6.3b Implicit Induction (Part 2)

Dienstag, 19. Januar 2016 08:30

To automate Thm 6.3.5: How can one find out the set of ground normal forms $NF(\mathcal{R})$ of a TRS $\mathcal{R}$ and how can one represent $NF(\mathcal{R})$ in a finite way?

In plus-example:

$NF(\mathcal{R}) = \{0, succ(0), succ(succ(0)), \dots\}$

i.e.: 0 and succ act as constructors to construct data objects

plus is an algorithm ($\hat{=}$ defined function symbol) which is "completely defined"

In this case, $NF(\mathcal{R})$ are all ground terms built from constructors.

Def 6.3.6. (Definition Principle)

Let $\Sigma^c \subseteq \Sigma$ be a set of constructors with $\Sigma^c_0 \neq \emptyset$ (i.e., there is at least 1 constant constructor) and $|\Sigma^c| \geq 2$ (i.e., there are at least 2 constructors).

Let $\Sigma^d = \Sigma \setminus \Sigma^c$ be the set of defined function symbols. A TRS $\mathcal{R}$ over Σ and $\mathcal{V}$ satisfies the definition principle for Σ^c iff

principle for Σ^c iff

(a) For all rules $l \to r \in \mathcal{R}$ we have $l \notin \mathcal{T}(\Sigma^c, \mathcal{V})$
(i.e., l contains at least one symbol from Σ^d)

(b) For all $f \in \Sigma^d$ and all $t_1, \ldots, t_n \in \mathcal{T}(\Sigma^c)$ there exists a rule $l \to r \in \mathcal{R}$ such that l matches $f(t_1, \ldots, t_n)$ (i.e., f is completely defined).

Ex. 637 The plus-TRS obviously satisfies the def. principle.

Thm 6.38 (Ground normal forms for definition principle)
Let $\mathcal{R}$ be a TRS that satisfies the def. principle for a set of constructors Σ^c. Then $NF(\mathcal{R}) = \mathcal{T}(\Sigma^c)$.

Proof: "$NF(\mathcal{R}) \subseteq \mathcal{T}(\Sigma^c)$":

Assume that there is a $q \in NF(\mathcal{R})$ which contains symbols from Σ^d. Then there exists an innermost subterm $f(t_1, \ldots, t_n)$ of q where $f \in \Sigma^d$, $t_1, \ldots, t_n \in \mathcal{T}(\Sigma^c)$. But since $\mathcal{R}$ satisfies the def. principle, by (b) $f(t_1, \ldots, t_n)$ is not in normal form. Thus, q is not in normal form either.

"$\mathcal{T}(\Sigma^c) \subseteq NF(\mathcal{R})$":

$T(\Sigma^c) \subseteq NF(\mathcal{R})$:

Since $\mathcal{R}$ satisfies the def. principle, (a) implies that no left-hand side of a rule matches a term without defined symbols.

Goal: Develop a technique which checks whether adding an equation $s \equiv t$ makes two (different) constructor ground terms equal.

Solution: Use a (slightly modified) version of the completion algorithm.

Usually: start with $(E \cup \{s \equiv t\}, \emptyset)$ and apply rules until one reaches $(\emptyset, \mathcal{R}_\omega)$

Now: start with $(\{s \equiv t\}, \mathcal{R})$

where $\mathcal{R}$ is a convergent TRS that is equivalent to E and $\mathcal{R}$ satisfies the definition principle.

In case of success, one again reaches a configuration $(\emptyset, \mathcal{R}_\omega)$.

Modifications: (together with the equations)

- ensure that the TRS satisfies the definition principle throughout the completion process

⇒ modify the "Orient" rule, ensure that left-hand sides of new rules always contain defined symbols

⇒ When adding a rule $s \to t$, we require that $s = f(\ldots)$ for some $f \in \Sigma^d$.

- try to detect inconsistencies (i.e., we try to find out when 2 different constructor ground terms are made equal)

  ⇒ add a new rule "Inconsistency" which stops and returns "False" if we have an equation of the form

$$c_1(\ldots) \equiv c_2(\ldots) \quad \text{for } c_1, c_2 \in \Sigma^c,\ c_1 \neq c_2$$

This does not hold for all x, → since there exist at least 2 constructors

$$c(\ldots) \equiv x \quad \text{for } c \in \Sigma,\ x \in \mathcal{V}$$

$$x \equiv c(\ldots) \quad \text{— " —}$$

- in order to make the procedure more powerful, we add another rule which exploits the fact that different constructor ground terms denote different objects and thus, constructors are injective

  ⇒ add a rule "Injectivity" which allows to replace

$$c(s_1, \ldots, s_n) \equiv c(t_1, \ldots, t_n) \quad \text{for } c \in \Sigma^c$$

  by $s_1 \equiv t_1, \ldots, s_n \equiv t_n$

Def 63g (Induction Procedure (HH 82))

Let $\mathcal{E}$ be a set of equations, $\mathcal{R}$ be a TRS over Σ and $\mathcal{V}$, let $\Sigma^c \subseteq \Sigma$, let $>$ be a reduction order.
Then the induction procedure consists of the 6 rules of the completion procedure (Def. 622), where "Orient" is modified as follows and we add 2 new rules "Inconsistency" and "Injectivity".

(see slide)

Ex 63.10. We use the induction procedure to check whether $\mathcal{E} \vdash_I s \equiv t$ for:

$\mathcal{E}$ = plus-equations

$\mathcal{R} = \{ plus(0, y) \to y,\ plus(s(x), y) \to s(plus(x, y)) \}$

"$s \equiv t$": $plus(x, s(y)) \equiv s(plus(x, y))$

Start with $(\underbrace{\{plus(x, s(y)) \equiv s(plus(x, y))\}}_{\mathcal{E}_0},\ \underbrace{\mathcal{R}}_{\mathcal{R}_0},\ \text{i.e. } \mathcal{R}_0 \setminus \mathcal{R} = \emptyset)$

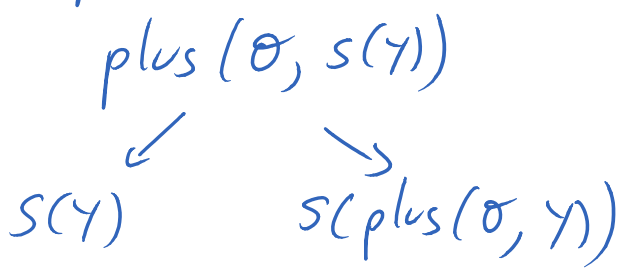
We use LPO or RPO with plus > succ.

$(\mathcal{E}_0, \mathcal{R}_0) \vdash_I (\underbrace{\emptyset}_{\mathcal{E}_1},\ \underbrace{\{plus(x, s(y)) \to s(plus(x, y))\} \cup \mathcal{R}}_{\mathcal{R}_1,\ \text{i.e., } \mathcal{R}_1 \setminus \mathcal{R} = \{plus(x, s(y)) \to \ldots\}})$

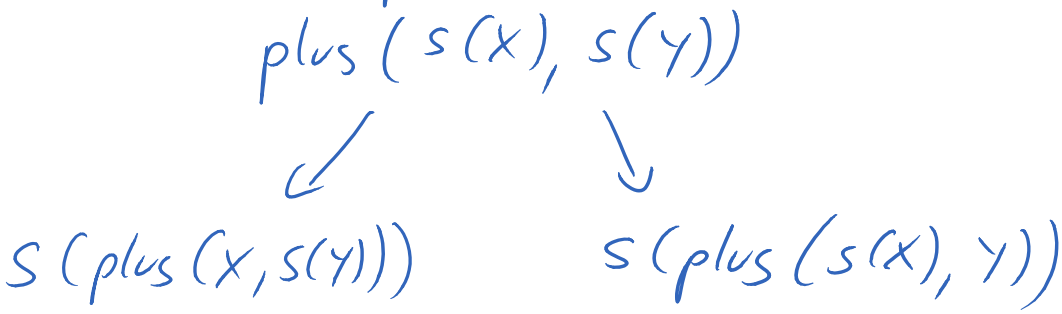
Of course, the completion sequence must be fair, i.e., we have to build all critical pairs of the persistent rules. Since $\mathcal{R}$ is convergent, we can assume that we already

built all critical pairs of $\mathcal{E}$ and only consider critical pairs resulting from new rules.

First new crit. pair:



Second new crit. pair:



Since the process ended with $(\emptyset, \mathcal{R}_\omega)$ without detecting inconsistencies and since the sequence was fair, this proves $\mathcal{E} \models_I plus(x, s(y)) \equiv s(plus(x,y))$

Where was the "induction"?

- The application of

  $plus(x, s(y)) \to s(plus(x,y))$

  to join the second crit. pair corresponds to the application of the induction hypothesis
- First crit. pair $\hat{=}$ Induction Base

Sec. – n – $\hat{=}$ Induction Step

Advantage: Choice of induction variable and ind. relation is done "automatically" be the modified completion procedure.

In a similar way, one can also prove associativity:

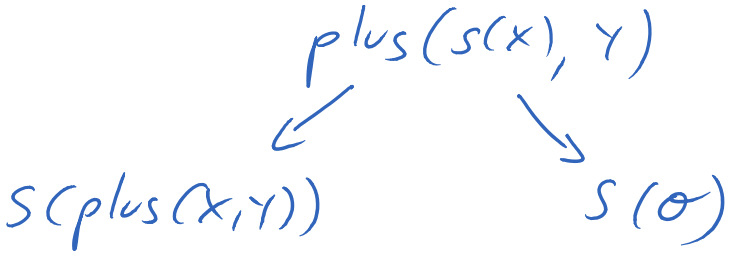
$\mathcal{E} \models_I plus(plus(x,y),z) \equiv plus(x, plus(y,z))$

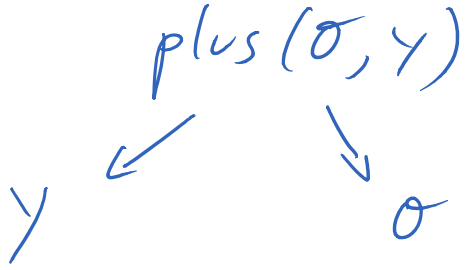
Ex 6.3.11

Check whether

$\mathcal{E} \models_I plus(succ(x), y) \equiv succ(0)$ holds.

Crit. Pair:





Since we detect an inconsistency, this proves

$\mathcal{E} \not\models_I plus(s(x), y) \equiv s(0)$

Thm 63.12 (Correctness of the Ind. Calculus)

Let $\mathcal{E}$ be a set of equations over Σ and $\mathcal{V}$,
let $\mathcal{R}$ be a convergent TRS that satisfies the def. principle for $\Sigma^c \subseteq \Sigma$ and is equivalent to $\mathcal{E}$,
let $s, t \in \mathcal{T}(\Sigma, \mathcal{V})$,
let $>$ be a rcl. order.
Let $(\{s \equiv t\}, \mathcal{R}) \vdash_I \ldots$ be a fair transformation with the ind. calculus of Def. 6.3.9.
If the transf. ends with "False", then $\mathcal{E} \not\models_I s \equiv t$.
If the transf. ends with $(\emptyset, \mathcal{R}_\omega)$, then $\mathcal{E} \models_I s \equiv t$.