• Lecture 1: Introduction, Abstract Rewriting
• Lecture 2: Term Rewriting
• Lecture 3: Combinatory Logic
• Lecture 4: Termination
• Lecture 5: Matching, Unification
• Lecture 6: Equational Reasoning, Completion
• Lecture 7: Confluence
• Lecture 8: Modularity
• Lecture 9: Strategies
• Lecture 10: Decidability
• Lecture 11: Infinitary Rewriting
Outline

- Overview
- Combinatory Logic
Termination
Introduction

Termination, Example 1

Example

\[ A(x, s(y)) \rightarrow s(A(x, y)) \]
\[ A(x, 0) \rightarrow x \]

Term: $A(0, 0)$

\[ \rightarrow_R \]

Term: $A(s(0), 0)$

\[ \rightarrow_R \]

Term: $s(A(0, 0))$

\[ \rightarrow_R \]

Term: $s(s(0))$

\[ \rightarrow_R \]

Term: $s(0)$

\[ \rightarrow_R \]
Introduction

Termination, Example 1

Example

\[ A(x, s(y)) \rightarrow s(A(x, y)) \]
\[ A(x, 0) \rightarrow x \]

Looks terminating:

- second rule makes terms smaller
- first rule makes ‘s’ move upwards
Introduction

Termination, Example 2

Example

\[ f(g(x)) \rightarrow g(f(x)) \]
Example

\[ f(g(x)) \rightarrow g(f(x)) \]

\[ f(f(g(f(g(x))))) \]
Example

\[ f(g(x)) \rightarrow g(f(x)) \]

\[ f(f(g(f(g(x)))))) \rightarrow f(f(g(g(f(x)))))) \]
Example

\[ f(g(x)) \rightarrow g(f(x)) \]

\[ f(f(g(f(g(x))))) \]
\[ \rightarrow f(f(g(g(f(x))))) \]
\[ \rightarrow f(g(f(g(f(x))))) \]
\[ \rightarrow g(g(f(g(f(x))))) \]
Example

\[
f(g(x)) \rightarrow g(f(x))
\]

\[
f(f(g(f(g(x))))))
\]

\[
\rightarrow f(f(g(g(f(x))))))
\]

\[
\rightarrow f(g(f(g(f(x))))))
\]

\[
\rightarrow g(f(f(g(f(x))))))
\]
Example

\[ f(g(x)) \rightarrow g(f(x)) \]

\[ f(f(g(f(g(x)))))) \]
\[ \rightarrow f(f(g(g(f(x)))))) \]
\[ \rightarrow f(g(f(g(f(x)))))) \]
\[ \rightarrow g(f(g(f(g(f(x)))))) \]
\[ \rightarrow g(f(g(f(g(f(x)))))) \]
Termination, Example 2

Example

\[ f(g(x)) \rightarrow g(f(x)) \]

\[ f(f(g(f(g(x))))) \]
\[ \rightarrow f(f(g(g(f(x))))) \]
\[ \rightarrow f(g(f(g(f(x))))) \]
\[ \rightarrow g(f(f(g(f(x))))) \]
\[ \rightarrow g(f(g(f(f(x))))) \]
\[ \rightarrow g(g(f(f(f(x))))) \]

Looks terminating:
• \( f \)'s move to the right
• \( g \)'s move to the left
Termination, Example 2

Example

\[ f(g(x)) \rightarrow g(f(x)) \]

\[ f(f(g(f(g(x)))))) \]
\[ \rightarrow f(f(g(g(f(x)))))) \]
\[ \rightarrow f(g(f(g(f(x)))))) \]
\[ \rightarrow g(f(f(g(f(x)))))) \]
\[ \rightarrow g(f(g(f(f(x)))))) \]
\[ \rightarrow g(g(f(f(f(x)))))) \]

Looks terminating:

- \( f \)'s move to the right
- \( g \)'s move to the left
Introduction

Termination, Example 3

Example

\[ f(g(x)) \rightarrow g(g(f(f(x)))) \]
Example

$$f(g(x)) \rightarrow g(g(f(f(x))))$$

Looks terminating:

- f’s move to the right
- g’s move to the left
Example

\[ f(g(x)) \rightarrow g(g(f(f(x)))) \]

Looks terminating:
- \( f \)'s move to the right
- \( g \)'s move to the left

But we have an infinite rewrite sequence:

\[
\begin{align*}
    f&\left( g(g(x)) \right) \\
&\rightarrow g(g(f(f(g(x)))))) \\
&\rightarrow g(g(f(g(g(f(f(x))))))) \\
&\rightarrow \ldots
\end{align*}
\]
Introduction

Termination, Example 3

Example

\[ f(g(x)) \rightarrow g(g(f(f(x)))) \]

Looks terminating:

- f’s move to the right
- g’s move to the left

But we have an infinite rewrite sequence:

\[
\begin{align*}
f(g(g(x))) & \\
\rightarrow g(g(f(f(g(x))))) & \\
\rightarrow g(g(f(g(g(f(f(x))))))) & \\
\rightarrow \ldots
\end{align*}
\]

We need proofs of termination!
Termination, Example 4

Example

\[ f(x, g(y)) \rightarrow f(x, x) \]
Example

$$f(x, g(y)) \rightarrow f(x, x)$$

Looks terminating:

- right side is smaller than the left side
Example

\[ f(x, g(y)) \rightarrow f(x, x) \]

Looks terminating:
- right side is smaller than the left side

But we have an infinite rewrite sequence:

\[ f(g(x), g(x)) \]
\[ \rightarrow f(g(x), g(x)) \]
\[ \rightarrow \ldots \]
Termination, Example 4

Example

\[ f(x, g(y)) \rightarrow f(x, x) \]

Looks terminating:
- right side is smaller than the left side

But we have an infinite rewrite sequence:

\[
\begin{align*}
  f(g(x), g(x)) & \rightarrow f(g(x), g(x)) \\
  & \rightarrow \ldots
\end{align*}
\]

We need proofs of termination!
Definition (Termination SN(R))

A rewrite system $R$ is terminating if there are no infinite rewrite sequences.
**Definition (Termination SN(R))**

A rewrite system \( R \) is terminating if there are no infinite rewrite sequences.

**Termination Methods 1967**

Knuth-Bendix order
Definition (Termination SN(R))

A rewrite system $R$ is terminating if there are no infinite rewrite sequences.

Termination Methods 1975

Knuth-Bendix order, polynomial interpretations
**Definition (Termination SN(R))**

A rewrite system $R$ is terminating if there are no infinite rewrite sequences.

**Termination Methods 1979**

Knuth-Bendix order, polynomial interpretations, multiset order, simple path order...
Definition (Termination SN(\(R\)))

A rewrite system \(R\) is terminating if there are no infinite rewrite sequences.

Termination Methods 1980s

Knuth-Bendix order, polynomial interpretations, multiset order, simple path order, lexicographic path order, semantic path order, recursive decomposition order, multiset path order, recursive path order, transformation order
**Definition (Termination SN(R))**

A rewrite system $R$ is terminating if there are no infinite rewrite sequences.

**Termination Methods 1990s**

Knuth-Bendix order, polynomial interpretations, multiset order, simple path order, lexicographic path order, semantic path order, recursive decomposition order, multiset path order, recursive path order, transformation order, elementary interpretations, type introduction, well-founded monotone algebras, general path order, semantic labeling, dummy elimination, dependency pairs, freezing, top-down labeling
Definition (Termination SN(\(R\)))

A rewrite system \(R\) is terminating if there are no infinite rewrite sequences.

Termination Methods 2000s

Knuth-Bendix order, polynomial interpretations, multiset order, simple path order, lexicographic path order, semantic path order, recursive decomposition order, multiset path order, recursive path order, transformation order, elementary interpretations, type introduction, well-founded monotone algebras, general path order, semantic labeling, dummy elimination, dependency pairs, freezing, top-down labeling, monotonic semantic path order, context-dependent interpretations, match-bounds, size-change principle, matrix interpretations, predictive labeling, uncurrying, bounded increase, quasi-periodic interpretations, arctic interpretations, increasing interpretations, root-labeling, ...
**Definition (Termination SN(R))**

A rewrite system $R$ is terminating if there are no infinite rewrite sequences.

**Termination Methods**

Knuth-Bendix order, *polynomial interpretations*, multiset order, simple path order, lexicographic path order, semantic path order, recursive decomposition order, multiset path order, recursive path order, transformation order, elementary interpretations, type introduction, *well-founded monotone algebras*, general path order, semantic labeling, dummy elimination, *dependency pairs*, freezing, top-down labeling, monotonic semantic path order, context-dependent interpretations, match-bounds, size-change principle, matrix interpretations, predictive labeling, uncurrying, bounded increase, quasi-periodic interpretations, arctic interpretations, increasing interpretations, root-labeling, ...
Termination Research

AProVE, Cariboo, CiME, Jambox, Termptation, Matchbox, MuTerm, NTI, Torpa, TPA, TTT2, VMTL, ...
Termination Research

Termination Tools

AProVE, Cariboo, CiME, Jambox, Termtation, Matchbox, MuTerm, NTI, Torpa, TPA, TTT², VMTL, ...
Outline

- Overview
- Combinatory Logic
**Lemma**

*TRS $R$ is terminating* 

$\iff$

$\exists$ well-founded order $>$ on terms such that $s > t$ whenever $s \rightarrow^*_R t$
**Lemma**

TRS $R$ is terminating

$\iff$

$\exists$ well-founded order $\succ$ on terms such that $s \succ t$ whenever $s \rightarrow_R t$

**Example**

- TRS

\[
0 + y \rightarrow y \quad s(x) + y \rightarrow s(x + y)
\]
Lemma

TRS $R$ is terminating

\[ \iff \exists \text{ well-founded order } > \text{ on terms such that } s > t \text{ whenever } s \rightarrow_R t \]

Example

- TRS

\[
0 + y \to y \quad s(x) + y \to s(x + y)
\]

- well-founded order $>$

\[
s > t \iff \varphi(s) > \varphi(t) \text{ with } \varphi(u) = \begin{cases} 
1 & \text{if } u = 0 \\
\varphi(v) + 1 & \text{if } u = s(v) \\
2\varphi(v) + \varphi(w) & \text{if } u = v + w \\
0 & \text{otherwise}
\end{cases}
\]
Lemma

**TRS** $R$ is terminating

$\iff$

$\exists$ well-founded order $>$ on terms such that $s > t$ whenever $s \to_R t$

Example

- **TRS**
  
  
  $0 + y \to y$  
  
  $s(x) + y \to s(x + y)$

- well-founded order $>$

  $s > t \iff \varphi(s) \succ \mathbb{N} \varphi(t)$ with $
  \varphi(u) = \begin{cases} 
  1 & \text{if } u = 0 \\
  \varphi(v) + 1 & \text{if } u = s(v) \\
  2\varphi(v) + \varphi(w) & \text{if } u = v + w \\
  0 & \text{otherwise} \end{cases}$

Remark

*(very) inconvenient to check all rewrite steps*
## Definitions

A **reduction order** is well-founded order $\succ$ on terms which is

- closed under contexts $s \succ t \implies C[s] \succ C[t]$
- closed under substitutions $s \succ t \implies s\sigma \succ t\sigma$
**Definitions**

A reduction order is well-founded order $>_{\text{on terms}}$ on terms which is

- closed under contexts $s > t \implies C[s] > C[t]$
- closed under substitutions $s > t \implies s\sigma > t\sigma$

**Notation**

$R \subseteq >$ if $\ell > r$ for all rules $\ell \rightarrow r$ in $R$
Definitions

A reduction order is well-founded order $\succ$ on terms which is

- closed under contexts $s \succ t \implies C[s] \succ C[t]$
- closed under substitutions $s \succ t \implies s\sigma \succ t\sigma$

Notation

$R \subseteq \succ$ if $\ell \succ r$ for all rules $\ell \rightarrow r$ in $R$

Theorem

$TRS R$ is terminating $\iff R \subseteq \succ$ for reduction order $\succ$
Theorem

TRS $R$ is terminating \iff $R \subseteq \succ$ for reduction order $\succ$

Proof $\Rightarrow$.

Let $R$ be terminating. We define $\succ = \rightarrow$. Then:

- $\succ$ is well-founded since $R$ is terminating,
- $\succ$ is closed under substitutions since $\rightarrow$ is, and
- $\succ$ is closed under contexts since $\rightarrow$ is.

Hence $\succ$ is a reduction order. Moreover $R \subseteq \rightarrow$. 
Theorem

TRS $R$ is terminating $\iff R \subseteq >$ for reduction order $>.$

Proof $\Rightarrow$.
Let $R$ be terminating.
**Theorem**

TRS $R$ is terminating $\iff R \subseteq >$ for reduction order $>.$

**Proof $\Rightarrow.$**

Let $R$ be terminating.

We define $> = \rightarrow.$
Theorem

TRS $R$ is terminating $\iff R \subseteq >$ for reduction order $>$

Proof $\Rightarrow$.

Let $R$ be terminating.

We define $> = \rightarrow$.

Then:

- $>$ is well-founded since $R$ is terminating,
Well-Founded Monotone Algebras

**Theorem**

\[
TRS \ R \text{ is terminating } \iff R \subseteq > \text{ for reduction order } >
\]

**Proof ⇒.**

Let \( R \) be terminating.

We define \( > = \rightarrow \).

Then:

- \( > \) is well-founded since \( R \) is terminating,
- \( > \) is closed under substitutions since \( \rightarrow \) is, and
Theorem

TRS \( R \) is terminating \( \iff \) \( R \subseteq > \) for reduction order \( > \)

Proof \( \Rightarrow \).

Let \( R \) be terminating.

We define \( > = \rightarrow \).

Then:

- \( > \) is well-founded since \( R \) is terminating,
- \( > \) is closed under substitutions since \( \rightarrow \) is, and
- \( > \) is closed under contexts since \( \rightarrow \) is.
Theorem

\[ \text{TRS } R \text{ is terminating } \iff R \subseteq > \text{ for reduction order } > \]

Proof \( \Rightarrow \).

Let \( R \) be terminating.

We define \( > = \rightarrow \).

Then:

- \( > \) is well-founded since \( R \) is terminating,
- \( > \) is closed under substitutions since \( \rightarrow \) is, and
- \( > \) is closed under contexts since \( \rightarrow \) is.

Hence \( > \) is a reduction order.
Theorem

\[ TRS \ R \ is \ terminating \quad \iff \quad R \subseteq > \ for \ reduction \ order \ > \]

Proof \( \Rightarrow \).

Let \( R \) be terminating.

We define \( > \) = \( \rightarrow \).

Then:

- \( > \) is well-founded since \( R \) is terminating,
- \( > \) is closed under substitutions since \( \rightarrow \) is, and
- \( > \) is closed under contexts since \( \rightarrow \) is.

Hence \( > \) is a reduction order. Moreover \( R \subseteq \rightarrow \).
Theorem

TRS $R$ is terminating $\iff R \subseteq >$ for reduction order $>$

Proof $\leq$.

Recall that $\rightarrow$ is the smallest relation $S$ such that:
- $R \subseteq S$,
- $S$ is closed under contexts, and
- $S$ is closed under substitutions.

Then $\rightarrow \subseteq >$ since $>$ has these properties.

Assume there exists an infinite rewrite sequence:

$t_0 \rightarrow t_1 \rightarrow t_2 \ldots$

Then also $t_0 > t_1 > t_2 \ldots$ since $\rightarrow \subseteq >$.

However, this contradicts well-foundedness of $>$. 
Well-Founded Monotone Algebras

**Theorem**

\[ TRS \ R \text{ is terminating} \iff R \subseteq > \text{ for reduction order } > \]

**Proof \( \Leftarrow \)**

Let \( > \) be reduction order such that \( R \subseteq > \).

\[ t_0 \rightarrow t_1 \rightarrow t_2 \ldots \]

Then also \( t_0 > t_1 > t_2 \ldots \) since \( \rightarrow \subseteq > \).

However, this contradicts well-foundedness of \( > \).
### Theorem

*TRS* $R$ is terminating $\iff R \subseteq >$ for reduction order $>$. 

### Proof $\Leftarrow$.

Let $>$ be reduction order such that $R \subseteq >$. 

Recall that $\rightarrow$ is the smallest relation $S$ such that:

- $R \subseteq S$,
- $S$ is closed under contexts, and
- $S$ is closed under substitutions.
Theorem

TRS $R$ is terminating $\iff R \subseteq >$ for reduction order $>$

Proof $\Leftarrow$.

Let $>$ be reduction order such that $R \subseteq >$.

Recall that $\rightarrow$ is the smallest relation $S$ such that:

- $R \subseteq S$,
- $S$ is closed under contexts, and
- $S$ is closed under substitutions.

Then $\rightarrow \subseteq >$ since $>$ has these properties.
Theorem

TRS $R$ is terminating $\iff R \subseteq >$ for reduction order $>$

Proof $\Leftarrow$.

Let $>$ be reduction order such that $R \subseteq >$.

Recall that $\rightarrow$ is the smallest relation $S$ such that:

- $R \subseteq S$,
- $S$ is closed under contexts, and
- $S$ is closed under substitutions.

Then $\rightarrow \subseteq >$ since $>$ has these properties.

Assume there exists an infinite rewrite sequence: $t_0 \rightarrow t_1 \rightarrow t_2 \ldots$
Well-Founded Monotone Algebras

Theorem

$\text{TRS } R \text{ is terminating } \iff R \subseteq > \text{ for reduction order } >$

Proof $\Leftarrow$.

Let $>$ be reduction order such that $R \subseteq >$.

Recall that $\rightarrow$ is the smallest relation $S$ such that:

- $R \subseteq S$,
- $S$ is closed under contexts, and
- $S$ is closed under substitutions.

Then $\rightarrow \subseteq >$ since $>$ has these properties.

Assume there exists an infinite rewrite sequence: $t_0 \rightarrow t_1 \rightarrow t_2 \ldots$

Then also $t_0 > t_1 > t_2 \ldots$ since $\rightarrow \subseteq >$
Theorem

TRS $R$ is terminating $\iff R \subseteq >$ for reduction order $>$. 

Proof $\Leftarrow$.

Let $>$ be reduction order such that $R \subseteq >$.

Recall that $\rightarrow$ is the smallest relation $S$ such that:

- $R \subseteq S$,
- $S$ is closed under contexts, and
- $S$ is closed under substitutions.

Then $\rightarrow \subseteq >$ since $>$ has these properties.

Assume there exists an infinite rewrite sequence: $t_0 \rightarrow t_1 \rightarrow t_2 \ldots$

Then also $t_0 > t_1 > t_2 \ldots$ since $\rightarrow \subseteq >$

However, this contradicts well-foundedness of $>$. 

Monotone algebras

Constructing Reduction Orders

Idea: give semantics to terms by interpreting them into an algebra.

Definition

A $\Sigma$-algebra $(A, [\cdot])$ consists of:

- a non-empty set $A$,
- and for every $f \in \Sigma$ an interpretation function $[f] : A^{ar(f)} \rightarrow A$.

Example (We use the $\Sigma$-algebra $(\mathbb{N}, [\cdot])$ with)

$[0] = 1$

$[s](x) = x + 1$

$[\cdot](x, y) = x + 2 \cdot y$
Constructing Reduction Orders

Idea: give semantics to terms by interpreting them into an algebra.

**Definition**

A \( \Sigma \)-algebra \((A, [\cdot])\) consists of:

- a non-empty set \( A \),
- and for every \( f \in \Sigma \) an **interpretation function** \([f]: A^{\text{ar}(f)} \rightarrow A\).
Constructing Reduction Orders

Idea: give semantics to terms by interpreting them into an algebra.

Definition

A $\Sigma$-algebra $(A, [\cdot])$ consists of:

- a non-empty set $A$,
- and for every $f \in \Sigma$ an interpretation function $[f]: A^{\text{ar}(f)} \rightarrow A$.

Example (We use the $\Sigma$-algebra $(\mathbb{N}, [\cdot])$ with)

$[0] = 1$
$[s](x) = x + 1$
$[A](x, y) = x + 2 \cdot y$
Monotone algebras

Interpretation of terms

Definition (Interpretation of Terms)
Let $\alpha : \mathcal{X} \to A$ be an interpretation of the variables. We define the evaluation of terms

$$[\cdot, \alpha] : \mathcal{T}(\Sigma, \mathcal{X}) \to A$$

inductively:

- $[x, \alpha] = \alpha(x)$ if $x \in \mathcal{X}$
- $[f(t_1, \ldots, t_n), \alpha] = [f][[t_1, \alpha], \ldots, [t_n, \alpha]]$

Example

Let $\alpha(x) = 1$, $\alpha(y) = 3$, we calculate:

- $[A(0, s(0)), \alpha] = 5$,
- $[A(s(x), y), \alpha] = 8$. 
Monotone algebras

Interpretation of terms

**Definition (Interpretation of Terms)**

Let $\alpha : \mathcal{X} \rightarrow A$ be an interpretation of the variables. We define the evaluation of terms

$\cdot, \alpha : T(\Sigma, \mathcal{X}) \rightarrow A$

inductively:

$x, \alpha = \alpha(x)$ if $x \in \mathcal{X}$

$f(t_1, \ldots, t_n), \alpha = f([t_1, \alpha], \ldots, [t_n, \alpha])$

**Example**

$[0] = 1 \quad [s](x) = x + 1 \quad [A](x, y) = x + 2 \cdot y$

Let $\alpha(x) = 1, \alpha(y) = 3$, we calculate:
Interpretation of terms

Definition (Interpretation of Terms)

Let $\alpha : \mathcal{X} \rightarrow A$ be an interpretation of the variables. We define the evaluation of terms

$$[\cdot, \alpha] : T(\Sigma, \mathcal{X}) \rightarrow A$$

inductively:

$$[x, \alpha] = \alpha(x) \quad \text{if} \ x \in \mathcal{X}$$

$$[f(t_1, \ldots, t_n), \alpha] = [f([t_1, \alpha], \ldots, [t_n, \alpha])$$

Example

$$[0] = 1 \quad [s](x) = x + 1 \quad [A](x, y) = x + 2 \cdot y$$

Let $\alpha(x) = 1$, $\alpha(y) = 3$, we calculate:

- $[A(0, s(0)), \alpha] =$

• $[A(0, s(0)), \alpha] =$

• $[A(0, s(0)), \alpha] =$
Interpretation of terms

**Definition (Interpretation of Terms)**

Let \( \alpha : \mathcal{X} \rightarrow A \) be an interpretation of the variables.

We define the evaluation of terms

\[
[\cdot, \alpha] : T(\Sigma, \mathcal{X}) \rightarrow A
\]

inductively:

\[
[x, \alpha] = \alpha(x) \quad \text{if } x \in \mathcal{X}
\]

\[
[f(t_1, \ldots, t_n), \alpha] = [f([t_1, \alpha], \ldots, [t_n, \alpha])]
\]

**Example**

\[
[0] = 1 \quad [s](x) = x + 1 \quad [A](x, y) = x + 2 \cdot y
\]

Let \( \alpha(x) = 1, \alpha(y) = 3 \), we calculate:

- \( [A(0, s(0)), \alpha] = 5 \),
Interpretation of terms

**Definition (Interpretation of Terms)**

Let $\alpha : \mathcal{X} \to A$ be an interpretation of the variables. We define the evaluation of terms

$$[\cdot, \alpha] : \mathcal{T}(\Sigma, \mathcal{X}) \to A$$

inductively:

$$[x, \alpha] = \alpha(x) \quad \text{if } x \in \mathcal{X}$$

$$[f(t_1, \ldots, t_n), \alpha] = [f([t_1, \alpha], \ldots, [t_n, \alpha])]$$

**Example**

$$[0] = 1 \quad [s](x) = x + 1 \quad [A](x, y) = x + 2 \cdot y$$

Let $\alpha(x) = 1$, $\alpha(y) = 3$, we calculate:

- $[A(0, s(0)), \alpha] = 5,$
- $[A(s(x), y), \alpha] = \ldots$
Monotone algebras

Interpretation of terms

Definition (Interpretation of Terms)

Let $\alpha : \mathcal{X} \to A$ be an interpretation of the variables. We define the evaluation of terms

$$[\cdot, \alpha] : \mathcal{T}(\Sigma, \mathcal{X}) \to A$$

inductively:

$$[x, \alpha] = \alpha(x) \quad \text{if } x \in \mathcal{X}$$

$$[f(t_1, \ldots, t_n), \alpha] = [f][[t_1, \alpha], \ldots, [t_n, \alpha]]$$

Example

$$[0] = 1 \quad [s](x) = x + 1 \quad [A](x, y) = x + 2 \cdot y$$

Let $\alpha(x) = 1$, $\alpha(y) = 3$, we calculate:

- $[A(0, s(0)), \alpha] = 5$,
- $[A(s(x), y), \alpha] = 8$. 
Monotone \(\Sigma\)-algebras

A function \([f]\) is **monotone** (w.r.t. \(>\)) if

\[
 a > b \text{ implies } [f](\ldots, a, \ldots) > [f](\ldots, b, \ldots)
\]
Monotone Σ-algebras

A function \([f]\) is **monotone** (w.r.t. \(>\)) if

\[a > b \text{ implies } [f](\ldots, a, \ldots) > [f](\ldots, b, \ldots)\]

**Definition**

A well-founded monotone Σ-algebra \((A, [\cdot], >)\):

- a Σ-algebra \((A, [\cdot])\),
- a well-founded order \(>\) on \(A\) (no infinite chains \(a_1 > a_2 > \ldots\)),
- such that for all \(f \in \Sigma\) the function \([f]_A\) is monotone.
Monotone \( \Sigma \)-algebras

A function \([f]\) is **monotone** (w.r.t. \(>\)) if

\[
a > b \text{ implies } [f](\ldots, a, \ldots) > [f](\ldots, b, \ldots)
\]

**Definition**

A well-founded monotone \( \Sigma \)-algebra \((A, [\cdot], >)\):

- a \( \Sigma \)-algebra \((A, [\cdot])\),
- a well-founded order \(>\) on \(A\)
  (no infinite chains \(a_1 > a_2 > \ldots\)),
- such that for all \(f \in \Sigma\) the function \([f]_A\) is monotone.

Relation \(>_A\) on terms: \(s > t\) if \([s, \alpha] > [t, \alpha]\) for all assignments \(\alpha\).
Monotone $\Sigma$-algebras

A function $[f]$ is monotone (w.r.t. $>$) if

$$a > b \implies [f](\ldots, a, \ldots) > [f](\ldots, b, \ldots)$$

Definition

A well-founded monotone $\Sigma$-algebra $(A, [\cdot], >)$:

- a $\Sigma$-algebra $(A, [\cdot])$,
- a well-founded order $>$ on $A$ (no infinite chains $a_1 > a_2 > \ldots$),
- such that for all $f \in \Sigma$ the function $[f]_A$ is monotone.

Relation $>_A$ on terms: $s > t$ if $[s, \alpha] > [t, \alpha]$ for all assignments $\alpha$.

Lemma

$>_A$ is a reduction order for every well-founded monotone $\Sigma$-algebra $(A, [\cdot], >)$
**Lemma**

Let $\mathcal{A}$ be a well-founded monotone $\Sigma$-algebra. Then $\triangleright_{\mathcal{A}}$ is closed under substitutions.

Let $\sigma$ be a substitution. We show $t\sigma \triangleright_{\mathcal{A}} s\sigma$.

That is: $[t\sigma, \alpha] > [s\sigma, \alpha]$ for every $\alpha : \mathcal{X} \rightarrow \mathcal{A}$.

Define $\beta : \mathcal{X} \rightarrow \mathcal{A}$ by $\beta(x) = [\sigma(x), \alpha]$. Claim: $[u\sigma, \alpha] = [u, \beta]$ for all $u$.

- Proof of the claim by induction over the term structure of $u$:

  
  $[x\sigma, \alpha] = [\sigma(x), \alpha] = [x, \beta]$

  $[f(t_1, \ldots, t_n)\sigma, \alpha] = [f([t_1\sigma, \alpha], \ldots, [t_n\sigma, \alpha])]$

  $\overset{IH}{=} [f([t_1, \beta], \ldots, [t_n, \beta])]$

  $= [f(t_1, \ldots, t_n), \beta]$  

Hence $[t\sigma, \alpha] = [t, \beta] > [s, \beta] = [s\sigma, \alpha]$. 

- - - - -
Lemma

Let \( A \) be a well-founded monotone \( \Sigma \)-algebra. Then \( >_A \) is closed under substitutions.

Proof.

Let \( s, t \in T(\Sigma, \mathcal{X}) \) such that \( t >_A s \).

...
**Lemma**

Let \( \mathcal{A} \) well-founded monotone \( \Sigma \)-algebra. Then \( \succ_{\mathcal{A}} \) is closed under substitutions.

**Proof.**

Let \( s, t \in \mathcal{T}(\Sigma, \mathcal{X}) \) such that \( t \succ_{\mathcal{A}} s \).

Let \( \sigma \) be a substitution. We show \( t\sigma \succ_{\mathcal{A}} s\sigma \).
Lemma

Let $\mathcal{A}$ be a well-founded monotone $\Sigma$-algebra. Then $>_\mathcal{A}$ is closed under substitutions.

Proof.

Let $s, t \in T(\Sigma, \mathcal{X})$ such that $t >_\mathcal{A} s$.

Let $\sigma$ be a substitution. We show $t\sigma >_\mathcal{A} s\sigma$.

That is: $[t\sigma, \alpha] > [s\sigma, \alpha]$ for every $\alpha : \mathcal{X} \to \mathcal{A}$. 
Lemma

Let $A$ be a well-founded monotone $\Sigma$-algebra. Then $\succ_{A}$ is closed under substitutions.

Proof.

Let $s, t \in T(\Sigma, X)$ such that $t \succ_{A} s$.

Let $\sigma$ be a substitution. We show $t\sigma \succ_{A} s\sigma$.

That is: $[t\sigma, \alpha] > [s\sigma, \alpha]$ for every $\alpha : X \rightarrow A$.

Define $\beta : X \rightarrow A$ by $\beta(x) = [\sigma(x), \alpha]$. 
Lemma

Let $\mathcal{A}$ well-founded monotone $\Sigma$-algebra. Then $\succ_{\mathcal{A}}$ is closed under substitutions.

Proof.

Let $s, t \in \mathcal{T}(\Sigma, \mathcal{X})$ such that $t \succ_{\mathcal{A}} s$.

Let $\sigma$ be a substitution. We show $t\sigma \succ_{\mathcal{A}} s\sigma$.

That is: $[t\sigma, \alpha] > [s\sigma, \alpha]$ for every $\alpha : \mathcal{X} \rightarrow A$.

Define $\beta : \mathcal{X} \rightarrow A$ by $\beta(x) = [\sigma(x), \alpha]$. Claim: $[u\sigma, \alpha] = [u, \beta]$ for all $u$. 

Term Rewriting Systems - Lecture 4

18/22
**Lemma**

*Let $\mathcal{A}$ be a well-founded monotone $\Sigma$-algebra. Then $\triangleright_\mathcal{A}$ is closed under substitutions.*

**Proof.**

Let $s, t \in T(\Sigma, \mathcal{X})$ such that $t \triangleright_\mathcal{A} s$.

Let $\sigma$ be a substitution. We show $t\sigma \triangleright_\mathcal{A} s\sigma$.

That is: $[t\sigma, \alpha] > [s\sigma, \alpha]$ for every $\alpha : \mathcal{X} \rightarrow \mathcal{A}$.

Define $\beta : \mathcal{X} \rightarrow \mathcal{A}$ by $\beta(x) = [\sigma(x), \alpha]$. Claim: $[u\sigma, \alpha] = [u, \beta]$ for all $u$.

- Proof of the claim by induction over the term structure of $u$: 
Lemma

Let $A$ be a well-founded monotone $\Sigma$-algebra. Then $\succ_A$ is closed under substitutions.

Proof.

Let $s, t \in T(\Sigma, X)$ such that $t \succ_A s$.

Let $\sigma$ be a substitution. We show $t\sigma \succ_A s\sigma$.

That is: $[t\sigma, \alpha] > [s\sigma, \alpha]$ for every $\alpha : X \to A$.

Define $\beta : X \to A$ by $\beta(x) = [\sigma(x), \alpha]$. Claim: $[u\sigma, \alpha] = [u, \beta]$ for all $u$.

- Proof of the claim by induction over the term structure of $u$:

  $[x\sigma, \alpha] = [\sigma(x), \alpha] = [x, \beta]$
Lemma

Let $\mathcal{A}$ well-founded monotone $\Sigma$-algebra. Then $\succ_{\mathcal{A}}$ is closed under substitutions.

Proof.

Let $s, t \in \mathcal{T}(\Sigma, \mathcal{X})$ such that $t \succ_{\mathcal{A}} s$.

Let $\sigma$ be a substitution. We show $t\sigma \succ_{\mathcal{A}} s\sigma$.

That is: $[t\sigma, \alpha] > [s\sigma, \alpha]$ for every $\alpha : \mathcal{X} \rightarrow \mathcal{A}$.

Define $\beta : \mathcal{X} \rightarrow \mathcal{A}$ by $\beta(x) = [\sigma(x), \alpha]$. Claim: $[u\sigma, \alpha] = [u, \beta]$ for all $u$.

- Proof of the claim by induction over the term structure of $u$:

\[
[x\sigma, \alpha] = [\sigma(x), \alpha] = [x, \beta] \\
[f(t_1, \ldots, t_n)\sigma, \alpha]
\]
Lemma

Let $\mathcal{A}$ be a well-founded monotone $\Sigma$-algebra. Then $\succ_\mathcal{A}$ is closed under substitutions.

Proof.

Let $s, t \in T(\Sigma, \mathcal{X})$ such that $t \succ_\mathcal{A} s$.
Let $\sigma$ be a substitution. We show $t\sigma \succ_\mathcal{A} s\sigma$.
That is: $[t\sigma, \alpha] > [s\sigma, \alpha]$ for every $\alpha : \mathcal{X} \to \mathcal{A}$.
Define $\beta : \mathcal{X} \to \mathcal{A}$ by $\beta(x) = [\sigma(x), \alpha]$. Claim: $[u\sigma, \alpha] = [u, \beta]$ for all $u$.

- Proof of the claim by induction over the term structure of $u$:
  
  $[x\sigma, \alpha] = [\sigma(x), \alpha] = [x, \beta]$
  
  $[f(t_1, \ldots, t_n)\sigma, \alpha] = [f][[t_1\sigma, \alpha], \ldots, [t_n\sigma, \alpha]]$
Lemma

Let $\mathcal{A}$ be a well-founded monotone $\Sigma$-algebra. Then $>_\mathcal{A}$ is closed under substitutions.

Proof.

Let $s, t \in T(\Sigma, \mathcal{X})$ such that $t >_\mathcal{A} s$.

Let $\sigma$ be a substitution. We show $t\sigma >_\mathcal{A} s\sigma$.

That is: $[t\sigma, \alpha] > [s\sigma, \alpha]$ for every $\alpha : \mathcal{X} \to \mathcal{A}$.

Define $\beta : \mathcal{X} \to \mathcal{A}$ by $\beta(x) = [\sigma(x), \alpha]$.

Claim: $[u\sigma, \alpha] = [u, \beta]$ for all $u$.

- Proof of the claim by induction over the term structure of $u$:
  
  $[x\sigma, \alpha] = [\sigma(x), \alpha] = [x, \beta]$

  $[f(t_1, \ldots, t_n)\sigma, \alpha] = [f([t_1\sigma, \alpha], \ldots, [t_n\sigma, \alpha])]$

  $\text{IH} \Rightarrow [f([t_1, \beta], \ldots, [t_n, \beta])]$
Lemma

Let \( \mathcal{A} \) well-founded monotone \( \Sigma \)-algebra. Then \( \succ_{\mathcal{A}} \) is closed under substitutions.

Proof.

Let \( s, t \in T(\Sigma, \mathcal{X}) \) such that \( t \succ_{\mathcal{A}} s \).

Let \( \sigma \) be a substitution. We show \( t\sigma \succ_{\mathcal{A}} s\sigma \).

That is: \([t\sigma, \alpha] > [s\sigma, \alpha]\) for every \( \alpha : \mathcal{X} \to A \).

Define \( \beta : \mathcal{X} \to A \) by \( \beta(x) = [\sigma(x), \alpha] \). Claim: \([u\sigma, \alpha] = [u, \beta]\) for all \( u \).

- Proof of the claim by induction over the term structure of \( u \):

  \[
  [x\sigma, \alpha] = [\sigma(x), \alpha] = [x, \beta] \\
  [f(t_1, \ldots, t_n)\sigma, \alpha] = [f]([t_1\sigma, \alpha], \ldots, [t_n\sigma, \alpha]) \\
  \overset{IH}= [f]([t_1, \beta], \ldots, [t_n, \beta]) \\
  = [f(t_1, \ldots, t_n), \beta]
  \]

Term Rewriting Systems - Lecture 4
Lemma

Let $\mathcal{A}$ well-founded monotone $\Sigma$-algebra. Then $>_{\mathcal{A}}$ is closed under substitutions.

Proof.

Let $s, t \in \mathcal{T}(\Sigma, \mathcal{X})$ such that $t >_{\mathcal{A}} s$.

Let $\sigma$ be a substitution. We show $t\sigma >_{\mathcal{A}} s\sigma$.

That is: $[t\sigma, \alpha] > [s\sigma, \alpha]$ for every $\alpha : \mathcal{X} \to \mathcal{A}$.

Define $\beta : \mathcal{X} \to \mathcal{A}$ by $\beta(x) = [\sigma(x), \alpha]$. Claim: $[u\sigma, \alpha] = [u, \beta]$ for all $u$.

- Proof of the claim by induction over the term structure of $u$:

  - $[x\sigma, \alpha] = [\sigma(x), \alpha] = [x, \beta]$  
  - $[f(t_1, \ldots, t_n)\sigma, \alpha] = [f([t_1\sigma, \alpha], \ldots, [t_n\sigma, \alpha])]$  
    \[= [f([t_1, \beta], \ldots, [t_n, \beta])] \quad \text{(IH)} \]
    \[= [f(t_1, \ldots, t_n), \beta] \]

Hence $[t\sigma, \alpha] = [t, \beta] > [s, \beta] = [s\sigma, \alpha]$. 

-
Lemma

Let $\mathcal{A}$ well-founded monotone $\Sigma$-algebra. Then $\succ_{\mathcal{A}}$ is closed under contexts.

We show $C[s] \succ_{\mathcal{A}} C[t]$ for every $C[\_]$.

That is: $[C[t], \alpha] > [C[s], \alpha]$ for every $\alpha : \mathcal{X} \rightarrow \mathcal{A}$.

Let $\alpha : \mathcal{X} \rightarrow \mathcal{A}$. We show $[C[t], \alpha] > [C[s], \alpha]$ by induction over structure of $C[\_]$:

- $C[\_] \equiv [\_] \Rightarrow [C[t]] = [t] > [s] = [C[s]]$
- $C[\_] = f(t_1, \ldots, t_n) \Rightarrow$ Let $i$ such that $t_i$ contains the hole $\Box$.

Then $C[u] = f(t_1, \ldots, t_i[u], \ldots, t_n)$. Hence

$[C[s], \alpha] = [f]([t_1, \alpha], \ldots, [t_i[s], \alpha], \ldots, [t_n, \alpha])$.

$[C[t], \alpha] = [f]([t_1, \alpha], \ldots, [t_i[t], \alpha], \ldots, [t_n, \alpha])$.

By IH $[t_i[t], \alpha] > [t_i[s], \alpha]$.

By monotonicity $[C[t], \alpha] > [C[s], \alpha]$.
Lemma

Let $\mathcal{A}$ be a well-founded monotone $\Sigma$-algebra. Then $\succ_{\mathcal{A}}$ is closed under contexts.

Proof.

Let $s, t \in T(\Sigma, \mathcal{X})$ such that $t \succ_{\mathcal{A}} s$. 

By induction over the structure of $C$:

- If $C \equiv ()$ then $C[t] = t > s = C[s]$.
- If $C \equiv f(t_1, \ldots, t_n)$, let $i$ such that $t_i$ contains a hole $\_$. Then $C[u] = f(t_1, \ldots, t_i[u], \ldots, t_n)$.

By IH, $t_i[u] > s = t_i[s]$. By monotonicity, $C[t] > C[s]$. 

Term Rewriting Systems - Lecture 4
**Lemma**

Let $\mathcal{A}$ be a well-founded monotone $\Sigma$-algebra. Then $\succ_\mathcal{A}$ is closed under contexts.

**Proof.**

Let $s, t \in T(\Sigma, \mathcal{X})$ such that $t \succ_\mathcal{A} s$. We show $C[s] \succ_\mathcal{A} C[t]$ for every $C[\ ]$. 

By IH $t_i[s] >_t t_i[s]$.

By monotonicity $C[t] > C[s]$.

Therefore, $C[s] \succ_\mathcal{A} C[t]$. 

Term Rewriting Systems - Lecture 4
Lemma

Let $A$ be a well-founded monotone $\Sigma$-algebra. Then $\succ_A$ is closed under contexts.

Proof.

Let $s, t \in T(\Sigma, X)$ such that $t \succ_A s$. We show $C[s] \succ_A C[t]$ for every $C[\ ]$.

That is: $[C[t], \alpha] \succ [C[s], \alpha]$ for every $\alpha : X \to A$. 
Lemma

Let $\mathcal{A}$ a well-founded monotone $\Sigma$-algebra. Then $>_\mathcal{A}$ is closed under contexts.

Proof.

Let $s, t \in T(\Sigma, \mathcal{X})$ such that $t >_\mathcal{A} s$. We show $C[s] >_\mathcal{A} C[t]$ for every $C[\ ]$.

That is: $[C[t], \alpha] > [C[s], \alpha]$ for every $\alpha : \mathcal{X} \rightarrow A$.

Let $\alpha : \mathcal{X} \rightarrow A$. 


Lemma

Let $\mathcal{A}$ be a well-founded monotone $\Sigma$-algebra. Then $\succ_{\mathcal{A}}$ is closed under contexts.

Proof.

Let $s, t \in T(\Sigma, \mathcal{X})$ such that $t \succ_{\mathcal{A}} s$. We show $C[s] \succ_{\mathcal{A}} C[t]$ for every $C[\ ]$.

That is: $[C[t], \alpha] > [C[s], \alpha]$ for every $\alpha : \mathcal{X} \to A$.

Let $\alpha : \mathcal{X} \to A$. We show $[C[t], \alpha] > [C[s], \alpha]$ by induction over the structure of $C[\ ]$:
Lemma

Let $\mathcal{A}$ be a well-founded monotone $\Sigma$-algebra. Then $\succ_{\mathcal{A}}$ is closed under contexts.

Proof.

Let $s, t \in T(\Sigma, \mathcal{X})$ such that $t \succ_{\mathcal{A}} s$. We show $C[s] \succ_{\mathcal{A}} C[t]$ for every $C[\ ]$.

That is: $[C[t], \alpha] \succ [C[s], \alpha]$ for every $\alpha : \mathcal{X} \rightarrow \mathcal{A}$.

Let $\alpha : \mathcal{X} \rightarrow \mathcal{A}$. We show $[C[t], \alpha] \succ [C[s], \alpha]$ by induction over structure of $C[\ ]$:

$C[\ ] \equiv [\ ] \Rightarrow \ldots \Rightarrow [C[t], \alpha] \succ [C[s], \alpha]$. 

By monotonicity $[C[t], \alpha] \succ [C[s], \alpha]$. 

By induction $[C[t], \alpha] \succ [C[s], \alpha]$.
Lemma

Let $\mathcal{A}$ a well-founded monotone $\Sigma$-algebra. Then $\succ_\mathcal{A}$ is closed under contexts.

Proof.

Let $s, t \in T(\Sigma, \mathcal{X})$ such that $t \succ_\mathcal{A} s$. We show $C[s] \succ_\mathcal{A} C[t]$ for every $C[\ ]$.

That is: $[C[t], \alpha] > [C[s], \alpha]$ for every $\alpha : \mathcal{X} \to \mathcal{A}$.

Let $\alpha : \mathcal{X} \to \mathcal{A}$. We show $[C[t], \alpha] > [C[s], \alpha]$ by induction over structure of $C[\ ]$:

$C[\ ] \equiv [\ ] \Rightarrow [C[t]] = [t] > [s] = [C[s]]$
**Lemma**

Let $\mathcal{A}$ be a well-founded monotone $\Sigma$-algebra. Then $>_{\mathcal{A}}$ is closed under contexts.

**Proof.**

Let $s, t \in T(\Sigma, \mathcal{X})$ such that $t >_{\mathcal{A}} s$. We show $C[s] >_{\mathcal{A}} C[t]$ for every $C[\ ]$.

That is: $[C[t], \alpha] > [C[s], \alpha]$ for every $\alpha : \mathcal{X} \to A$.

Let $\alpha : \mathcal{X} \to A$. We show $[C[t], \alpha] > [C[s], \alpha]$ by induction over structure of $C[\ ]$:

- $C[\ ] \equiv [ ] \Rightarrow [C[t]] = [t] > [s] = [C[s]]$
- $C[\ ] \equiv f(t_1, \ldots, t_n) \Rightarrow$
**Lemma**

Let $\mathcal{A}$ well-founded monotone $\Sigma$-algebra. Then $>_\mathcal{A}$ is closed under contexts.

**Proof.**

Let $s, t \in T(\Sigma, \mathcal{X})$ such that $t >_\mathcal{A} s$. We show $C[s] >_\mathcal{A} C[t]$ for every $C[\ ]$.

That is: $[C[t], \alpha] > [C[s], \alpha]$ for every $\alpha : \mathcal{X} \rightarrow A$.

Let $\alpha : \mathcal{X} \rightarrow A$. We show $[C[t], \alpha] > [C[s], \alpha]$ by induction over structure of $C[\ ]$:

- $C[\ ] \equiv [\ ] \Rightarrow [C[t]] = [t] > [s] = [C[s]]$
- $C[\ ] \equiv f(t_1, \ldots, t_n) \Rightarrow$ Let $i$ such that $t_i$ contains the hole $\square$. 


Lemma

Let $\mathcal{A}$ be a well-founded monotone $\Sigma$-algebra. Then $\succ_{\mathcal{A}}$ is closed under contexts.

Proof.

Let $s, t \in T(\Sigma, \mathcal{X})$ such that $t \succ_{\mathcal{A}} s$. We show $C[s] \succ_{\mathcal{A}} C[t]$ for every $C[\ ]$.

That is: $[C[t], \alpha] \succ [C[s], \alpha]$ for every $\alpha : \mathcal{X} \to A$.

Let $\alpha : \mathcal{X} \to A$. We show $[C[t], \alpha] \succ [C[s], \alpha]$ by induction over structure of $C[\ ]$:

- $C[\ ] \equiv [\ ] \Rightarrow [C[t]] = [t] > [s] = [C[s]]$
- $C[\ ] \equiv f(t_1, \ldots, t_n) \Rightarrow$ Let $i$ such that $t_i$ contains the hole $\square$.
  
  Then $C[u] = f(t_1, \ldots, t_i[u], \ldots, t_n)$. Hence
Lemma

Let $\mathcal{A}$ be a well-founded monotone $\Sigma$-algebra. Then $>_\mathcal{A}$ is closed under contexts.

Proof.

Let $s, t \in T(\Sigma, \mathcal{X})$ such that $t >_\mathcal{A} s$. We show $C[s] >_\mathcal{A} C[t]$ for every $C[\ ]$.

That is: $[C[t], \alpha] > [C[s], \alpha]$ for every $\alpha : \mathcal{X} \to \mathcal{A}$.

Let $\alpha : \mathcal{X} \to \mathcal{A}$. We show $[C[t], \alpha] > [C[s], \alpha]$ by induction over structure of $C[\ ]$:

- $C[\ ] \equiv [ ]$ $\Rightarrow$ $[C[t]] = [t] > [s] = [C[s]]$
- $C[\ ] \equiv f(t_1, \ldots, t_n)$ $\Rightarrow$ Let $i$ such that $t_i$ contains the hole $\square$.
  Then $C[u] = f(t_1, \ldots, t_i[u], \ldots, t_n)$. Hence
  $[C[s], \alpha] = [f](\ldots, [t_i[s], \alpha], \ldots, [t_n, \alpha]).$
Lemma

Let $\mathcal{A}$ well-founded monotone $\Sigma$-algebra. Then $>_\mathcal{A}$ is closed under contexts.

Proof.

Let $s, t \in T(\Sigma, \mathcal{X})$ such that $t >_\mathcal{A} s$. We show $C[s] >_\mathcal{A} C[t]$ for every $C[\ ]$.

That is: $[C[t], \alpha] > [C[s], \alpha]$ for every $\alpha : \mathcal{X} \to \mathcal{A}$.

Let $\alpha : \mathcal{X} \to \mathcal{A}$. We show $[C[t], \alpha] > [C[s], \alpha]$ by induction over structure of $C[\ ]$:

- $C[\ ] \equiv [\ ] \Rightarrow [C[t]] = [t] > [s] = [C[s]]$
- $C[\ ] \equiv f(t_1, \ldots, t_n) \Rightarrow$ Let $i$ such that $t_i$ contains the hole $\square$.
  - Then $C[u] = f(t_1, \ldots, t_i[u], \ldots, t_n)$. Hence
    
    $[C[s], \alpha] = [f([t_1, \alpha], \ldots, [t_i[s], \alpha], \ldots, [t_n, \alpha])].$
    
    $[C[t], \alpha] = [f([t_1, \alpha], \ldots, [t_i[t], \alpha], \ldots, [t_n, \alpha])].$
Lemma

Let $\mathcal{A}$ well-founded monotone $\Sigma$-algebra. Then $\succ_{\mathcal{A}}$ is closed under contexts.

Proof.

Let $s, t \in \mathcal{T}(\Sigma, \mathcal{X})$ such that $t \succ_{\mathcal{A}} s$. We show $C[s] \succ_{\mathcal{A}} C[t]$ for every $C[\ ]$.

That is: $[C[t], \alpha] \succ [C[s], \alpha]$ for every $\alpha : \mathcal{X} \to \mathcal{A}$.

Let $\alpha : \mathcal{X} \to \mathcal{A}$. We show $[C[t], \alpha] \succ [C[s], \alpha]$ by induction over structure of $C[\ ]$:

- $C[\ ] \equiv [\ ] \Rightarrow [C[t]] = [t] \succ [s] = [C[s]]$
- $C[\ ] \equiv f(t_1, \ldots, t_n) \Rightarrow$ Let $i$ such that $t_i$ contains the hole $\Box$.

Then $C[u] = f(t_1, \ldots, t_i[u], \ldots, t_n)$. Hence

$[C[s], \alpha] = [f]( [t_1, \alpha], \ldots, [t_i[s], \alpha], \ldots, [t_n, \alpha])$.

$[C[t], \alpha] = [f]( [t_1, \alpha], \ldots, [t_i[t], \alpha], \ldots, [t_n, \alpha])$.

By IH $[t_i[t], \alpha] \succ [t_i[s], \alpha]$. 


Monotone algebras

Lemma

Let $\mathcal{A}$ be a well-founded monotone $\Sigma$-algebra. Then $\succ_{\mathcal{A}}$ is closed under contexts.

Proof.

Let $s, t \in T(\Sigma, \mathcal{X})$ such that $t \succ_{\mathcal{A}} s$. We show $C[s] \succ_{\mathcal{A}} C[t]$ for every $C[\ ]$.

That is: $[C[t], \alpha] \succ [C[s], \alpha]$ for every $\alpha : \mathcal{X} \rightarrow A$.

Let $\alpha : \mathcal{X} \rightarrow A$. We show $[C[t], \alpha] \succ [C[s], \alpha]$ by induction over structure of $C[\ ]$:

- $C[\ ] \equiv [$ $\Rightarrow [C[t]] = [t] \succ [s] = [C[s]]$
- $C[\ ] \equiv f(t_1, \ldots, t_n) \Rightarrow$ Let $i$ such that $t_i$ contains the hole $\square$.

Then $C[u] = f(t_1, \ldots, t_i[u], \ldots, t_n)$. Hence

$[C[s], \alpha] = [f([t_1, \alpha], \ldots, [t_i[s], \alpha], \ldots, [t_n, \alpha])]$.

$[C[t], \alpha] = [f([t_1, \alpha], \ldots, [t_i[t], \alpha], \ldots, [t_n, \alpha])]$.

By IH $[t_i[t], \alpha] > [t_i[s], \alpha]$.

By monotonicity $[C[t], \alpha] > [C[s], \alpha]$. 

\[\square\]
Lemma

Let $\mathcal{A}$ be a well-founded monotone $\Sigma$-algebra. Then $\succ_{\mathcal{A}}$ is well-founded.

Let $\alpha : \mathcal{X} \to \mathcal{A}$. Then by definition of $\succ_{\mathcal{A}}$: $[t_1, \alpha] > [t_2, \alpha] > [t_3, \alpha] > \ldots$. This infinite decreasing $\succ$ chain contradicts well-foundedness of $\succ$. 

Term Rewriting Systems - Lecture 4
**Lemma**

Let $\mathcal{A}$ a well-founded monotone $\Sigma$-algebra. Then $>_\mathcal{A}$ is well-founded.

**Proof.**

Assume there is an infinite sequence $t_1 >_\mathcal{A} t_2 >_\mathcal{A} t_3 >_\mathcal{A} \ldots$. 

Let $\alpha : X \to \mathcal{A}$. Then by definition of $>_\mathcal{A}$:

\[
\alpha(t_1) >_\mathcal{A} \alpha(t_2) >_\mathcal{A} \alpha(t_3) >_\mathcal{A} \ldots
\]

This infinite decreasing chain contradicts well-foundedness of $>_\mathcal{A}$.

We have shown that:

- $>_\mathcal{A}$ is closed under substitutions
- $>_\mathcal{A}$ is closed under contexts
- $>_\mathcal{A}$ is well-founded

Hence we have proven that: $\mathcal{A}$ is a reduction order for every well-founded monotone $\Sigma$-algebra $(\mathcal{A}, [\cdot], >_\mathcal{A})$.
Lemma

Let $\mathcal{A}$ well-founded monotone $\Sigma$-algebra. Then $\succ_{\mathcal{A}}$ is well-founded.

Proof.

Assume there is an infinite sequence $t_1 \succ_{\mathcal{A}} t_2 \succ_{\mathcal{A}} t_3 \succ_{\mathcal{A}} \ldots$.
Let $\alpha : \mathcal{X} \rightarrow A$. Then by definition of $\succ_{\mathcal{A}}$: $[t_1, \alpha] \succ [t_2, \alpha] \succ [t_3, \alpha] > \ldots$. 
Lemma

Let \( \mathcal{A} \) well-founded monotone \( \Sigma \)-algebra. Then \( \succ_{\mathcal{A}} \) is well-founded.

Proof.

Assume there is an infinite sequence \( t_1 \succ_{\mathcal{A}} t_2 \succ_{\mathcal{A}} t_3 \succ_{\mathcal{A}} \ldots \).

Let \( \alpha : \mathcal{X} \rightarrow \mathcal{A} \). Then by definition of \( \succ_{\mathcal{A}} \): \([t_1, \alpha] \succ [t_2, \alpha] \succ [t_3, \alpha] \succ \ldots \).

This infinite decreasing \( \succ \) chain contradicts well-foundedness of \( \succ \).
Lemma

Let $\mathcal{A}$ be a well-founded monotone $\Sigma$-algebra. Then $\succ_{\mathcal{A}}$ is well-founded.

Proof.

Assume there is an infinite sequence $t_1 \succ_{\mathcal{A}} t_2 \succ_{\mathcal{A}} t_3 \succ_{\mathcal{A}} \ldots$. Let $\alpha : \mathcal{X} \rightarrow \mathcal{A}$. Then by definition of $\succ_{\mathcal{A}}$: $[t_1, \alpha] \succ [t_2, \alpha] \succ [t_3, \alpha] \succ \ldots$. This infinite decreasing $\succ$ chain contradicts well-foundedness of $\succ$.

We have shown that:

• $\succ_{\mathcal{A}}$ is closed under substitutions
• $\succ_{\mathcal{A}}$ is closed under contexts
• $\succ_{\mathcal{A}}$ is well-founded

Hence we have proven that: Lemma $\mathcal{A}$ is a reduction order for every well-founded monotone $\Sigma$-algebra $(\mathcal{A}, [\cdot], \succ)$.
Lemma

Let $\mathcal{A}$ be a well-founded monotone $\Sigma$-algebra. Then $\succ_{\mathcal{A}}$ is well-founded.

Proof.

Assume there is an infinite sequence $t_1 \succ_{\mathcal{A}} t_2 \succ_{\mathcal{A}} t_3 \succ_{\mathcal{A}} \ldots$.

Let $\alpha : \mathcal{X} \to A$. Then by definition of $\succ_{\mathcal{A}}$:

$[t_1, \alpha] \succ [t_2, \alpha] \succ [t_3, \alpha] \succ \ldots$.

This infinite decreasing $\succ$ chain contradicts well-foundedness of $\succ$.

We have shown that:

- $\succ_{\mathcal{A}}$ is closed under substitutions
**Lemma**

Let $\mathcal{A}$ well-founded monotone $\Sigma$-algebra. Then $>_{\mathcal{A}}$ is well-founded.

**Proof.**

Assume there is an infinite sequence $t_1 >_{\mathcal{A}} t_2 >_{\mathcal{A}} t_3 >_{\mathcal{A}} \ldots$.

Let $\alpha : \mathcal{X} \rightarrow \mathcal{A}$. Then by definition of $>_{\mathcal{A}}$: $[t_1, \alpha] > [t_2, \alpha] > [t_3, \alpha] > \ldots$.

This infinite decreasing $>$ chain contradicts well-foundedness of $>$. \(\blacksquare\)

We have shown that:

- $>_{\mathcal{A}}$ is closed under substitutions
- $>_{\mathcal{A}}$ is closed under contexts
Lemma

Let $\mathcal{A}$ well-founded monotone $\Sigma$-algebra. Then $>_{\mathcal{A}}$ is well-founded.

Proof.

Assume there is an infinite sequence $t_1 >_{\mathcal{A}} t_2 >_{\mathcal{A}} t_3 >_{\mathcal{A}} \ldots$.

Let $\alpha : \mathcal{X} \to \mathcal{A}$. Then by definition of $>_{\mathcal{A}}$: $[t_1, \alpha] > [t_2, \alpha] > [t_3, \alpha] > \ldots$.

This infinite decreasing $>$ chain contradicts well-foundedness of $>$.

We have shown that:

- $>_{\mathcal{A}}$ is closed under substitutions
- $>_{\mathcal{A}}$ is closed under contexts
- $>_{\mathcal{A}}$ is well-founded
Lemma

Let $A$ well-founded monotone $\Sigma$-algebra. Then $\succ_A$ is well-founded.

Proof.

Assume there is an infinite sequence $t_1 \succ_A t_2 \succ_A t_3 \succ_A \ldots$.

Let $\alpha : \mathcal{X} \to A$. Then by definition of $\succ_A$: $[t_1, \alpha] > [t_2, \alpha] > [t_3, \alpha] > \ldots$.

This infinite decreasing $\succ$ chain contradicts well-foundedness of $\succ$.

We have shown that:

- $\succ_A$ is closed under substitutions
- $\succ_A$ is closed under contexts
- $\succ_A$ is well-founded

Hence we have proven that:

Lemma

$\succ_A$ is a reduction order for every well-founded monotone $\Sigma$-algebra $(A, [\cdot], \succ)$
Theorem

TRS \( R \) is terminating \( \iff R \subseteq >_A \) for well-founded monotone algebra \((A, >)\)

\( \Leftarrow \) Follows since we have shown that \( >_A \) is a reduction order.

\( \Rightarrow \) As \( \Sigma \)-algebra we take \((A, [\cdot], >)\) with

1. \( A = \mathcal{T}(\Sigma, \mathcal{X}) \),
2. \( [f](t_1, \ldots, t_n) = f(t_1, \ldots, t_n) \),
3. \( > := \rightarrow \).

Then \( > \) is well-founded since \( R \) is terminating.

Monotonicity of \( > \) follows from closure of \( \rightarrow \) under contexts:

\[ s > t \Rightarrow [f](\ldots, s, \ldots) = f(\ldots, s, \ldots) > f(\ldots, t, \ldots) = [f](\ldots, t, \ldots) \]

\( R \subseteq >_A \) since for all \( \ell \rightarrow r \in R \) and \( \alpha : \mathcal{X} \rightarrow A \):

\[ [\ell, \alpha] = \ell \alpha > r \alpha = [r, \alpha]. \]
Monotone algebras

**Theorem**

\[
\text{TRS } R \text{ is terminating } \iff R \subseteq >_A \text{ for well-founded monotone algebra } (A, >)
\]

**Proof.**

\[
\begin{align*}
\text{\[f\] } t_1, \ldots, t_n &= f(t_1, \ldots, t_n) \\
\rightarrow &= >
\end{align*}
\]

Then \(>\) is well-founded since \(R\) is terminating.

Monotonicity of \(>\) follows from closure of \(\rightarrow\) under contexts:

\[
\begin{align*}
s > t &\implies \text{[
    f
    
    \ldots,
    s,
    
    \ldots,
    
]
    = f(
    \ldots,
    s,
    
    \ldots,
    
) >
\text{[
    f
    
    \ldots,
    t,
    
    \ldots,
    
] = f(
    \ldots,
    t,
    
    \ldots,
    
)}
\end{align*}
\]

\(R \subseteq >_A\) since for all \(\ell \rightarrow r \in R\) and \(\alpha : X \rightarrow A\):

\[
\begin{align*}
\text{[
    \ell, \alpha
    
] = \ell \alpha > r \alpha &= \text{[
    r, \alpha
    
]}
\end{align*}
\]
Monotone algebras

Theorem

TRS $R$ is terminating $\iff R \subseteq >_A$ for well-founded monotone algebra $(A, >)$

Proof.

$\Leftarrow$ Follows since we have shown that $>_A$ is a reduction order.
Theorem

\[ \text{TRS } R \text{ is terminating } \iff R \subseteq >_A \text{ for well-founded monotone algebra } (A, >) \]

Proof.

\( \Leftarrow \) Follows since we have shown that \( >_A \) is a reduction order.

\( \Rightarrow \) As \( \Sigma \)-algebra we take \((A, [\cdot], >)\) with

\( R \subseteq >_A \) since for all \( \ell \rightarrow r \in R \) and \( \alpha : X \rightarrow A \):

\[ [\ell, \alpha] = \ell \alpha > r \alpha = [r, \alpha] \]
Theorem

TRS $R$ is terminating $\iff R \subseteq >_\mathcal{A}$ for well-founded monotone algebra $(\mathcal{A}, >)$

Proof.

$\Leftarrow$ Follows since we have shown that $>_\mathcal{A}$ is a reduction order.

$\Rightarrow$ As $\Sigma$-algebra we take $(\mathcal{A}, [\cdot], >)$ with

- $A = \mathcal{T}(\Sigma, \mathcal{X})$, 

Theorem

*TRS* $R$ is terminating $\iff R \subseteq >_A$ for well-founded monotone algebra $(A, >)$

Proof.

$\Leftarrow$ Follows since we have shown that $>_A$ is a reduction order.

$\Rightarrow$ As $\Sigma$-algebra we take $(A, [\cdot], >)$ with

- $A = \mathcal{T}(\Sigma, \mathcal{X})$,
- $[f](t_1, \ldots, t_n) = f(t_1, \ldots, t_n)$,
Theorem

TRS $R$ is terminating $\iff R \subseteq >_{\mathcal{A}}$ for well-founded monotone algebra $(\mathcal{A}, >)$

Proof.

$\Leftarrow$ Follows since we have shown that $>_{\mathcal{A}}$ is a reduction order.

$\Rightarrow$ As $\Sigma$-algebra we take $(\mathcal{A}, [\cdot], >)$ with

- $A = T(\Sigma, \mathcal{A})$,
- $[f](t_1, \ldots, t_n) = f(t_1, \ldots, t_n)$,
- $> := \rightarrow$.
Monotone algebras

Theorem

TRS $R$ is terminating $\iff R \subseteq >_A$ for well-founded monotone algebra $(A, >)$

Proof.

$\Leftarrow$ Follows since we have shown that $>_A$ is a reduction order.

$\Rightarrow$ As $\Sigma$-algebra we take $(A, [\cdot], >)$ with

- $A = \mathcal{T}(\Sigma, \mathcal{X})$,
- $[f](t_1, \ldots, t_n) = f(t_1, \ldots, t_n)$,
- $> := \rightarrow$.

Then $>$ is well-founded since $R$ is terminating.
Theorem

TRS $R$ is terminating $\iff R \subseteq A$ for well-founded monotone algebra $(A, >)$

Proof.

$\Leftarrow$ Follows since we have shown that $>_A$ is a reduction order.

$\Rightarrow$ As $\Sigma$-algebra we take $(A, [\cdot], >)$ with

- $A = T(\Sigma, \lambda')$,
- $[f](t_1, \ldots, t_n) = f(t_1, \ldots, t_n)$,
- $> := \rightarrow$.

Then $>$ is well-founded since $R$ is terminating.

Monotonicity of $>$ follows from closure of $\rightarrow$ under contexts:

$s > t \Rightarrow [f](\ldots, s, \ldots) = f(\ldots, s, \ldots) > f(\ldots, t, \ldots) = [f](\ldots, t, \ldots)$
Theorem

TRS $R$ is terminating $\iff R \subseteq >_A$ for well-founded monotone algebra $(A, >)$

Proof.

$\Leftarrow$ Follows since we have shown that $>_A$ is a reduction order.

$\Rightarrow$ As $\Sigma$-algebra we take $(A, [\cdot], >)$ with

- $A = T(\Sigma, \mathcal{A})$,
- $[f](t_1, \ldots, t_n) = f(t_1, \ldots, t_n)$,
- $> := \rightarrow$.

Then $>$ is well-founded since $R$ is terminating.

Monotonicity of $>$ follows from closure of $\rightarrow$ under contexts:

$s > t \Rightarrow [f](\ldots, s, \ldots) = f(\ldots, s, \ldots) > f(\ldots, t, \ldots) = [f](\ldots, t, \ldots)$

$R \subseteq >_A$ since for all $\ell \rightarrow r \in R$ and $\alpha : \mathcal{X} \rightarrow A$:

$[\ell, \alpha] = \ell \alpha > r \alpha = [r, \alpha]$. 
Well-Founded Monotone Algebras

used in termination proofs/tools:

- polynomial interpretations over $\mathbb{N}$
Monotone algebras

Well-Founded Monotone Algebras

used in termination proofs/tools:

- polynomial interpretations over $\mathbb{N}$
- polynomial interpretations over $\mathbb{Q}$ and $\mathbb{R}$
Monotone algebras

Well-Founded Monotone Algebras

used in termination proofs/tools:

• polynomial interpretations over $\mathbb{N}$
• polynomial interpretations over $\mathbb{Q}$ and $\mathbb{R}$
• matrix interpretations over $\mathbb{N}$
Monotone algebras

Well-Founded Monotone Algebras

used in termination proofs/tools:

- polynomial interpretations over \( \mathbb{N} \)
- polynomial interpretations over \( \mathbb{Q} \) and \( \mathbb{R} \)
- matrix interpretations over \( \mathbb{N} \)
- matrix interpretations over \( \mathbb{N} \cup \{-\infty\} \)
Well-Founded Monotone Algebras

used in termination proofs/tools:

- polynomial interpretations over $\mathbb{N}$
- polynomial interpretations over $\mathbb{Q}$ and $\mathbb{R}$
- matrix interpretations over $\mathbb{N}$
- matrix interpretations over $\mathbb{N} \cup \{-\infty\}$
- ...
Well-Founded Monotone Algebras

used in termination proofs/tools:

- polynomial interpretations over $\mathbb{N}$
- polynomial interpretations over $\mathbb{Q}$ and $\mathbb{R}$
- matrix interpretations over $\mathbb{N}$
- matrix interpretations over $\mathbb{N} \cup \{-\infty\}$
- ...
Outline

- Overview

- Combinatory Logic
A polynomial interpretations over $\mathbb{N}$ consists of:

- $\Sigma$-algebra is $(\mathbb{N}, [:])$,
- $>$ defined as usual on $\mathbb{N}$,
- the interpretations $[f]$ are polynomials.

However, to prove termination we need to check:

- monotonicity of the polynoms $[f]$, and
- $R \subseteq >$, that is, $[\ell, \alpha] > [r, \alpha]$ for all $\ell \rightarrow r \in R$ and $\alpha : \mathcal{X} \rightarrow A$. 

Polynomial interpretations

Definition

A polynomial interpretations over \( \mathbb{N} \) consists of:

- \( \Sigma \)-algebra is \( (\mathbb{N}, [\cdot]) \),
- \( > \) defined as usual on \( \mathbb{N} \),
- the interpretations \([f] \) are polynomials.

The first two conditions of well-founded monotone \( \Sigma \)-algebras are fulfilled:

- monotonicity of the polynoms \([f] \), and
- \( R \subseteq > \), that is, \([\ell, \alpha] > [r, \alpha] \) for all \( \ell \rightarrow r \in R \) and \( \alpha : X \rightarrow A \).
Definition

A polynomial interpretations over \( \mathbb{N} \) consists of:

- \( \Sigma \)-algebra is \((\mathbb{N}, [\cdot])\),
- \( > \) defined as usual on \( \mathbb{N} \),
- the interpretations \([f]\) are polynomials.

The first two conditions of well-founded monotone \( \Sigma \)-algebras are fulfilled:

- \((\mathbb{N}, [\cdot])\) is a \( \Sigma \)-algebra
- monotonicity of the polynoms \([f]\), and
- \( R \subseteq > \), that is, \([\ell, \alpha] > [r, \alpha]\) for all \( \ell \rightarrow r \in R \) and \( \alpha : X \rightarrow A \).
Definition

A polynomial interpretations over \( \mathbb{N} \) consists of:

- \( \Sigma \)-algebra is \((\mathbb{N}, [\cdot])\),
- \( > \) defined as usual on \( \mathbb{N} \),
- the interpretations \([f]\) are polynomials.

The first two conditions of well-founded monotone \( \Sigma \)-algebras are fulfilled:

- \((\mathbb{N}, [\cdot])\) is a \( \Sigma \)-algebra
- \( > \) is well-founded

- monotonicity of the polynoms \([f]\), and
- \( R \subseteq > \), that is, \([\ell, \alpha] > [r, \alpha]\) for all \( \ell \rightarrow r \in R \) and \( \alpha : \mathcal{X} \rightarrow A \)
Polynomial interpretations

**Definition**

A polynomial interpretations over $\mathbb{N}$ consists of:

- $\Sigma$-algebra is $(\mathbb{N}, [\cdot])$,
- $>$ defined as usual on $\mathbb{N}$,
- the interpretations $[f]$ are polynomials.

The first two conditions of well-founded monotone $\Sigma$-algebras are fulfilled:

- $(\mathbb{N}, [\cdot])$ is a $\Sigma$-algebra
- $>$ is well-founded

However, to prove termination we need to check:
Polynomial Interpretations

Definition

A polynomial interpretations over \( \mathbb{N} \) consists of:

- \( \Sigma \)-algebra is (\( \mathbb{N}, [\cdot] \)),
- \( > \) defined as usual on \( \mathbb{N} \),
- the interpretations \([f]\) are polynomials.

The first two conditions of well-founded monotone \( \Sigma \)-algebras are fulfilled:

- (\( \mathbb{N}, [\cdot] \)) is a \( \Sigma \)-algebra
- \( > \) is well-founded

However, to prove termination we need to check:

- monotonicity of the polynoms \([f]\), and
Definition

A polynomial interpretations over $\mathbb{N}$ consists of:

- $\Sigma$-algebra is $(\mathbb{N}, [\cdot])$,
- $>$ defined as usual on $\mathbb{N}$,
- the interpretations $[f]$ are polynomials.

The first two conditions of well-founded monotone $\Sigma$-algebras are fulfilled:

- $(\mathbb{N}, [\cdot])$ is a $\Sigma$-algebra
- $>$ is well-founded

However, to prove termination we need to check:

- monotonicity of the polynoms $[f]$, and
- $R \subseteq >$, that is, $[\ell, \alpha] > [r, \alpha]$ for all $\ell \rightarrow r \in R$ and $\alpha : \mathcal{X} \rightarrow A$
Example 1

Example

\[ f(g(x)) \rightarrow f(f(x)) \]

Find a polynomial interpretation over \( \mathbb{N} \) which proves termination:

\[ [f](x) = ??? \quad [g](x) = ??? \]

• Are the functions \([f]\) monotone?
  Yes, since whenever \( a > b \), then \( [f](a) = a > b = [f](b) \), \([g](a) = a + 1 > b + 1 = [g](b) \),

• Does \([\ell, \alpha] > [r, \alpha]\) hold?
  Yes since, \([f](g(x), \alpha) = [f](g(\alpha(x))) = \alpha(x) + 1 \geq \alpha(x) = [f](f(x), \alpha)\]

Hence we have proven termination.
Example 1

**Example**

\[ f(g(x)) \rightarrow f(f(x)) \]

Find a polynomial interpretation over \( \mathbb{N} \) which proves termination:

\[ [f](x) = x \quad [g](x) = x + 1 \]
Example

\[ f(g(x)) \rightarrow f(f(x)) \]

Find a polynomial interpretation over \( \mathbb{N} \) which proves termination:

\[ [f](x) = x \quad \quad \quad \quad [g](x) = x + 1 \]

- Are the functions \([f]\) monotone?
Example 1

\[ f(g(x)) \rightarrow f(f(x)) \]

Find a polynomial interpretation over \( \mathbb{N} \) which proves termination:

\[ [f](x) = x \quad \quad \quad [g](x) = x + 1 \]

- **Are the functions \([f]\) monotone?**
  
  Yes, since whenever \( a > b \), then
  
  \[ [f](a) = a > b = [f](b), \]
  
  \[ [g](a) = a + 1 > b + 1 = [g](b), \]
Example 1

Example

\[ f(g(x)) \rightarrow f(f(x)) \]

Find a polynomial interpretation over \( \mathbb{N} \) which proves termination:

\[ [f](x) = x \quad \quad [g](x) = x + 1 \]

- Are the functions \([f]\) monotone?
  Yes, since whenever \( a > b \), then
  \[ [f](a) = a > b = [f](b), \]
  \[ [g](a) = a + 1 > b + 1 = [g](b), \]

- Does \([\ell, \alpha] > [r, \alpha]\) hold?
Example

\[ f(g(x)) \rightarrow f(f(x)) \]

Find a polynomial interpretation over \( \mathbb{N} \) which proves termination:

\[
[f](x) = x \\
[g](x) = x + 1
\]

- Are the functions \([f]\) monotone?
  Yes, since whenever \( a > b \), then
  \[
  [f](a) = a > b = [f](b), \\
  [g](a) = a + 1 > b + 1 = [g](b),
  \]

- Does \([\ell, \alpha] > [r, \alpha]\) hold?
  Yes since,
  \[
  [f(g(x)), \alpha] = [f([g](\alpha(x))) = \alpha(x) + 1 > \alpha(x) = [f(f(x)), \alpha]
  \]
Polynomial Interpretations

Example 1

Example

\[ f(g(x)) \rightarrow f(f(x)) \]

Find a polynomial interpretation over \( \mathbb{N} \) which proves termination:

\[
[f](x) = x \\
[g](x) = x + 1
\]

• Are the functions \([f]\) monotone?
  Yes, since whenever \( a > b \), then
  \[
  [f](a) = a > b = [f](b), \\
  [g](a) = a + 1 > b + 1 = [g](b),
  \]

• Does \([\ell, \alpha] > [r, \alpha]\) hold?
  Yes since,
  \[
  [f(g(x)), \alpha] = [f]([g](\alpha(x))) = \alpha(x) + 1 > \alpha(x) = [f(f(x)), \alpha]
  \]

Hence we have proven termination.
Example 2

Example

\[ f(g(x)) \rightarrow g(f(x)) \]

Find a polynomial interpretation over \( \mathbb{N} \) which proves termination:

\[ [f](x) = ??? \quad [g](x) = ??? \]
Example 2

Example

\[ f(g(x)) \rightarrow g(f(x)) \]

Find a polynomial interpretation over \( \mathbb{N} \) which proves termination:

\[ [f](x) = 2 \cdot x \quad \quad \quad [g](x) = x + 1 \]
Polynomial Interpretations

Example 2

Example

\[ f(g(x)) \rightarrow g(f(x)) \]

Find a polynomial interpretation over \( \mathbb{N} \) which proves termination:

\[
[f](x) = 2 \cdot x \\
g(x) = x + 1
\]

- Are the functions \([f]\) monotone?

Hence we have proven termination.
Example 2

Example

\[ f(g(x)) \rightarrow g(f(x)) \]

Find a polynomial interpretation over \( \mathbb{N} \) which proves termination:

\[
[f](x) = 2 \cdot x \quad \quad \quad [g](x) = x + 1
\]

- Are the functions \([f]\) monotone?
  
  Yes, since whenever \( a > b \), then
  
  \[
  [f](a) = 2 \cdot a > 2 \cdot b = [f](b),
  \]
  
  \[
  [g](a) = a + 1 > b + 1 = [g](b),
  \]
Example 2

Example

\[ f(g(x)) \rightarrow g(f(x)) \]

Find a polynomial interpretation over \( \mathbb{N} \) which proves termination:

\[ [f](x) = 2 \cdot x \quad \quad \quad [g](x) = x + 1 \]

- Are the functions \([f]\) monotone?
  Yes, since whenever \( a > b \), then
  \[ [f](a) = 2 \cdot a > 2 \cdot b = [f](b), \]
  \[ [g](a) = a + 1 > b + 1 = [g](b), \]

- Does \([\ell, \alpha] > [r, \alpha]\) hold?
Example

\[ f(g(x)) \rightarrow g(f(x)) \]

Find a polynomial interpretation over \( \mathbb{N} \) which proves termination:

\[
\begin{align*}
[f](x) &= 2 \cdot x \\
g](x) &= x + 1
\end{align*}
\]

- Are the functions \([f]\) monotone?
  Yes, since whenever \( a > b \), then
  \[
  [f](a) = 2 \cdot a > 2 \cdot b = [f](b),
  \]
  \[
  [g](a) = a + 1 > b + 1 = [g](b),
  \]

- Does \([\ell, \alpha] > [r, \alpha]\) hold?
  Yes since,
  \[
  [f(g(x)), \alpha] = 2 \cdot (\alpha(x) + 1) > 2 \cdot \alpha(x) + 1 = [g(f(x)), \alpha]
  \]
Example 2

Example

\[ f(g(x)) \rightarrow g(f(x)) \]

Find a polynomial interpretation over \( \mathbb{N} \) which proves termination:

\[ [f](x) = 2 \cdot x \quad \quad [g](x) = x + 1 \]

- Are the functions \( [f] \) monotone?
  Yes, since whenever \( a > b \), then
  \[ [f](a) = 2 \cdot a > 2 \cdot b = [f](b), \]
  \[ [g](a) = a + 1 > b + 1 = [g](b), \]

- Does \( [\ell, \alpha] > [r, \alpha] \) hold?
  Yes since,
  \[ [f(g(x)), \alpha] = 2 \cdot (\alpha(x) + 1) > 2 \cdot \alpha(x) + 1 = [g(f(x)), \alpha] \]

Hence we have proven termination.
Example

\[ A(x, 0) \to x \]
\[ A(x, s(y)) \to s(A(x, y)) \]

Find a polynomial interpretation over \( \mathbb{N} \) which proves termination:

\[ [0] = ??? \quad [s](x) = ??? \quad [A](x, y) = ??? \]
Example

\[ A(x, 0) \rightarrow x \]
\[ A(x, s(y)) \rightarrow s(A(x, y)) \]

Find a polynomial interpretation over \( \mathbb{N} \) which proves termination:

\[ [0] = 1 \quad [s](x) = x + 1 \quad [A](x, y) = x + 2 \cdot y \]
Example 3

Example

\[ A(x, 0) \rightarrow x \]
\[ A(x, s(y)) \rightarrow s(A(x, y)) \]

Find a polynomial interpretation over \( \mathbb{N} \) which proves termination:

\[ [0] = 1 \quad [s](x) = x + 1 \quad [A](x, y) = x + 2 \cdot y \]

- Are the functions \([f]\) monotone?
Example 3

Example

\[ A(x, 0) \to x \]
\[ A(x, s(y)) \to s(A(x, y)) \]

Find a polynomial interpretation over \( \mathbb{N} \) which proves termination:

\[ [0] = 1 \quad [s](x) = x + 1 \quad [A](x, y) = x + 2 \cdot y \]

- Are the functions \([f]\) monotone?

Yes, since whenever \( a > b \), then

\[ [s](a) = a + 1 > b + 1 = [s](b), \]
\[ [A](a, y) = a + 2 \cdot y > b + 2 \cdot y = [A](b, y), \]
\[ [A](x, a) = x + 2 \cdot a > x + 2 \cdot b = [A](x, b), \]
Example 3

Example

\[ A(x, 0) \rightarrow x \]
\[ A(x, s(y)) \rightarrow s(A(x, y)) \]

Find a polynomial interpretation over \( \mathbb{N} \) which proves termination:

\[ [0] = 1 \quad [s](x) = x + 1 \quad [A](x, y) = x + 2 \cdot y \]

- Does \([\ell, \alpha] > [r, \alpha]\) hold?
### Example 3

**Example**

\[
A(x, 0) \rightarrow x \\
A(x, s(y)) \rightarrow s(A(x, y))
\]

Find a polynomial interpretation over \(\mathbb{N}\) which proves termination:

\[
[0] = 1 \\
[s](x) = x + 1 \\
[A](x, y) = x + 2 \cdot y
\]

- **Does \([\ell, \alpha] > [r, \alpha]\) hold?**
  
  Yes since,

  \[
  [A(x, s(y)), \alpha] = [A](\alpha(x), [s](\alpha(y))) \\
  = \alpha(x) + 2 \cdot \alpha(y) + 2 \\
  > \alpha(x) + 2 \cdot \alpha(y) + 1 \\
  = [s(+ (x, y)), \alpha] \\
  [A(x, 0), \alpha] = \alpha(x) + 2 \cdot 1 > \alpha(x) = [x, \alpha]
  \]
Example 3

Example

\begin{align*}
A(x, 0) & \rightarrow x \\
A(x, s(y)) & \rightarrow s(A(x, y))
\end{align*}

Find a polynomial interpretation over \( \mathbb{N} \) which proves termination:

\begin{align*}
[0] &= 1 \\
[s](x) &= x + 1 \\
[A](x, y) &= x + 2 \cdot y
\end{align*}

• Does \([\ell, \alpha] > [r, \alpha]\) hold?

Yes since,

\begin{align*}
[A(x, s(y)), \alpha] &= [A](\alpha(x), [s](\alpha(y))) \\
&= \alpha(x) + 2 \cdot \alpha(y) + 2 \\
&> \alpha(x) + 2 \cdot \alpha(y) + 1 \\
&= [s(+ (x, y)), \alpha] \\
[A(x, 0), \alpha] &= \alpha(x) + 2 \cdot 1 > \alpha(x) = [x, \alpha]
\end{align*}

Hence we have proven termination.
Example ($\triangleright$ not well-founded)

Let $R = \{ f(x) \rightarrow f(f(x)) \}$ with the $\Sigma$-algebra $(\mathbb{Z}, [\cdot])$ and

$$[f](x) = x - 1$$

and $\triangleright$ as usual on $\mathbb{Z}$. 
Why do we need the conditions?

Example (> not well-founded)

Let \( R = \{ f(x) \rightarrow f(f(x)) \} \) with the \( \Sigma \)-algebra \((\mathbb{Z}, [\cdot])\) and

\[
[f](x) = x - 1
\]

and \( > \) as usual on \( \mathbb{Z} \).

Then \( R \) is not terminating

\[
f(x) \rightarrow f(f(x)) \rightarrow f(f(f(x))) \rightarrow \ldots
\]
Polynomial Interpretations

Why do we need the conditions?

Example (> not well-founded)

Let $R = \{ f(x) \rightarrow f(f(x)) \}$ with the $\Sigma$-algebra $(\mathbb{Z}, [\cdot])$ and

$$[f](x) = x - 1$$

and $>$ as usual on $\mathbb{Z}$.

Then $R$ is not terminating

$$f(x) \rightarrow f(f(x)) \rightarrow f(f(f(x))) \rightarrow \ldots$$

but

- $[f]$ is monotone, and
- $[f(x), \alpha] = \alpha(x) - 1 > \alpha(x) - 2 = [f(f(x)), \alpha]$. 
Why do we need the conditions?

Example (> not well-founded)

Let $R = \{ f(x) \to f(f(x)) \}$ with the $\Sigma$-algebra $(\mathbb{Z}, [\cdot])$ and

$$[f](x) = x - 1$$

and $>$ as usual on $\mathbb{Z}$.

Then $R$ is not terminating

$$f(x) \to f(f(x)) \to f(f(f(x))) \to \ldots$$

but

- $[f]$ is monotone, and
- $[f(x), \alpha] = \alpha(x) - 1 > \alpha(x) - 2 = [f(f(x)), \alpha]$.

Hence $>$ needs to be well-founded!
Why do we need the conditions?

Example ([f] not monotone)

Let $R = \{ f(x) \to g(f(x)) \}$ with the $\Sigma$-algebra $(\mathbb{N}, [\cdot])$ and

$$[f](x) = x + 1 \quad \quad \quad [g](x) = 0$$

and $>$ as usual on $\mathbb{N}$. 
Polynomial Interpretations

Why do we need the conditions?

Example ([f] not monotone)

Let $R = \{ f(x) \rightarrow g(f(x)) \}$ with the $\Sigma$-algebra $(\mathbb{N}, [\cdot])$ and

$$[f](x) = x + 1 \quad \quad [g](x) = 0$$

and $\succ$ as usual on $\mathbb{N}$.

Then $R$ is not terminating

$$f(x) \rightarrow g(f(x)) \rightarrow g(g(f(x))) \rightarrow \ldots$$

Hence the functions $[f]$ need to be monotone!
Why do we need the conditions?

Example ([f] not monotone)

Let \( R = \{ f(x) \rightarrow g(f(x)) \} \) with the \( \Sigma \)-algebra \((\mathbb{N}, [\cdot])\) and

\[
[f](x) = x + 1 \quad \quad \quad \quad [g](x) = 0
\]

and \( \cdot > \) as usual on \( \mathbb{N} \).

Then \( R \) is not terminating

\[
f(x) \rightarrow g(f(x)) \rightarrow g(g(f(x))) \rightarrow \ldots
\]

but

- \( \cdot > \) is well-founded, and

- \([f(x), \alpha] = \alpha(x) + 1 > 0 = [g(f(x)), \alpha] \).
Why do we need the conditions?

Example (\([f]\) not monotone)

Let \( R = \{ f(x) \rightarrow g(f(x)) \} \) with the \( \Sigma \)-algebra \((\mathbb{N}, [\cdot])\) and

\[
[f](x) = x + 1 \quad \quad \quad \quad \quad \quad [g](x) = 0
\]

and \( > \) as usual on \( \mathbb{N} \).

Then \( R \) is not terminating

\[
f(x) \rightarrow g(f(x)) \rightarrow g(g(f(x))) \rightarrow \ldots
\]

but

- \( > \) is well-founded, and
- \([f(x), \alpha] = \alpha(x) + 1 > 0 = [g(f(x)), \alpha]\).

Hence the functions \([f]\) need to be monotone!
Question

How to find suitable polynomials?
Question
How to find suitable polynomials?

Modern Approach
(a) choose abstract polynomial interpretations (linear, quadratic, ...)

Polynomial Interpretations
Question
How to find suitable polynomials?

Modern Approach
(a) choose abstract polynomial interpretations (linear, quadratic, . . .)
(b) transform rewrite rules into polynomial ordering constraints
Question
How to find suitable polynomials?

Modern Approach
(a) choose abstract polynomial interpretations (linear, quadratic, . . .)
(b) transform rewrite rules into polynomial ordering constraints
(c) add monotonicity and well-definedness constraints
**Question**

How to find suitable polynomials?

**Modern Approach**

(a) choose abstract polynomial interpretations (linear, quadratic, ...)

(b) transform rewrite rules into polynomial ordering constraints

(c) add monotonicity and well-definedness constraints

(d) eliminate universally quantified variables
Question
How to find suitable polynomials?

Modern Approach
(a) choose abstract polynomial interpretations (linear, quadratic, ...)
(b) transform rewrite rules into polynomial ordering constraints
(c) add monotonicity and well-definedness constraints
(d) eliminate universally quantified variables
(e) translate resulting constraints to SAT or SMT problem
Example

- rewrite system

\[
\begin{align*}
0 + y & \rightarrow y \\
s(x) + y & \rightarrow s(x + y)
\end{align*}
\]
Example

- rewrite system
  \[0 + y \rightarrow y\]
  \[s(x) + y \rightarrow s(x + y)\]

- interpretations
  \[0_A = a\]
  \[s_A(x) = bx + c\]
  \[+_A(x, y) = dx + ey + f\]
Example

• rewrite system

\[ 0 + y \rightarrow y \]
\[ s(x) + y \rightarrow s(x + y) \]

• interpretations

\[ 0_A = a \]
\[ s_A(x) = bx + c \]
\[ +_A(x, y) = dx + ey + f \]

• polynomial constraints

\[ \forall x, y \in \mathbb{N} \]
\[ da + ey + f > y \]
\[ d(bx + c) + ey + f > b(dx + ey + f) + c \]
Example

• rewrite system

\[ 0 + y \rightarrow y \]
\[ s(x) + y \rightarrow s(x + y) \]

• interpretations

\[ 0_A = a \]
\[ s_A(x) = bx + c \]
\[ +_A(x, y) = dx + ey + f \]

• polynomial constraints

\[ \forall x, y \in \mathbb{N} \]
\[ da + ey + f > y \]
\[ d(bx + c) + ey + f > b(dx + ey + f) + c \]
\[ a \geq 0 \quad b \geq 1 \quad c \geq 0 \quad d \geq 1 \quad e \geq 1 \quad f \geq 0 \]
Example

• rewrite system

\[
0 + y \rightarrow y \\
\text{s}(x) + y \rightarrow \text{s}(x + y)
\]

• interpretations

\[
0_A = a \\
\text{s}_A(x) = bx + c \\
+ A(x, y) = dx + ey + f
\]

• polynomial constraints \( \forall x, y \in \mathbb{N} \)

\[
(e - 1)y + da + f > 0 \\
(e - be)y + dc + f - bf - c > 0 \\
\]

\[
a \geq 0 \quad b \geq 1 \quad c \geq 0 \quad d \geq 1 \quad e \geq 1 \quad f \geq 0
\]
Example

- rewrite system
  
  \[ 0 + y \rightarrow y \]
  
  \[ s(x) + y \rightarrow s(x + y) \]

- interpretations
  
  \[ 0_A = a \]
  
  \[ s_A(x) = bx + c \]
  
  \[ +_A(x, y) = dx + ey + f \]

- diophantine constraints
  
  \[ \forall x, y \in \mathbb{N} \]
  
  \[ e - 1 \geq 0 \quad da + f > 0 \]
  
  \[ e - be \geq 0 \quad dc + f - bf - c > 0 \]
  
  \[ a \geq 0 \quad b \geq 1 \quad c \geq 0 \quad d \geq 1 \quad e \geq 1 \quad f \geq 0 \]
Example

• rewrite system

\[ 0 + y \rightarrow y \]
\[ s(x) + y \rightarrow s(x + y) \]

• interpretations

\[ 0_A = a \]
\[ s_A(x) = bx + c \]
\[ +_A(x, y) = dx + ey + f \]

• diophantine constraints

\[ \forall x, y \in \mathbb{N} \]
\[ e - 1 \geq 0 \quad da + f > 0 \]
\[ e - be \geq 0 \quad dc + f - bf - c > 0 \]
\[ a \geq 0 \quad b \geq 1 \quad c \geq 0 \quad d \geq 1 \quad e \geq 1 \quad f \geq 0 \]

• possible solution

\[ a = 0 \quad b = 1 \quad c = 1 \quad d = 2 \quad e = 1 \quad f = 1 \]
Example

- rewrite system
  
  \[ 0 + y \rightarrow y \]
  
  \[ s(x) + y \rightarrow s(x + y) \]

- interpretations
  
  \[ 0_A = 0 \]
  
  \[ s_A(x) = x + 1 \]
  
  \[ +_A(x, y) = 2x + y + 1 \]

- diophantine constraints
  
  \( \forall x, y \in \mathbb{N} \)
  
  \[ e - 1 \geq 0 \quad da + f > 0 \]
  
  \[ e - be \geq 0 \quad dc + f - bf - c > 0 \]
  
  \[ a \geq 0 \quad b \geq 1 \quad c \geq 0 \quad d \geq 1 \quad e \geq 1 \quad f \geq 0 \]

- possible solution
  
  \[ a = 0 \quad b = 1 \quad c = 1 \quad d = 2 \quad e = 1 \quad f = 1 \]
Example

• TRS

\[
\begin{align*}
0 + y & \rightarrow y \\
\text{let } s(x) & \rightarrow s(x + y) \\
0 \times y & \rightarrow 0 \\
\text{let } s(x) \times y & \rightarrow y + (x \times y)
\end{align*}
\]
Example

- **TRS**
  
  \[
  \begin{align*}
  0 + y & \rightarrow y \\
  s(x) + y & \rightarrow s(x + y) \\
  0 \times y & \rightarrow 0 \\
  s(x) \times y & \rightarrow y + (x \times y)
  \end{align*}
  \]

- **interpretations in** \( \mathbb{N} \)
  
  \[
  \begin{align*}
  0_A &= 1 \\
  s_A(x) &= x + 1 \\
  +_A(x, y) &= 2x + y \\
  \times_A(x, y) &= 2xy + x + y + 1
  \end{align*}
  \]
Example

- **TRS**
  
  \[
  0 + y \rightarrow y \quad \rightarrow 0 \\
  s(x) + y \rightarrow s(x + y) \\
  s(x) \times y \rightarrow y + (x \times y)
  \]

- **interpretations in** \(\mathbb{N}\)
  
  \[
  0_A = 1 \\
  s_A(x) = x + 1 \\
  +_A(x, y) = 2x + y \\
  \times_A(x, y) = 2xy + x + y + 1
  \]

- **constraints** \(\forall x, y \in \mathbb{N}\)
  
  \[
  y + 2 > y \\
  2x + y + 2 > 2x + y + 1 \\
  3y + 2 > 1 \\
  2xy + x + 3y + 2 > 2xy + x + 3y + 1
  \]
Polynomial Interpretations

Example

- **TRS**
  
  \[
  \begin{align*}
  0 + y & \rightarrow y \\
  s(x) + y & \rightarrow s(x + y) \\
  0 \times y & \rightarrow 0 \\
  s(x) \times y & \rightarrow y + (x \times y)
  \end{align*}
  \]

- **interpretations in** \( \mathbb{N} \)
  
  \[
  \begin{align*}
  0_A &= 1 \\
  s_A(x) &= x + 1 \\
  +_A(x, y) &= 2x + y \\
  \times_A(x, y) &= 2xy + x + y + 1
  \end{align*}
  \]

- **constraints** \( \forall x, y \in \mathbb{N} \)
  
  \[
  \begin{align*}
  2 &> 0 \\
  1 &> 0 \\
  3y + 1 &> 0 \\
  1 &> 0
  \end{align*}
  \]
Example

• TRS

\[ 0 + y \rightarrow y \quad 0 \times y \rightarrow 0 \]
\[ s(x) + y \rightarrow s(x + y) \quad s(x) \times y \rightarrow y + (x \times y) \]

• interpretations in \( \mathbb{N} \)

\[ 0_A = 1 \quad +_A(x, y) = 2x + y \]
\[ s_A(x) = x + 1 \quad \times_A(x, y) = 2xy + x + y + 1 \]

• constraints

\[ \forall x, y \in \mathbb{N} \]
\[ 2 > 0 \quad 3y + 1 > 0 \]
\[ 1 > 0 \quad 1 > 0 \]

• \( s(0) \times s(s(0)) \rightarrow s(s(0)) + (0 \times s(s(0))) \rightarrow s(s(0)) + 0 \rightarrow s(s(0) + 0) \)

\[ \rightarrow s(s(0 + 0)) \rightarrow s(s(0)) \]
Example

- **TRS**
  
  \[
  0 + y \rightarrow y \\
  s(x) + y \rightarrow s(x + y) \\
  0 \times y \rightarrow 0 \\
  s(x) \times y \rightarrow y + (x \times y)
  \]

- **interpretations in** \(\mathbb{N}\)
  
  \[
  0_A = 1 \\
  s_A(x) = x + 1 \\
  +_A(x, y) = 2x + y \\
  \times_A(x, y) = 2xy + x + y + 1
  \]

- **constraints** \(\forall x, y \in \mathbb{N}\)
  
  \[
  2 > 0 \\
  1 > 0 \\
  3y + 1 > 0 \\
  1 > 0
  \]

- **s(0) \times s(s(0)) \rightarrow s(s(0)) + (0 \times s(s(0))) \rightarrow s(s(0)) + 0 \rightarrow s(s(0) + 0) \rightarrow s(s(0 + 0)) \rightarrow s(s(0)) \rightarrow s(s(0)) > 18 \rightarrow 17 \rightarrow 7 > 6 > 5 > 3
Example

- TRS

\[ \partial(x + y) \rightarrow \partial(x) + \partial(y) \]
\[ \partial(x - y) \rightarrow \partial(x) - \partial(y) \]
\[ \partial(x \times y) \rightarrow (\partial(x) \times y) + (x \times \partial(y)) \]
\[ \partial(x \div y) \rightarrow ((\partial(x) \times y) - (x \times \partial(y))) \div (y \times y) \]
**Example**

- **TRS**
  \[
  \partial(x + y) \rightarrow \partial(x) + \partial(y) \\
  \partial(x - y) \rightarrow \partial(x) - \partial(y) \\
  \partial(x \times y) \rightarrow (\partial(x) \times y) + (x \times \partial(y)) \\
  \partial(x \div y) \rightarrow ((\partial(x) \times y) - (x \times \partial(y))) \div (y \times y)
  \]

- **interpretations in** \(\mathbb{N}\)
  \[
  \alpha_\mathcal{A} = \beta_\mathcal{A} = 0_\mathcal{A} = 1_\mathcal{A} = 1 \\
  +_\mathcal{A}(x, y) = -_\mathcal{A}(x, y) = \times_\mathcal{A}(x, y) = \div_\mathcal{A}(x, y) = x + y + 3 \\
  \partial_\mathcal{A}(x) = x^2 + 6x + 6
  \]
Example

- TRS

\[ \partial(x + y) \rightarrow \partial(x) + \partial(y) \quad \partial(\alpha) = 1 \]
\[ \partial(x - y) \rightarrow \partial(x) - \partial(y) \quad \partial(\beta) = 0 \]
\[ \partial(x \times y) \rightarrow (\partial(x) \times y) + (x \times \partial(y)) \]
\[ \partial(x \div y) \rightarrow ((\partial(x) \times y) - (x \times \partial(y))) \div (y \times y) \]

- interpretations in \( \mathbb{N} \)

\[ \alpha_A = \beta_A = 0_A = 1_A = 1 \]
\[ +_A(x, y) = -_A(x, y) = \times_A(x, y) = \div_A(x, y) = x + y + 3 \]
\[ \partial_A(x) = x^2 + 6x + 6 \]

- constraints \( \forall x, y \in \mathbb{N} \)

\[ x^2 + y^2 + 2xy + 12x + 12y + 33 > x^2 + y^2 + 6x + 6y + 15 \quad 13 > 1 \]
\[ x^2 + y^2 + 2xy + 12x + 12y + 33 > x^2 + y^2 + 6x + 6y + 15 \quad 13 > 1 \]
\[ x^2 + y^2 + 2xy + 12x + 12y + 33 > x^2 + y^2 + 7x + 7y + 21 \]
\[ x^2 + y^2 + 2xy + 12x + 12y + 33 > x^2 + y^2 + 7x + 9y + 27 \]
Example

• TRS

\[
\begin{align*}
\partial(x + y) & \rightarrow \partial(x) + \partial(y) & \partial(\alpha) = 1 \\
\partial(x - y) & \rightarrow \partial(x) - \partial(y) & \partial(\beta) = 0 \\
\partial(x \times y) & \rightarrow (\partial(x) \times y) + (x \times \partial(y)) \\
\partial(x \div y) & \rightarrow ((\partial(x) \times y) - (x \times \partial(y))) \div (y \times y)
\end{align*}
\]

• interpretations in \( \mathbb{N} \)

\[
\begin{align*}
\alpha_A &= \beta_A = 0_A = 1_A = 1 \\
+_A(x, y) &= -_A(x, y) = \times_A(x, y) = \div_A(x, y) = x + y + 3 \\
\partial_A(x) &= x^2 + 6x + 6
\end{align*}
\]

• constraints \( \forall x, y \in \mathbb{N} \)

\[
\begin{align*}
2xy + 6x + 6y + 18 & > 0 & 13 & > 1 \\
2xy + 6x + 6y + 18 & > 0 & 13 & > 1 \\
2xy + 5x + 5y + 12 & > 0 \\
2xy + 5x + 3y + 6 & > 0
\end{align*}
\]
Remark

Numerous terminating TRSs are not polynomially terminating.
Remark

Numerous terminating TRSs are not polynomially terminating.
A non-simply terminating TRS: \( f(f(x)) \rightarrow f(g(f(x))) \)

Example \((S = \{ f(f(x)) \rightarrow f(g(f(x))) \})\)

There exists no monotone \(\Sigma\)-algebra \(A\) with \(A = \mathbb{N}\) proving \(SN(S)\).
Polynomial Interpretations

A non-simply terminating TRS: \( f(f(x)) \rightarrow f(g(f(x))) \)

Example \((S = \{ f(f(x)) \rightarrow f(g(f(x))) \})\)

There exists no monotone \( \Sigma \)-algebra \( A \) with \( A = \mathbb{N} \) proving SN(S).

Assume contrary it would exist. Let \( \alpha : \mathcal{X} \rightarrow A \), then:

\[
[f(f(x)), \alpha] > [f(g(f(x))), \alpha].
\]
A non-simply terminating TRS: \( f(f(x)) \rightarrow f(g(f(x))) \)

Example \( S = \{ f(f(x)) \rightarrow f(g(f(x))) \} \)

There exists no monotone \( \Sigma \)-algebra \( A \) with \( A = \mathbb{N} \) proving \( SN(S) \).

Assume contrary it would exist. Let \( \alpha : \mathcal{X} \rightarrow A \), then:

\[
[f(f(x)), \alpha] > [f(g(f(x))), \alpha].
\]

But then also

\[
[f(x), \alpha] > [g(f(x)), \alpha]
\]

since otherwise:

- \( [f(x), \alpha] = [g(f(x)), \alpha] \) implies \( [f(f(x)), \alpha] = [f(g(f(x))), \alpha] \),
- \( [f(x), \alpha] < [g(f(x)), \alpha] \) implies \( [f(f(x)), \alpha] < [f(g(f(x))), \alpha] \)

by monotonicity.
A non-simply terminating TRS: $f(f(x)) \rightarrow f(g(f(x)))$

Example ($S = \{ f(f(x)) \rightarrow f(g(f(x))) \}$)

There exists no monotone $\Sigma$-algebra $A$ with $A = \mathbb{N}$ proving $SN(S)$.

Assume contrary it would exist. Let $\alpha : \mathcal{X} \rightarrow A$, then:

$$[f(f(x)), \alpha] > [f(g(f(x))), \alpha].$$

But then also

$$[f(x), \alpha] > [g(f(x)), \alpha]$$

since otherwise:

- $[f(x), \alpha] = [g(f(x)), \alpha]$ implies $[f(f(x)), \alpha] = [f(g(f(x))), \alpha]$,
- $[f(x), \alpha] < [g(f(x)), \alpha]$ implies $[f(f(x)), \alpha] < [f(g(f(x))), \alpha]$ by monotonicity.

But then $A$ would also prove termination of $f(x) \rightarrow g(f(x))$. 
A non-simply terminating TRS: $f(f(x)) \rightarrow f(g(f(x)))$

Example ($S = \{ f(f(x)) \rightarrow f(g(f(x))) \}$)

There exists no monotone $\Sigma$-algebra $A$ with $A = \mathbb{N}$ proving $SN(S)$.

Assume contrary it would exist. Let $\alpha : \mathcal{X} \rightarrow A$, then:

$$[f(f(x)), \alpha] > [f(g(f(x))), \alpha].$$

But then also

$$[f(x), \alpha] > [g(f(x)), \alpha]$$

since otherwise:

- $[f(x), \alpha] = [g(f(x)), \alpha]$ implies $[f(f(x)), \alpha] = [f(g(f(x))), \alpha]$,
- $[f(x), \alpha] < [g(f(x)), \alpha]$ implies $[f(f(x)), \alpha] < [f(g(f(x))), \alpha]$ by monotonicity.

But then $A$ would also prove termination of $f(x) \rightarrow g(f(x))$.

Thus we need another $\Sigma$-algebra, for example $A = \mathbb{N}^2$. 
A non-simply terminating TRS: $f(f(x)) \rightarrow f(g(f(x)))$

**Example ($S = \{ f(f(x)) \rightarrow f(g(f(x))) \}$)**

We choose the $\Sigma$-algebra $(\mathbb{N}^2, [\cdot])$ with:

$$[f](\vec{x}) = \begin{pmatrix} 1 & 1 \\ 0 & 0 \end{pmatrix} \cdot \vec{x} + \begin{pmatrix} 0 \\ 1 \end{pmatrix} \quad \quad [g](\vec{x}) = \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} \cdot \vec{x} + \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$
A non-simply terminating TRS: \( f(f(x)) \rightarrow f(g(f(x))) \)

**Example \((S = \{ f(f(x)) \rightarrow f(g(f(x))) \})\)**

We choose the \( \Sigma \)-algebra \((\mathbb{N}^2, [\cdot])\) with:

\[
[f](\vec{x}) = \begin{pmatrix} 1 & 1 \\ 0 & 0 \end{pmatrix} \cdot \vec{x} + \begin{pmatrix} 0 \\ 1 \end{pmatrix} \\
[g](\vec{x}) = \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} \cdot \vec{x} + \begin{pmatrix} 0 \\ 0 \end{pmatrix}
\]

where \( > \) on \( \mathbb{N}^2 \) is defined as follows:

\[
\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} > \begin{pmatrix} y_1 \\ y_2 \end{pmatrix} \iff x_1 > y_1 \text{ and } x_2 \geq y_2
\]
Polynomial Interpretations

A non-simply terminating TRS: \( f(f(x)) \rightarrow f(g(f(x))) \)

**Example \((S = \{ f(f(x)) \rightarrow f(g(f(x))) \})\)**

We choose the \( \Sigma \)-algebra \((\mathbb{N}^2, [\cdot])\) with:

\[
[f](\vec{x}) = \begin{pmatrix} 1 & 1 \\ 0 & 0 \end{pmatrix} \cdot \vec{x} + \begin{pmatrix} 0 \\ 1 \end{pmatrix} \quad \quad [g](\vec{x}) = \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} \cdot \vec{x} + \begin{pmatrix} 0 \\ 0 \end{pmatrix}
\]

where \(>\) on \( \mathbb{N}^2 \) is defined as follows:

\[
\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} > \begin{pmatrix} y_1 \\ y_2 \end{pmatrix} \iff x_1 > y_1 \text{ and } x_2 \geq y_2
\]

Let \( \alpha: \mathcal{X} \rightarrow A \) be arbitrary; write \( \vec{x} = \alpha(x) \). We obtain

\[
[f(f(x))] = \begin{pmatrix} 1 & 1 \\ 0 & 0 \end{pmatrix} \cdot \vec{x} + \begin{pmatrix} 1 \end{pmatrix} > \begin{pmatrix} 1 & 1 \\ 0 & 0 \end{pmatrix} \cdot \vec{x} + \begin{pmatrix} 0 \\ 1 \end{pmatrix} = [f(g(f(x)))]
\]
A non-simply terminating TRS: $f(f(x)) \rightarrow f(g(f(x)))$

**Example ($S = \{ f(f(x)) \rightarrow f(g(f(x))) \}$)**

We choose the Σ-algebra $(\mathbb{N}^2, [\cdot])$ with:

$$[f](\vec{x}) = \begin{pmatrix} 1 & 1 \\ 0 & 0 \end{pmatrix} \cdot \vec{x} + \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

$$[g](\vec{x}) = \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} \cdot \vec{x} + \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

where $>\,\text{on}\, \mathbb{N}^2$ is defined as follows:

$$\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} > \begin{pmatrix} y_1 \\ y_2 \end{pmatrix} \iff x_1 > y_1 \,\text{and}\, x_2 \geq y_2$$

Let $\alpha: \mathcal{X} \rightarrow A$ be arbitrary; write $\vec{x} = \alpha(x)$. We obtain

$$[f(f(x))] = \begin{pmatrix} 1 & 1 \\ 0 & 0 \end{pmatrix} \cdot \vec{x} + \begin{pmatrix} 1 \\ 1 \end{pmatrix} > \begin{pmatrix} 1 & 1 \\ 0 & 0 \end{pmatrix} \cdot \vec{x} + \begin{pmatrix} 0 \\ 1 \end{pmatrix} = [f(g(f(x)))].$$

Hence $S$ is terminating.
Termination via Dependency Pairs
We call a term minimal if all its strict subterms are terminating.

We prove: every non-terminating term \( t \) contains a minimal subterm.

We use induction on the term structure of \( t \):

- Base case: all strict subterms of \( t \) are terminating. Then \( t \) is minimal itself.
- Induction step: \( t \) has a strict subterm \( t' \) which is not terminating.

Then by IH the term \( t' \) contains a minimal, non-terminating subterm \( t'' \). The term \( t'' \) is a minimal non-terminating subterm of \( t \).
We call a term is minimal if all its strict subterms are terminating.

**Theorem**

Let \( R \) be a non-terminating TRS. Then there exists a minimal term which is non-terminating.

We prove: every non-terminating term \( t \) contains a minimal subterm.

We use induction on the term structure of \( t \):

- **Base case:** all strict subterms of \( t \) are terminating. Then \( t \) is minimal itself.
- **Induction step:** \( t \) has a strict subterm \( t' \) which is not terminating.

Then by IH the term \( t' \) contains a minimal, non-terminating subterm \( t'' \). The term \( t'' \) is a minimal non-terminating subterm of \( t \).
We call a term is \textit{minimal} if all its strict subterms are terminating.

**Theorem**

\textit{Let }$R$\textit{ be a non-terminating TRS. Then there exists a minimal term which is non-terminating.}

**Proof.**

Since $R$ is non-terminating, there exists a non-terminating term $t$. 
We call a term is **minimal** if all its strict subterms are terminating.

**Theorem**

Let $R$ be a non-terminating TRS.

Then there exists a **minimal** term which is non-terminating.

**Proof.**

Since $R$ is non-terminating, there exists a non-terminating term $t$.

We prove: every non-terminating term $t$ contains a minimal subterm.
Minimal Terms

We call a term is **minimal** if all its strict subterms are terminating.

**Theorem**

*Let R be a non-terminating TRS.*

*Then there exists a minimal term which is non-terminating.*

**Proof.**

Since $R$ is non-terminating, there exists a non-terminating term $t$.

We prove: every non-terminating term $t$ contains a minimal subterm.

We use induction on the term structure of $t$: 
We call a term is minimal if all its strict subterms are terminating.

**Theorem**

Let $R$ be a non-terminating TRS. Then there exists a minimal term which is non-terminating.

**Proof.**

Since $R$ is non-terminating, there exists a non-terminating term $t$.

We prove: every non-terminating term $t$ contains a minimal subterm.

We use induction on the term structure of $t$:

- Base case: all strict subterms of $t$ are terminating.
We call a term is **minimal** if all its strict subterms are terminating.

**Theorem**

*Let R be a non-terminating TRS. Then there exists a minimal term which is non-terminating.*

**Proof.**

Since R is non-terminating, there exists a non-terminating term $t$.

We prove: every non-terminating term $t$ contains a minimal subterm.

We use induction on the term structure of $t$:

- Base case: all strict subterms of $t$ are terminating. Then $t$ is minimal itself.
Minimal Terms

We call a term is minimal if all its strict subterms are terminating.

Theorem

Let $R$ be a non-terminating TRS.
Then there exists a minimal term which is non-terminating.

Proof.

Since $R$ is non-terminating, there exists a non-terminating term $t$.

We prove: every non-terminating term $t$ contains a minimal subterm.

We use induction on the term structure of $t$:

- Base case: all strict subterms of $t$ are terminating. Then $t$ is minimal itself.
- Induction step: $t$ has a strict subterm $t'$ which is not terminating.
We call a term is minimal if all its strict subterms are terminating.

Theorem

Let $R$ be a non-terminating TRS. Then there exists a minimal term which is non-terminating.

Proof.

Since $R$ is non-terminating, there exists a non-terminating term $t$.

We prove: every non-terminating term $t$ contains a minimal subterm.

We use induction on the term structure of $t$:

- Base case: all strict subterms of $t$ are terminating. Then $t$ is minimal itself.
- Induction step: $t$ has a strict subterm $t'$ which is not terminating.

Then by IH the term $t'$ contains a minimal, non-terminating subterm $t''$. 
We call a term is minimal if all its strict subterms are terminating.

**Theorem**

Let \( R \) be a non-terminating TRS.
Then there exists a minimal term which is non-terminating.

**Proof.**

Since \( R \) is non-terminating, there exists a non-terminating term \( t \).

We prove: every non-terminating term \( t \) contains a minimal subterm.

We use induction on the term structure of \( t \):

- **Base case:** all strict subterms of \( t \) are terminating. Then \( t \) is minimal itself.
- **Induction step:** \( t \) has a strict subterm \( t' \) which is not terminating.
  
  Then by IH the term \( t' \) contains a minimal, non-terminating subterm \( t'' \). The term \( t'' \) is a minimal non-terminating subterm of \( t \).
Steps below the root do not change minimality:

Then since \( t \) is minimal, it follows that all \( t_k \) are terminating. Since \( t_k \rightarrow^* s_k \), we obtain \( s_k \) is terminating for all \( 1 \leq k \leq n \). Hence \( s \) is minimal.
Steps below the root do not change minimality:

**Lemma**

Let \( t \) be minimal and \( t \rightarrow s \) a rewrite step below the root. Then \( s \) is minimal.

Then since \( t \) is minimal, it follows that all \( t_k \) are terminating.

Since \( t_k \rightarrow^* s_k \), we obtain \( s_k \) is terminating for all \( 1 \leq k \leq n \).

Hence \( s \) is minimal.
Steps below the root do not change minimality:

**Lemma**

*Let $t$ be minimal and $t \rightarrow s$ a rewrite step below the root. Then $s$ is minimal.*

**Proof.**

We have $t = f(t_1, \ldots, t_n)$ and $s = f(s_1, \ldots, s_n)$ and $i \in \mathbb{N}$ such that:

- $t_i \rightarrow s_i$, and
- $t_j = s_j$ for all $j \neq i$. 
Steps below the root do not change minimality:

**Lemma**

Let $t$ be minimal and $t \rightarrow s$ a rewrite step below the root. Then $s$ is minimal.

**Proof.**

We have $t = f(t_1, \ldots, t_n)$ and $s = f(s_1, \ldots, s_n)$ and $i \in \mathbb{N}$ such that:

- $t_i \rightarrow s_i$, and
- $t_j = s_j$ for all $j \neq i$.

Then since $t$ is minimal, it follows that all $t_k$ are terminating.
Steps below the root do not change minimality:

**Lemma**

Let $t$ be minimal and $t \rightarrow s$ a rewrite step below the root. Then $s$ is minimal.

**Proof.**

We have $t = f(t_1, \ldots, t_n)$ and $s = f(s_1, \ldots, s_n)$ and $i \in \mathbb{N}$ such that:

- $t_i \rightarrow s_i$, and
- $t_j = s_j$ for all $j \neq i$.

Then since $t$ is minimal, it follows that all $t_k$ are terminating.

Since $t_k \rightarrow^* s_k$, we obtain $s_k$ is terminating for all $1 \leq k \leq n$. 
Steps below the root do not change minimality:

**Lemma**

Let $t$ be minimal and $t \rightarrow s$ a rewrite step below the root. Then $s$ is minimal.

**Proof.**

We have $t = f(t_1, \ldots, t_n)$ and $s = f(s_1, \ldots, s_n)$ and $i \in \mathbb{N}$ such that:

- $t_i \rightarrow s_i$, and
- $t_j = s_j$ for all $j \neq i$.

Then since $t$ is minimal, it follows that all $t_k$ are terminating.

Since $t_k \rightarrow^* s_k$, we obtain $s_k$ is terminating for all $1 \leq k \leq n$.

Hence $s$ is minimal.
Assume $R$ is non-terminating.
Assume $R$ is non-terminating. There exists a minimal, non-terminating term $t_0$.

\[ t_0 \rightarrow_R t_1 \rightarrow_R t_2 \rightarrow_R \ldots \]
Dependency Pairs, Introduction

Assume $R$ is non-terminating. There exists a minimal, non-terminating term $t_0$.

\[ t_0 \xrightarrow{R} t_1 \xrightarrow{R} t_2 \xrightarrow{R} \ldots \]

At some point there must be a root step $t_i \rightarrow t_{i+1}$.
Assume $R$ is non-terminating. There exists a minimal, non-terminating term $t_0$.

$$t_0 \rightarrow_R t_1 \rightarrow_R t_2 \rightarrow_R \ldots$$

At some point there must be a root step $t_i \rightarrow t_{i+1}$.

**Proof.**

Assume all steps would be below the root. Then:

- all $t_i$ are minimal,
- all rewrite steps are in terminating terminating.
Assume $R$ is non-terminating. There exists a minimal, non-terminating term $t_0$.

\[ t_0 \rightarrow_R t_1 \rightarrow_R t_2 \rightarrow_R \ldots \]

At some point there must be a root step $t_i \rightarrow t_{i+1}$.

**Proof.**

Assume all steps would be below the root. Then:

- all $t_i$ are minimal,
- all rewrite steps are in terminating.

Notation: we use $\top \rightarrow_R$ to denote root rewrite steps.
Dependency Pairs, Introduction

Let $t_0$ be a minimal and $t_0 \rightarrow_R t_1 \rightarrow_R t_2 \rightarrow_R \ldots$ an infinite rewrite sequence.

We consider the first root rewrite step $t_i \rightarrow t_{i+1}$:
Let $t_0$ be a minimal and $t_0 \rightarrow_R t_1 \rightarrow_R t_2 \rightarrow_R \ldots$ an infinite rewrite sequence.

We consider the first root rewrite step $t_i \rightarrow t_{i+1}$:

- The term $t_{i+1}$ contains a minimal, non-terminating subterm $s$.
- The root of $s$ lies in the pattern of $r$.
- Hence there exists a non-variable subterm $r'$ of $r$ such that $s = r'\sigma$.

Idea: add a rule $\ell \rightarrow r'$ then $t_i \rightarrow s$. 

\[ t_i \equiv \ell \sigma \]
\[ t_{i+1} \equiv r \sigma \]

SN

SN & minimal

SN

SN
Let $t_0$ be a minimal and $t_0 \rightarrow_R t_1 \rightarrow_R t_2 \rightarrow_R \ldots$ an infinite rewrite sequence.

We consider the first root rewrite step $t_i \rightarrow t_{i+1}$:

- The term $t_{i+1}$ contains a minimal, non-terminating subterm $s$. 

\[ t_i \equiv \ell \sigma \]
\[ t_{i+1} \equiv r \sigma \]

\[ \neg \text{SN} \]
\[ \neg \text{SN} \text{ & minimal} \]

\[ \text{SN} \]
\[ \ell \sigma \]

\[ r \sigma \]

\[ s \sigma \]

\[ \text{SN} \]

\[ \text{SN} \]
Let $t_0$ be a minimal and $t_0 \rightarrow_R t_1 \rightarrow_R t_2 \rightarrow_R \ldots$ an infinite rewrite sequence.

We consider the first root rewrite step $t_i \rightarrow t_{i+1}$:

- The term $t_{i+1}$ contains a minimal, non-terminating subterm $s$.
- The root of $s$ lies in the pattern of $r$. 

\[ t_i \equiv \ell \sigma \]
\[ t_{i+1} \equiv r \sigma \]
Let $t_0$ be a minimal and $t_0 \rightarrow_R t_1 \rightarrow_R t_2 \rightarrow_R \ldots$ an infinite rewrite sequence.

We consider the first root rewrite step $t_i \rightarrow t_{i+1}$:

- The term $t_{i+1}$ contains a minimal, non-terminating subterm $s$.
- The root of $s$ lies in the pattern of $r$.

Hence there exists a non-variable subterm $r'$ of $r$ such that $s = r' \sigma$. 
Let \( t_0 \) be a minimal and \( t_0 \rightarrow^R t_1 \rightarrow^R t_2 \rightarrow^R \ldots \) an infinite rewrite sequence.

We consider the first root rewrite step \( t_i \rightarrow t_{i+1} \):

- The term \( t_{i+1} \) contains a minimal, non-terminating subterm \( s \).
- The root of \( s \) lies in the pattern of \( r \).

Hence there exists a non-variable subterm \( r' \) of \( r \) such that \( s = r'\sigma \).

Idea: add a rule \( \ell \rightarrow r' \) then \( t_i \rightarrow s \).
Dependency Pairs

For every $f \in \Sigma$ let $f#$ be a fresh symbol with the same arity as $f$.

$DP(R) = \{ f#(x) \rightarrow g#(f(x)),
\quad f#(x) \rightarrow f#(x) \}$
For every $f \in \Sigma$ let $f_\#$ be a fresh symbol with the same arity as $f$. By $t_\#$ we denote $f_\#(t_1, \ldots, t_n)$ for $t = f(t_1, \ldots, t_n) \in \mathcal{T}(\Sigma, \mathcal{X})$.

$$\text{DP}(R) = \{ f_\#(x) \rightarrow g_\#(f(x)), f_\#(x) \rightarrow f_\#(x) \}$$
For every \( f \in \Sigma \) let \( f\# \) be a fresh symbol with the same arity as \( f \).

By \( t\# \) we denote \( f\#(t_1, \ldots, t_n) \) for \( t = f(t_1, \ldots, t_n) \in T(\Sigma, \mathcal{X}) \).

**Definition (Dependency Pairs)**

\[
DP(R) = \{ \ell\# \to r'\# \mid \ell \to r \in R, \ r' \sqsubseteq r \text{ with } r' \notin \mathcal{X} \}
\]
For every \( f \in \Sigma \) let \( f_\# \) be a fresh symbol with the same arity as \( f \).
By \( t_\# \) we denote \( f_\#(t_1, \ldots, t_n) \) for \( t = f(t_1, \ldots, t_n) \in \mathcal{T}(\Sigma, \mathcal{X}) \).

**Definition (Dependency Pairs)**

\[
\text{DP}(R) = \{ \ell_\# \to r'_\# \mid \ell \to r \in R, \ r' \sqsubseteq r \text{ with } r' \notin \mathcal{X} \}
\]

**Example**

\[
R = \{ f(x) \to g(f(x)) \}
\]
For every \( f \in \Sigma \) let \( f_\# \) be a fresh symbol with the same arity as \( f \). By \( t_\# \) we denote \( f_\#(t_1, \ldots, t_n) \) for \( t = f(t_1, \ldots, t_n) \in \mathcal{T}(\Sigma, \mathcal{X}) \).

**Definition (Dependency Pairs)**

\[
\text{DP}(R) = \{ \ell_\# \rightarrow r'_\# \mid \ell \rightarrow r \in R, \ r' \sqsubseteq r \text{ with } r' \not\in \mathcal{X} \}
\]

**Example**

\[
R = \{ f(x) \rightarrow g(f(x)) \}
\]

\[
\text{DP}(R) = \{ f_\#(x) \rightarrow g_\#(f(x)), \ f_\#(x) \rightarrow f_\#(x) \}
\]
Dependency Pairs

Let $t_0$ be a minimal and $t_0 \rightarrow_R t_1 \rightarrow_R t_2 \rightarrow_R \ldots$ an infinite rewrite sequence.

We consider the first root rewrite step $t_i \rightarrow t_{i+1}$:

\[ t_i \equiv \ell \sigma \]

\[ t_{i+1} \equiv r \sigma \]

\[ \neg \text{SN} \& \text{minimal} \]

\[ \text{SN} \]

\[ \text{SN} \]

\[ \ell \]

\[ r \]

\[ s \]

\[ \sigma \]

\[ \text{DP}(R) \]
Let $t_0$ be a minimal and $t_0 \rightarrow_R t_1 \rightarrow_R t_2 \rightarrow_R \ldots$ an infinite rewrite sequence.

We consider the first root rewrite step $t_i \rightarrow t_{i+1}$:

Then

$$t_0,\# \rightarrow_R \ldots \rightarrow_R t_i,\# \rightarrow_{\text{DP}(R)} t_{i+1},\#$$
Let $t_0$ be a minimal and $t_0 \rightarrow_R t_1 \rightarrow_R t_2 \rightarrow_R \ldots$ an infinite rewrite sequence.

We consider the first root rewrite step $t_i \rightarrow t_{i+1}$:

Then

$$t_0,\# \rightarrow_R \ldots \rightarrow_R t_i,\# \rightarrow_{\text{DP}(R)}^{\text{top}} s\#$$

Repeating the construction with $s$ yields:

$$t_0,\# \rightarrow^* \rightarrow_{\text{DP}(R)}^{\text{top}} \rightarrow^* \rightarrow_{\text{DP}(R)}^{\text{top}} \rightarrow^* \rightarrow_{\text{DP}(R)}^{\text{top}} \ldots$$

an infinite rewrite sequence containing infinitely many $\text{DP}(R)$ steps.
Lemma

Let $R$ be a non-terminating TRS. Then there exists a rewrite sequence:

$$t_0, \# \rightarrow^* R \cdot \rightarrow^\text{top} \text{DP}(R) \cdot \rightarrow^* R \cdot \rightarrow^\text{top} \text{DP}(R) \cdot \rightarrow^* R \cdot \rightarrow^\text{top} \text{DP}(R) \cdots$$

such that:

- the sequence contains infinitely many $\text{DP}(R)$ steps,
- all $R$ steps are below the root, and
- all $\text{DP}(R)$ steps are at the root.
Example

\[ R = \{ f(g(x)) \rightarrow g(g(f(f(x)))) \} \]
Example

\[ R = \{ f(g(x)) \rightarrow g(g(f(f(x)))) \} \]

\[ \text{DP}(R) = \{ f_\#(g(x)) \rightarrow g_\#(g(f(f(x)))) ,\]
\[ f_\#(g(x)) \rightarrow g_\#(f(f(x))),\]
\[ f_\#(g(x)) \rightarrow f_\#(f(x)),\]
\[ f_\#(g(x)) \rightarrow f_\#(x) \} \]
Dependency Pairs, Examples

Example

\[ R = \{ \ f(g(x)) \rightarrow g(g(f(f(x)))) \} \]

\[ \text{DP}(R) = \{ \ f#(g(x)) \rightarrow g#(g(f(f(x))))), \]
\[ f#(g(x)) \rightarrow g#(f(f(x))), \]
\[ f#(g(x)) \rightarrow f#(f(x)), \]
\[ f#(g(x)) \rightarrow f#(x) \} \]

\( R \) is non-terminating:

\[ f(g(g(x))) \rightarrow g(g(f(g(x)))) \rightarrow g(g(f(g(g(f(f(x))))))) \rightarrow \ldots \]
Dependency Pairs, Examples

Example

\[ R = \{ f(g(x)) \rightarrow g(g(f(f(x)))) \} \]

\[ \text{DP}(R) = \{ f\#(g(x)) \rightarrow g\#(g(f(f(x)))), \]
\[ f\#(g(x)) \rightarrow g\#(f(f(x))), \]
\[ f\#(g(x)) \rightarrow f\#(f(x)), \]
\[ f\#(g(x)) \rightarrow f\#(x) \} \]

\( R \) is non-terminating:

\[ f(g(g(x))) \rightarrow g(g(f(g(f(x)))))) \rightarrow g(g(f(g(g(g(f(f(x)))))))) \rightarrow \ldots \]

How does an infinite \( R \cup \xrightarrow{\text{top}}_{\text{DP}(R)} \) rewrite sequence look like?
Dependency Pairs, Examples

Example

\[ R = \{ f(g(x)) \rightarrow g(g(f(f(x)))) \} \]

\[ \text{DP}(R) = \{ f\#(g(x)) \rightarrow g\#(g(f(f(x)))), \]
\[ f\#(g(x)) \rightarrow g\#(f(x)), \]
\[ f\#(g(x)) \rightarrow f\#(x) \} \]

\( R \) is non-terminating:

\[ f(g(g(x))) \rightarrow g(g(f(g(x)))) \rightarrow g(g(g(f(g(x)))))) \rightarrow \ldots \]

How does an infinite \( \rightarrow_R \cup \rightarrow_{\text{DP}(R)} \) rewrite sequence look like?

\[ f\#(g(g(x))) \rightarrow_{\text{DP}(R)} f\#(f(g(x))) \rightarrow_R f\#(g(g(f(f(x)))))) \rightarrow \ldots \]
Example

\[ R = \{ \text{A}(x, s(y)) \rightarrow s(A(x, y)), \]
\[ \text{A}(x, 0) \rightarrow x \} \]
Dependency Pairs, Examples

Example

\[
\begin{align*}
R &= \{ \text{A}(x, s(y)) \rightarrow s(\text{A}(x, y)), \\
&\quad \text{A}(x, 0) \rightarrow x \} \\
\text{DP}(R) &= \{ \text{A}_\#(x, s(y)) \rightarrow s_\#(\text{A}(x, y)), \\
&\quad \text{A}_\#(x, s(y)) \rightarrow \text{A}_\#(x, y) \}
\end{align*}
\]
Definition (Relative Termination)

A relation \( \rightarrow_1 \) is called terminating relative to \( \rightarrow_2 \), denoted \( \text{SN}(\rightarrow_1 / \rightarrow_2) \), if every \( \rightarrow_1 \cup \rightarrow_2 \) rewrite sequence contains only finitely many \( \rightarrow_1 \) steps.
**Definition (Relative Termination)**

A relation $\rightarrow_1$ is called terminating relative to $\rightarrow_2$, denoted $SN(\rightarrow_1 / \rightarrow_2)$, if every $\rightarrow_1 \cup \rightarrow_2$ rewrite sequence contains only finitely many $\rightarrow_1$ steps.

**Lemma**

$$SN(\rightarrow_1 / \rightarrow_2) \iff SN(\rightarrow_2^* \cdot \rightarrow_1 \cdot \rightarrow_2^*)$$
Dependency Pairs, Main Theorem

**Definition (Relative Termination)**

A relation $\rightarrow_1$ is called **terminating relative to** $\rightarrow_2$, denoted $\text{SN}(\rightarrow_1 / \rightarrow_2)$, if every $\rightarrow_1 \cup \rightarrow_2$ rewrite sequence contains only finitely many $\rightarrow_1$ steps.

**Lemma**

$$\text{SN}(\rightarrow_1 / \rightarrow_2) \iff \text{SN}(\rightarrow_2^* \cdot \rightarrow_1 \cdot \rightarrow_2^*)$$

The main theorem from dependency pairs is:

**Theorem**

$$\text{SN}(R) \iff \text{SN}(\text{DP}(R)_{top}/R)$$
**Definition (Relative Termination)**

A relation $\rightarrow_1$ is called **terminating relative to** $\rightarrow_2$, denoted $\text{SN}(\rightarrow_1 / \rightarrow_2)$, if every $\rightarrow_1 \cup \rightarrow_2$ rewrite sequence contains only finitely many $\rightarrow_1$ steps.

**Lemma**

\[
\text{SN}(\rightarrow_1 / \rightarrow_2) \iff \text{SN}(\rightarrow_2^* \cdot \rightarrow_1 \cdot \rightarrow_2^*)
\]

The main theorem from dependency pairs is:

**Theorem**

\[
\text{SN}(R) \iff \text{SN}(\text{DP}(R)_{\text{top}} / R)
\]

That is, a TRS $R$ is terminating if and only if $\rightarrow_{\text{DP}(R)}^{\text{top}}$ terminates relative to $\rightarrow_R$. 
Dependency Pairs, Termination Proofs

Definition

A well-founded weakly monotone $\Sigma$-algebra $(A, [\cdot], >, \succeq)$ consists of:

- a $\Sigma$-algebra $(A, [\cdot])$ with relations $>$, $\succeq$ on $A$
- $>$ is well-founded,
- $> \cdot \succeq \subseteq >$ (compatibility),
- for all $f \in \Sigma$ the function $[f]$ is monotone w.r.t. $\succeq$. 

Theorem

$SN(DP(R)_{top}/R)$ if there exists a weakly monotone $\Sigma$-algebra s.t.

- $DP(R)$ is well-founded,
- $R \subseteq \succeq$ that is, $[\ell, \alpha] \succeq [r, \alpha] \quad \forall \alpha, \ell \rightarrow r \in DP(R)$

Advantages: no monotonicity for $>$, and $\succeq$ not well-founded.

Term Rewriting Systems - Lecture 4
**Definition**

A well-founded weakly monotone $\Sigma$-algebra $(A, [\cdot], >, \succeq)$ consists of:

- a $\Sigma$-algebra $(A, [\cdot])$ with relations $>$, $\succeq$ on $A$
- $>$ is well-founded,
- $> \cdot \succeq \subseteq >$ (compatibility),
- for all $f \in \Sigma$ the function $[f]$ is monotone w.r.t. $\succeq$

**Theorem**

$SN(DP(R)_{top}/R)$ if there exists a weakly monotone $\Sigma$-algebra s.t.

- $DP(R) \subseteq >$ 
  that is, $[\ell, \alpha] > [r, \alpha] \quad \forall \alpha, \ell \rightarrow r \in DP(R)$
- $R \subseteq \succeq$ 
  that is, $[\ell, \alpha] \succeq [r, \alpha] \quad \forall \alpha, \ell \rightarrow r \in R$
**Definition**

A well-founded weakly monotone \( \Sigma \)-algebra \((A, [\cdot], >, \succeq)\) consists of:

- a \( \Sigma \)-algebra \((A, [\cdot])\) with relations \(>\), \(\succeq\) on \(A\)
- \(>\) is well-founded,
- \(> \cdot \succeq \subseteq >\) (compatibility),
- for all \(f \in \Sigma\) the function \([f]\) is monotone w.r.t. \(\succeq\)

**Theorem**

\(\text{SN}(\text{DP}(R))_{top}/R)\) if there exists a weakly monotone \(\Sigma\)-algebra s.t.

- \(\text{DP}(R) \subseteq >\)  
  that is, \([\ell, \alpha] > [r, \alpha]\) \(\forall \alpha, \ell \rightarrow r \in \text{DP}(R)\)
- \(R \subseteq \succeq\)  
  that is, \([\ell, \alpha] \succeq [r, \alpha]\) \(\forall \alpha, \ell \rightarrow r \in R\)

Advantages: no monotonicity for \(>\), and \(\succeq\) not well-founded.
Frequently used are polynomial interpretations over \( \mathbb{N} \):

- \( > \) as usual on \( \mathbb{N} \) and \( \geq := \geq \)
- the interpretations \([f]\) are polynomials
Frequently used are polynomial interpretations over $\mathbb{N}$:

- $>$ as usual on $\mathbb{N}$ and $\succeq := \geq$
- the interpretations $[f]$ are polynomials

We will see some examples...
Example: \( f(f(x)) \rightarrow f(g(f(x))) \)
Example: $f(f(x)) \rightarrow f(g(f(x)))$

Example

$$DP(R) = \{ f#(f(x)) \rightarrow f#(g(f(x))), \\
               f#(f(x)) \rightarrow g#(f(x)), \\
               f#(f(x)) \rightarrow f#(x) \}$$
Example: \( f(f(x)) \rightarrow f(g(f(x))) \)

Example

\[
\text{DP}(R) = \{ f#(f(x)) \rightarrow f#(g(f(x))), \\
f#(f(x)) \rightarrow g#(f(x)), \\
f#(f(x)) \rightarrow f#(x) \}
\]

\[
[f](x) = ???, \quad [f#](x) = ???, \quad [g](x) = ???, \quad [g#](x) = ???
\]
Example: $f(f(x)) \rightarrow f(g(f(x)))$

Example

$$DP(R) = \{ f_#(f(x)) \rightarrow f_#(g(f(x))),$$
$$f_#(f(x)) \rightarrow g_#(f(x)),$$
$$f_#(f(x)) \rightarrow f_#(x) \}$$

\[ f(x) = x + 1 \quad f_#(x) = x + 1 \quad g(x) = 0 \quad g_#(x) = 0 \]
Example: $f(f(x)) \rightarrow f(g(f(x)))$

Example

$$DP(R) = \{ f_#(f(x)) \rightarrow f_#(g(f(x))), \\ f_#(f(x)) \rightarrow g_#(f(x)), \\ f_#(f(x)) \rightarrow f_#(x) \}$$

$$[f](x) = x + 1 \quad [f_#](x) = x + 1 \quad [g](x) = 0 \quad [g_#](x) = 0$$

• Are the functions $[f]$ monotone w.r.t. $\geq$?
Example: \( f(f(x)) \rightarrow f(g(f(x))) \)

Example

\[
\text{DP}(R) = \{ \begin{align*}
& f_\#(f(x)) \rightarrow f_\#(g(f(x))), \\
& f_\#(f(x)) \rightarrow g_\#(f(x)), \\
& f_\#(f(x)) \rightarrow f_\#(x) \}
\]

\[
[f](x) = x + 1 \quad [f_\#](x) = x + 1 \quad [g](x) = 0 \quad [g_\#](x) = 0
\]

• Are the functions \([f]\) monotone w.r.t. \(\geq\)?
  
  Yes, since whenever \(a \geq b\), then
  
  \[
  [f](a) = a \geq b = [f](b),
  [g](a) = 0 \geq 0 = [g](b),
  \]
Example: \( f(f(x)) \rightarrow f(g(f(x))) \)

Example

\[
\text{DP}(R) = \{ \begin{align*}
&f_\#(f(x)) \rightarrow f_\#(g(f(x))), \\
&f_\#(f(x)) \rightarrow g_\#(f(x)), \\
&f_\#(f(x)) \rightarrow f_\#(x)
\end{align*} \}
\]

\[
[f](x) = x + 1 \quad [f_\#](x) = x + 1 \quad [g](x) = 0 \quad [g_\#](x) = 0
\]

• Does \([\ell, \alpha] > [r, \alpha]\) for all \(\ell \rightarrow r \in \text{DP}(R)\) hold?
Example: $f(f(x)) \rightarrow f(g(f(x)))$

Dependency Pairs

Example

$$DP(R) = \{ f_\#(f(x)) \rightarrow f_\#(g(f(x))), \\
      f_\#(f(x)) \rightarrow g_\#(f(x)), \\
      f_\#(f(x)) \rightarrow f_\#(x) \}$$

$$[f](x) = x + 1 \quad [f_\#](x) = x + 1 \quad [g](x) = 0 \quad [g_\#](x) = 0$$

• Does $[\ell, \alpha] > [r, \alpha]$ for all $\ell \rightarrow r \in DP(R)$ hold?

  $[f_\#(f(x)), \alpha] = \alpha(x) + 2 > 1 = [f_\#(g(f(x))), \alpha]$

  $[f_\#(f(x)), \alpha] = \alpha(x) + 2 > 0 = [g_\#(f(x)), \alpha]$

  $[f_\#(f(x)), \alpha] = \alpha(x) + 2 > \alpha(x) + 1 = [f_\#(x), \alpha]$
Dependency Pairs

Example: \( f(f(x)) \rightarrow f(g(f(x))) \)

Example

\[
\text{DP}(R) = \{ f_#(f(x)) \rightarrow f_#(g(f(x))), \\
f_#(f(x)) \rightarrow g_#(f(x)), \\
f_#(f(x)) \rightarrow f_#(x) \} \\
\]

\[ f(x) = x + 1 \quad f_#(x) = x + 1 \quad g(x) = 0 \quad g_#(x) = 0 \]

• Does \([\ell, \alpha] > [r, \alpha]\) for all \(\ell \rightarrow r \in \text{DP}(R)\) hold?

  \[ [f_#(f(x)), \alpha] = \alpha(x) + 2 > 1 = [f_#(g(f(x))), \alpha] \]

  \[ [f_#(f(x)), \alpha] = \alpha(x) + 2 > 0 = [g_#(f(x)), \alpha] \]

  \[ [f_#(f(x)), \alpha] = \alpha(x) + 2 > \alpha(x) + 1 = [f_#(x), \alpha] \]

• Does \([\ell, \alpha] \geq [r, \alpha]\) for all \(\ell \rightarrow r \in R\) hold?
Example: \( f(f(x)) \rightarrow f(g(f(x))) \)

\[
\text{DP}(R) = \{ f_\#(f(x)) \rightarrow f_\#(g(f(x))), \\
               f_\#(f(x)) \rightarrow g_\#(f(x)), \\
               f_\#(f(x)) \rightarrow f_\#(x) \}
\]

\[
[f](x) = x + 1 \quad [f_\#](x) = x + 1 \quad [g](x) = 0 \quad [g_\#](x) = 0
\]

• Does \([\ell, \alpha] > [r, \alpha]\) for all \(\ell \rightarrow r \in \text{DP}(R)\) hold?
  \[
  [f_\#(f(x)), \alpha] = \alpha(x) + 2 > 1 = [f_\#(g(f(x))), \alpha] \\
  [f_\#(f(x)), \alpha] = \alpha(x) + 2 > 0 = [g_\#(f(x)), \alpha] \\
  [f_\#(f(x)), \alpha] = \alpha(x) + 2 > \alpha(x) + 1 = [f_\#(x), \alpha]
  \]

• Does \([\ell, \alpha] \geq [r, \alpha]\) for all \(\ell \rightarrow r \in R\) hold?
  \[
  [f(f(x)), \alpha] = \alpha(x) + 2 \geq 1 = [f(g(f(x))), \alpha]
  \]
Dependency Pairs

Example: \( f(f(x)) \rightarrow f(g(f(x))) \)

Example

\[
\text{DP}(R) = \{ f\#(f(x)) \rightarrow f\#(g(f(x))), \\
f\#(f(x)) \rightarrow g\#(f(x)), \\
f\#(f(x)) \rightarrow f\#(x) \}
\]

\[
[f](x) = x + 1 \quad [f\#](x) = x + 1 \quad [g](x) = 0 \quad [g\#](x) = 0
\]

• Does \([\ell, \alpha] > [r, \alpha]\) for all \(\ell \rightarrow r \in \text{DP}(R)\) hold?
  \[
  [f\#(f(x)), \alpha] = \alpha(x) + 2 > 1 = [f\#(g(f(x))), \alpha]
  \]
  \[
  [f\#(f(x)), \alpha] = \alpha(x) + 2 > 0 = [g\#(f(x)), \alpha]
  \]
  \[
  [f\#(f(x)), \alpha] = \alpha(x) + 2 > \alpha(x) + 1 = [f\#(x), \alpha]
  \]

• Does \([\ell, \alpha] \geq [r, \alpha]\) for all \(\ell \rightarrow r \in R\) hold?
  \[
  [f(f(x)), \alpha] = \alpha(x) + 2 \geq 1 = [f(g(f(x))), \alpha]
  \]

Hence we have proven termination.
Stepwise Termination Proofs

Stepwise termination proofs with monotone $\Sigma$-algebras:

**Theorem**

If there exists a monotone $\Sigma$-algebra $(A, [], >)$ s.t.

- $R \subseteq \geq$, and
- $R' \subseteq >$, and

where $\geq := > \cup =$. Then

$$\text{SN}(R) \implies \text{SN}(R \cup R')$$

This theorem allows us to stepwise remove rules until none are left.
Stepwise Termination Proofs

Stepwise termination proofs with monotone $\Sigma$-algebras:

**Theorem**

If there exists a monotone $\Sigma$-algebra $(A, [\cdot], >)$ s.t.

- $R \subseteq \geq$, and
- $R' \subseteq >$, and

where $\geq := > \cup =$. Then

$$SN(R) \implies SN(R \cup R')$$

This theorem allows us to stepwise remove rules until none are left.

**Remark**

Instead of $\geq$ we can more generally use a monotone relation $\succeq$ with $> \cdot \succeq \subseteq >$.
Example

\[ f(f(x)) \rightarrow g(x) \quad \text{and} \quad f(g(x)) \rightarrow g(f(x)) \]
Example

\[ f(f(x)) \rightarrow g(x) \quad f(g(x)) \rightarrow g(f(x)) \]

We use the following interpretation:

\[ [f](x) = x + 1 \quad [g](x) = x + 1 \]
Example

\[ f(f(x)) \rightarrow g(x) \quad f(g(x)) \rightarrow g(f(x)) \]

We use the following interpretation:

\[ [f](x) = x + 1 \quad [g](x) = x + 1 \]

We get the following interpretation of rules:

\[ [f(g(x)), \alpha] = \alpha(x) + 2 > \alpha(x) + 1 = [g(x), \alpha] \]

\[ [f(g(x)), \alpha] = \alpha(x) + 2 \geq \alpha(x) + 2 = [g(f(x)), \alpha] \]
Example

\[ f(f(x)) \rightarrow g(x) \quad f(g(x)) \rightarrow g(f(x)) \]

We use the following interpretation:

\[ [f(x)] = x + 1 \quad [g(x)] = x + 1 \]

We get the following interpretation of rules:

\[ [f(g(x)), \alpha] = \alpha(x) + 2 > \alpha(x) + 1 = [g(x), \alpha] \]
\[ [f(g(x)), \alpha] = \alpha(x) + 2 \geq \alpha(x) + 2 = [g(f(x)), \alpha] \]

The first rule is strictly decreasing, hence we can remove it.
Stepwise Termination Proofs

Example

\[ f(f(x)) \rightarrow g(x) \quad \text{f}(g(x)) \rightarrow g(f(x)) \]

We use the following interpretation:

\[ [f](x) = x + 1 \quad [g](x) = x + 1 \]

We get the following interpretation of rules:

\[ [f(g(x)), \alpha] = \alpha(x) + 2 > \alpha(x) + 1 = [g(x), \alpha] \]
\[ [f(g(x)), \alpha] = \alpha(x) + 2 \geq \alpha(x) + 2 = [g(f(x)), \alpha] \]

The first rule is strictly decreasing, hence we can remove it.

Thus for termination it suffices to show \(\text{SN}(f(g(x)) \rightarrow g(f(x))).\)
Example

We use the following interpretation:

\[
[f](x) = x + 1 \\
[g](x) = x + 1
\]

We get the following interpretation of rules:

\[
[f(g(x)), \alpha] = \alpha(x) + 2 > \alpha(x) + 1 = [g(x), \alpha] \\
[f(g(x)), \alpha] = \alpha(x) + 2 \geq \alpha(x) + 2 = [g(f(x)), \alpha]
\]

The first rule is strictly decreasing, hence we can remove it.

Thus for termination it suffices to show \(\text{SN}(f(g(x)) \rightarrow g(f(x)))\).

We have already shown this a few slides ago. Hence we have proven termination.
Stepwise Termination Proofs

Stepwise termination proofs with dependency pairs:

**Theorem**

*If there exists a weakly monotone \( \Sigma \)-algebra s.t.*

- \( T_1 \subseteq \succ \)
- \( T_2 \cup R \subseteq \succeq \)

*Then*

\[
\text{SN}(T_{2,\text{top}}/R) \implies \text{SN}((T_1 \cup T_2)_{\text{top}}/R)
\]

That is, we may remove the strictly decreasing top-rules.
Stepwise termination proofs with dependency pairs:

**Theorem**

*If there exists a weakly monotone \( \Sigma \)-algebra s.t.*

\[
\begin{align*}
& \bullet \ T_1 \subseteq > \\
& \bullet \ T_2 \cup R \subseteq \succeq
\end{align*}
\]

*Then*

\[
\text{SN}(T_2,_{\text{top}}/R) \implies \text{SN}((T_1 \cup T_2)_{\text{top}}/R)
\]

That is, we may remove the strictly decreasing top-rules.

Typically, \( T_1, T_2 \subseteq \text{DP}(R) \). But we can also tackle other top-termination problems.
Stepwise termination proofs with dependency pairs:

**Theorem**

If there exists a weakly monotone $\Sigma$-algebra s.t.

- $T_1 \subseteq >$
- $T_2 \cup R \subseteq \succeq$

Then

$$\text{SN}(T_2, \text{top}/R) \implies \text{SN}((T_1 \cup T_2)\text{top}/R)$$

That is, we may remove the strictly decreasing top-rules.

Typically, $T_1, T_2 \subseteq \text{DP}(R)$. But we can also tackle other top-termination problems.

We are **not** allowed to remove strictly decreasing rules in $R$!
Stepwise termination proofs with dependency pairs:

**Theorem**

If there exists a weakly monotone $\Sigma$-algebra s.t.

- $T_1 \subseteq >$
- $T_2 \cup R \subseteq \succeq$

Then

$$\text{SN}(T_{2,\text{top}}/R) \implies \text{SN}((T_1 \cup T_2)_{\text{top}}/R)$$

That is, we may remove the strictly decreasing top-rules.

Typically, $T_1, T_2 \subseteq \text{DP}(R)$. But we can also tackle other top-termination problems.

We are not allowed to remove strictly decreasing rules in $R$!

(for removing from $R$ we need monotonic interpretations)
Example

\[
\begin{align*}
\text{minus}(x, 0) & \rightarrow x \\
\text{minus}(s(x), s(y)) & \rightarrow \text{minus}(x, y) \\
\text{quot}(0, s(y)) & \rightarrow 0 \\
\text{quot}(s(x), s(y)) & \rightarrow s(\text{quot(\text{minus}(x, y), s(y))))
\end{align*}
\]
Example

\[
\begin{align*}
\text{minus}(x, 0) & \rightarrow x \\
\text{minus}(s(x), s(y)) & \rightarrow \text{minus}(x, y) \\
\text{quot}(0, s(y)) & \rightarrow 0 \\
\text{quot}(s(x), s(y)) & \rightarrow s(\text{quot}(\text{minus}(x, y), s(y)))
\end{align*}
\]

\[\text{DP}(R) = \{ \text{minus}_#(s(x), s(y)) \rightarrow \text{minus}_#(x, y) \]
\[\quad \text{quot}_#(0, s(y)) \rightarrow 0_# \]
\[\quad \text{quot}_#(s(x), s(y)) \rightarrow s_#(\text{quot}(\text{minus}(x, y), s(y))) \]
\[\quad \text{quot}_#(s(x), s(y)) \rightarrow \text{quot}_#(\text{minus}(x, y), s(y)) \]
\[\quad \text{quot}_#(s(x), s(y)) \rightarrow \text{minus}_#(x, y) \]
\[\quad \text{quot}_#(s(x), s(y)) \rightarrow s_#(y) \} \]
Example

\[
\begin{align*}
\text{minus}(x, 0) & \rightarrow x \\
\text{minus}(s(x), s(y)) & \rightarrow \text{minus}(x, y) \\
\text{quot}(0, s(y)) & \rightarrow 0 \\
\text{quot}(s(x), s(y)) & \rightarrow s(\text{quot}(\text{minus}(x, y), s(y)))
\end{align*}
\]

\[
\text{DP}(R) = \{ \text{minus}_{\#}(s(x), s(y)) \rightarrow \text{minus}_{\#}(x, y) \\
\text{quot}_{\#}(0, s(y)) \rightarrow 0_{\#} \\
\text{quot}_{\#}(s(x), s(y)) \rightarrow s_{\#}(\text{quot}(\text{minus}(x, y), s(y))) \\
\text{quot}_{\#}(s(x), s(y)) \rightarrow \text{quot}_{\#}(\text{minus}(x, y), s(y)) \\
\text{quot}_{\#}(s(x), s(y)) \rightarrow \text{minus}_{\#}(x, y) \\
\text{quot}_{\#}(s(x), s(y)) \rightarrow s_{\#}(y) \}
\]

We use the interpretation:

\[
[\text{minus}_{\#}](x, y) = 1 \quad [\text{quot}_{\#}](x, y) = 1 \quad [\text{minus}](x, y) = x \quad [s](x) = x \\
[f](\vec{x}) = 0 \text{ for all other symbols } f
\]
Example

\[
\begin{align*}
\text{minus}(x, 0) & \rightarrow x \\
\text{minus}(s(x), s(y)) & \rightarrow \text{minus}(x, y) \\
\text{quot}(0, s(y)) & \rightarrow 0 \\
\text{quot}(s(x), s(y)) & \rightarrow s(\text{quot}(\text{minus}(x, y), s(y)))
\end{align*}
\]

\[
\text{DP}(R) = \{ \text{minus}_{\#}(s(x), s(y)) \rightarrow \text{minus}_{\#}(x, y) \\
\quad \text{quot}_{\#}(0, s(y)) \rightarrow 0_{\#} \\
\quad \text{quot}_{\#}(s(x), s(y)) \rightarrow s_{\#}(\text{quot}(\text{minus}(x, y), s(y))) \\
\quad \text{quot}_{\#}(s(x), s(y)) \rightarrow \text{quot}_{\#}(\text{minus}(x, y), s(y)) \\
\quad \text{quot}_{\#}(s(x), s(y)) \rightarrow \text{minus}_{\#}(x, y) \\
\quad \text{quot}_{\#}(s(x), s(y)) \rightarrow s_{\#}(y) \}
\]

The following interpretation removes the remaining DP rules and proves:

\[
\begin{align*}
[\text{minus}_{\#}] (x, y) = [\text{minus}] (x, y) = [\text{quot}_{\#}] (x, y) = [\text{quot}] (x, y) = x \\
[s] (x) = x + 1 \\
[0] = 0
\end{align*}
\]
Example

\[
\begin{align*}
\text{minus}(x, 0) & \rightarrow x \\
\text{minus}(s(x), s(y)) & \rightarrow \text{minus}(x, y) \\
\text{quot}(0, s(y)) & \rightarrow 0 \\
\text{quot}(s(x), s(y)) & \rightarrow s(\text{quot}(\text{minus}(x, y), s(y)))
\end{align*}
\]

\[
\text{DP}(R) = \{ \text{minus}_\#(s(x), s(y)) \rightarrow \text{minus}_\#(x, y) \}
\]

\[
\begin{align*}
\text{quot}_\#(0, s(y)) & \rightarrow 0_\# \\
\text{quot}_\#(s(x), s(y)) & \rightarrow s_\#(\text{quot}(\text{minus}(x, y), s(y))) \\
\text{quot}_\#(s(x), s(y)) & \rightarrow \text{quot}_\#(\text{minus}(x, y), s(y)) \\
\text{quot}_\#(s(x), s(y)) & \rightarrow \text{minus}_\#(x, y) \\
\text{quot}_\#(s(x), s(y)) & \rightarrow s_\#(y)
\end{align*}
\]

The following interpretation removes the remaining DP rules and proves:

\[
\begin{align*}
[\text{minus}_\#](x, y) & = [\text{minus}](x, y) = [\text{quot}_\#](x, y) = [\text{quot}](x, y) = x \\
[s](x) & = x + 1 \\
[0] & = 0
\end{align*}
\]
Example

\[
\begin{align*}
\text{minus}(x, 0) & \rightarrow x \\
\text{minus}(s(x), s(y)) & \rightarrow \text{minus}(x, y) \\
\text{quot}(0, s(y)) & \rightarrow 0 \\
\text{quot}(s(x), s(y)) & \rightarrow s(\text{quot}(\text{minus}(x, y), s(y)))
\end{align*}
\]
**Example**

\[
\begin{align*}
\text{minus}(x, 0) & \rightarrow x \\
\text{minus}(s(x), s(y)) & \rightarrow \text{minus}(x, y) \\
\text{quot}(0, s(y)) & \rightarrow 0 \\
\text{quot}(s(x), s(y)) & \rightarrow s(\text{quot}(\text{minus}(x, y), s(y)))
\end{align*}
\]

Dependency graph: analysis which DP-rules may follow each other

1. \(\text{minus}(s(x), s(y)) \rightarrow \text{minus}(x, y)\)
2. \(\text{quot}(s(x), s(y)) \rightarrow \text{quot}(\text{minus}(x, y), s(y))\)
3. \(\text{quot}(s(x), s(y)) \rightarrow \text{quot}(\text{minus}(x, y), s(y))\)
Example

\[
\begin{align*}
\text{minus}(x, 0) & \rightarrow x \\
\text{minus}(s(x), s(y)) & \rightarrow \text{minus}(x, y) \\
\text{quot}(0, s(y)) & \rightarrow 0 \\
\text{quot}(s(x), s(y)) & \rightarrow s(\text{quot}(\text{minus}(x, y), s(y)))
\end{align*}
\]

Dependency graph: analysis which DP-rules may follow each other

\[
\begin{align*}
(1) & \quad \text{minus}_#(s(x), s(y)) \rightarrow \text{minus}_#(x, y) \\
(2) & \quad \text{quot}_#(s(x), s(y)) \rightarrow \text{minus}_#(x, y) \\
(3) & \quad \text{quot}_#(s(x), s(y)) \rightarrow \text{quot}_#(\text{minus}(x, y), s(y))
\end{align*}
\]

Idea: consider only strongly connected components \(\text{SN}(\{1\}_{\text{top}}/R), \text{SN}(\{3\}_{\text{top}}/R)\).
Subterm Criterion

**Theorem (Subterm Criterion)**

Let $R$ be a TRS, $T_1, T_2 \subseteq \text{DP}(R)$, and $\pi : \Sigma_\# \rightarrow \mathbb{N}$ such that:

- $s_{\pi}(f_\#) \triangleright t_{\pi}(g_\#)$ for every rule $f_\#(s_1, \ldots, s_n) \rightarrow g_\#(t_1, \ldots, t_m) \in T_1$
- $s_{\pi}(f_\#) = t_{\pi}(g_\#)$ for every rule $f_\#(s_1, \ldots, s_n) \rightarrow g_\#(t_1, \ldots, t_m) \in T_2$

Then:

$$\text{SN}(T_{2,\text{top}}/R) \implies \text{SN}((T_1 \cup T_2)_{\text{top}}/R)$$
Theorem (Subterm Criterion)

Let $R$ be a TRS, $T_1, T_2 \subseteq \text{DP}(R)$, and $\pi : \Sigma \rightarrow \mathbb{N}$ such that:

- $s_{\pi}(f) \triangleright t_{\pi}(g)$ for every rule $f(s_1, \ldots, s_n) \rightarrow g(t_1, \ldots, t_m) \in T_1$
- $s_{\pi}(f) = t_{\pi}(g)$ for every rule $f(s_1, \ldots, s_n) \rightarrow g(t_1, \ldots, t_m) \in T_2$

Then:

$$\text{SN}(T_{2,\text{top}}/R) \implies \text{SN}((T_1 \cup T_2)_{\text{top}}/R)$$

Proof.

After the dependency pairs transformation, we consider only minimal terms.
Subterm Criterion

**Theorem (Subterm Criterion)**

Let $R$ be a TRS, $T_1, T_2 \subseteq \text{DP}(R)$, and $\pi : \Sigma \rightarrow \mathbb{N}$ such that:

- $s_{\pi}(f_{\#}) \triangleright t_{\pi}(g_{\#})$ for every rule $f_{\#}(s_1, \ldots, s_n) \rightarrow g_{\#}(t_1, \ldots, t_m) \in T_1$
- $s_{\pi}(f_{\#}) = t_{\pi}(g_{\#})$ for every rule $f_{\#}(s_1, \ldots, s_n) \rightarrow g_{\#}(t_1, \ldots, t_m) \in T_2$

Then:

$$\text{SN}(T_{2,\text{top}}/R) \implies \text{SN}((T_1 \cup T_2)_{\text{top}}/R)$$

**Proof.**

After the dependency pairs transformation, we consider only minimal terms. We can only finitely often make a terminating term smaller ($\triangleright$). 

\[\square\]
Example: Ackermann function

Example

\[
\text{Ack}(0, y) \rightarrow s(y)
\]
\[
\text{Ack}(s(x), 0) \rightarrow \text{Ack}(x, s(0))
\]
\[
\text{Ack}(s(x), s(y)) \rightarrow \text{Ack}(x, \text{Ack}(s(x), y))
\]
Subterm Criterion

Example: Ackermann function

Example

\[
\begin{align*}
\text{Ack}(0, y) &\to s(y) \\
\text{Ack}(s(x), 0) &\to \text{Ack}(x, s(0)) \\
\text{Ack}(s(x), s(y)) &\to \text{Ack}(x, \text{Ack}(s(x), y))
\end{align*}
\]

\[DP(R) = \{ \text{Ack}_\#(0, y) \to s_\#(y) \]
\[
\text{Ack}_\#(s(x), 0) \to \text{Ack}_\#(x, s(0)) \\
\text{Ack}_\#(s(x), 0) \to s_\#(0) \\
\text{Ack}_\#(s(x), 0) \to 0_\#
\]
\[
\text{Ack}_\#(s(x), s(y)) \to \text{Ack}_\#(x, \text{Ack}(s(x), y)) \\
\text{Ack}_\#(s(x), s(y)) \to \text{Ack}_\#(s(x), y) \\
\text{Ack}_\#(s(x), s(y)) \to s_\#(x) \}
\]
Example: Ackermann function

Example

\[
\begin{align*}
\text{Ack}(0, y) & \rightarrow s(y) \\
\text{Ack}(s(x), 0) & \rightarrow \text{Ack}(x, s(0)) \\
\text{Ack}(s(x), s(y)) & \rightarrow \text{Ack}(x, \text{Ack}(s(x), y))
\end{align*}
\]

\[
\text{DP}(R) = \{ \text{Ack}_#(0, y) \rightarrow s_#(y) \}
\]

\[
\begin{align*}
\text{Ack}_#(s(x), 0) & \rightarrow \text{Ack}_#(x, s(0)) \\
\text{Ack}_#(s(x), 0) & \rightarrow s_#(0) \\
\text{Ack}_#(s(x), 0) & \rightarrow 0_# \\
\text{Ack}_#(s(x), s(y)) & \rightarrow \text{Ack}_#(x, \text{Ack}(s(x), y)) \\
\text{Ack}_#(s(x), s(y)) & \rightarrow \text{Ack}_#(s(x), y) \\
\text{Ack}_#(s(x), s(y)) & \rightarrow s_#(x)
\end{align*}
\]

We use the interpretation:

\[
\begin{align*}
[\text{Ack}_#](x, y) = 1 & \quad [\text{Ack}](x, y) = 0 & \quad [s](x) = 0 & \quad [0] = 0 & \quad s_#(x) = 0
\end{align*}
\]
Example: Ackermann function

Example

\[
\begin{align*}
\text{Ack}(0, y) & \rightarrow s(y) \\
\text{Ack}(s(x), 0) & \rightarrow \text{Ack}(x, \text{s}(0)) \\
\text{Ack}(s(x), s(y)) & \rightarrow \text{Ack}(x, \text{Ack}(s(x), y))
\end{align*}
\]

\[
\text{DP}(R) = \{ \begin{align*}
\text{Ack}_#(0, y) & \rightarrow s_#(y) \\
\text{Ack}_#(s(x), 0) & \rightarrow \text{Ack}_#(x, \text{s}(0)) \\
\text{Ack}_#(s(x), 0) & \rightarrow s_#(0) \\
\text{Ack}_#(s(x), 0) & \rightarrow 0_# \\
\text{Ack}_#(s(x), s(y)) & \rightarrow \text{Ack}_#(x, \text{Ack}(s(x), y)) \\
\text{Ack}_#(s(x), s(y)) & \rightarrow \text{Ack}_#(s(x), y) \\
\text{Ack}_#(s(x), s(y)) & \rightarrow s_#(x) \}
\]

We use the interpretation:

\[
\begin{align*}
[\text{Ack}_#](x, y) & = 1 \\
[\text{Ack}](x, y) & = 0 \\
[s](x) & = 0 \\
[0] & = 0 \\
[s_#](x) & = 0
\end{align*}
\]
Example: Ackermann function

Example

\[
\begin{align*}
\text{Ack}(0, y) & \to s(y) \\
\text{Ack}(s(x), 0) & \to \text{Ack}(x, s(0)) \\
\text{Ack}(s(x), s(y)) & \to \text{Ack}(x, \text{Ack}(s(x), y)) \\
\end{align*}
\]

\[
\text{DP}(R) = \{ \text{Ack}_\#(s(x), 0) \to \text{Ack}_\#(x, s(0)) \\
\text{Ack}_\#(s(x), s(y)) \to \text{Ack}_\#(x, \text{Ack}(s(x), y)) \\
\text{Ack}_\#(s(x), s(y)) \to \text{Ack}_\#(s(x), y) \} 
\]

We use the subterm criterion:

\[
\pi(\text{Ack}_\#) = ?
\]
Example: Ackermann function

Example

\[
\begin{align*}
&\text{Ack}(0, y) \rightarrow s(y) \\
&\text{Ack}(s(x), 0) \rightarrow \text{Ack}(x, s(0)) \\
&\text{Ack}(s(x), s(y)) \rightarrow \text{Ack}(x, \text{Ack}(s(x), y))
\end{align*}
\]

\[
\text{DP}(R) = \{ \text{Ack}_#(s(x), 0) \rightarrow \text{Ack}_#(x, s(0)) \\
\quad \text{Ack}_#(s(x), s(y)) \rightarrow \text{Ack}_#(x, \text{Ack}(s(x), y)) \\
\quad \text{Ack}_#(s(x), s(y)) \rightarrow \text{Ack}_#(s(x), y) \}
\]

We use the subterm criterion:

\[
\pi(\text{Ack}_#) = 1
\]
Subterm Criterion

Example: Ackermann function

Example

\[
\begin{align*}
\text{Ack}(0, y) & \rightarrow s(y) \\
\text{Ack}(s(x), 0) & \rightarrow \text{Ack}(x, s(0)) \\
\text{Ack}(s(x), s(y)) & \rightarrow \text{Ack}(x, \text{Ack}(s(x), y))
\end{align*}
\]

\[
\text{DP}(R) = \{ \text{Ack}#(s(x), 0) \rightarrow \text{Ack}#(x, s(0)), \\
\text{Ack}#(s(x), s(y)) \rightarrow \text{Ack}#(x, \text{Ack}(s(x), y)) \\
\text{Ack}#(s(x), s(y)) \rightarrow \text{Ack}#(s(x), y) \}\n\]

We use the subterm criterion:

\[
\pi(\text{Ack}#) = 1
\]

We can remove the first two DP-rules.
Example: Ackermann function

\[
\begin{align*}
\text{Ack}(0, y) & \rightarrow s(y) \\
\text{Ack}(s(x), 0) & \rightarrow \text{Ack}(x, s(0)) \\
\text{Ack}(s(x), s(y)) & \rightarrow \text{Ack}(x, \text{Ack}(s(x), y)) \\
\end{align*}
\]

\[
\text{DP}(R) = \{ \text{Ack}_\#(s(x), s(y)) \rightarrow \text{Ack}_\#(s(x), y) \}
\]

We use the subterm criterion:

\[
\pi(\text{Ack}_\#) = ?
\]
Example: Ackermann function

<table>
<thead>
<tr>
<th>Example</th>
<th>Ackermann Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ack(0, y) → s(y)</td>
<td></td>
</tr>
<tr>
<td>Ack(s(x), 0) → Ack(x, s(0))</td>
<td></td>
</tr>
<tr>
<td>Ack(s(x), s(y)) → Ack(x, Ack(s(x), y))</td>
<td></td>
</tr>
<tr>
<td>DP(R) = { Ack(#)(s(x), s(y)) → Ack(#)(s(x), y) }</td>
<td></td>
</tr>
</tbody>
</table>

We use the subterm criterion:

\[ \pi(\text{Ack#}) = 2 \]
Example: Ackermann function

Ackermann function

\[\text{Ack}(0, y) \rightarrow s(y)\]
\[\text{Ack}(s(x), 0) \rightarrow \text{Ack}(x, s(0))\]
\[\text{Ack}(s(x), s(y)) \rightarrow \text{Ack}(x, \text{Ack}(s(x), y))\]

\[\text{DP}(R) = \{ \text{Ack}\#(s(x), s(y)) \rightarrow \text{Ack}\#(s(x), y) \} \]

We use the subterm criterion:

\[\pi(\text{Ack}\#) = 2\]

We can remove the remaining DP-rule.
Example: Ackermann function

\begin{align*}
\text{Ack}(0, y) & \rightarrow s(y) \\
\text{Ack}(s(x), 0) & \rightarrow \text{Ack}(x, s(0)) \\
\text{Ack}(s(x), s(y)) & \rightarrow \text{Ack}(x, \text{Ack}(s(x), y))
\end{align*}

\[ \text{DP}(R) = \{ \text{Ack}_{\#}(s(x), s(y)) \rightarrow \text{Ack}_{\#}(s(x), y) \} \]

We use the subterm criterion:

\[ \pi(\text{Ack}_{\#}) = 2 \]

We can remove the remaining DP-rule.

Hence we have proven termination.
Iterative Lexicographic Path Order (ILPO)
Iterative Lexicographic Path Order

ILPO... Historical overview

Kamin and Lévy [1980] (lexicographic path order, LPO):

- Kruskal’s Tree Theorem was used in the original proofs
- Buchholz [1995] simplified the proof: Kruskal not needed

Bergsta and Klop [1985]:

- Iterative version of RPO: ‘star method’
- Operational definition of reduction order via an auxiliary term rewriting system (with stars)

Klop, van Oostrom and de Vrijer [2005]:

- Extension of the star method to LPO
- Iterative lexicographic path order (ILPO)
Given a terminating relation $\succ$ on signature $\Sigma$, define TRS $\mathcal{Lex}_\succ$.
Given a terminating relation $\succ$ on signature $\Sigma$, define TRS $\mathcal{L}ex_{\succ}$

- **Signature:** $\Sigma \cup \Sigma^*$, where $\Sigma^* = \{f^* \mid f \in \Sigma\}$
  - $f^*$ is fresh and has the same arity as $f$
Given a terminating relation $\succ$ on signature $\Sigma$, define TRS $\mathcal{Lex}_\succ$

- **Signature:** $\Sigma \cup \Sigma^*$, where $\Sigma^* = \{f^* \mid f \in \Sigma\}$
  - $f^*$ is fresh and has the same arity as $f$
- **Reduction rules** (four types, for arbitrary $f, g \in \Sigma$):

  $f(\vec{x}) \rightarrow \text{put } f^*(\vec{x})$
  
  $f^*(\vec{x}) \rightarrow \text{select } x_i$
  
  $f^*(\vec{x}) \rightarrow \text{copy } g(f^*(\vec{x}), \ldots, f^*(\vec{x}))$ if $f \succ g$
  
  $f^*(\vec{x}, g(\vec{y}), \vec{z}) \rightarrow \text{lex } f(\vec{x}, g^*(\vec{y}), \vec{l}, \ldots, l)$ where $l = f^*(\vec{x}, g(\vec{y}), \vec{z})$
Given a terminating relation $\succ$ on signature $\Sigma$, define TRS $\mathcal{L}ex_\succ$

- **Signature:** $\Sigma \cup \Sigma^*$, where $\Sigma^* = \{f^* \mid f \in \Sigma\}$
  
  $f^*$ is fresh and has the same arity as $f$

- **Reduction rules** (four types, for arbitrary $f, g \in \Sigma$):

  $f(\vec{x}) \rightarrow_{\text{put}} f^*(\vec{x})$

  $f^*(\vec{x}) \rightarrow_{\text{select}} x_i$

  $g(\vec{y}) \rightarrow_{\text{copy}} g(f^*(\vec{x}), \ldots, f^*(\vec{x})) \text{ if } f \succ g$

  $f^*(\vec{x}, g(\vec{y}), \vec{z}) \rightarrow_{\text{lex}} f(\vec{x}, g^*(\vec{y}), l, \ldots, l)$ where $l = f^*(\vec{x}, g(\vec{y}), \vec{z})$
Given a terminating relation $\succ$ on signature $\Sigma$, define TRS $\mathcal{Lex}_\succ$

- **Signature**: $\Sigma \cup \Sigma^*$, where $\Sigma^* = \{ f^* \mid f \in \Sigma \}$
  - $f^*$ is fresh and has the same arity as $f$
- **Reduction rules** (four types, for arbitrary $f, g \in \Sigma$):
  - $f(\vec{x}) \rightarrow_{\text{put}} f^*(\vec{x})$
  - $f^*(\vec{x}) \rightarrow_{\text{select}} x_i$
  - $g \succ f^*(\vec{x})$, where $l = f^*(\vec{y}), \vec{z}$
ILPO... The star TRS $\mathcal{L}ex_{\succ}$

Given a terminating relation $\succ$ on signature $\Sigma$, define TRS $\mathcal{L}ex_{\succ}$

- **Signature:** $\Sigma \cup \Sigma^*$, where $\Sigma^* = \{ f^* \mid f \in \Sigma \}$
  
  $f^*$ is fresh and has the same arity as $f$

- **Reduction rules** (four types, for arbitrary $f, g \in \Sigma$):
  
  \[
  f(\vec{x}) \rightarrow_{\text{put}} f^*(\vec{x})
  \]

  \[
  f^*(\vec{x}) \rightarrow_{\text{select}} x_i
  \]

  \[
  f^*(\vec{x}) \rightarrow_{\text{copy}} g(f^*(\vec{x}), \ldots, f^*(\vec{x})) \quad \text{if } f \succ g
  \]

  \[
  f^*(\vec{x}) \rightarrow g(\vec{\bar{y}}), \bar{z}
  \]

  where $l = \bar{l} \Rightarrow g(\vec{\bar{y}}), \bar{z}$
Given a terminating relation $\succ$ on signature $\Sigma$, define TRS $\text{Lex}_{\succ}$

- **Signature:** $\Sigma \cup \Sigma^*$, where $\Sigma^* = \{ f^* \mid f \in \Sigma \}$
  
  $f^*$ is fresh and has the same arity as $f$

- **Reduction rules** (four types, for arbitrary $f, g \in \Sigma$):

  - $f(\vec{x}) \rightarrow_{\text{put}} f^*(\vec{x})$
  - $f^*(\vec{x}) \rightarrow_{\text{select}} x_i$
  - $f^*(\vec{x}) \rightarrow_{\text{copy}} g(f^*(\vec{x}), \ldots, f^*(\vec{x}))$ if $f \succ g$
  - $f^*(\vec{x}, g(\vec{y}), \vec{z}) \rightarrow_{\text{lex}} f(\vec{x}, g^*(\vec{y}), l, \ldots, l)$ where $l = f^*(\vec{x}, g(\vec{y}), \vec{z})$
Iterative Lexicographic Path Order

**ILPO... The star TRS \( \mathcal{L}_{\text{Lex}} \)**

Given a terminating relation \( \succ \) on signature \( \Sigma \), define TRS \( \mathcal{L}_{\text{Lex}} \)

- **Signature**: \( \Sigma \cup \Sigma^* \), where \( \Sigma^* = \{ f^* \mid f \in \Sigma \} \)
  
  \( f^* \) is fresh and has the same arity as \( f \)

- **Reduction rules** (four types, for arbitrary \( f, g \in \Sigma \)):
  
  \[
  f(x) \rightarrow_{\text{put}} f^*(x) \]
  
  \[
  f^*(x) \rightarrow_{\text{select}} x_i \]
  
  \[
  f^*(x) \rightarrow_{\text{copy}} g(f^*(x), \ldots, f^*(x)) \quad \text{if } f \succ g \]
  
  \[
  f^*(x, g(y), z) \rightarrow_{\text{lex}} f(x, g^*(y), l, \ldots, l) \quad \text{where } l = f^*(x, g(y), z) \]

**Definition (ILPO)**

\( \succ_{\text{ilpo}} \) is the restriction of \( \rightarrow^+_{\mathcal{L}_{\text{Lex}}} \) to terms over \( \Sigma \), i.e.

\[
t \succ_{\text{ilpo}} s \iff t \rightarrow^+_{\mathcal{L}_{\text{Lex}}} s \quad \text{and} \quad t, s \in T(\Sigma \cup V)
\]
Claim

≻_{ilpo} is a reduction order, that is:

- ≻_{ilpo} is well-founded, and
- closed under substitution and contexts

The closure under substitutions and contexts is immediate since →_{Lex} ≻_{ilpo} is. Only required: proof of termination of ≻_{ilpo}.

Corollary

A TRS \( R \) is terminating if \( R \subseteq ≻_{ilpo} \).

Proof.

≻_{ilpo} is a reduction order with \( R \subseteq ≻_{ilpo} \).
Claim

\(\succ_{ilpo}\) is a reduction order, that is:

- \(\succ_{ilpo}\) is well-founded, and
- closed under substitution and contexts

The closure under substitutions and contexts is immediate since \(\to^{+\text{Lex}}\) is.
Claim

$\succ_{ilpo}$ is a reduction order, that is:

- $\succ_{ilpo}$ is well-founded, and
- closed under substitution and contexts

The closure under substitutions and contexts is immediate since $\rightarrow_{\text{Lex},\succ}$ is.

Only required: proof of termination of $\succ_{ilpo}$. 
Claim

\(\succeq_{ilpo}\) is a reduction order, that is:

- \(\succeq_{ilpo}\) is well-founded, and
- closed under substitution and contexts

The closure under substitutions and contexts is immediate since \(\rightarrow^+_{\text{Lex}_{\succeq}}\) is.

Only required: proof of termination of \(\succeq_{ilpo}\).

Corollary

A TRS \(\mathcal{R}\) is terminating if \(\mathcal{R} \subseteq \succeq_{ilpo}\).
Claim

\(\succ_{ilpo}\) is a reduction order, that is:

- \(\succ_{ilpo}\) is well-founded, and
- closed under substitution and contexts

The closure under substitutions and contexts is immediate since \(\rightarrow^{+}_{\mathcal{Lex}}\) is.

Only required: proof of termination of \(\succ_{ilpo}\).

Corollary

A TRS \(\mathcal{R}\) is terminating if \(\mathcal{R} \subseteq \succ_{ilpo}\).

Proof. \(\succ_{ilpo}\) is a reduction order with \(\mathcal{R} \subseteq \succ_{ilpo}\).
Example, Addition and multiplication

\[\begin{align*}
A(x, 0) & \rightarrow x \\
A(x, S(y)) & \rightarrow S(A(x, y)) \\
M(x, 0) & \rightarrow 0 \\
M(x, S(y)) & \rightarrow A(x, M(x, y))
\end{align*}\]
Iterative Lexicographic Path Order

Example, Addition and multiplication

\[
\begin{align*}
A(x, 0) & \rightarrow x \\
A(x, S(y)) & \rightarrow S(A(x, y)) \\
M(x, 0) & \rightarrow 0 \\
M(x, S(y)) & \rightarrow A(x, M(x, y))
\end{align*}
\]

Use relation \( R \) given by \( M \succ A \) and \( A \succ S \).
Example, Addition and multiplication

\[
\begin{align*}
A(x, 0) & \rightarrow x \\
A(x, S(y)) & \rightarrow S(A(x, y)) \\
M(x, 0) & \rightarrow 0 \\
M(x, S(y)) & \rightarrow A(x, M(x, y))
\end{align*}
\]

Use relation \( R \) given by \( M \succ A \) and \( A \succ S \).

For each reduction rule a corresponding Lex-reductions:
Example, Addition and multiplication

\[
\begin{align*}
A(x, 0) & \rightarrow x \\
A(x, S(y)) & \rightarrow S(A(x, y)) \\
M(x, 0) & \rightarrow 0 \\
M(x, S(y)) & \rightarrow A(x, M(x, y))
\end{align*}
\]

Use relation \( R \) given by \( M \gg A \) and \( A \gg S \).

For each reduction rule a corresponding Lex-reductions:

\( A(x, 0) \)
Example, Addition and multiplication

<table>
<thead>
<tr>
<th>Rule</th>
<th>Transformation</th>
</tr>
</thead>
<tbody>
<tr>
<td>$A(x, 0)$</td>
<td>$\rightarrow x$</td>
</tr>
<tr>
<td>$A(x, S(y))$</td>
<td>$\rightarrow S(A(x, y))$</td>
</tr>
<tr>
<td>$M(x, 0)$</td>
<td>$\rightarrow 0$</td>
</tr>
<tr>
<td>$M(x, S(y))$</td>
<td>$\rightarrow A(x, M(x, y))$</td>
</tr>
</tbody>
</table>

Use relation $R$ given by $M \succ A$ and $A \succ S$.

For each reduction rule a corresponding Lex-reductions:

$A(x, 0) \rightarrow_{\text{put}} A^*(x, 0)$
Example, Addition and multiplication

\[
\begin{align*}
A(x, 0) & \rightarrow x \\
A(x, S(y)) & \rightarrow S(A(x, y)) \\
M(x, 0) & \rightarrow 0 \\
M(x, S(y)) & \rightarrow A(x, M(x, y))
\end{align*}
\]

Use relation \( R \) given by \( M \succ A \) and \( A \succ S \).

For each reduction rule a corresponding Lex-reductions:

\[
A(x, 0) \rightarrow_{\text{put}} A^*(x, 0) \rightarrow_{\text{select}} x
\]
Example, Addition and multiplication

\[
\begin{align*}
A(x, 0) & \rightarrow x \\
A(x, S(y)) & \rightarrow S(A(x, y)) \\
M(x, 0) & \rightarrow 0 \\
M(x, S(y)) & \rightarrow A(x, M(x, y))
\end{align*}
\]

Use relation \( R \) given by \( M \succ A \) and \( A \succ S \).

For each reduction rule a corresponding Lex-reductions:

\[
\begin{align*}
A(x, 0) & \rightarrow_{\text{put}} A^*(x, 0) \rightarrow_{\text{select}} x \\
A(x, S(y)) &
\end{align*}
\]
Example, Addition and multiplication

\[
\begin{align*}
A(x, 0) & \rightarrow x \\
A(x, S(y)) & \rightarrow S(A(x, y)) \\
M(x, 0) & \rightarrow 0 \\
M(x, S(y)) & \rightarrow A(x, M(x, y))
\end{align*}
\]

Use relation \( R \) given by \( M \succ A \) and \( A \succ S \).

For each reduction rule a corresponding Lex-reductions:

\[
\begin{align*}
A(x, 0) & \rightarrow_{\text{put}} A^*(x, 0) \rightarrow_{\text{select}} x \\
A(x, S(y)) & \rightarrow_{\text{put}} A^*(x, S(y))
\end{align*}
\]
Example, Addition and multiplication

\[
\begin{align*}
A(x, 0) &\rightarrow x \\
A(x, S(y)) &\rightarrow S(A(x, y)) \\
M(x, 0) &\rightarrow 0 \\
M(x, S(y)) &\rightarrow A(x, M(x, y))
\end{align*}
\]

Use relation \( R \) given by \( M \succ A \) and \( A \succ S \).

For each reduction rule a corresponding Lex-reductions:

\[
\begin{align*}
A(x, 0) &\rightarrow_{\text{put}} A^*(x, 0) \rightarrow_{\text{select}} x \\
A(x, S(y)) &\rightarrow_{\text{put}} A^*(x, S(y)) \rightarrow_{\text{copy}} S(A^*(x, S(y)))
\end{align*}
\]
Example, Addition and multiplication

\[
\begin{align*}
A(x, 0) & \rightarrow x \\
A(x, S(y)) & \rightarrow S(A(x, y)) \\
M(x, 0) & \rightarrow 0 \\
M(x, S(y)) & \rightarrow A(x, M(x, y))
\end{align*}
\]

Use relation $R$ given by $M \succ A$ and $A \succ S$.

For each reduction rule a corresponding Lex-reductions:

\[
\begin{align*}
A(x, 0) & \xrightarrow{\text{put}} A^*(x, 0) \xrightarrow{\text{select}} x \\
A(x, S(y)) & \xrightarrow{\text{put}} A^*(x, S(y)) \xrightarrow{\text{copy}} S(A^*(x, S(y))) \\
& \xrightarrow{\text{lex}} S(A(x, S^*(y)))
\end{align*}
\]
Example, Addition and multiplication

\[
\begin{align*}
A(x, 0) & \rightarrow x \\
A(x, S(y)) & \rightarrow S(A(x, y)) \\
M(x, 0) & \rightarrow 0 \\
M(x, S(y)) & \rightarrow A(x, M(x, y))
\end{align*}
\]

Use relation \( R \) given by \( M \succ A \) and \( A \succ S \).

For each reduction rule a corresponding Lex-reductions:

\[
\begin{align*}
A(x, 0) & \rightarrow_{\text{put}} A^*(x, 0) \rightarrow_{\text{select}} x \\
A(x, S(y)) & \rightarrow_{\text{put}} A^*(x, S(y)) \rightarrow_{\text{copy}} S(A^*(x, S(y))) \\
& \rightarrow_{\text{lex}} S(A(x, S^*(y))) \rightarrow_{\text{select}} S(A(x, y))
\end{align*}
\]
Iterative Lexicographic Path Order

Example, Addition and multiplication

\[
\begin{align*}
A(x, 0) & \to x \\
A(x, S(y)) & \to S(A(x, y)) \\
M(x, 0) & \to 0 \\
M(x, S(y)) & \to A(x, M(x, y))
\end{align*}
\]

Use relation \( R \) given by \( M \succ A \) and \( A \succ S \).

For each reduction rule a corresponding Lex-reductions:

\[
\begin{align*}
A(x, 0) & \to_{\text{put}} A^*(x, 0) \to_{\text{select}} x \\
A(x, S(y)) & \to_{\text{put}} A^*(x, S(y)) \to_{\text{copy}} S(A^*(x, S(y))) \\
& \to_{\text{lex}} S(A(x, S^*(y))) \to_{\text{select}} S(A(x, y)) \\
M(x, 0) &
\end{align*}
\]
Example, Addition and multiplication

\[
\begin{array}{l}
A(x,0) \rightarrow x \\
A(x, S(y)) \rightarrow S(A(x, y)) \\
M(x, 0) \rightarrow 0 \\
M(x, S(y)) \rightarrow A(x, M(x, y))
\end{array}
\]

Use relation \( R \) given by \( M \succ A \) and \( A \succ S \).

For each reduction rule a corresponding Lex-reductions:

\[
\begin{align*}
A(x, 0) & \rightarrow_{\text{put}} A^*(x, 0) \rightarrow_{\text{select}} x \\
A(x, S(y)) & \rightarrow_{\text{put}} A^*(x, S(y)) \rightarrow_{\text{copy}} S(A^*(x, S(y))) \\
 & \rightarrow_{\text{lex}} S(A(x, S^*(y))) \rightarrow_{\text{select}} S(A(x, y)) \\
M(x, 0) & \rightarrow_{\text{put}} M^*(x, 0)
\end{align*}
\]
Iterative Lexicographic Path Order

Example, Addition and multiplication

\[
\begin{align*}
A(x, 0) & \rightarrow x \\
A(x, S(y)) & \rightarrow S(A(x, y)) \\
M(x, 0) & \rightarrow 0 \\
M(x, S(y)) & \rightarrow A(x, M(x, y))
\end{align*}
\]

Use relation \( R \) given by \( M \succ A \) and \( A \succ S \).

For each reduction rule a corresponding Lex-reductions:

\[
\begin{align*}
A(x, 0) & \rightarrow_{\text{put}} A^*(x, 0) \rightarrow_{\text{select}} x \\
A(x, S(y)) & \rightarrow_{\text{put}} A^*(x, S(y)) \rightarrow_{\text{copy}} S(A^*(x, S(y))) \\
& \quad \rightarrow_{\text{lex}} S(A(x, S^*(y))) \rightarrow_{\text{select}} S(A(x, y)) \\
M(x, 0) & \rightarrow_{\text{put}} M^*(x, 0) \rightarrow_{\text{select}} 0
\end{align*}
\]
Example, Addition and multiplication

\[
\begin{align*}
A(x, 0) & \rightarrow x \\
A(x, S(y)) & \rightarrow S(A(x, y)) \\
M(x, 0) & \rightarrow 0 \\
M(x, S(y)) & \rightarrow A(x, M(x, y))
\end{align*}
\]

Use relation \( R \) given by \( M \succ A \) and \( A \succ S \).

For each reduction rule a corresponding Lex-reductions:

\[
\begin{align*}
A(x, 0) & \rightarrow_{\text{put}} A^*(x, 0) \rightarrow_{\text{select}} x \\
A(x, S(y)) & \rightarrow_{\text{put}} A^*(x, S(y)) \rightarrow_{\text{copy}} S(A^*(x, S(y))) \\
& \rightarrow_{\text{lex}} S(A(x, S^*(y))) \rightarrow_{\text{select}} S(A(x, y)) \\
M(x, 0) & \rightarrow_{\text{put}} M^*(x, 0) \rightarrow_{\text{select}} 0 \\
M(x, S(y)) &
\end{align*}
\]
Iterative Lexicographic Path Order

Example, Addition and multiplication

\[
\begin{align*}
A(x, 0) & \rightarrow x \\
A(x, S(y)) & \rightarrow S(A(x, y)) \\
M(x, 0) & \rightarrow 0 \\
M(x, S(y)) & \rightarrow A(x, M(x, y))
\end{align*}
\]

Use relation \( R \) given by \( M \succ A \) and \( A \succ S \).

For each reduction rule a corresponding Lex-reductions:

\[
\begin{align*}
A(x, 0) & \rightarrow_{\text{put}} A^*(x, 0) \rightarrow_{\text{select}} x \\
A(x, S(y)) & \rightarrow_{\text{put}} A^*(x, S(y)) \rightarrow_{\text{copy}} S(A^*(x, S(y))) \\
& \quad \rightarrow_{\text{lex}} S(A(x, S^*(y))) \rightarrow_{\text{select}} S(A(x, y)) \\
M(x, 0) & \rightarrow_{\text{put}} M^*(x, 0) \rightarrow_{\text{select}} 0 \\
M(x, S(y)) & \rightarrow_{\text{put}} M^*(x, S(y))
\end{align*}
\]
Example, Addition and multiplication

\[
\begin{align*}
A(x, 0) & \rightarrow x \\
A(x, S(y)) & \rightarrow S(A(x, y)) \\
M(x, 0) & \rightarrow 0 \\
M(x, S(y)) & \rightarrow A(x, M(x, y))
\end{align*}
\]

Use relation $R$ given by $M \succ A$ and $A \succ S$.

For each reduction rule a corresponding Lex-reductions:

\[
\begin{align*}
A(x, 0) & \rightarrow_{\text{put}} \ A^*(x, 0) \rightarrow_{\text{select}} x \\
A(x, S(y)) & \rightarrow_{\text{put}} \ A^*(x, S(y)) \rightarrow_{\text{copy}} S(A^*(x, S(y))) \\
& \quad \rightarrow_{\text{lex}} S(A(x, S^*(y))) \rightarrow_{\text{select}} S(A(x, y)) \\
M(x, 0) & \rightarrow_{\text{put}} \ M^*(x, 0) \rightarrow_{\text{select}} 0 \\
M(x, S(y)) & \rightarrow_{\text{put}} M^*(x, S(y)) \rightarrow_{\text{copy}} A(M^*(x, S(y)), M^*(x, S(y)))
\end{align*}
\]
Example, Addition and multiplication

\[
\begin{align*}
A(x, 0) &\rightarrow x \\
A(x, S(y)) &\rightarrow S(A(x, y)) \\
M(x, 0) &\rightarrow 0 \\
M(x, S(y)) &\rightarrow A(x, M(x, y))
\end{align*}
\]

Use relation \( R \) given by \( M \succ A \) and \( A \succ S \).

For each reduction rule a corresponding Lex-reductions:

\[
\begin{align*}
A(x, 0) &\rightarrow_{\text{put}} A^*(x, 0) \rightarrow_{\text{select}} x \\
A(x, S(y)) &\rightarrow_{\text{put}} A^*(x, S(y)) \rightarrow_{\text{copy}} S(A^*(x, S(y))) \\
&\rightarrow_{\text{lex}} S(A(x, S^*(y))) \rightarrow_{\text{select}} S(A(x, y)) \\
M(x, 0) &\rightarrow_{\text{put}} M^*(x, 0) \rightarrow_{\text{select}} 0 \\
M(x, S(y)) &\rightarrow_{\text{put}} M^*(x, S(y)) \rightarrow_{\text{copy}} A(M^*(x, S(y)), M^*(x, S(y))) \\
&\rightarrow_{\text{select}} A(x, M^*(x, S(y)))
\end{align*}
\]
Example, Addition and multiplication

\[
\begin{align*}
A(x, 0) & \rightarrow x \\
A(x, S(y)) & \rightarrow S(A(x, y)) \\
M(x, 0) & \rightarrow 0 \\
M(x, S(y)) & \rightarrow A(x, M(x, y))
\end{align*}
\]

Use relation \( R \) given by \( M \succ A \) and \( A \succ S \).

For each reduction rule a corresponding Lex-reductions:

\[
\begin{align*}
A(x, 0) & \rightarrow_{\text{put}} A^*(x, 0) \rightarrow_{\text{select}} x \\
A(x, S(y)) & \rightarrow_{\text{put}} A^*(x, S(y)) \rightarrow_{\text{copy}} S(A^*(x, S(y))) \\
& \rightarrow_{\text{lex}} S(A(x, S^*(y))) \rightarrow_{\text{select}} S(A(x, y)) \\
M(x, 0) & \rightarrow_{\text{put}} M^*(x, 0) \rightarrow_{\text{select}} 0 \\
M(x, S(y)) & \rightarrow_{\text{put}} M^*(x, S(y)) \rightarrow_{\text{copy}} A(M^*(x, S(y)), M^*(x, S(y))) \\
& \rightarrow_{\text{select}} A(x, M^*(x, S(y))) \rightarrow_{\text{lex}} A(x, M(x, S^*(y)))
\end{align*}
\]
Example, Addition and multiplication

\[
\begin{align*}
A(x, 0) &\rightarrow x \\
A(x, S(y)) &\rightarrow S(A(x, y)) \\
M(x, 0) &\rightarrow 0 \\
M(x, S(y)) &\rightarrow A(x, M(x, y))
\end{align*}
\]

Use relation \( R \) given by \( M \succ A \) and \( A \succ S \).

For each reduction rule a corresponding Lex-reductions:

\[
\begin{align*}
A(x, 0) &\rightarrow_{\text{put}} A^*(x, 0) \rightarrow_{\text{select}} x \\
A(x, S(y)) &\rightarrow_{\text{put}} A^*(x, S(y)) \rightarrow_{\text{copy}} S(A^*(x, S(y))) \\
&\rightarrow_{\text{lex}} S(A(x, S^*(y))) \rightarrow_{\text{select}} S(A(x, y)) \\
M(x, 0) &\rightarrow_{\text{put}} M^*(x, 0) \rightarrow_{\text{select}} 0 \\
M(x, S(y)) &\rightarrow_{\text{put}} M^*(x, S(y)) \rightarrow_{\text{copy}} A(M^*(x, S(y)), M^*(x, S(y))) \\
&\rightarrow_{\text{select}} A(x, M^*(x, S(y))) \rightarrow_{\text{lex}} A(x, M(x, S^*(y))) \\
&\rightarrow_{\text{select}} A(x, M(x, y))
\end{align*}
\]
But $\mathcal{Lex}$ is in general not terminating, e.g. if $A > S$, then

$$A(x, y) \rightarrow_{\text{put}} A^*(x, y)$$
$$\rightarrow_{\text{copy}} S(A^*(x, y))$$
$$\rightarrow_{\text{copy}} S(S(A^*(x, y)))$$
$$\ldots$$
But \( \mathcal{L}ex \) is in general not terminating, e.g. if \( A > S \), then

\[
A(x, y) \to_{\text{put}} A^*(x, y) \to_{\text{copy}} S(A^*(x, y)) \to_{\text{copy}} S(S(A^*(x, y))) \ldots
\]

Starred symbol \( A^* \) is ‘used’ infinitely often.
Termination for $\succ_{ilpo}$ via termination of $\text{Lex}^\omega$

But $\text{Lex}$ is in general not terminating, e.g. if $A > S$, then

$$A(x, y) \rightarrow_{\text{put}} A^\ast(x, y)$$
$$\rightarrow_{\text{copy}} S(A^\ast(x, y))$$
$$\rightarrow_{\text{copy}} S(S(A^\ast(x, y)))$$

... 

Starred symbol $A^\ast$ is ‘used’ infinitely often.

This is essential in any infinite reduction!
Termination for \(\succ_{ilpo}\) via termination of \(\mathcal{L}ex^\omega\)

But \(\mathcal{L}ex\) is in general not terminating, e.g. if \(A \succ S\), then

\[
A(x, y) \rightarrow_{\text{put}} A^*(x, y) \\
\rightarrow_{\text{copy}} S(A^*(x, y)) \\
\rightarrow_{\text{copy}} S(S(A^*(x, y))) \\
\ldots
\]

Starred symbol \(A^*\) is ‘used’ infinitely often.

This is essential in any infinite reduction!

Idea:

- use numbers instead of stars,
- the numbers fix how often a symbol can be used.
Termination for $\succ_{ilpo}$ via termination of $\text{Lex}^\omega$

But $\text{Lex}$ is in general not terminating, e.g. if $A > S$, then

$$A(x, y) \rightarrow_{\text{put}} A^*(x, y) \rightarrow_{\text{copy}} S(A^*(x, y)) \rightarrow_{\text{copy}} S(S(A^*(x, y))) \ldots$$

Starred symbol $A^*$ is ‘used’ infinitely often.

This is essential in any infinite reduction!

Idea:

- use numbers instead of stars,
- the numbers fix how often a symbol can be used.

$\Rightarrow$ Yields a terminating TRS $\text{Lex}^\omega$. 
Given a terminating relation $\succ$ on signature $\Sigma$, define TRS $\text{Lex}^\omega$

- **Signature:** $\Sigma \uplus \Sigma^\omega$, where $\Sigma^\omega = \{ f^n \mid f \in \Sigma, \; n \in \mathbb{N} \}$
  
  $f^n$ is fresh and has same arity as $f$

- **Reduction rules:**

  $f(\vec{x}) \rightarrow_{\text{put}} f^n(\vec{x})$

  $f^n(\vec{x}) \rightarrow_{\text{select}} x_i$

  $f^{n+1}(\vec{x}) \rightarrow_{\text{copy}} g(f^n(\vec{x}), \ldots, f^n(\vec{x}))$ \quad \text{if } f \succ g

  $f^{n+1}(\vec{x}, g(\vec{y}), \vec{z}) \rightarrow_{\text{lex}} f(\vec{x}, g^n(\vec{y}), l, \ldots, l)$ \quad \text{where } l = f^n(\vec{x}, g(\vec{y}), \vec{z})
Back and forth between $\text{Lex}$ and $\text{Lex}^\omega$

From $\rightarrow \text{Lex}^\omega$ to $\rightarrow \text{Lex}$:

- every reduction can be transformed by replacing $f^n$ by $f^*$. 
Back and forth between $\text{Lex}$ and $\text{Lex}^{\omega}$

From $\rightarrow_{\text{Lex}^{\omega}}$ to $\rightarrow_{\text{Lex}}$:

- every reduction can be transformed by replacing $f^n$ by $f^*$.

From $\rightarrow_{\text{Lex}}$ to $\rightarrow_{\text{Lex}^{\omega}}$:

- every finite reduction can be lifted, in particular
  - every reduction between two starless terms can be lifted.
Back and forth between $\text{Lex}$ and $\text{Lex}^\omega$

From $\rightarrow_{\text{Lex}^\omega}$ to $\rightarrow_{\text{Lex}}$:

- every reduction can be transformed by replacing $f^n$ by $f^\ast$.

From $\rightarrow_{\text{Lex}}$ to $\rightarrow_{\text{Lex}^\omega}$:

- every finite reduction can be lifted, in particular
- every reduction between two starless terms can be lifted.

For example:

$M(x, S(y)) \rightarrow_{\text{put}} M^\ast(x, S(y)) \rightarrow_{\text{copy}} A(M^\ast(x, S(y)), M^\ast(x, S(y)))$

$\rightarrow_{\text{select}} A(x, M^\ast(x, S(y))) \rightarrow_{\text{lex}} A(x, M(x, S^\ast(y)))$

$\rightarrow_{\text{select}} A(x, M(x, y))$

becomes:

$M(x, S(y)) \rightarrow_{\text{put}} M^2(x, S(y)) \rightarrow_{\text{copy}} A(M^1(x, S(y)), M^1(x, S(y)))$

$\rightarrow_{\text{select}} A(x, M^1(x, S(y))) \rightarrow_{\text{lex}} A(x, M(x, S^0(y)))$

$\rightarrow_{\text{select}} A(x, M(x, y))$
Theorem

\[ \rightarrow_{\text{Lex}}^{+} \text{ and } \rightarrow_{\text{Lex}^\omega}^{+} \text{ coincide on } T(\Sigma \cup V) \]
Theorem
\[ \rightarrow^+_{\text{Lex}_\triangleright} \text{ and } \rightarrow^+_{\text{Lex}_\omega} \text{ coincide on } T(\Sigma \uplus V) \]

Note that the infinite reduction

\[
A(x, y) \rightarrow_{\text{put}} A^*(x, y) \\
\rightarrow_{\text{copy}} S(A^*(x, y)) \\
\rightarrow_{\text{copy}} S(S(A^*(x, y))) \\
\ldots
\]

cannot be lifted:

\[
A(x, y) \rightarrow_{\text{put}} A?^*(x, y) \\
\rightarrow_{\text{copy}} S(A?^{−1}(x, y)) \\
\rightarrow_{\text{copy}} S(S(A?^{−2}(x, y))) \\
\ldots
\]
Prove the implication

\[ t_1, \ldots, t_n \text{ are terminating} \implies f^\ell(t_1, \ldots, t_n) \text{ is terminating} \]

by induction on triple \( \langle f, \vec{t}, \ell \rangle \) in ordering \( \langle \succ, (\rightarrow \text{Lex}_\omega)^n, \rangle \).
Prove the implication

$$t_1, \ldots, t_n \text{ are terminating } \implies f^\ell(t_1, \ldots, t_n) \text{ is terminating}$$

by induction on triple $\langle f, \vec{t}, \ell \rangle$ in ordering $\langle \succ, (\rightarrow_{\text{Lex}^\omega})^n, > \rangle$.

- Here $(\rightarrow_{\text{Lex}^\omega})^n$ is the lexicographic order on $n$-tuples
Prove the implication

\[ t_1, \ldots, t_n \text{ are terminating} \implies f^\ell(t_1, \ldots, t_n) \text{ is terminating} \]

by induction on triple \( \langle f, \vec{t}, \ell \rangle \) in ordering \( \langle \succ, (\to_{\text{Lex}^\omega})^n, \rangle \).

- Here \((\to_{\text{Lex}^\omega})^n\) is the lexicographic order on \(n\)-tuples.
- No label \(\ell\) counts as \(\infty\) with \(\infty > n\).
Prove the implication

\[ t_1, \ldots, t_n \text{ are terminating} \implies f^\ell(t_1, \ldots, t_n) \text{ is terminating} \]

by induction on triple \( \langle f, \vec{t}, \ell \rangle \) in ordering \( \langle \succ, (\rightarrow_{\text{Lex}^\omega})^n, \rangle \).

- Here \( (\rightarrow_{\text{Lex}^\omega})^n \) is the lexicographic order on \( n \)-tuples
- No label \( \ell \) counts as \( \infty \) with \( \infty > n \).

In general a term is SN if all one-step reducts are SN.
Prove the implication

\[ t_1, \ldots, t_n \text{ are terminating} \implies f^\ell(t_1, \ldots, t_n) \text{ is terminating} \]

by induction on triple \( \langle f, \vec{t}, \ell \rangle \) in ordering \( \langle \succ, (\rightarrow_{\text{Lex}^\omega})^n, > \rangle \).

- Here \( (\rightarrow_{\text{Lex}^\omega})^n \) is the lexicographic order on \( n \)-tuples
- No label \( \ell \) counts as \( \infty \) with \( \infty > n \).

In general a term is SN if all one-step reducts are SN.

\[ \Rightarrow \text{ We check all one step reducts of } f^\ell(t_1, \ldots, t_n). \]
Prove the implication

\[ t_1, \ldots, t_n \text{ are terminating} \implies f^\ell(t_1, \ldots, t_n) \text{ is terminating} \]

by induction on triple \( \langle f, \vec{t}, \ell \rangle \) in ordering \( \langle \succ, (\to_{\text{Lex}^\omega})^n, \rangle \).
Iterative Lexicographic Path Order

Termination of $\text{Lex}^\omega$ à la Buchholz

Prove the implication

\[ t_1, \ldots, t_n \text{ are terminating } \implies f^\ell(t_1, \ldots, t_n) \text{ is terminating} \]

by induction on triple $\langle f, \vec{t}, \ell \rangle$ in ordering $\langle \succ, (\rightarrow_{\text{Lex}^\omega})^n, \rangle$.

**Case 1.** Internal step $f^\ell(\ldots, t_i, \ldots) \rightarrow f^\ell(\ldots, t'_i, \ldots)$.

The triple decreases in the second component.
Iterative Lexicographic Path Order

Termination of $\text{Lex}^\omega$ à la Buchholz

Prove the implication

\[ t_1, \ldots, t_n \text{ are terminating} \implies f^\ell(t_1, \ldots, t_n) \text{ is terminating} \]

by induction on triple $\langle f, \vec{t}, \ell \rangle$ in ordering $\langle \succ, (\to_{\text{Lex}^\omega})^n, \rangle$.

**Case 1.** Internal step $f^\ell(\ldots, t_i, \ldots) \to f^\ell(\ldots, t'_i, \ldots)$.
The triple decreases in the second component.

**Case 2.** $f(t_1, \ldots, t_n) \to_{\text{put}} f^n(t_1, \ldots, t_n)$.
We have a decrease in the third component.
Prove the implication

\[ t_1, \ldots, t_n \text{ are terminating} \implies f^\ell(t_1, \ldots, t_n) \text{ is terminating} \]

by induction on triple \( \langle f, \vec{t}, \ell \rangle \) in ordering \( \langle \succ, (\to \operatorname{Lex}_\omega)^n, \rangle \).

**Case 1.** Internal step \( f^\ell(\ldots, t_i, \ldots) \to f^\ell(\ldots, t'_i, \ldots) \).

The triple decreases in the second component.

**Case 2.** \( f(t_1, \ldots, t_n) \to_{\text{put}} f^n(t_1, \ldots, t_n) \).

We have a decrease in the third component.

**Case 3.** \( f(t_1, \ldots, t_n) \to_{\text{select}} t_i \).

By assumption \( t_i \) is SN.
Iterative Lexicographic Path Order

Termination of $\text{Lex}^\omega$ à la Buchholz

Prove the implication

$$t_1, \ldots, t_n \text{ are terminating } \implies f^\ell(t_1, \ldots, t_n) \text{ is terminating}$$

by induction on triple $\langle f, \vec{t}, \ell \rangle$ in ordering $\langle \succ, (\rightarrow \text{Lex}^\omega)^n, > \rangle$.

Case 4. $f^{n+1}(\vec{t}) \rightarrow_{\text{copy}} g(f^n(\vec{t}), \ldots, f^n(\vec{t}))$.

By IH the arguments $f^n(\vec{t})$ of $g$ are SN since $n + 1 > n$.

Again by IH the term $g(\ldots)$ itself is SN, since $f \succ g$. 
Prove the implication

\[ t_1, \ldots, t_n \text{ are terminating } \implies f^\ell(t_1, \ldots, t_n) \text{ is terminating} \]

by induction on triple \( \langle f, \vec{t}, \ell \rangle \) in ordering \( \langle \succ, (\to \text{Lex}^\omega)^n, \rangle \).

Case 4. \( f^{n+1}(\vec{t}) \to_{\text{copy}} g(f^n(\vec{t}), \ldots, f^n(\vec{t})) \).
By IH the arguments \( f^n(\vec{t}) \) of \( g \) are SN since \( n + 1 > n \).
Again by IH the term \( g(\ldots) \) itself is SN, since \( f \succ g \).

Case 5. \( f^{n+1}(\vec{t}, g(\vec{s}), \vec{r}) \to_{\text{lex}} f(\vec{t}, g^n(\vec{s}), l, \ldots, l), l = f^n(\vec{t}, g(\vec{y}), \vec{r}) \).
By assumption the arguments \( \vec{t} \) and \( g(\vec{s}) \) are SN.
By IH we get \( l \) is SN.
Since \( g(\vec{s}) \to_{\text{put}} g^n(\vec{s}) \) we get
\begin{itemize}
  \item \( g^n(\vec{s}) \) is SN, and
  \item the triple decreases in the second component.
\end{itemize}
Thus by IH \( f(\vec{t}, g^n(\vec{s}), l, \ldots, l) \) is SN.
Hence we have proven:

**Theorem**

\[ \rightarrow_{\text{Lex}^\omega} \text{ is terminating} \]
Hence we have proven:

**Theorem**

\[ \rightarrow_{\text{Lex}^\omega} \text{ is terminating} \]

**Corollary**

\[ \rightarrow_{\text{Lex}}^+ \text{ is terminating on } T(\Sigma \cup V) \]

**Proof.** \[ \rightarrow_{\text{Lex}}^+ \text{ and } \rightarrow_{\text{Lex}^\omega}^+ \text{ coincide on } T(\Sigma \cup V) \]
Example, Ackermann Function

Example

\[
\begin{align*}
\text{Ack}(0, y) & \rightarrow s(y) \\
\text{Ack}(\text{s}(x), 0) & \rightarrow \text{Ack}(x, s(0)) \\
\text{Ack}(\text{s}(x), \text{s}(y)) & \rightarrow \text{Ack}(x, \text{Ack}(\text{s}(x), y))
\end{align*}
\]

Find an order \( \succ \) on \( \Sigma \) which proves termination.
Example, Ackermann Function

Example

\[ \text{Ack}(0, y) \rightarrow s(y) \]
\[ \text{Ack}(s(x), 0) \rightarrow \text{Ack}(x, s(0)) \]
\[ \text{Ack}(s(x), s(y)) \rightarrow \text{Ack}(x, \text{Ack}(s(x), y)) \]

Find an order \( \succ \) on \( \Sigma \) which proves termination.

\[ \text{Ack} \succ s \]
Example, Ackermann Function

Example

\[\text{Ack}(0, y) \rightarrow s(y)\]
\[\text{Ack}(s(x), 0) \rightarrow \text{Ack}(x, s(0))\]
\[\text{Ack}(s(x), s(y)) \rightarrow \text{Ack}(x, \text{Ack}(s(x), y))\]

Find an order \(\succ\) on \(\Sigma\) which proves termination.

\[\text{Ack} \succ s\]

We get the following derivations:

\[\text{Ack}(0, y)\]
Example, Ackermann Function

Example

\[
\begin{align*}
\text{Ack}(0, y) &\rightarrow s(y) \\
\text{Ack}(s(x), 0) &\rightarrow \text{Ack}(x, s(0)) \\
\text{Ack}(s(x), s(y)) &\rightarrow \text{Ack}(x, \text{Ack}(s(x), y))
\end{align*}
\]

Find an order \(\succ\) on \(\Sigma\) which proves termination.

\[\text{Ack} \succ s\]

We get the following derivations:

\[\text{Ack}(0, y) \rightarrow_{\text{put}} \text{Ack}^*(0, y)\]
Example, Ackermann Function

Example

\[
\begin{align*}
\text{Ack}(0, y) & \rightarrow s(y) \\
\text{Ack}(s(x), 0) & \rightarrow \text{Ack}(x, s(0)) \\
\text{Ack}(s(x), s(y)) & \rightarrow \text{Ack}(x, \text{Ack}(s(x), y))
\end{align*}
\]

Find an order \( \succ \) on \( \Sigma \) which proves termination.

\( \text{Ack} \succ s \)

We get the following derivations:

\[
\text{Ack}(0, y) \rightarrow_{\text{put}} \text{Ack}^*(0, y) \rightarrow_{\text{copy}} s(\text{Ack}^*(0, y))
\]
Example, Ackermann Function

\[
\begin{align*}
\text{Ack}(0, y) & \rightarrow s(y) \\
\text{ Ack}(s(x), 0) & \rightarrow \text{ Ack}(x, s(0)) \\
\text{ Ack}(s(x), s(y)) & \rightarrow \text{ Ack}(x, \text{ Ack}(s(x), y))
\end{align*}
\]

Find an order \( \succ \) on \( \Sigma \) which proves termination.

\[
\text{ Ack} \succ s
\]

We get the following derivations:

\[
\text{ Ack}(0, y) \rightarrow_{\text{ put }} \text{ Ack}^*(0, y) \rightarrow_{\text{ copy }} s(\text{ Ack}^*(0, y)) \rightarrow_{\text{ select }} s(y)
\]
Example, Ackermann Function

```
Example

\[
\begin{align*}
  \text{Ack}(0, y) & \to \text{s}(y) \\
  \text{Ack}(\text{s}(x), 0) & \to \text{Ack}(x, \text{s}(0)) \\
  \text{Ack}(\text{s}(x), \text{s}(y)) & \to \text{Ack}(x, \text{Ack}(\text{s}(x), y))
\end{align*}
\]

Find an order \( \succ \) on \( \Sigma \) which proves termination.

\[
\text{Ack} \succ \text{s}
\]

We get the following derivations:

\[
\begin{align*}
  \text{Ack}(0, y) & \xrightarrow{\text{put}} \text{Ack}^*(0, y) \xrightarrow{\text{copy}} \text{s}(\text{Ack}^*(0, y)) \xrightarrow{\text{select}} \text{s}(y) \\
  \text{Ack}(\text{s}(x), 0) & \xrightarrow{} \text{s}(\text{s}(\text{Ack}(\text{s}(x), 0)))
\end{align*}
\]

Hence we have proven termination.
```
Example

\[
\begin{align*}
\text{Ack}(0, y) & \rightarrow s(y) \\
\text{Ack}(s(x), 0) & \rightarrow \text{Ack}(x, s(0)) \\
\text{Ack}(s(x), s(y)) & \rightarrow \text{Ack}(x, \text{Ack}(s(x), y))
\end{align*}
\]

Find an order \( \succ \) on \( \Sigma \) which proves termination.

\[
\text{Ack} \succ s
\]

We get the following derivations:

\[
\begin{align*}
\text{Ack}(0, y) & \rightarrow_{\text{put}} \text{Ack}^*(0, y) \rightarrow_{\text{copy}} s(\text{Ack}^*(0, y)) \rightarrow_{\text{select}} s(y) \\
\text{Ack}(s(x), 0) & \rightarrow_{\text{put}} \text{Ack}^*(s(x), 0)
\end{align*}
\]
Example, Ackermann Function

Example

\[
\begin{align*}
\text{Ack}(0, y) & \to s(y) \\
\text{Ack}(s(x), 0) & \to \text{Ack}(x, s(0)) \\
\text{Ack}(s(x), s(y)) & \to \text{Ack}(x, \text{Ack}(s(x), y))
\end{align*}
\]

Find an order \( \succ \) on \( \Sigma \) which proves termination.

\[\text{Ack} \succ s\]

We get the following derivations:

\[
\begin{align*}
\text{Ack}(0, y) & \to_{\text{put}} \text{Ack}^*(0, y) \to_{\text{copy}} s(\text{Ack}^*(0, y)) \to_{\text{select}} s(y) \\
\text{Ack}(s(x), 0) & \to_{\text{put}} \text{Ack}^*(s(x), 0) \to_{\text{lex}} \text{Ack}(s^*(x), \text{Ack}^*(s(x), 0))
\end{align*}
\]
Example, Ackermann Function

Example

\[ \text{Ack}(0, y) \rightarrow s(y) \]
\[ \text{Ack}(s(x), 0) \rightarrow \text{Ack}(x, \text{s}(0)) \]
\[ \text{Ack}(s(x), s(y)) \rightarrow \text{Ack}(x, \text{Ack}(s(x), y)) \]

Find an order \( \succ \) on \( \Sigma \) which proves termination.

\[ \text{Ack} \succ s \]

We get the following derivations:

\[ \text{Ack}(0, y) \rightarrow_{\text{put}} \text{Ack}^*(0, y) \rightarrow_{\text{copy}} s(\text{Ack}^*(0, y)) \rightarrow_{\text{select}} s(y) \]
\[ \text{Ack}(s(x), 0) \rightarrow_{\text{put}} \text{Ack}^*(s(x), 0) \rightarrow_{\text{lex}} \text{Ack}(s^*(x), \text{Ack}^*(s(x), 0)) \rightarrow_{\text{select}} \text{Ack}(x, \text{Ack}^*(s(x), 0)) \]
Example, Ackermann Function

Example

\[
\begin{align*}
\text{Ack}(0, y) & \rightarrow s(y) \\
\text{Ack}(s(x), 0) & \rightarrow \text{Ack}(x, s(0)) \\
\text{Ack}(s(x), s(y)) & \rightarrow \text{Ack}(x, \text{Ack}(s(x), y))
\end{align*}
\]

Find an order $\succ$ on $\Sigma$ which proves termination.

\[\text{Ack} \succ s\]

We get the following derivations:

\[
\begin{align*}
\text{Ack}(0, y) & \rightarrow_{\text{put}} \text{Ack}^*(0, y) \rightarrow_{\text{copy}} s(\text{Ack}^*(0, y)) \rightarrow_{\text{select}} s(y) \\
\text{Ack}(s(x), 0) & \rightarrow_{\text{put}} \text{Ack}^*(s(x), 0) \rightarrow_{\text{lex}} \text{Ack}(s^*(x), \text{Ack}^*(s(x), 0)) \\
& \rightarrow_{\text{select}} \text{Ack}(x, \text{Ack}^*(s(x), 0)) \rightarrow_{\text{copy}} \text{Ack}(x, s(\text{Ack}^*(s(x), 0)))
\end{align*}
\]
Example, Ackermann Function

Example

\[
\begin{align*}
\text{Ack}(0, y) & \rightarrow s(y) \\
\text{Ack}(s(x), 0) & \rightarrow \text{Ack}(x, s(0)) \\
\text{Ack}(s(x), s(y)) & \rightarrow \text{Ack}(x, \text{Ack}(s(x), y))
\end{align*}
\]

Find an order \( \succ \) on \( \Sigma \) which proves termination.

\[
\text{Ack} \succ s
\]

We get the following derivations:

\[
\begin{align*}
\text{Ack}(0, y) & \rightarrow_{\text{put}} \text{Ack}^*(0, y) \rightarrow_{\text{copy}} s(\text{Ack}^*(0, y)) \rightarrow_{\text{select}} s(y) \\
\text{Ack}(s(x), 0) & \rightarrow_{\text{put}} \text{Ack}^*(s(x), 0) \rightarrow_{\text{lex}} \text{Ack}(s^*(x), \text{Ack}^*(s(x), 0)) \\
& \rightarrow_{\text{select}} \text{Ack}(x, \text{Ack}^*(s(x), 0)) \rightarrow_{\text{copy}} \text{Ack}(x, s(\text{Ack}^*(s(x), 0))) \\
& \rightarrow_{\text{select}} \text{Ack}(x, s(0))
\end{align*}
\]
Example, Ackermann Function

Example

\[
\begin{align*}
\text{Ack}(0, y) & \rightarrow s(y) \\
\text{Ack}(s(x), 0) & \rightarrow \text{Ack}(x, s(0)) \\
\text{Ack}(s(x), s(y)) & \rightarrow \text{Ack}(x, \text{Ack}(s(x), y))
\end{align*}
\]

Find an order $\succ$ on $\Sigma$ which proves termination.

\[
\text{Ack} \succ s
\]

We get the following derivations:

\[
\text{Ack}(s(x), s(y))
\]
Example, Ackermann Function

Example

\[
\begin{align*}
\text{Ack}(0, y) & \to s(y) \\
\text{Ack}(s(x), 0) & \to \text{Ack}(x, s(0)) \\
\text{Ack}(s(x), s(y)) & \to \text{Ack}(x, \text{Ack}(s(x), y))
\end{align*}
\]

Find an order \(\succ\) on \(\Sigma\) which proves termination.

\[\text{Ack} \succ s\]

We get the following derivations:

\[\text{Ack}(s(x), s(y)) \rightarrow_{\text{put}} \text{Ack}^*(s(x), s(y))\]
Example, Ackermann Function

Example

\[
\begin{align*}
\text{Ack}(0, y) & \rightarrow s(y) \\
\text{Ack}(s(x), 0) & \rightarrow \text{Ack}(x, s(0)) \\
\text{Ack}(s(x), s(y)) & \rightarrow \text{Ack}(x, \text{Ack}(s(x), y))
\end{align*}
\]

Find an order \( \succ \) on \( \Sigma \) which proves termination.

\[ \text{Ack} \succ s \]

We get the following derivations:

\[
\text{Ack}(s(x), s(y)) \rightarrow_{\text{put}} \text{Ack}^*(s(x), s(y)) \rightarrow_{\text{lex}} \text{Ack}(s^*(x), \text{Ack}^*(s(x), s(y)))
\]
Example, Ackermann Function

Example

\[
\begin{align*}
\text{Ack}(0, y) & \to \text{s}(y) \\
\text{Ack}(\text{s}(x), 0) & \to \text{Ack}(x, \text{s}(0)) \\
\text{Ack}(\text{s}(x), \text{s}(y)) & \to \text{Ack}(x, \text{Ack}(\text{s}(x), y))
\end{align*}
\]

Find an order \( \succ \) on \( \Sigma \) which proves termination.

\[ \text{Ack} \succ \text{s} \]

We get the following derivations:

\[
\text{Ack}(\text{s}(x), \text{s}(y)) \to_{\text{put}} \text{ Ack}^*(\text{s}(x), \text{s}(y))
\to_{\text{lex}} \text{ Ack}(\text{s}^*(x), \text{ Ack}^*(\text{s}(x), \text{s}(y))))
\to_{\text{select}} \text{ Ack}(x, \text{ Ack}^*(\text{s}(x), \text{s}(y))))
\]
Example, Ackermann Function

Example

\[
\begin{align*}
\text{Ack}(0, y) & \rightarrow s(y) \\
\text{Ack}(s(x), 0) & \rightarrow \text{Ack}(x, s(0)) \\
\text{Ack}(s(x), s(y)) & \rightarrow \text{Ack}(x, \text{Ack}(s(x), y))
\end{align*}
\]

Find an order \( \succ \) on \( \Sigma \) which proves termination.

\[ \text{Ack} \succ s \]

We get the following derivations:

\[
\begin{align*}
\text{Ack}(s(x), s(y)) & \rightarrow_{\text{put}} \text{Ack}^*(s(x), s(y)) \\
& \rightarrow_{\text{lex}} \text{Ack}(s^*(x), \text{Ack}^*(s(x), s(y))) \\
& \rightarrow_{\text{select}} \text{Ack}(x, \text{Ack}^*(s(x), s(y))) \\
& \rightarrow_{\text{lex}} \text{Ack}(x, \text{Ack}(s(x), s^*(y)))
\end{align*}
\]

Hence we have proven termination.
Iterative Lexicographic Path Order

Example, Ackermann Function

Example

\[ \text{Ack}(0, y) \rightarrow s(y) \]
\[ \text{Ack}(s(x), 0) \rightarrow \text{Ack}(x, s(0)) \]
\[ \text{Ack}(s(x), s(y)) \rightarrow \text{Ack}(x, \text{Ack}(s(x), y)) \]

Find an order \( \succ \) on \( \Sigma \) which proves termination.

\[ \text{Ack} \succ s \]

We get the following derivations:

\[ \text{Ack}(s(x), s(y)) \rightarrow_{\text{put}} \text{Ack}^*(s(x), s(y)) \]
\[ \rightarrow_{\text{lex}} \text{Ack}(s^*(x), \text{Ack}^*(s(x), s(y))) \]
\[ \rightarrow_{\text{select}} \text{Ack}(x, \text{Ack}^*(s(x), s(y))) \]
\[ \rightarrow_{\text{lex}} \text{Ack}(x, \text{Ack}(s(x), s^*(y))) \]
\[ \rightarrow_{\text{select}} \text{Ack}(x, \text{Ack}(s(x), y)) \]

Hence we have proven termination.
Iterative Lexicographic Path Order

Recursive definition of LPO

Let $\succ$ be a strict order on signature $\Sigma$

Define $\succ_{lpo}$ on $T(\Sigma, V)$ by: $s \succ_{lpo} t$ iff

(LPO1) $t \in \text{Var}(s)$ and $s \neq t$, or

(LPO2) $s = f(s_1, \ldots, s_m)$, $t = g(t_1, \ldots, t_n)$, and

(LPO2a) $\exists 1 \leq i \leq m$, with $s_i = t$ or $s_i \succ_{lpo} t$, or

(LPO2b) $f \succ g$ and $s \succ_{lpo} t_j$ for all $1 \leq j \leq n$, or

(LPO2c) $f = g$, and

$s \succ_{lpo} t_j$ for all $1 \leq j \leq n$, and there exists $1 \leq i \leq m$, s.t.

$s_1 = t_1$, $\ldots$, $s_{i-1} = t_{i-1}$ and $s_i \succ_{lpo} t_i$. 
Iterative Lexicographic Path Order

Recursive definition of LPO

Let \( \succ \) be a strict order on signature \( \Sigma \)

Define \( \succ_{lpo} \) on \( T(\Sigma, V) \) by: \( s \succ_{lpo} t \) iff

(LPO1) \( t \in \text{Var}(s) \) and \( s \neq t \), or

(LPO2) \( s = f(s_1, \ldots, s_m), t = g(t_1, \ldots, t_n) \), and

   (LPO2a) \( \exists 1 \leq i \leq m, \) with \( s_i = t \) or \( s_i \succ_{lpo} t \), or

   (LPO2b) \( f \succ g \) and \( s \succ_{lpo} t_j \) for all \( 1 \leq j \leq n \), or

   (LPO2c) \( f = g, \) and

   \( s \succ_{lpo} t_j \) for all \( 1 \leq j \leq n \), and

there exists \( 1 \leq i \leq m, \) s.t.

\( s_1 = t_1, \ldots, s_{i-1} = t_{i-1} \) and \( s_i \succ_{lpo} t_i \).

Theorem

\( \succ_{ilpo} \) is equivalent with \( \succ_{lpo} \)