Semantic Path Order and Dependency Pairs - Handout

August 21, 2013

1 Lexicographic Semantic Path Order

Let $>, \approx$ be some strict partial order and equivalence relation (respectively) on the set of terms, that are compatible, i.e.: $> \circ \approx \; \subseteq \; >$ and $\approx \circ > \; \subseteq \; >$. Usually their definitions involves some "semantics".

Definition 1.1. 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 $1 \leq i \leq n$.

Definition 1.2. 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 $1 \leq i \leq m$, where $\gtrsim \; = \; > \cup \approx$ (as usual), and $\approx$ is The semantic path equivalence induced by $\approx$.

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 $\approx$).

It is straightforward to verify that $\approx$ is an equivalence relation, $>$ is a strict partial order (transitive and irreflexive), and that $> \circ \approx \; \subseteq \; >$ and $\approx \circ > \; \subseteq \; >$. For this purpose, the following proposition is useful.

Proposition 1.3 (Subterm Property)**.** If t is a proper subterm of s, then $s > t$.

Proof. By induction on the structure of s: Suppose that for all immediate subterms s_i of s, we have $s_i > t$ whenever t is a proper subterm of s_i. Assume that t is a proper subterm of s. Then t is either an immediate subterm s_i of s, or a proper subterm of an immediate subterm s_i of s. In the first case, $s_i \gtrsim t$ (since $\gtrsim$ is reflexive). In the latter, $s_i > t$ by the induction hypothesis. Therefore, in both cases we have $s > t$ by condition 1. □

Example 1.4. 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 in this language to be its natural numerical value, where 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 $F(S(x)) > F(P(S(x)))$ (since $[\![F(S(x))]\!] = ([\![x]\!] + 1)! > ([\![x]\!] + 1 - 1)! = [\![F(P(S(x)))]\!]$), and $F(S(x)) > P(S(x))$.
- $F(S(x)) > S(x) * F(P(S(x)))$, since $F(S(x)) > S(x) * F(P(S(x)))$, $F(S(x)) > S(x)$, and $F(S(x)) > F(P(S(x)))$.

Theorem 1.5. If $>$ is well-founded, then the lexicographic semantic path order $>$ induced by $\langle >, \approx \rangle$ is also well-founded.

Proof. Suppose that $>$ is not well-founded. We take a minimal counterexample $t_1 = f_1(s_1^1, \ldots, s_1^{m_1}) > t_2 = f_2(s_2^1, \ldots, s_2^{m_2}) > \ldots$ (i.e. t_n at each step is minimal in size of all counterexamples starting with $t_1, t_2, \ldots, t_{n-1}$). Note that for every i, $s \not\gtrsim t_{i+1}$ for every subterm s of t_i (otherwise, we could take s instead of t_i in the i'th stage and obtain a "shorter" counterexample).

Therefore, for every i, $s_i^j > t_{i+1}$ for all $1 \leq j \leq m_i$, and either $t_i > t_{i+1}$ or ($t_{i+1} \approx t_i$ and $\langle s_i^1, \ldots, s_i^{m_i} \rangle >_{lex} \langle s_{i+1}^1, \ldots, s_{i+1}^{m_{i+1}} \rangle$). Since $>$ is well-founded and $> \circ \approx \subseteq >$, there is some N, such that $t_i \approx t_{i+1}$ for every $i \geq N$. Hence, $\langle s_i^1, \ldots, s_i^{m_i} \rangle >_{lex} \langle s_{i+1}^1, \ldots, s_{i+1}^{m_{i+1}} \rangle$ for every $i \geq N$. The properties of lexicographic ordering ensure that there is some point from which a certain component constantly decreases. In other words, the is some $M \geq N$ and $1 \leq j \leq m_M$ such that $s_M^j > s_{M+1}^j > s_{M+2}^j > \ldots$. Since $t_{M-1} > s_M^j$ (for the reason that $t_{M-1} > t_M$ and by the subterm property $t_M > s_M^j$), we obtain that $t_1 > t_2 > \ldots > t_{M-1} > s_M^j > s_{M+1}^j > s_{M+2}^j > \ldots$. However, this is a "shorter" counterexample. Contradiction. □

Note that usual lexicographic path order is obtained as a special instance by considering $>$ and $\approx$ that are completely determined according to symbol at the heads of the compared terms.

2 Proving Termination Using Lexicographic Semantic Path Orders

Given a rewrite system and two terms s, t, we write $s \to t$ if t can be obtained from s by one rewrite step. To prove that a rewrite system terminates, one has to identify relations $>$ and $\approx$ for which the following hold:

1. $>$ is well-founded.

2. For every two terms, s and t, 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 rewrite rule $l \to r$ and substitution σ: $\sigma l > \sigma r$, where $>$ is the lexicographic semantic path order induced by $\langle >, \approx \rangle$.

Lemma 2.1. Conditions $2 - 3$ above ensure that if $s \to t$ then $s > t$.

Proof. Suppose that $s \to t$. Then there is some context c, rule $l \to r$, and substitution σ, such that $s = c[\sigma l]$ and $t = c[\sigma r]$. We prove that $s > t$, by induction on the depth of the context. For depth 0 (a top-rewrite of s into t), we have $s > t$ directly by condition 3. Now suppose that $s = f(u_1, \ldots, u_{j-1}, s', u_{j+1}, \ldots, u_n)$ is rewritten into $t = f(u_1, \ldots, u_{j-1}, t', u_{j+1}, \ldots, u_n)$. Thus $s' \to t'$, and by the induction hypothesis, $s' > t'$. Hence, $\langle u_1, \ldots, u_{j-1}, s', u_{j+1}, \ldots, u_n \rangle >_{lex} \langle u_1, \ldots, u_{j-1}, t', u_{j+1}, \ldots, u_n \rangle$. In addition, $s > t'$ (since $s > s'$ by the subterm property and $s' > t'$), and $s > u_i$ for every other immediate subterm of t (since it is also an immediate subterm of s). Hence, it suffices to show that either $s > t$ or $s \approx t$. This follows by condition 2. □

Theorem 2.2. Under conditions $1 - 3$ above, the rewrite system is terminating.

Proof. By Lemma 2.1, $s \to t$ implies $s > t$. The claim follows since $>$ is well-founded (Theorem 1.5). □

Example 2.3. Consider the following rewrite system for *factorial*:

- $P(S(x)) \to x$

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

Define $>$ as in Example 1.4 ($t > s$ if either t is headed by F and s is not, or both are headed by F and $[\![t]\!] > [\![s]\!]$), and define $\approx$ by $t \approx s$, if both are headed by the same symbol, and $[\![t]\!] = [\![s]\!]$. Then, $>$ and $\approx$ are compatible. $>$ is clearly well-founded (since the natural order of natural numbers is well-founded). Following Example 1.4, we have $\sigma l > \sigma r$ for every rewrite rule $l \to r$ of this system and substitution σ. In addition, since $[\![\sigma l]\!] = [\![\sigma r]\!]$ for every substitution σ and rule $l \to r$, we have that $[\![s]\!] = [\![t]\!]$ whenever $s \to t$. It follows that $s \to t$ implies that $f(\cdots, s, \cdots) \approx f(\cdots, t, \cdots)$ for every symbol f. Consequently, by Theorem 2.2, this system is terminating.

Example 2.4. Consider the following rewrite system:

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

Define $>$ by $t > s$ if the number of $\vee, \wedge$ in t is greater than their number in s. Similarly, define $\approx$ by $t \approx s$ if the number of $\vee, \wedge$ in t is equal to their number in s. Then, $>$ and $\approx$ are compatible, and $>$ is clearly well-founded. Now, we have:

- For every two terms, s and t, if $s \to t$ then $f(\cdots, s, \cdots) \approx f(\cdots, t, \cdots)$ for every symbol f, since all rules of the system preserve the number of $\vee$ and $\wedge$.
- $\sigma l > \sigma r$ for every rewrite rule $l \to r$ and substitution σ. For $\neg\neg x \to x$, this follows by the subterm property. For $\neg(x \vee y) \to \neg\neg\neg x \wedge \neg\neg\neg y$, note that $\neg(x \vee y) > \neg^i x$ for every i, x and y (proof by induction, using the subterm property for the base case and $\neg(x \vee y) > \neg^i x$ in the induction step), and similarly $\neg(x \vee y) > \neg^i y$ for every i,x and y. Thus, since $\neg(x \vee y) \approx \neg\neg\neg x \wedge \neg\neg\neg y$ and $\langle x \vee y \rangle >_{lex} \langle \neg\neg\neg x, \neg\neg\neg y \rangle$ for every x, y, we have $\neg(x \vee y) > \neg\neg\neg x \wedge \neg\neg\neg y$ for every x, y. The proof for the third rule is exactly the same.

Consequently, by Theorem 2.2, this system is terminating.

3 Other Semantic Path Orders

In addition to the lexicographic semantic path order defined above, various different semantic path orders can be used for proving termination of rewrite systems. For example, by replacing in Definition 1.2 $>_{lex}$ by $>_{multiset}$, that denotes the multiset partial order induced by $>$ and $\approx$, we obtain "multiset semantic path order". Each of the proofs above can be modified in a straightforward way, showing that $>$ is well-founded (provided that $>$ is well-founded), and that conditions 1 – 3 guarantee the termination of a given rewrite system.

Similarly, it is possible to use lexicographic ordering as in Definition 1.2 with different ordering of the subterms, possibly ignoring or duplicating some of them. To do so, one has to associate a list of indices $i^f_1, \ldots, i^f_{m_f}$ to every symbol f, and instead of $\langle s_1, \ldots, s_m \rangle >_{lex} \langle t_1, \ldots, t_n \rangle$ (in Definition 1.2) have $\langle s_{i^f_1}, \ldots, s_{i^f_{m_f}} \rangle >_{lex} \langle t_{i^g_1}, \ldots, t_{i^g_{m_g}} \rangle$. We call the resulting ordering *modified lexicographic semantic path ordering.* The proof that $>$ is well-founded (provided that $>$ is well-founded) remains exactly the same. The proof that conditions 1 – 3 above guarantee termination requires several modifications. Indeed, since we allow some immediate subterms to be discarded in the lexicographic ordering, we might not have $s > t$ whenever $s \to t$. The modified proof is given below.

Lemma 3.1. Let $\gtrsim$ be a modified lexicographic semantic path ordering. Conditions 2 – 3 above ensure that if $s \to t$ then $s \gtrsim t$.

Proof. Suppose that $s \to t$. Then there is some context c, rule $l \to r$, and substitution σ, such that $s = c[\sigma l]$ and $t = c[\sigma r]$. We prove that $s > t$, by induction on the depth of the context. For depth 0 (a top-rewrite of s into t), we have $s > t$ directly by condition 3. Now suppose that $s = f(u_1, \ldots, u_n)$ is rewritten into $t = f(u'_1, \ldots, u'_n)$. Thus there is some $1 \leq j \leq n$, such that $u_j \to u'_j$, and for every $i \neq j$, $u_i = u'_i$. By the induction hypotheses, $u_j \gtrsim u'_j$. Hence, $\langle u_{i^f_1}, \ldots, u_{i^f_{m_f}} \rangle >_{lex} \langle u'_{i^f_1}, \ldots, u'_{i^f_{m_f}} \rangle$, or $u_{i^f_k} \approx u'_{i^f_k}$ for every $1 \leq k \leq m_f$. In addition, $s > u'_j$ (since $s > u_j$ by the subterm property, and $u_j \gtrsim u'_j$) and $s > u'_i$ for every other immediate subterm of t (since it is also an immediate subterm of s). By condition 2, we have $s > t$ or $s \approx t$. It follows that $s > t$ or $s \approx t$.

□

Theorem 3.2. Under conditions 1 – 3 above, where $>_{lex}$ is a modified lexicographic semantic path ordering, the rewrite system is terminating

Proof. By condition 3, we have $s > t$ whenever $s \to t$ by a top-level rewriting. By Lemma 3.1, $s \gtrsim t$ whenever $s \to t$ by an inner-level rewriting. Since $>$ is well-founded (Theorem 1.5), these conditions guarantee that the system is terminating. This is proved as for quasi-simplification orderings. For completeness, we reproduce the argument for that.

Say that a term s_0 *initiates* an infinite derivation, if there are some $s_1, s_2, \ldots$ such that $s_0 \to s_1 \to s_2 \to \ldots$. We observe that if s_0 initiates an infinite derivation, then there exists some t that initiates an infinite derivation and satisfies $s_0 > t$. Indeed, if $s_n \to s_{n+1}$ is obtained by a top-level rewrite for some n, then $s_0 \gtrsim s_1 \gtrsim \ldots \gtrsim s_n > s_{n+1}$, and by choosing $t = s_{n+1}$ we have $s_0 > t$. If there is no top-level rewrite in $s_0 \to s_1 \to s_2 \to \ldots$, then there is some immediate subterm of s_0 that initiates an infinite derivation (since s_0 has a finite number of subterms), and we can take t to be this subterm. By the subterm property, we have $s_0 > t$.

Now, suppose that $s_1 \to s_2 \to \ldots$ is an infinite derivation in the system. We recursively construct and infinite sequence of terms $t_1, t_2, \ldots$, such that each t_i initiates an infinite derivation, and $t_1 > t_2 > \ldots$ (contradicting the fact that $>$ is well-founded). Choose $t_1 = s_1$. Suppose that t_n was defined. Since t_n initiates an infinite derivation, by the observation above, there some t that initiates an infinite derivation as well, and satisfies $t_n > t$. Take $t_{n+1} = t$. $\square$

3.1 Extended Semantic Path Order

Inspecting the proofs above, we note that we did not used the fact that $\approx$ is an equivalence relation. Indeed, everything works exactly the same if we replace $\approx$ by some quasi-order, according the next definition:

Definition 3.3. Let $>, \gtrsim$ be some strict partial order and quasi-order (respectively) on the set of terms, that are compatible, i.e.: $> \circ \gtrsim\ \subseteq\ >$.

1. The semantic path equivalence $\approx$ induced by $\gtrsim$ is identical to the semantic path equivalence induced by $\approx = \gtrsim \cap \lesssim$ (see Definition 1.1).

2. The lexicographic semantic path order $>$ induced by $\langle >, \gtrsim \rangle$ is defined exactly as in Definition 1.2, where $\approx$ is replaced by $\gtrsim$.

Note that the subterm property still holds as well as the next theorem (with the same proofs):

Theorem 3.4. A rewrite system is terminating if there are compatible strict partial order $>$ and quasi-order $\gtrsim$ satisfying:

1. $>$ is well-founded.

2. For every two terms, s and t, if $s \to t$ then $f(\cdots, s, \cdots) > f(\cdots, t, \cdots)$ or $f(\cdots, s, \cdots) \gtrsim f(\cdots, t, \cdots)$ for every symbol f.

3. For every rewrite rule $l \to r$ and substitution σ: $\sigma l > \sigma r$, where $>$ is the lexicographic semantic path order induced by $\langle >, \gtrsim \rangle$.

4 Dependency Pairs

In this section we briefly introduce the Dependency Pairs method, and show its correctness using a semantic path order.

Definition 4.1. A *constructor* (in some rewrite system) is a symbol that never appears at the head of a left-hand side of any rule.

Definition 4.2. 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 4.3. Consider the following rewrite system:

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

Then s and 0 are constructors, and the dependency pairs are:

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

Definition 4.4. A rewrite system is called *normal* if each variable occurring in a right side of some rule also occurs in its left side, and no rule has the form $x \to r$ for some variable x.

Obviously, if some variable occurs only in a right side of some rule, then the system is not terminating. In addition, rules of the form $x \to r$ can always be replaced by equivalent rules with non-variable left side.

Theorem 4.5. A normal rewrite system[1] 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, i.e.: $s \gtrsim t$ implies $f(\cdots, s, \cdots) \gtrsim f(\cdots, t, \cdots)$ for every symbol f.

Example 4.6. Consider the rewrite system from Example 4.3. Define a numerical interpretation $[\![\cdot]\!]$ for terms in this language as follows: $[\![0]\!] = 0$; $[\![s(x)]\!] = [\![x]\!] + 1$; $[\![x - y]\!] = [\![x]\!]$; $[\![x \div y]\!] = [\![x]\!]$. Define $\gtrsim$ by $s \gtrsim t$ iff $[\![s]\!] \geq [\![t]\!]$; and $>$ by $s > t$ iff $[\![s]\!] > [\![t]\!]$. Clearly, $\gtrsim \circ > \ \subseteq \ >$. It is easy to verify that $\sigma l \gtrsim \sigma r$ for each rule $l \to r$ and substitution σ, and $\sigma l > \sigma r$ for each dependency pairs $l \to r$ and substitution σ. In addition, $>$ is well-founded, and $\gtrsim$ is weakly monotonic (since in the suggested interpretation all symbols are interpreted by a weakly monotonic function). By Theorem 4.5 this system is terminating.

To prove Theorem 4.5, we use the following lemma.

Lemma 4.7. If conditions 1 – 5 hold for some $\gtrsim$ and $>$ (for some normal rewrite system), then they also hold for some $\gtrsim'$ and $>'$, such that $s >' t$ whenever t is headed by a constructor and s is not.

Proof. We obtain $>'$ from $>$ by adding all pairs whose left-side is a non-constructor term and right-side is a constructor term, and removing from $>$ any pair whose left-side is a constructor term. $\gtrsim'$ is obtained from $\gtrsim$ by removing any pair with left-side a constructor and right-side not. We still have:

1. $\gtrsim' \circ >' \ \subseteq \ >'$.

2. $\sigma l \gtrsim' \sigma r$ for each rule $l \to r$ and substitution σ, since constructors by definition never appear at the head of a left-side. Here we also use the fact that l is not a variable, and thus *sigmal* is headed by a constructor iff l is.

[1]

3. $\sigma l >' \sigma r$ for each dependency pair $l \to r$ and substitution σ, for the same reason.

4. $>'$ is well-founded.

5. $\gtrsim'$ is weakly monotonic. Indeed, suppose that $s \gtrsim' t$, and let f be some symbol. Then $s \gtrsim t$, and the weak monotonicity of $\gtrsim$ entails that $f(\cdots, s, \cdots) \gtrsim f(\cdots, t, \cdots)$. Since both sides are headed by the same symbol, we also have $f(\cdots, s, \cdots) \gtrsim' f(\cdots, t, \cdots)$.

□

Proof of Theorem 4.5. We use a semantic path order to show that the system is terminating (Theorem 3.4). By Lemma 4.7, we can suppose w.l.o.g. that in $>$ all terms headed by constructors are smaller than all those that are not. Let $> = \gtrsim \circ > \circ \gtrsim$ (its transitivity can be proved using the fact that $\gtrsim \circ > \subseteq >$). Clearly, $> \circ \gtrsim \subseteq >$, so $>$ and $\gtrsim$ are compatible. We show that $>$ and $\gtrsim$ satisfy all conditions from Theorem 3.4:

- $>$ is well-founded since $\gtrsim \circ > \subseteq >$ and $>$ is well-founded.

- We prove that for every two terms, s and t, if $s \to t$ then $f(\cdots, s, \cdots) \gtrsim f(\cdots, t, \cdots)$ for every symbol f. Since $\gtrsim$ is weakly monotonic, it suffices to show that $s \to t$ implies that $s \gtrsim t$. Suppose that $s \to t$. Then there is some context c, rule $l \to r$, and substitution σ, such that $s = c[\sigma l]$ and $t = c[\sigma r]$. We prove that $s \gtrsim t$, by induction on the depth of the context c. For depth 0 (a top-rewrite of s into t), we have $s \gtrsim t$ by definition. Now suppose that $s = f(u_1, \ldots, u_n)$ is rewritten into $t = f(u'_1, \ldots, u'_n)$. Thus there is some $1 \leq j \leq n$, such that $u_j \to u'_j$, and for every $i \neq j$, $u_i = u'_i$. By the induction hypotheses, $u_j \gtrsim u'_j$. and since $\gtrsim$ is weakly monotonic, $s \gtrsim t$ as well.

- We prove that $\sigma l > \sigma r$ for every rule $l \to r$ and substitution σ. Consider a rule $l \to r$ and a substitution σ. If r is a proper subterm of l (in particular, if r is a variable), then $\sigma l > \sigma r$ by the subterm property of $>$. If not, we have $\sigma l > \sigma r$ (if r is headed by a constructor, this holds by our assumption, and otherwise it holds since $l \to r$ is a dependency pair), and so $\sigma l > \sigma r$. To show that $\sigma l > \sigma r$, it remains to prove that $\sigma l > \sigma r'$ for every immediate subterm r' of r. We show that by induction on the structure of r' that $\sigma l > \sigma r'$ for every subterm r' 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, using the induction hypothesis, it suffices to show that $\sigma l > \sigma r'$. Since $> \subseteq >$, we can show that $\sigma l > \sigma r'$. If r' is headed by a constructor, then $\sigma l > \sigma r'$ by our assumption, and if not, then $l \to r'$ is a dependency pair, and again $\sigma l > \sigma r'$. □

Example 4.8. According to the last proof, the system from Example 4.6 can be proved to be terminating by Theorem 3.4, using $\langle >, \gtrsim \rangle$ 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.

References

[Arts and Giesl, 2000] Thomas Arts and Jürgen Giesl. Termination of term rewriting using dependency pairs. *Theoretical Computer Science*, 236:133–178, 2000. Preliminary version available at `http://verify.rwth-aachen.de/giesl/papers/ibn-97-46.ps`.

[Dershowitz and Hoot, 1995] Nachum Dershowitz and Charles Hoot. Natural termination. *Theoretical Computer Science*, 142(2):179–207, 1995. Available at `http://nachum.org/papers/natural-sterm94.pdf`.

[Dershowitz, 1982] Nachum Dershowitz. Orderings for term-rewriting systems. *Theoretical Computer Science*, 17(3):279–301, 1982. Available at `http://nachum.org/papers/Orderings4TRS.pdf`.

[Dershowitz, 2013] Nachum Dershowitz. Dependency Pairs are a Simple Semantic Path Ordering. Unpublished note.

[Kamin and Lévy, 1980] Sam Kamin and Jean-Jacques Lévy. Two generalizations of the recursive path ordering. Unpublished letter to Nachum Dershowitz, Department of Computer Science, University of Illinois, Urbana, IL, February 1980. Available at `http://nachum.org/term/kamin-levy80spo.pdf`.