The embedding order is too weak, but it is a good starting point for the development of suitable reduction relations. \( \Rightarrow \) Special notion for reduction orders that contain the embedding order.

**Def 4.4.1 (Simplification Order (Dershowitz, 1987))**

A reduction order \( \succ \) where \( S \succ t \) holds whenever \( S \succ \text{emb} t \), is called a simplification order.

If an order contains the embedding order, then there is a "simple" way to prove its well-foundedness: one only has to check that the order is irreflexive (\( t \nRightarrow t \) for all \( t \)).

Reason: Kruskal's Theorem.

**Thm 4.4.2 (Kruskal's Theorem)**

As usual, the signature \( \Sigma \) is finite.

(a) For any infinite sequence of ground terms \( t_0, t_1, t_2, ... \), there exist \( i, j \in \mathbb{N} \) with \( i < j \) such that \( t_i \text{emb} t_j \).

(b) Every stable monotonic transitive...

\[ \text{minus}(0, 0), \text{succ}(0), 0, \text{minus}(s(0), 0) \]
(b) Every stable, monotonic, transitive relation > that satisfies the subterm property \( f(x_1, \ldots, x_n) > x_i \) for all \( 1 \leq i \leq n \) contains the embedding order (i.e., \( s \succemb t \Rightarrow s > t \)).

(c) Every stable, monotonic, transitive, irreflexive relation that satisfies the subterm property is well founded. Thus, it is a simplification order.

Proof: (a) See literature.

(b) To show: \( S \succemb T \) implies \( S > T \)

(Simple proof by structural induction on \( S \).)

(c) \( > \) is stable, monotonic, transitive, contains \( \succemb \), irreflexive.

To show: \( > \) is well founded.
Assume that there is an infinite sequence
\[ t_0 > t_1 > t_2 > \ldots \]
Let \( \sigma \) replace all variables in \( t_0, t_1, \ldots \) by ground terms. Then stability of \( > \) implies:
\[ t_0 > t_1 \sigma, t_2 \sigma > \ldots \]

By Knuth’s Theorem (a) there exist \( i, j \) such that
\[ t_i \sigma \succemb t_j \sigma \]

By Knuth’s Theorem (a) there exist \( i, j \) such that
\[ t_i \sigma \succemb t_j \sigma \]
Since $\tau$ contains $\tau_2$, we also have: $t_1 \tau \leq t_j \tau$.

Thus: $t_i \tau_2 \leq t_j \tau_2 \leq t_i \tau$

By transitivity of $\tau$: $t_i \tau > t_i \tau$

which contradicts irreflexivity of $\tau$.

**Nax:** Define suitable relations and prove that they satisfy stability, monotonicity, transitivity, irreflexivity, subterm property.

**Ideas:** Take the embedding order and improve its two main drawbacks:

- Weak comparison of terms $S$ and $T$ if
  \[ S = f(...), T = g(...), \]
  weak comparison of terms $S$ and $T$ if
  \[ S = f(...), T = f(...). \] Here, one needs better ways to compare tuples of terms.

There are two main ways to compare tuples:

- lexicographically or

- as multisets

Lexicographic combination of two relations allows us to compare tuples of objects.

**Def 443 (Lexicographic Combination of Relations)**

Let $\succ_1$ be a relation on a set $T_1$ and $\succ_2$ be a relation
Let \(\succeq_1\) be a relation on a set \(T_1\) and \(\succeq_2\) be a relation on a set \(T_2\) (i.e., \(\succeq_i \subseteq T_i \times T_i\)). Then the **lexicographic combination** \(\succeq_{1 \times 2}\) is a relation on \(T_1 \times T_2\), which is defined as follows:

\[(S_1, S_2) \succeq_{1 \times 2} (t_1, t_2) \text{ iff } S_1 \succeq_1 t_1 \text{ or } (S_1 = t_1 \text{ and } S_2 \succeq_2 t_2)\]

Similarly, one can define the lexicographic combination of arbitrary many relations \(\succeq_1, \ldots, \succeq_n:\)

\[(S_1, \ldots, S_n) \succeq_{1 \times \ldots \times n} (t_1, \ldots, t_n) \text{ iff there exists an } i \in \{1, \ldots, n\} \text{ with } S_i \succeq_i t_i \text{ and } S_j = t_j \text{ for all } 1 \leq j \neq i.

The \(n\)-fold lexicographic combination of a relation \(\succeq\) with itself is denoted \(\succeq^n\). (This is the case where \(\succeq = \succeq_1 = \ldots = \succeq_n\)).

**Example 4.4.4.** \((3, 5) \ (\succeq^2_{1 \times}) \lex (2, 6) \ (\succeq^2_{1 \times}) \lex (2, 5)\)

Words in a lexicon are also ordered lexicographically. Let \(\succeq_{\text{alph}}\) be the order of letters in the alphabet, i.e., \(\alpha \succ_{\text{alph}} \beta \succ_{\text{alph}} \ldots \succ_{\text{alph}} \zeta\).

\(\text{hans} \ (\succeq_{\text{alph}})^4 \lex \text{hugo} \ (\succeq_{\text{alph}})^4 \lex \text{juli}.

**Well-foundedness is maintained under lexicographic combination.**
Graphic combinations.

But the order in a lexicon is not well founded:

\[ a > ba > bba > bbb a > \ldots \]

Reason: Here, the length of the tuples is not bounded.

Then 4.45 (Well-Foundedness of Lexicographic Combinations)

Let \( \succ_{\alpha} \) be a relation on \( T_1 \neq \emptyset \) and \( \succ_{\beta} \) be a relation on \( T_2 \neq \emptyset \). Then \( \succ_{\alpha} \) and \( \succ_{\beta} \) are well founded if their lexicographic combination \( \succ_{\alpha \times \beta} \) is well founded.

Proof: "\( \Leftarrow \)" Let \( \succ_{\alpha \times \beta} \) be well founded.

If \( \succ_{\alpha} \) were not well founded, then there would exist a sequence \( \mu_0 \succ_{\alpha} \mu_1 \succ_{\alpha} \mu_2 \succ_{\alpha} \mu_3 \succ_{\alpha} \ldots \)

Let \( v \in T_2 \). Then \((\mu_0, v) \succ_{\alpha \times \beta} (\mu_1, v) \succ_{\alpha \times \beta} (\mu_2, v) \succ_{\alpha \times \beta} \ldots \)

Similarly, if \( \succ_{\beta} \) were not well founded, then there would exist a sequence \( v_0 \succ_{\beta} v_1 \succ_{\beta} v_2 \succ_{\beta} \ldots \)

Let \( \mu \in T_1 \). Then \((\mu, v_0) \succ_{\alpha \times \beta} (\mu, v_1) \succ_{\alpha \times \beta} (\mu, v_2) \succ_{\alpha \times \beta} \ldots \)

"\( \Rightarrow \)" Assume that \( \succ_{\alpha \times \beta} \) were not well founded:

\((\mu_0, v_0) \succ_{\alpha \times \beta} (\mu_1, v_1) \succ_{\alpha \times \beta} (\mu_2, v_2) \succ_{\alpha \times \beta} \ldots \)

Thus: \( \mu_0 \succ_{\alpha} \mu_1 \succ_{\alpha} \mu_2 \succ_{\alpha} \ldots \)

Since \( \succ_{\alpha} \) is well founded, there is an \( i \in \mathbb{N} \) such that
\[ M_i = M_{i+1} = M_{i+2} = \ldots \]

Thus: \[ V_i \succ V_{i+1} \succ V_{i+2} \succ \ldots \]

This contradicts well-foundedness of \( \succ \).

Now we define the lexicographic path order, which is a more powerful simplification order than the embedding order.

- \( \mathcal{LPO} \) should again contain the subterm relation \( \mathcal{D} \).
- First condition for \( \mathcal{LPO} \) is the same as for \( \mathcal{D} \).
- \( \mathcal{D} \) is weak when comparing terms \( f(s_1, \ldots, s_n) \) and \( g(t_1, \ldots, t_m) \), e.g., \( \text{plus}(\text{succ}(x), y) \) \( \succ \) \( \text{succ}(\text{plus}(x, y)) \).

Solution: Assign different weights to function symbols.

We use an order \( I \) on function symbols (precedence).

If \( f \succ g \), then \( f(s_1, \ldots, s_n) \succ \mathcal{LPO} g(t_1, \ldots, t_m) \).

To make \( \mathcal{LPO} \) well-founded, \( I \) must also be well founded.

If \( f \succ g \), then: \( f(x) \succ \mathcal{LPO} g(f(x)) \)

By first condition: \( g(f(x)) \succ \mathcal{LPO} f(x) \)

Solution:

\[ f(s_1, \ldots, s_n) \succ \mathcal{LPO} g(t_1, \ldots, t_m) \]

if \( f \succ g \) and
\[ f(s_1, \ldots, s_n) \gtrdot_{lpo} t_1, \ldots, f(s_1, \ldots, s_n) \gtrdot_{lpo} t_m \]

\( S_{\text{lex}} \) is also weak when comparing terms that start with the same \( \text{fct. symbol} \):

\[ f(s_1, \ldots, s_n) \gtrdot_{lpo} f(t_1, \ldots, t_n) \text{ if } (s_1, \ldots, s_n) \gtrdot_{lpo} (t_1, \ldots, t_n) \]

This means: \( s_1 = t_1, s_2 = t_2, \ldots, s_{i-1} = t_{i-1}, S_i \gtrdot_{lpo} t_i \)

Is this enough to guarantee well-foundedness of \( lpo \)?

\[ f(\text{succ}(0), 0) \gtrdot_{lpo} f(0, f(\text{succ}(0), 0)) \]

Since \( \text{succ}(0) \gtrdot_{lpo} 0 \)

To prevent this, we define:

\[ f(s_1, \ldots, s_n) \gtrdot_{lpo} f(t_1, \ldots, t_n) \text{ if } \]

\[ s_1 = t_1, \ldots, s_{i-1} = t_{i-1}, S_i \gtrdot_{lpo} t_i, \]

\[ f(s_1, \ldots, s_n) \gtrdot_{lpo} t_{i+1}, \ldots, f(s_1, \ldots, s_n) \gtrdot_{lpo} t_n \]

\[ \text{Def 4.4.6 (lexicographic Path Order, Kamin+Levy 1980)} \]

- see slide -

We will show that the \( \text{CPO} \) is a simplification
order \Rightarrow LPO can be used for termination proofs of TRSs.

\text{Ex 4.47}

\text{plus} (\sigma, y) \rightarrow y
\text{plus} (s(x), y) \rightarrow s(\text{plus}(x, y))
\text{times} (\sigma, y) \rightarrow \sigma
\text{times} (s(x), y) \rightarrow \text{plus}(y, \text{times}(x, y))

Rules 1 and 3 are decreasing w.r.t. \text{Lpo} since \text{Lpo} contains 0.
\text{plus}(s(x), y) \succ_{\text{Lpo}} s(\text{plus}(x, y))
\text{times}(s(x), y) \succ_{\text{Lpo}} \text{plus}(y, \text{times}(x, y))
\text{times}(s(x), y) \succ_{\text{Lpo}} \text{plus}(y, \text{times}(x, y))
\text{times}(s(x), y) \succ_{\text{Lpo}} \text{times}(x, y)

So this shows that all rules can be oriented by LPO if one uses a precedence with \text{times} \preceq \text{plus} \succ \text{succ}.

To prove termination with LPO:
• start with empty precedence I.
• Orient one rule after another and extend I on demand.
• Whenever I is extended, make sure that I remains well founded.

Checking whether a TNS can be oriented with some LPO is decidable (since \( \Sigma \) is finite and thus there are only finitely many possible precedences). (This is an NP-complete problem that can be implemented efficiently using SAT solvers.)

Ex. 4.48

\[
\text{sum}(0, y) \rightarrow y
\]

\[
\text{sum}(s(x), y) \rightarrow \text{sum}(x, s(y))
\]

The embedding order fails for the second rule: we have \( s(x) \geq_{emb} x \), but \( y \not\geq_{emb} s(y) \).

But:

\[
\text{sum}(s(x), y) \geq_{epo} \text{sum}(x, s(y)),
\]

Since \( s(x) \geq_{epo} x \), \( \text{sum}(s(x), y) \geq_{epo} s(y) \) requires \( \text{sum} I \succ_{epo} \).

Now we prove that LPO can indeed be used for termination proofs (i.e., that it is a simplification order).
Theorem 4.4.9 (Properties of LPO)

The lexicographic path order is a simplification order.

Proof: We have to prove that LPO

- has subterm property
- is monotonic
- is stable
- is transitive
- is irreflexive

This implies that LPO is a simplification order by Thm 4.4.2.

Subterm Property

\[ f(x_1, \ldots, x_n) \geq_{LPO} x_i, \text{ since } x_i \geq_{LPO} x_i. \]

Monotonicity

Show that \( s \geq_{LPO} t \) implies \( q[s]_\pi \geq_{LPO} q[t]_\pi \).

Can easily be proved by structural induction on \( \pi \).

Stability

Show that \( s \geq_{LPO} t \) implies \( s_0 \geq_{LPO} t_0 \).

We prove this claim by structural induction with the relation \( \geq^{2}_{lex} \). It is well-founded by Thm 4.4.5.

This means: When proving the claim for \((s, t)\), we can use as induction hypothesis that it already holds for all \((s', t')\) where \( s \geq^{2}_{lex} s' \) or \( s = s' \) and \( t \geq^{2}_{lex} t' \).

Case analysis according to the def. of LPO.
We have \( s \geq_{lep_0} t \) and want to show \( s \\geq_{lep_0} t \).

**Case 1:** \( s = f(s_1, \ldots, s_n), \ s_i \geq_{lep_0} t \)

By ind. hyp.: \( s_i \geq_{lep_0} t \)

Thus: \( s = f(s_1, \ldots, s_n) \geq_{lep_0} t \).

**Case 2:** \( s = f(s_1, \ldots, s_n), \ t = g(t_1, \ldots, t_m), \ f \geq_{lep_0} g \)

By ind. hyp.: \( s \geq_{lep_0} t \)

Thus: \( s = f(s_1, \ldots, s_n), \ t = g(t_1, \ldots, t_m), \ f \geq_{lep_0} g, \ s \geq_{lep_0} t \)

\( \Rightarrow \) \( s \geq_{lep_0} t \)

**Case 3:** \( s = f(s_1, \ldots, s_i, \ldots, s_n), \ t = f(s_1, \ldots, s_i, \ldots, s_n), \ s_i \geq_{lep_0} t_i \)

By ind. hyp.: \( s_i \geq_{lep_0} t_i \)

Thus: \( s \geq_{lep_0} t \)

\( \Rightarrow \) \( s \geq_{lep_0} t \)

**Transitivity:** can be proved by induction on \( \Delta_3 \).

**Irreflexivity:** We show that \( s \nleq_{lep_0} s \) holds by structural induction on \( s \).

**Case 1:** \( s = f(s_1, \ldots, s_n), \ s_i \geq_{lep_0} s \)

We also have \( s \geq_{lep_0} s_i \).

This implies \( s_i \geq_{lep_0} s_i \) by transitivity. \( y \) to the ind. hypothesis.

**Case 2:** \( s = f(s_1, \ldots, s_n), \ f \geq f \) would contradict well-founded.
Contradict well-foundedness of $\mathcal{I}$.

Case 3: $S = f(S_1, \ldots, S_n)$ with $S_i \geq_{\text{po}} S_i \stackrel{\text{ind. hyp.}}{\Rightarrow} S_i$.

LPO compares arguments lexicographically from left to right. But one could also compare arguments from right to left (or in any other permutation).

\[
\begin{align*}
pred(O) & \rightarrow O \\
pred(succ(x)) & \rightarrow x \\
minus(x, O) & \rightarrow x \\
minus(x, succ(y)) & \rightarrow minus(pred(x), y)
\end{align*}
\]

The last rule is not decreasing with $\geq_{\text{po}}$, because $x \not\geq_{\text{po}} pred(x)$.

Solution: for minus, the arguments should be compared from right to left. Then

\[
\begin{align*}
\minus(x, s(y)) & \geq \minus(p(x), y), \\
\text{since } S(y) & \geq y \text{ and } \minus(x, s(y)) \geq p(y) \text{ (if } minus \rightarrow pred) 
\end{align*}
\]

To make LPO stronger: LPOS (LPO with status).

- every function symbol of arity $n$ gets a
status (a permutation of 1,..,n)

- when comparing two terms \( f(\ldots) \) and \( g(\ldots) \),
  use lexicographic comparison of the arguments
  where the status of \( f \) determines in which order
  the arguments are compared.

In the example: minus would need status \( < 2,1 \) \)

\[
\text{first compare the second arguments, then the first}
\]

sum would need status \( < 1,2 \)