Semantic Path Order and Dependency Pairs

August 21, 2013

Semantic Path Order

- $>$ is a strict partial order (on terms).
- $\approx$ is an equivalence relation (on terms).
- They are compatible, i.e.: $> \circ \approx \ \subseteq \ >$ and $\approx \circ > \ \subseteq \ >$.
- Usually their definition involves some "semantics".

Semantic Path Order

Definition

The semantic path equivalence $\approx$ induced by $\approx$ is the equivalence relation between terms (inductively defined) by: $f(s_1, \ldots, s_m) \approx g(t_1, \ldots, t_n)$ iff $f(s_1, \ldots, s_m) \approx g(t_1, \ldots, t_n)$, $m = n$, and $s_i \approx t_i$ for every $1 \leq i \leq n$.

Semantic Path Order

Definition

The lexicographic semantic path order $>$ induced by $\langle >, \approx \rangle$ is the strict partial order between terms, recursively defined as follows: For two terms $s = f(s_1, \ldots, s_m)$ and $t = g(t_1, \ldots, t_n)$, $s > t$ if at least one of the following hold:

1. $s_i \gtrsim t$ for some $1 \leq i \leq m$.
2. $s > t_i$ for all $1 \leq i \leq n$, and $s > t$.
3. $s > t_i$ for all $1 \leq i \leq n$, $s \approx t$, and $\langle s_1, \ldots, s_m \rangle >_{lex} \langle t_1, \ldots, t_n \rangle$ ($>_{lex}$ is the lexicographic ordering induced by $>$ and $\approx$).

Properties

Subterm Property

If t is a proper subterm of s, then $s > t$.

- $s > t$ is a strict partial order.
- $s \approx t$ is an equivalence relation.
- They are compatible: $> \circ \approx \;\subseteq\; >$ and $\approx \circ > \;\subseteq\; >$.

Example

- Consider a language with a constant 0 (nullary symbol), three unary symbols S, P, F, and one binary symbol $*$.
- Define the semantic interpretation $[\![t]\!]$ of a term t to be its natural numerical value (S is successor, P is previous and $[\![P(0)]\!] = 0$, $*$ is multiplication, and F is factorial).
- Define $>$ by $t > s$ if either t is headed by F and s is not, or both are headed by F and $[\![t]\!] > [\![s]\!]$.
- Then, for every term x:
  - $F(0) > S(0)$, since $F(0) > S(0)$ and $F(0) > 0$.
  - $F(S(x)) > P(S(x))$, since $F(S(x)) > P(S(x))$ and $F(S(x)) > S(x)$.
  - $F(S(x)) > F(P(S(x)))$ since $F(S(x)) > F(P(S(x)))$ and $F(S(x)) > P(S(x))$.
  - $F(S(x)) > S(x) * F(P(S(x)))$. Why?

Basic Theorem

Theorem

If $>$ is well-founded, then the lexicographic semantic path order $>$ induced by $\langle >, \approx \rangle$ is also well-founded.

Proof Outline.

1. Take a minimal counterexample
   $t_1 = f_1(s_1^1, \ldots, s_1^{m_1}) > t_2 = f_2(s_2^1, \ldots, s_2^{m_2}) > \ldots$
2. Observe that $s_j^i \gtrsim t_{i+1}$ cannot hold (by minimality)
3. Therefore, $t_i \gtrsim t_{i+1}$ for every i
4. Since $>$ is well-founded and $> \circ \approx \subseteq >$, $t_i \approx t_{i+1}$ from some point on
5. From that point: $\langle s_i^1, \ldots, s_i^{m_i} \rangle >_{lex} \langle s_{i+1}^1, \ldots, s_{i+1}^{m_{i+1}} \rangle$
6. From some M a certain component j constantly decreases
7. $t_1 > \ldots > t_{M-1} > s_M^j > s_{M+1}^j > s_{M+2}^j > \ldots$ is a "shorter" counterexample

□

Proving Termination Using Lexicographic Semantic Path Orders

To prove that a rewrite system terminates, one has to identify relations $>$ and $\approx$ for which the following hold:

1. $>$ is well-founded.
2. If $s \to t$ then $f(\cdots, s, \cdots) \gtrsim f(\cdots, t, \cdots)$ for every symbol f (as usual, $\gtrsim = > \cup \approx$).
3. For every rule $l \to r$ and substitution σ: $\sigma l > \sigma r$, where $>$ is the lexicographic semantic path order induced by $\langle >, \approx \rangle$.

Example

- $P(S(x)) \to x$
- $F(0) \to S(0)$
- $F(S(x)) \to S(x) * F(P(S(x)))$

- Take $>$ as before, and $t \approx s$ iff both are headed by the same symbol, and $[\![t]\!] = [\![s]\!]$
- $>$ and $\approx$ are compatible.
- $>$ is well-founded.
- if $s \to t$ then $f(\cdots, s, \cdots) \approx f(\cdots, t, \cdots)$ (since all rules preserve the numerical value).
- $\sigma l > \sigma r$ for every rule $l \to r$ and σ.

Therefore, this system is terminating.

Proof of Termination

- Conditions $2 - 3$ above ensure that if $s \rightarrow t$ then $s > t$ (by induction on the depth of the rewrite step)
- The claim follows since $>$ is well-founded

Example

- $\neg\neg x \to x$
- $\neg(x \vee y) \to \neg\neg\neg x \wedge \neg\neg\neg y$
- $\neg(x \wedge y) \to \neg\neg\neg x \vee \neg\neg\neg y$

- $t > s$ iff the number of $\vee, \wedge$ in t is greater than their number in s
- $t \approx s$ iff the number of $\vee, \wedge$ in t is equal to their number in s
- $>$ and $\approx$ are compatible, and $>$ is well-founded
- $s \to t$ implies $f(\cdots, s, \cdots) \approx f(\cdots, t, \cdots)$ (since all rules of the system preserve the number of $\vee$ and $\wedge$).
- $\sigma l > \sigma r$ for every rule $l \to r$ and σ (verify).

Consequently, this system is terminating.

Other Semantic Path Orders

- Replace $>_{lex}$ by $>_{multiset}$ to obtain *"multiset semantic path order"*.
  - All proofs remain the same.

- Use $>_{lex}$ with different ordering of the subterms, possibly ignoring or duplicating some of them.
  - Associate a list of indices $i^f_1, \ldots, i^f_{m_f}$ to every symbol f.
  - Replace $\langle s_1, \ldots, s_m \rangle >_{lex} \langle t_1, \ldots, t_n \rangle$ by $\langle s_{i^f_1}, \ldots, s_{i^f_{m_f}} \rangle >_{lex} \langle t_{i^g_1}, \ldots, t_{i^g_{m_g}} \rangle$.
  - The proof that $>$ is well-founded remains the same.
  - The termination proof does not work: we might have $s \to t$ and $s \approx t$.
  - However, we can still prove termination by showing:
    - $s \to t$ implies $s \gtrsim t$.
    - Top-rewrite of $s \to t$ implies $s > t$.

Slightly Extended Semantic Path Orders

Observation

We did not used the fact that $\approx$ is an equivalence relation.

- We can take any quasi order $\gtrsim$ instead of $\approx$.
- Require compatibility: $> \circ \gtrsim \; \subseteq \; >$.
- Same definition of a semantic path order.

Slightly Extended Semantic Path Orders

Theorem

A rewrite system is terminating if there are compatible strict partial order $>$ and quasi-order $\gtrsim$ satisfying:

1. $>$ *is well-founded.*
2. $s \to t$ *implies* $f(\cdots, s, \cdots) > f(\cdots, t, \cdots)$ *or* $f(\cdots, s, \cdots) \gtrsim f(\cdots, t, \cdots)$.
3. *For every rule* $l \to r$ *and substitution* σ*:* $\sigma l > \sigma r$, *where* $>$ *is the (lexicographic) semantic path order induced by* $\langle >, \gtrsim \rangle$.

- The proof remains the same.

Dependency Pairs

Definition

A *constructor* (in some rewrite system) is a symbol that never appears at the head of a left-hand side of any rule.

Definition

The dependency pairs of a rewrite system consist of all pairs $l \to u$ for every rule $l \to r$ and non-variable (not necessarily proper) subterm u of r that is not headed by a constructor.

Example

- $x - 0 \to x$
- $s(x) - s(y) \to x - y$
- $0 \div s(y) \to 0$
- $s(x) \div s(y) \to s((x - y) \div s(y))$

- s and 0 are constructors.
- The dependency pairs are:
  - $s(x) - s(y) \to x - y$
  - $s(x) \div s(y) \to (x - y) \div s(y)$
  - $s(x) \div s(y) \to x - y$

Proving Termination Using Dependency Pairs

Theorem

A rewrite system[a] *is terminating if there exist a quasi-order* $\gtrsim$ *and a strict partial order* $>$ *that satisfy the following conditions:*

1. $\gtrsim \circ > \;\subseteq\; >$
2. $\sigma l \gtrsim \sigma r$ *for each rule* $l \to r$ *and substitution* σ
3. $\sigma l > \sigma r$ *for each dependency pair* $l \to r$ *and substitution* σ
4. $>$ *is well-founded*
5. $\gtrsim$ *is weakly monotonic:* $s \gtrsim t$ *implies* $f(\cdots, s, \cdots) \gtrsim f(\cdots, t, \cdots)$

---

[a] We assume that each variable occurring in a right side of some rule also occurs in its left side, and that no rule has the form $x \to r$ for some variable x.

Example

$x - 0 \to x$	Dependency pairs:
$s(x) - s(y) \to x - y$	$s(x) - s(y) \to x - y$
$0 \div s(y) \to 0$	$s(x) \div s(y) \to (x - y) \div s(y)$
$s(x) \div s(y) \to s((x - y) \div s(y))$	$s(x) \div s(y) \to x - y$

- Define $[\![\cdot]\!]$ by: $[\![0]\!] = 0$; $[\![s(x)]\!] = [\![x]\!] + 1$; $[\![x - y]\!] = [\![x]\!]$; $[\![x \div y]\!] = [\![x]\!]$
- Define: $s \gtrsim t$ iff $[\![s]\!] \geq [\![t]\!]$; $s > t$ iff $[\![s]\!] > [\![t]\!]$

- $\gtrsim \circ > \ \subseteq \ >$
- $\sigma l \gtrsim \sigma r$ for each rule $l \to r$ and σ
- $\sigma l > \sigma r$ for each dependency pair $l \to r$ and σ
- $>$ is well-founded
- $\gtrsim$ is weakly monotonic (since in the suggested interpretation all symbols are interpreted by a weakly monotonic function)

Consequently, this system is terminating.

Termination Proof (Using Semantic Path Order)

Lemma

If the conditions above hold for some $\gtrsim$ and $>$, then they also hold for some $\gtrsim'$ and $>'$, such that $s >' t$ whenever t is headed by a constructor and s is not.

Proof Outline.

Obtain $>'$ from $>$ by:

- Adding all pairs whose left-side is a non-constructor term and right-side is a constructor term
- Removing any pair whose left-side is a constructor term

Obtain $\gtrsim'$ from $\gtrsim$ by:

- Removing any pair with left-side a constructor and right-side not.

Show that the conditions above hold for $\gtrsim'$ and $>'$. □

Termination Proof (Using Semantic Path Order)

- By the previous lemma, we can suppose that in $>$ all terms headed by constructors are smaller than all those that are not.
- Let $>$ be $\gtrsim \circ > \circ \gtrsim$.
- $> \circ \gtrsim \subseteq >$, so $>$ and $\gtrsim$ are compatible.
- We show that $>$ and $\gtrsim$ meet all conditions required to prove termination using the lexicographic semantic path order induced by $\langle >, \gtrsim \rangle$.

Termination Proof (Using Semantic Path Order)

- $>$ is well-founded since $\gtrsim \circ > \ \subseteq \ >$ and $>$ is well-founded.
- Suppose that $s \rightarrow t$, and show that $f(\cdots, s, \cdots) \gtrsim f(\cdots, t, \cdots)$. Since $\gtrsim$ is weakly monotonic, it suffices to show that $s \rightarrow t$ implies that $s \gtrsim t$. This is proven by induction on the depth of the rewrite step $s \rightarrow t$ (again, using weak monotonicity for the induction step).

Termination Proof (Using Semantic Path Order)

- Consider a rule $l \to r$ and a substitution σ. We show that $\sigma l > \sigma r$.
  - If r is a proper subterm of l (in particular, if r is a variable), then $\sigma l > \sigma r$ by the subterm property.
  - Otherwise r is headed by a constructor or $l \to r$ is a dependency pair.
  - In both cases, $\sigma l > \sigma r$ and so $\sigma l > \sigma r$.
  - To show that $\sigma l > \sigma r$, it suffices to prove that $\sigma l > \sigma r'$ for every subterm r' of r.
  - Use induction on the structure of r':
    - Suppose that for all subterms r'' of r' we have $\sigma l > \sigma r''$.
    - If r' is a subterm of l (in particular, if r' is a variable), then $\sigma l > \sigma r'$ by the subterm property.
    - Otherwise r' is headed by a constructor or $l \to r'$ is a dependency pair.
    - In both cases, $\sigma l > \sigma r'$ and so $\sigma l > \sigma r'$.
    - By the induction hypothesis, $\sigma l > \sigma r'$.

Revisiting Example Above

$x - 0 \rightarrow x$	Dependency pairs:
$s(x) - s(y) \rightarrow x - y$	$s(x) - s(y) \rightarrow x - y$
$0 \div s(y) \rightarrow 0$	$s(x) \div s(y) \rightarrow (x - y) \div s(y)$
$s(x) \div s(y) \rightarrow s((x - y) \div s(y))$	$s(x) \div s(y) \rightarrow x - y$

- Define $[\![\cdot]\!]$ by: $[\![0]\!] = 0$; $[\![s(x)]\!] = [\![x]\!] + 1$; $[\![x - y]\!] = [\![x]\!]$; $[\![x \div y]\!] = [\![x]\!]$.
- Define $\gtrsim$ by: $s \gtrsim t$ iff $[\![s]\!] \geq [\![t]\!]$.
- Define $>$ by: $s > t$ iff $[\![s]\!] > [\![t]\!]$.

According to the last proof, termination can be proved using the lexicographic semantic path order induced by $\langle >, \gtrsim \rangle$, that are defined by:

- $s > t$ if s is not headed by S or 0, and either $[\![s]\!] > [\![t]\!]$ or t is headed by S or 0.
- $s \gtrsim t$ if $[\![s]\!] \geq [\![t]\!]$, and either s is not headed by S or 0 or t is headed by S or 0.