$
$\to_{\beta}\ \mathsf{f}(\mathsf{c}(\lambda x.\mathsf{f}xx))(\mathsf{c}(\lambda x.\mathsf{f}xx))$
$\to_{\mathcal{R}} \ldots$

Monotony of $\sigma(\alpha_1, \ldots, \alpha_m)$

can always be satisfied by taking $\sigma(\alpha_1, \ldots, \alpha_m) = \infty$!

Monotony of $\sigma(\alpha_1, \ldots, \alpha_m)$

can always be satisfied by taking $\sigma(\alpha_1, \ldots, \alpha_m) = \infty$!

$$\begin{array}{rcl} \mathsf{sub}\ x\ 0 & \to & x \\ \mathsf{sub}\ 0\ y & \to & 0 \\ \mathsf{sub}\ (\mathsf{s}\ x)\ (\mathsf{s}\ y) & \to & \mathsf{sub}\ x\ y \\ \mathsf{f}\ (\mathsf{s}\ 0) & \to & \mathsf{f}\ (\mathsf{sub}\ (\mathsf{s}\ 0)\ 0) \end{array}$$

$0 : \mathsf{N}_0$, $\mathsf{sub} : \mathsf{N}_\alpha \Rightarrow \mathsf{N}_\beta \Rightarrow \mathsf{N}_{\alpha-\beta}$, $\mathsf{f} : \mathsf{N}_\alpha \Rightarrow \mathsf{N}$, $\varphi = \{(\alpha, \mathtt{s}\,0)\}$

$\vdash^{\mathsf{f}}_{\varphi} \mathsf{s}\,0 : \mathsf{N}_{\mathtt{s}\,0}$
$\vdash^{\mathsf{f}}_{\varphi} 0 : \mathsf{N}_0 \leq \mathsf{N}_{\mathtt{s}\,0}$
$\vdash^{\mathsf{f}}_{\varphi} \mathsf{sub}\ (\mathsf{s}\ 0)\ 0 : \mathsf{N}_0$
$\vdash^{\mathsf{f}}_{\varphi} \mathsf{f}\ (\mathsf{sub}\ (\mathsf{s}\ 0)\ 0) : \mathsf{N}$ since $0 < \mathtt{s}\,0$

Minimality of φ

$$\mathsf{f}\ x \quad \to_{\mathcal{R}} \quad \mathsf{f}\ x$$

$\mathsf{f} : \mathsf{N}_\alpha \Rightarrow \mathsf{N}_\alpha,\ \Gamma = x : \mathsf{N}_{\beta_x},\ \varphi = \{(\alpha, \mathtt{s}\,\beta_x)\}$

$\Gamma \vdash^{\mathsf{f}}_{\varphi} \mathsf{f}\,x : \mathsf{N}_\alpha\varphi = \mathsf{N}_{\mathtt{s}\,\beta_x}$ since $\beta_x < \mathtt{s}\,\beta_x$

Minimality of φ

easy to satisfy in an algebra with a `max` operator

Minimality of φ

easy to satisfy in an algebra with a `max` operator

but not always satisfied in the successor algebra:

Minimality of φ

easy to satisfy in an algebra with a `max` operator

but not always satisfied in the successor algebra:

- $\mathsf{c} : \mathsf{A}_\alpha \Rightarrow \mathsf{A}_{\mathtt{s}\,\alpha}$, $\mathsf{b} : \mathsf{A}_\alpha \Rightarrow \mathsf{A}_\alpha \Rightarrow \mathsf{A}_{\mathtt{s}\,\alpha}$, $\mathsf{f} : \mathsf{A}_\alpha \Rightarrow \mathsf{A}_\beta \Rightarrow T$

Minimality of φ

easy to satisfy in an algebra with a `max` operator

but not always satisfied in the successor algebra:

- $\mathsf{c} : \mathsf{A}_\alpha \Rightarrow \mathsf{A}_{\mathbf{s}\,\alpha}$, $\mathsf{b} : \mathsf{A}_\alpha \Rightarrow \mathsf{A}_\alpha \Rightarrow \mathsf{A}_{\mathbf{s}\,\alpha}$, $\mathsf{f} : \mathsf{A}_\alpha \Rightarrow \mathsf{A}_\beta \Rightarrow T$
- $\mathsf{f}\,(\mathsf{c}x)\,(\mathsf{b}\,(\mathsf{c}\,x)\,(\mathsf{c}\,y)) \;\to_{\mathcal{R}}\; t, \quad \Gamma = x : \mathsf{A}_\gamma, y : \mathsf{A}_\gamma$

Minimality of φ

easy to satisfy in an algebra with a `max` operator

but not always satisfied in the successor algebra:

- $\mathsf{c} : \mathsf{A}_\alpha \Rightarrow \mathsf{A}_{\mathtt{s}\,\alpha}$, $\mathsf{b} : \mathsf{A}_\alpha \Rightarrow \mathsf{A}_\alpha \Rightarrow \mathsf{A}_{\mathtt{s}\,\alpha}$, $\mathsf{f} : \mathsf{A}_\alpha \Rightarrow \mathsf{A}_\beta \Rightarrow T$
- $\mathsf{f}\,(\mathsf{c}x)\,(\mathsf{b}\,(\mathsf{c}\,x)\,(\mathsf{c}\,y)) \;\to_{\mathcal{R}}\; t, \quad \Gamma = x : \mathsf{A}_\gamma, y : \mathsf{A}_\gamma$
- φ minimal if, for all t, u, there is $\gamma\mu$ such that

Minimality of φ

easy to satisfy in an algebra with a `max` operator

but not always satisfied in the successor algebra:

- $\mathsf{c} : \mathsf{A}_\alpha \Rightarrow \mathsf{A}_{\mathtt{s}\,\alpha}$, $\mathsf{b} : \mathsf{A}_\alpha \Rightarrow \mathsf{A}_\alpha \Rightarrow \mathsf{A}_{\mathtt{s}\,\alpha}$, $\mathsf{f} : \mathsf{A}_\alpha \Rightarrow \mathsf{A}_\beta \Rightarrow T$
- $\mathsf{f}\,(\mathsf{c}x)\,(\mathsf{b}\,(\mathsf{c}\,x)\,(\mathsf{c}\,y)) \;\to_{\mathcal{R}}\; t, \quad \Gamma = x : \mathsf{A}_\gamma, y : \mathsf{A}_\gamma$
- φ minimal if, for all t, u, there is $\gamma\mu$ such that
  1. $\|\mathsf{c}\,t\| = \|t\| + 1 = [\![\alpha\varphi]\!]\mu$

Minimality of φ

easy to satisfy in an algebra with a `max` operator

but not always satisfied in the successor algebra:

- $\mathsf{c} : \mathsf{A}_\alpha \Rightarrow \mathsf{A}_{\mathtt{s}\,\alpha}$, $\mathsf{b} : \mathsf{A}_\alpha \Rightarrow \mathsf{A}_\alpha \Rightarrow \mathsf{A}_{\mathtt{s}\,\alpha}$, $\mathsf{f} : \mathsf{A}_\alpha \Rightarrow \mathsf{A}_\beta \Rightarrow T$
- $\mathsf{f}\,(\mathsf{c}x)\,(\mathsf{b}\,(\mathsf{c}\,x)\,(\mathsf{c}\,y)) \;\to_{\mathcal{R}}\; t, \quad \Gamma = x : \mathsf{A}_\gamma, y : \mathsf{A}_\gamma$
- φ minimal if, for all t, u, there is $\gamma\mu$ such that
  1. $\|\mathsf{c}\,t\| = \|t\| + 1 = [\![\alpha\varphi]\!]\mu$
  2. $\|\mathsf{b}\,(\mathsf{c}\,t)\,(\mathsf{c}\,u)\| = \max\{\|t\|, \|u\|\} + 2 = [\![\beta\varphi]\!]\mu$

Minimality of φ

easy to satisfy in an algebra with a `max` operator

but not always satisfied in the successor algebra:

- $\mathsf{c} : \mathsf{A}_\alpha \Rightarrow \mathsf{A}_{\mathtt{s}\,\alpha}$, $\mathsf{b} : \mathsf{A}_\alpha \Rightarrow \mathsf{A}_\alpha \Rightarrow \mathsf{A}_{\mathtt{s}\,\alpha}$, $\mathsf{f} : \mathsf{A}_\alpha \Rightarrow \mathsf{A}_\beta \Rightarrow T$
- $\mathsf{f}\,(\mathsf{c}x)\,(\mathsf{b}\,(\mathsf{c}\,x)\,(\mathsf{c}\,y)) \;\to_{\mathcal{R}}\; t, \quad \Gamma = x : \mathsf{A}_\gamma, y : \mathsf{A}_\gamma$
- φ minimal if, for all t, u, there is $\gamma\mu$ such that
  1. $\|\mathsf{c}\,t\| = \|t\| + 1 = [\![\alpha\varphi]\!]\mu$
  2. $\|\mathsf{b}\,(\mathsf{c}\,t)\,(\mathsf{c}\,u)\| = \max\{\|t\|, \|u\|\} + 2 = [\![\beta\varphi]\!]\mu$
  3. $\max\{\|t\|, \|u\|\} \leq \gamma\mu$

Minimality of φ

easy to satisfy in an algebra with a `max` operator

but not always satisfied in the successor algebra:

- $\mathsf{c} : \mathsf{A}_\alpha \Rightarrow \mathsf{A}_{\mathtt{s}\,\alpha}$, $\mathsf{b} : \mathsf{A}_\alpha \Rightarrow \mathsf{A}_\alpha \Rightarrow \mathsf{A}_{\mathtt{s}\,\alpha}$, $\mathsf{f} : \mathsf{A}_\alpha \Rightarrow \mathsf{A}_\beta \Rightarrow T$
- $\mathsf{f}\,(\mathsf{c}x)\,(\mathsf{b}\,(\mathsf{c}\,x)\,(\mathsf{c}\,y)) \;\to_{\mathcal{R}}\; t, \quad \Gamma = x : \mathsf{A}_\gamma, y : \mathsf{A}_\gamma$
- φ minimal if, for all t, u, there is $\gamma\mu$ such that
  1. $\|\mathsf{c}\,t\| = \|t\| + 1 = [\![\alpha\varphi]\!]\mu$
  2. $\|\mathsf{b}\,(\mathsf{c}\,t)\,(\mathsf{c}\,u)\| = \max\{\|t\|, \|u\|\} + 2 = [\![\beta\varphi]\!]\mu$
  3. $\max\{\|t\|, \|u\|\} \leq \gamma\mu$
- there is no minimal φ since $\|t\| \neq \max\{\|t\|, \|u\|\}$

2nd part

Outline

Decidability of $\vdash^{\mathsf{f}}_{\varphi}$?

problem: given Γ, t and T, check whether $\Gamma \vdash^{\mathsf{f}}_{\varphi} t : T$

$$\text{(app-decr)}\ \frac{\begin{array}{c}(\forall i)\,\Gamma \vdash^{\mathsf{f}}_{\varphi} t_i : T_i \\ (h, \vec{T} \Rightarrow T) \in \Gamma \cup \Theta\psi \\ h <_{\mathcal{F}} \mathsf{f} \vee (h \simeq_{\mathcal{F}} \mathsf{f} \wedge \psi <_{\mathbf{A}} \varphi)\end{array}}{\Gamma \vdash^{\mathsf{f}}_{\varphi} h\vec{t} : T}$$

Decidability of $\vdash^{\mathsf{f}}_{\varphi}$?

problem: given Γ, t and T, check whether $\Gamma \vdash^{\mathsf{f}}_{\varphi} t : T$

$$\text{(app-decr)}\ \frac{\begin{array}{c}(\forall i)\,\Gamma \vdash^{\mathsf{f}}_{\varphi} t_i : T_i \\ (h, \vec{T} \Rightarrow T) \in \Gamma \cup \Theta\psi \\ h <_{\mathcal{F}} \mathsf{f} \vee (h \simeq_{\mathcal{F}} \mathsf{f} \wedge \psi <_{\mathbf{A}} \varphi)\end{array}}{\Gamma \vdash^{\mathsf{f}}_{\varphi} h\vec{t} : T}$$

$$\rightsquigarrow \quad \text{(app)}\ \frac{\begin{array}{c}(\forall i)\,\Gamma \vdash t_i : T_i \\ (h, \vec{T} \Rightarrow T) \in \Gamma \cup \Theta\psi\end{array}}{\Gamma \vdash h\vec{t} : T}$$

1. check whether $\Gamma \vdash t : T$
2. check decreasingness condition

Decidability of $\vdash$ without subtyping [Hindley 1969]

- $\{T \mid \Gamma \vdash t : T\}$ has a smallest element (if not empty) wrt the instantiation ordering $T \sqsubseteq U$ if $(\exists\theta)\, T\theta = U$

Decidability of $\vdash$ without subtyping [Hindley 1969]

- $\{T \mid \Gamma \vdash t : T\}$ has a smallest element (if not empty) wrt the instantiation ordering $T \sqsubseteq U$ if $(\exists\theta)\, T\theta = U$
- the most general type can be computed by using unification:

$$\text{(inf-lam)}\ \frac{\Gamma, x : U \vdash v \uparrow V}{\Gamma \vdash \lambda x^U.v \uparrow U \Rightarrow V}$$

$$\text{(inf-app)}\ \frac{\begin{array}{c}(\forall i)\, \Gamma \vdash t_i \uparrow U_i \\ (h, \vec{V} \Rightarrow V) \in \Gamma \cup \Theta \\ \eta = \mathrm{mgs}(\{U_1 =^? V_1, \ldots, U_n =^? V_n\})\end{array}}{\Gamma \vdash h\vec{t} \uparrow V\eta}$$

Decidability of $\vdash$ with subtyping?

extend Hindley's method by assuming:

Decidability of $\vdash$ with subtyping?

extend Hindley's method by assuming:

1. every solvable subtyping problem has a most general solution wrt subtyping composed with instanciation: $T \sqsubseteq U$ if $(\exists\theta)T\theta \leq U$

Decidability of $\vdash$ with subtyping?

extend Hindley's method by assuming:

1. every solvable subtyping problem has a most general solution wrt subtyping composed with instanciation: $T \sqsubseteq U$ if $(\exists\theta)T\theta \leq U$
2. there is an algorithm for deciding whether a subtyping problem is solvable and, if so, computing its most general solution

Decidability of $\vdash$ with subtyping?

extend Hindley's method by assuming:

1. every solvable subtyping problem has a most general solution wrt subtyping composed with instanciation: $T \sqsubseteq U$ if $(\exists\theta)\, T\theta \leq U$
2. there is an algorithm for deciding whether a subtyping problem is solvable and, if so, computing its most general solution

$$\text{(inf-lam)}\ \frac{\Gamma, x : U \vdash v \uparrow V}{\Gamma \vdash \lambda x^{U}.v \uparrow U \Rightarrow V}$$

$$\text{(inf-app)}\ \frac{\begin{array}{c}(\forall i)\, \Gamma \vdash t_i \uparrow U_i \\ (h, \vec{V} \Rightarrow V) \in \Gamma \cup \Theta \\ \eta = \mathrm{mgs}(\{U_1 \leq^{?} V_1, \ldots, U_n \leq^{?} V_n\})\end{array}}{\Gamma \vdash h\vec{t} \uparrow V\eta}$$

Reducing subtyping problems $\leq^?$ to size problems $\leq_{\mathtt{A}}^{\infty ?}$

$$\text{(sub-refl)}\ \frac{}{T \leq T} \qquad \text{(sub-trans)}\ \frac{T \leq U \quad U \leq V}{T \leq V}$$

$$\text{(sub-base)}\ \frac{a \leq_{\mathtt{A}}^{\infty} b}{L_a \leq L_b} \qquad \text{(sub-arrow)}\ \frac{T' \leq T \quad U \leq U'}{T \Rightarrow U \leq T' \Rightarrow U'}$$

Reducing subtyping problems $\leq^?$ to size problems $\leq^{\infty?}_{\mathtt{A}}$

$$\text{(sub-refl)}\ \frac{}{T \leq T} \qquad \text{(sub-trans)}\ \frac{T \leq U \quad U \leq V}{T \leq V}$$

$$\text{(sub-base)}\ \frac{a \leq^{\infty}_{\mathtt{A}} b}{L_a \leq L_b} \qquad \text{(sub-arrow)}\ \frac{T' \leq T \quad U \leq U'}{T \Rightarrow U \leq T' \Rightarrow U'}$$

theorem: (sub-refl) and (sub-trans) are redundant

Reducing subtyping problems $\leq^?$ to size problems $\leq_{\mathtt{A}}^{\infty?}$

$$\text{(sub-refl)}\ \frac{}{T \leq T} \qquad \text{(sub-trans)}\ \frac{T \leq U \quad U \leq V}{T \leq V}$$

$$\text{(sub-base)}\ \frac{a \leq_{\mathtt{A}}^{\infty} b}{L_a \leq L_b} \qquad \text{(sub-arrow)}\ \frac{T' \leq T \quad U \leq U'}{T \Rightarrow U \leq T' \Rightarrow U'}$$

theorem: (sub-refl) and (sub-trans) are redundant

hence, any subtyping problem $\{T_1 \leq^? U_1, \ldots, T_n \leq^? U_n\}$
is equivalent to a size problem $\{a_1 \leq_{\mathtt{A}}^{\infty?} b_1, \ldots, a_k \leq_{\mathtt{A}}^{\infty?} b_k\}$

Outline

Successor algebra

1. function symbols:
   - successor $\mathtt{s}$
   - constants $\mathtt{c}, \mathtt{d}, \ldots$ for the size variables used in Γ

every term a is of the form ∞, $\mathtt{s}^k\alpha$ or $\mathtt{s}^k\mathtt{c}$
its base symbol $\mathrm{base}(a)$ is ∞, α and $\mathtt{c}$ respectively

Successor algebra

1. function symbols:
   - successor $\mathtt{s}$
   - constants $\mathtt{c}, \mathtt{d}, \ldots$ for the size variables used in Γ

   every term a is of the form ∞, $\mathtt{s}^k\alpha$ or $\mathtt{s}^k\mathtt{c}$
   its base symbol $\mathrm{base}(a)$ is ∞, α and $\mathtt{c}$ respectively

2. ordering:

$$\frac{}{a \leq_{\mathtt{A}} a} \qquad \frac{a <_{\mathtt{A}} b}{a \leq_{\mathtt{A}} b} \qquad \frac{}{a <_{\mathtt{A}} \mathtt{s}\, a} \qquad \frac{a <_{\mathtt{A}} b \quad b <_{\mathtt{A}} c}{a <_{\mathtt{A}} c}$$

Graph representation of a problem in the successor algebra

remark 1: every constraint has at most 2 variables

$$\mathbf{s}^k a \leq^? \mathbf{s}^l b \in P \quad \Leftrightarrow \quad a \xrightarrow{k-l} b \in \mathrm{Graph}(P)$$

Graph representation of a problem in the successor algebra

remark 1: every constraint has at most 2 variables

$$\mathtt{s}^k a \leq^? \mathtt{s}^l b \in P \quad \Leftrightarrow \quad a \xrightarrow{k-l} b \in \mathrm{Graph}(P)$$

example:

$\mathtt{s}\alpha \leq^{?}_{\mathtt{A}}{}^{\infty} \beta$

Graph representation of a problem in the successor algebra

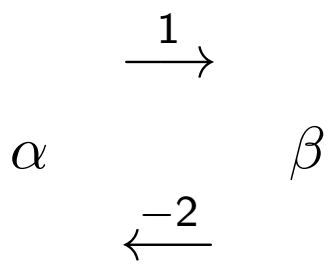
remark 1: every constraint has at most 2 variables

$$\mathtt{s}^k a \leq^? \mathtt{s}^l b \in P \quad \Leftrightarrow \quad a \xrightarrow{k-l} b \in \mathrm{Graph}(P)$$

example:

$\mathtt{s}\alpha \leq^{\infty ?}_{\mathtt{A}} \beta$

$\beta \leq^{\infty ?}_{\mathtt{A}} \mathtt{s}^2\alpha$



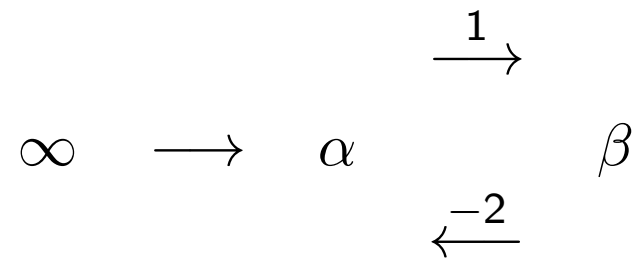
Graph representation of a problem in the successor algebra

remark 1: every constraint has at most 2 variables

$$\mathtt{s}^k a \leq^? \mathtt{s}^l b \in P \quad \Leftrightarrow \quad a \xrightarrow{k-l} b \in \mathrm{Graph}(P)$$

example:

$$\begin{array}{l} \mathtt{s}\alpha \leq_{\mathtt{A}}^{\infty ?} \beta \\ \beta \leq_{\mathtt{A}}^{\infty ?} \mathtt{s}^2\alpha \\ \infty \leq_{\mathtt{A}}^{\infty ?} \alpha \end{array}$$



Graph representation of a problem in the successor algebra

remark 1: every constraint has at most 2 variables

$$\mathtt{s}^k a \leq^? \mathtt{s}^l b \in P \quad \Leftrightarrow \quad a \xrightarrow{k-l} b \in \mathrm{Graph}(P)$$

example:

$$\begin{array}{l} \mathtt{s}\alpha \leq_{\mathtt{A}}^{\infty ?} \beta \\ \beta \leq_{\mathtt{A}}^{\infty ?} \mathtt{s}^2\alpha \\ \infty \leq_{\mathtt{A}}^{\infty ?} \alpha \\ \beta \leq_{\mathtt{A}}^{\infty ?} \mathtt{sc} \end{array} \qquad \infty \longrightarrow \alpha \underset{\xleftarrow{-2}}{\xrightarrow{1}} \beta \xrightarrow{-1} \mathtt{c}$$

Graph representation of a problem in the successor algebra

remark 1: every constraint has at most 2 variables

$$\mathtt{s}^k a \leq^? \mathtt{s}^l b \in P \quad \Leftrightarrow \quad a \xrightarrow{k-l} b \in \mathrm{Graph}(P)$$

example:

$$\begin{array}{l} \mathtt{s}\alpha \leq_{\mathtt{A}}^{\infty ?} \beta \\ \beta \leq_{\mathtt{A}}^{\infty ?} \mathtt{s}^2\alpha \\ \infty \leq_{\mathtt{A}}^{\infty ?} \alpha \\ \beta \leq_{\mathtt{A}}^{\infty ?} \mathtt{sc} \end{array} \qquad \infty \longrightarrow \alpha \underset{\xleftarrow{-2}}{\xrightarrow{1}} \beta \xrightarrow{-1} \mathtt{c}$$

remark 2: if $a \simeq_P b$ and $\varphi \in \mathrm{Sol}(P)$ then $\mathrm{base}(a\varphi) = \mathrm{base}(b\varphi)$

$\leq_P$ smallest qo s.t. $a \leq_P b$ if there is a constraint $\mathtt{s}^k a \leq^? \mathtt{s}^l b \in P$

Satisfiability of a problem in the successor algebra

if there is no constant, a problem in $\mathtt{A} \cup \{\infty\}$ with n variables is equivalent to a problem $ax \oplus b \leq x$ in the idempotent semi-ring $(\overline{\mathbb{Z}}_{\max}^{n \times n}, \oplus, \otimes)$ where $\overline{\mathbb{Z}}_{\max} = \mathbb{Z} \cup \{\pm\infty\}$, $\oplus = \max$ and $\otimes = +$

Satisfiability of a problem in the successor algebra

if there is no constant, a problem in $\mathtt{A} \cup \{\infty\}$ with n variables is equivalent to a problem $ax \oplus b \leq x$ in the idempotent semi-ring $(\overline{\mathbb{Z}}_{\max}^{n\times n}, \oplus, \otimes)$ where $\overline{\mathbb{Z}}_{\max} = \mathbb{Z} \cup \{\pm\infty\}$, $\oplus = \max$ and $\otimes = +$

example: $\{\mathtt{s}\alpha \leq_{\mathtt{A}}^{\infty ?} \beta, \beta \leq_{\mathtt{A}}^{\infty ?} \mathtt{s}^2\alpha, \infty \leq_{\mathtt{A}}^{\infty ?} \alpha\}$ is equivalent to $ax \oplus b \leq x$ where $a = \begin{pmatrix} -\infty & -2 \\ 1 & -\infty \end{pmatrix}$, $x = \begin{pmatrix} \alpha \\ \beta \end{pmatrix}$ and $b = \begin{pmatrix} +\infty \\ -\infty \end{pmatrix}$

Satisfiability of a problem in the successor algebra

if there is no constant, a problem in $\mathtt{A} \cup \{\infty\}$ with n variables is equivalent to a problem $ax \oplus b \leq x$ in the idempotent semi-ring $(\overline{\mathbb{Z}}_{\max}^{n\times n}, \oplus, \otimes)$ where $\overline{\mathbb{Z}}_{\max} = \mathbb{Z} \cup \{\pm\infty\}$, $\oplus = \max$ and $\otimes = +$

example: $\{\mathtt{s}\alpha \leq_{\mathtt{A}}^{\infty?} \beta, \beta \leq_{\mathtt{A}}^{\infty?} \mathtt{s}^2\alpha, \infty \leq_{\mathtt{A}}^{\infty?} \alpha\}$ is equivalent to $ax \oplus b \leq x$ where $a = \begin{pmatrix} -\infty & -2 \\ 1 & -\infty \end{pmatrix}$, $x = \begin{pmatrix} \alpha \\ \beta \end{pmatrix}$ and $b = \begin{pmatrix} +\infty \\ -\infty \end{pmatrix}$

$$\begin{array}{rcl} \alpha + 1 & \leq & \beta \\ \beta & \leq & \alpha + 2 \\ +\infty & \leq & \alpha \end{array}$$

Satisfiability of a problem in the successor algebra

if there is no constant, a problem in $\mathtt{A} \cup \{\infty\}$ with n variables is equivalent to a problem $ax \oplus b \leq x$ in the idempotent semi-ring $(\overline{\mathbb{Z}}_{\max}^{n\times n}, \oplus, \otimes)$ where $\overline{\mathbb{Z}}_{\max} = \mathbb{Z} \cup \{\pm\infty\}$, $\oplus = \max$ and $\otimes = +$

example: $\{\mathtt{s}\alpha \leq_{\mathtt{A}}^{\infty?} \beta, \beta \leq_{\mathtt{A}}^{\infty?} \mathtt{s}^2\alpha, \infty \leq_{\mathtt{A}}^{\infty?} \alpha\}$ is equivalent to $ax \oplus b \leq x$ where $a = \begin{pmatrix} -\infty & -2 \\ 1 & -\infty \end{pmatrix}$, $x = \begin{pmatrix} \alpha \\ \beta \end{pmatrix}$ and $b = \begin{pmatrix} +\infty \\ -\infty \end{pmatrix}$

$$\begin{array}{rcl} \alpha + 1 & \leq & \beta \\ \beta & \leq & \alpha + 2 \\ +\infty & \leq & \alpha \end{array} \quad \rightsquigarrow \quad \begin{array}{rcl} \beta - 2 & \leq & \alpha \\ +\infty & \leq & \alpha \\ \alpha + 1 & \leq & \beta \end{array}$$

Satisfiability of a problem in the successor algebra

if there is no constant, a problem in $\mathtt{A} \cup \{\infty\}$ with n variables is equivalent to a problem $ax \oplus b \leq x$ in the idempotent semi-ring $(\overline{\mathbb{Z}}_{\max}^{n\times n}, \oplus, \otimes)$ where $\overline{\mathbb{Z}}_{\max} = \mathbb{Z} \cup \{\pm\infty\}$, $\oplus = \max$ and $\otimes = +$

example: $\{\mathtt{s}\alpha \leq_{\mathtt{A}}^{\infty?} \beta, \beta \leq_{\mathtt{A}}^{\infty?} \mathtt{s}^2\alpha, \infty \leq_{\mathtt{A}}^{\infty?} \alpha\}$ is equivalent to $ax \oplus b \leq x$ where $a = \begin{pmatrix} -\infty & -2 \\ 1 & -\infty \end{pmatrix}$, $x = \begin{pmatrix} \alpha \\ \beta \end{pmatrix}$ and $b = \begin{pmatrix} +\infty \\ -\infty \end{pmatrix}$

$$\begin{array}{rcl} \alpha + 1 & \leq & \beta \\ \beta & \leq & \alpha + 2 \\ +\infty & \leq & \alpha \end{array} \rightsquigarrow \begin{array}{rcl} \beta - 2 & \leq & \alpha \\ +\infty & \leq & \alpha \\ \alpha + 1 & \leq & \beta \end{array} \rightsquigarrow \begin{array}{rcl} -2 \otimes \beta \oplus +\infty & \leq & \alpha \\ 1 \otimes \alpha & \leq & \beta \end{array}$$

Satisfiability of a problem in the successor algebra

if there is no constant, a problem in $\mathtt{A} \cup \{\infty\}$ with n variables is equivalent to a problem $ax \oplus b \leq x$ in the idempotent semi-ring $(\overline{\mathbb{Z}}_{\max}^{n\times n}, \oplus, \otimes)$ where $\overline{\mathbb{Z}}_{\max} = \mathbb{Z} \cup \{\pm\infty\}$, $\oplus = \max$ and $\otimes = +$

example: $\{\mathtt{s}\alpha \leq_{\mathtt{A}}^{\infty?} \beta, \beta \leq_{\mathtt{A}}^{\infty?} \mathtt{s}^2\alpha, \infty \leq_{\mathtt{A}}^{\infty?} \alpha\}$ is equivalent to $ax \oplus b \leq x$ where $a = \begin{pmatrix} -\infty & -2 \\ 1 & -\infty \end{pmatrix}$, $x = \begin{pmatrix} \alpha \\ \beta \end{pmatrix}$ and $b = \begin{pmatrix} +\infty \\ -\infty \end{pmatrix}$

$$\begin{array}{rcl} \alpha + 1 & \leq & \beta \\ \beta & \leq & \alpha + 2 \\ +\infty & \leq & \alpha \end{array} \rightsquigarrow \begin{array}{rcl} \beta - 2 & \leq & \alpha \\ +\infty & \leq & \alpha \\ \alpha + 1 & \leq & \beta \end{array} \rightsquigarrow \begin{array}{rcl} -2 \otimes \beta \oplus +\infty & \leq & \alpha \\ 1 \otimes \alpha & \leq & \beta \end{array}$$

$$\rightsquigarrow \begin{array}{rcl} -\infty \otimes \alpha \oplus -2 \otimes \beta \oplus +\infty & \leq & \alpha \\ 1 \otimes \alpha \oplus -\infty \otimes \beta \oplus -\infty & \leq & \beta \end{array}$$

Satisfiability of a problem in the successor algebra

after [C79], [BCOQ92], [ABG14] (Hankbook of linear algebra):

Satisfiability of a problem in the successor algebra

after [C79], [BCOQ92], [ABG14] (Hankbook of linear algebra):

theorem 1: $ax \oplus b \leq x$ has smallest solution a^*b with $a^* = \bigoplus_{i=0}^{\infty} a^i$

(in $(\mathbb{R}, +, \times)$, we have $x = \frac{b}{1-a} = b \cdot \sum_{i=0}^{\infty} a^i$)

Satisfiability of a problem in the successor algebra

after [C79], [BCOQ92], [ABG14] (Hankbook of linear algebra):

theorem 1: $ax \oplus b \leq x$ has smallest solution a^*b with $a^* = \bigoplus_{i=0}^{\infty} a^i$

(in $(\mathbb{R}, +, \times)$, we have $x = \frac{b}{1-a} = b \cdot \sum_{i=0}^{\infty} a^i$)

theorem 2: $a^* = \bigoplus_{i=0}^{n} a^i$ if there is no positive cycle

Dealing with constants

solutions of $\{\alpha \leq^? \mathtt{s}'\mathsf{c}\}$?

Dealing with constants

solutions of $\{\alpha \leq^? \mathtt{s}^l\mathsf{c}\}$? substitutions $\{(\alpha, \mathtt{s}^k\mathsf{c})\}$ with $0 \leq k \leq l$

Dealing with constants

solutions of $\{\alpha \leq^? \mathtt{s}^l\mathtt{c}\}$? substitutions $\{(\alpha, \mathtt{s}^k\mathtt{c})\}$ with $0 \leq k \leq l$

idea 1: extend $\mathtt{A}$ with

- a new sort $\mathtt{N}$ for non-negative integers
- a new function symbol $\mathtt{s} : \mathtt{N} \Rightarrow \mathtt{A} \Rightarrow \mathtt{A}$ for successor iteration

Dealing with constants

solutions of $\{\alpha \leq^? \mathtt{s}^l\mathtt{c}\}$? substitutions $\{(\alpha, \mathtt{s}^k\mathtt{c})\}$ with $0 \leq k \leq l$

idea 1: extend $\mathtt{A}$ with

- a new sort $\mathtt{N}$ for non-negative integers
- a new function symbol $\mathtt{s} : \mathtt{N} \Rightarrow \mathtt{A} \Rightarrow \mathtt{A}$ for successor iteration

now, every term is either ∞, $\mathtt{s}^k\alpha$ or $\mathtt{s}^e\mathtt{c}$ with $e \in \{k, k + \mathrm{x}_\alpha\}$

Dealing with constants

solutions of $\{\alpha \leq^? \mathtt{s}^l\mathtt{c}\}$? substitutions $\{(\alpha, \mathtt{s}^k\mathtt{c})\}$ with $0 \leq k \leq l$

idea 1: extend $\mathtt{A}$ with

- a new sort $\mathtt{N}$ for non-negative integers
- a new function symbol $\mathtt{s} : \mathtt{N} \Rightarrow \mathtt{A} \Rightarrow \mathtt{A}$ for successor iteration

now, every term is either ∞, $\mathtt{s}^k\alpha$ or $\mathtt{s}^e\mathtt{c}$ with $e \in \{k, k + \mathrm{x}_\alpha\}$

$$\{\alpha \leq^? \mathtt{s}^l\mathtt{c}\} \text{ is equivalent to } \{\alpha =^? \mathtt{s}^{\mathrm{x}_\alpha}\mathtt{c}, \mathrm{x}_\alpha \leq^? l\}$$

Dealing with constants

solutions of $\{\alpha \leq^? \mathtt{s}^l\mathsf{c}\}$? substitutions $\{(\alpha, \mathtt{s}^k\mathsf{c})\}$ with $0 \leq k \leq l$

idea 1: extend $\mathtt{A}$ with

- a new sort $\mathtt{N}$ for non-negative integers
- a new function symbol $\mathtt{s} : \mathtt{N} \Rightarrow \mathtt{A} \Rightarrow \mathtt{A}$ for successor iteration

now, every term is either ∞, $\mathtt{s}^k\alpha$ or $\mathtt{s}^e\mathsf{c}$ with $e \in \{k, k + \mathrm{x}_\alpha\}$

$$\{\alpha \leq^? \mathtt{s}^l\mathsf{c}\} \text{ is equivalent to } \{\alpha =^? \mathtt{s}^{\mathrm{x}_\alpha}\mathsf{c}, \mathrm{x}_\alpha \leq^? l\}$$

idea 2: transform P into an equivalent problem on $\overline{\mathbb{Z}}_{\mathsf{max}}$

Satisfiability of a problem in the successor algebra

simplification rules:

$$
\begin{array}{rcl}
P \uplus \{\mathtt{s}\, a \leq^? \mathtt{s}\, b\} & \to_1 & P \cup \{a \leq^? b\} \\
P \uplus \{\mathtt{s}^e \mathtt{c} \leq^? \mathtt{s}^f \mathtt{c}\} & \to_2 & P \cup \{e \leq^? f\} \\
P \uplus \{\infty \leq^? \mathtt{s}^{\mathtt{s}e} \alpha\} & \to_3 & P \cup \{\infty \leq^? \alpha\}
\end{array}
$$

Satisfiability of a problem in the successor algebra

simplification rules:

$$P \uplus \{\mathtt{s}\, a \leq^? \mathtt{s}\, b\} \quad \to_1 \quad P \cup \{a \leq^? b\}$$
$$P \uplus \{\mathtt{s}^e \mathtt{c} \leq^? \mathtt{s}^f \mathtt{c}\} \quad \to_2 \quad P \cup \{e \leq^? f\}$$
$$P \uplus \{\infty \leq^? \mathtt{s}^{\mathtt{s}e} \alpha\} \quad \to_3 \quad P \cup \{\infty \leq^? \alpha\}$$

constraints always true:

$$P \uplus \{a \leq^? \infty\} \quad \to_4 \quad P \cup \{\alpha \leq^? \infty \mid \alpha \in \mathrm{Var}(a) - \mathrm{Var}(P)\}$$
if $a \notin \mathtt{X} \vee a \in \mathrm{Var}(P)$
$$P \uplus \{a \leq^? \mathtt{s}^e a\} \quad \to_5 \quad P \cup \{\alpha \leq^? \infty \mid \alpha \in \mathrm{Var}(a) - \mathrm{Var}(P)\}$$

Satisfiability of a problem in the successor algebra

simplification rules:

$$P \uplus \{\mathtt{s}\, a \leq^? \mathtt{s}\, b\} \to_1 P \cup \{a \leq^? b\}$$
$$P \uplus \{\mathtt{s}^e \mathtt{c} \leq^? \mathtt{s}^f \mathtt{c}\} \to_2 P \cup \{e \leq^? f\}$$
$$P \uplus \{\infty \leq^? \mathtt{s}^{\mathtt{s}e} \alpha\} \to_3 P \cup \{\infty \leq^? \alpha\}$$

constraints always true:

$$P \uplus \{a \leq^? \infty\} \to_4 P \cup \{\alpha \leq^? \infty \mid \alpha \in \mathrm{Var}(a) - \mathrm{Var}(P)\}$$
if $a \notin \mathtt{X} \vee a \in \mathrm{Var}(P)$
$$P \uplus \{a \leq^? \mathtt{s}^e a\} \to_5 P \cup \{\alpha \leq^? \infty \mid \alpha \in \mathrm{Var}(a) - \mathrm{Var}(P)\}$$

constraints always false:

$$P \uplus \{\infty \leq^? \mathtt{s}^e \mathtt{c}\} \to_6 \bot$$
$$P \uplus \{\mathtt{s}^{\mathtt{s}e} \alpha \leq^? \mathtt{c}\} \to_7 \bot$$
$$P \uplus \{\mathtt{s}^e \mathtt{c} \leq^? \mathtt{s}^f \mathtt{d}\} \to_8 \bot \text{ if } \mathtt{c} \neq \mathtt{d}$$
$$P \uplus Q \to_9 \bot \text{ if } \mathrm{Var}_{\mathtt{A}}(Q) = \emptyset \text{ and } \mathrm{Sol}(Q) = \emptyset$$

Satisfiability of a problem in the successor algebra

$$P \uplus \{\infty \leq^? \alpha\} \quad \to_{10} \quad P\{(\alpha, \infty)\} \cup \{\infty \leq^? \alpha\} \quad \text{if } \alpha \in \mathrm{Var}(P)$$

variables that must be set to ∞:

Satisfiability of a problem in the successor algebra

variables that must be set to ∞:

$$P \uplus \{\infty \leq^? \alpha\} \quad \to_{10} \quad P\{(\alpha, \infty)\} \cup \{\infty \leq^? \alpha\} \quad \text{if } \alpha \in \mathrm{Var}(P)$$

$$P \uplus \{\mathtt{s}^{\mathtt{se}}\alpha \leq^? \alpha\} \quad \to_{11} \quad P\{(\alpha, \infty)\} \cup \{\infty \leq^? \alpha\}$$

Satisfiability of a problem in the successor algebra

		variables that must be set to ∞:
$P \uplus \{\infty \leq^? \alpha\}$	$\to_{10}$	$P\{(\alpha, \infty)\} \cup \{\infty \leq^? \alpha\}$ if $\alpha \in \mathrm{Var}(P)$
$P \uplus \{\mathtt{s}^{\mathtt{se}}\alpha \leq^? \alpha\}$	$\to_{11}$	$P\{(\alpha, \infty)\} \cup \{\infty \leq^? \alpha\}$
$P \uplus Q$	$\to_{12}$	$P\{(\alpha, \infty) \mid \alpha \in \mathrm{Var}(Q)\}$
		$\cup\{\infty \leq^? \alpha \mid \alpha \in \mathrm{Var}(Q)\}$
		if $\mathrm{Graph}(Q)$ is a positive cycle

Satisfiability of a problem in the successor algebra

		variables that must be set to ∞:
$P \uplus \{\infty \leq^? \alpha\}$	$\to_{10}$	$P\{(\alpha, \infty)\} \cup \{\infty \leq^? \alpha\}$ if $\alpha \in \mathrm{Var}(P)$
$P \uplus \{\mathtt{s}^{\mathtt{se}}\alpha \leq^? \alpha\}$	$\to_{11}$	$P\{(\alpha, \infty)\} \cup \{\infty \leq^? \alpha\}$
$P \uplus Q$	$\to_{12}$	$P\{(\alpha, \infty) \mid \alpha \in \mathrm{Var}(Q)\}$
		$\cup\{\infty \leq^? \alpha \mid \alpha \in \mathrm{Var}(Q)\}$
		if $\mathrm{Graph}(Q)$ is a positive cycle
P	$\to_{14}$	$P\{(\alpha, \infty)\} \cup \{\infty \leq^? \alpha\}$
		if $\mathsf{c} \leq_P \alpha, \mathsf{d} \leq_P \alpha, \mathsf{c} \neq \mathsf{d}$

Satisfiability of a problem in the successor algebra

variables that must be set to ∞:

$$P \uplus \{\infty \leq^? \alpha\} \to_{10} P\{(\alpha, \infty)\} \cup \{\infty \leq^? \alpha\} \quad \text{if } \alpha \in \mathrm{Var}(P)$$

$$P \uplus \{\mathtt{s}^{\mathtt{se}}\alpha \leq^? \alpha\} \to_{11} P\{(\alpha, \infty)\} \cup \{\infty \leq^? \alpha\}$$

$$P \uplus Q \to_{12} P\{(\alpha, \infty) \mid \alpha \in \mathrm{Var}(Q)\} \cup \{\infty \leq^? \alpha \mid \alpha \in \mathrm{Var}(Q)\} \quad \text{if } \mathrm{Graph}(Q) \text{ is a positive cycle}$$

$$P \to_{14} P\{(\alpha, \infty)\} \cup \{\infty \leq^? \alpha\} \quad \text{if } \mathsf{c} \leq_P \alpha, \mathsf{d} \leq_P \alpha, \mathsf{c} \neq \mathsf{d}$$

variables that must be set to a constant:

$$P \uplus \{\mathtt{s}^k\alpha \leq^? \mathtt{s}^e\mathsf{c}\} \to_{13} P\{(\alpha, \mathtt{s}^{\mathrm{x}_\alpha}\mathsf{c})\} \cup \{\alpha =^? \mathtt{s}^{\mathrm{x}_\alpha}\mathsf{c}, k + \mathrm{x}_\alpha \leq^? e\} \quad \text{if } (k, e) \neq (0, \mathrm{x}_\alpha)$$

Satisfiability of a problem in the successor algebra

variables that must be set to ∞:

$$P \uplus \{\infty \leq^? \alpha\} \to_{10} P\{(\alpha, \infty)\} \cup \{\infty \leq^? \alpha\} \quad \text{if } \alpha \in \mathrm{Var}(P)$$

$$P \uplus \{\mathtt{s}^{\mathtt{se}}\alpha \leq^? \alpha\} \to_{11} P\{(\alpha, \infty)\} \cup \{\infty \leq^? \alpha\}$$

$$P \uplus Q \to_{12} P\{(\alpha, \infty) \mid \alpha \in \mathrm{Var}(Q)\} \cup \{\infty \leq^? \alpha \mid \alpha \in \mathrm{Var}(Q)\} \quad \text{if } \mathrm{Graph}(Q) \text{ is a positive cycle}$$

$$P \to_{14} P\{(\alpha, \infty)\} \cup \{\infty \leq^? \alpha\} \quad \text{if } \mathtt{c} \leq_P \alpha, \mathtt{d} \leq_P \alpha, \mathtt{c} \neq \mathtt{d}$$

variables that must be set to a constant:

$$P \uplus \{\mathtt{s}^k\alpha \leq^? \mathtt{s}^e\mathtt{c}\} \to_{13} P\{(\alpha, \mathtt{s}^{\mathrm{x}_\alpha}\mathtt{c})\} \cup \{\alpha =^? \mathtt{s}^{\mathrm{x}_\alpha}\mathtt{c}, k + \mathrm{x}_\alpha \leq^? e\} \quad \text{if } (k, e) \neq (0, \mathrm{x}_\alpha)$$

theorem: this rewrite system terminates

Normal forms

after normalization, we get $\bot$ or an *affine* problem:

- constraints are of the form $\mathtt{s}^k\alpha \leq^? \mathtt{s}^l\beta$ or $\mathtt{s}^e\mathtt{c} \leq^? \mathtt{s}^m\beta$
- there is no positive cycle
- there is no tuple $(\alpha, \mathtt{c}, \mathtt{d})$ such that $\mathtt{c} \leq_P \alpha$, $\mathtt{d} \leq_P \alpha$ and $\mathtt{c} \neq \mathtt{d}$
  $\Rightarrow$ an equivalence class modulo $\simeq_P$ contains at most 1 constant

Normal forms

after normalization, we get $\bot$ or an *affine* problem:

- constraints are of the form $\mathtt{s}^k\alpha \leq^? \mathtt{s}^l\beta$ or $\mathtt{s}^e\mathtt{c} \leq^? \mathtt{s}^m\beta$
- there is no positive cycle
- there is no tuple $(\alpha, \mathtt{c}, \mathtt{d})$ such that $\mathtt{c} \leq_P \alpha$, $\mathtt{d} \leq_P \alpha$ and $\mathtt{c} \neq \mathtt{d}$
  $\Rightarrow$ an equivalence class modulo $\simeq_P$ contains at most 1 constant

theorem 0: an affine problem is always satisfiable (with $\varphi = \infty$)

Normal forms

after normalization, we get $\bot$ or an *affine* problem:

- constraints are of the form $\mathtt{s}^k\alpha \leq^? \mathtt{s}^l\beta$ or $\mathtt{s}^e\mathtt{c} \leq^? \mathtt{s}^m\beta$
- there is no positive cycle
- there is no tuple $(\alpha, \mathtt{c}, \mathtt{d})$ such that $\mathtt{c} \leq_P \alpha$, $\mathtt{d} \leq_P \alpha$ and $\mathtt{c} \neq \mathtt{d}$
  $\Rightarrow$ an equivalence class modulo $\simeq_P$ contains at most 1 constant

theorem 0: an affine problem is always satisfiable (with $\varphi = \infty$)

theorem 1: the finite solutions are isomorphic to the solutions of the integer problem obtained by substituting constants with 0

Normal forms

after normalization, we get $\bot$ or an *affine* problem:

- constraints are of the form $\mathtt{s}^k\alpha \leq^? \mathtt{s}^l\beta$ or $\mathtt{s}^e\mathtt{c} \leq^? \mathtt{s}^m\beta$
- there is no positive cycle
- there is no tuple $(\alpha, \mathtt{c}, \mathtt{d})$ such that $\mathtt{c} \leq_P \alpha$, $\mathtt{d} \leq_P \alpha$ and $\mathtt{c} \neq \mathtt{d}$
  $\Rightarrow$ an equivalence class modulo $\simeq_P$ contains at most 1 constant

theorem 0: an affine problem is always satisfiable (with $\varphi = \infty$)

theorem 1: the finite solutions are isomorphic to the solutions of the integer problem obtained by substituting constants with 0

example: $P = \{\mathtt{s}^k\alpha \leq^? \mathtt{s}^l\beta, \mathtt{s}^e\mathtt{c} \leq^? \mathtt{s}^m\beta\}$ is associated to
$Q = \{k + \alpha \leq^? l + \beta, e \leq^? m + \beta\}$
a solution ψ of Q gives a solution $\widehat{\psi}$ of P such that $\alpha\widehat{\psi} = \mathtt{s}^{\alpha\psi}\alpha^*$ where α^* = representative picked in $[\alpha]_{\simeq_P}$ with $\alpha^* = \mathtt{c}$ if $\alpha \simeq_P \mathtt{c}$

Normal forms

after normalization, we get $\bot$ or an *affine* problem:

- constraints are of the form $\mathtt{s}^k\alpha \leq^? \mathtt{s}^l\beta$ or $\mathtt{s}^e\mathtt{c} \leq^? \mathtt{s}^m\beta$
- there is no positive cycle
- there is no tuple $(\alpha, \mathtt{c}, \mathtt{d})$ such that $\mathtt{c} \leq_P \alpha$, $\mathtt{d} \leq_P \alpha$ and $\mathtt{c} \neq \mathtt{d}$
  $\Rightarrow$ an equivalence class modulo $\simeq_P$ contains at most 1 constant

theorem 0: an affine problem is always satisfiable (with $\varphi = \infty$)

theorem 1: the finite solutions are isomorphic to the solutions of the integer problem obtained by substituting constants with 0

example: $P = \{\mathtt{s}^k\alpha \leq^? \mathtt{s}^l\beta, \mathtt{s}^e\mathtt{c} \leq^? \mathtt{s}^m\beta\}$ is associated to
$Q = \{k + \alpha \leq^? l + \beta, e \leq^? m + \beta\}$
a solution ψ of Q gives a solution $\widehat{\psi}$ of P such that $\alpha\widehat{\psi} = \mathtt{s}^{\alpha\psi}\alpha^*$ where α^* = representative picked in $[\alpha]_{\simeq_P}$ with $\alpha^* = \mathtt{c}$ if $\alpha \simeq_P \mathtt{c}$

theorem 2: an integer problem with at most 2 variables per constraint and no positive cycle has a smallest solution

Satisfiability of a problem in the successor algebra

1. the above rules terminate, are correct and complete
2. P is satisfiable iff any normal form of P is distinct from $\bot$
3. satisfiability is decidable in polynomial time
4. every satisfiable problem has a smallest solution
5. the smallest solution is computable in polynomial time

Conclusion

extension of Hughes-Pareto-Sabry (POPL'96):

1. user-defined notions of size
2. arbitrary size algebra
3. rewriting-based function definitions
4. general type-checking algorithm
5. completeness in the successor algebra
6. prototype implementation: HOT
   (best termination tool for HOR in 2012)

Possible directions of research

- study other algebra (*e.g.* max-successor, max-plus)

Possible directions of research

- study other algebra (*e.g.* max-successor, max-plus)
- extension to dependent and polymorphic types

Possible directions of research

- study other algebra (*e.g.* max-successor, max-plus)
- extension to dependent and polymorphic types
- re-implementation of HOT using SAT/SMT solver

Possible directions of research

- study other algebra (*e.g.* max-successor, max-plus)
- extension to dependent and polymorphic types
- re-implementation of HOT using SAT/SMT solver
- combination with CPO, the last version of HORPO

Possible directions of research

- study other algebra (*e.g.* max-successor, max-plus)
- extension to dependent and polymorphic types
- re-implementation of HOT using SAT/SMT solver
- combination with CPO, the last version of HORPO
- existential and constrained sized types & conditional rewriting [B.-Riba 2006]