• 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
Strategies
**Definition**

A *rewrite strategy* $S$ is mapping that assigns to every reducible term $t$ a nonempty set of finite nonempty rewrite sequences starting from $t$.
A rewrite strategy $S$ is a mapping that assigns to every reducible term $t$ a nonempty set of finite nonempty rewrite sequences starting from $t$.

- $S$ is **deterministic** if $|S(t)| = 1$ for every reducible term $t$.
A rewrite strategy $S$ is a mapping that assigns to every reducible term $t$ a nonempty set of finite nonempty rewrite sequences starting from $t$.

- $S$ is deterministic if $|S(t)| = 1$ for every reducible term $t$.
- $S$ normalizes term $t$ if there are no infinite $S$ rewrite sequences starting from $t$.
- $S$ is normalizing if it normalizes every term that has a normal form.
- $S$ is perpetual if every maximal $S$ rewrite sequence starting from any non-terminating term is infinite.
A rewrite strategy $S$ is mapping that assigns to every reducible term $t$ a nonempty set of finite nonempty rewrite sequences starting from $t$.

- $S$ is deterministic if $|S(t)| = 1$ for every reducible term $t$
- $S$ normalizes term $t$ if there are no infinite $S$ rewrite sequences starting from $t$
- $S$ is normalizing if it normalizes every term that has a normal form

**Definition**

A rewrite sequence is maximal if it is infinite, or it ends in a normal form.

**Lemma**

For terminating TRSs every strategy is normalizing and perpetual.
A rewrite strategy $S$ is mapping that assigns to every reducible term $t$ a nonempty set of finite nonempty rewrite sequences starting from $t$.

- $S$ is deterministic if $|S(t)| = 1$ for every reducible term $t$
- $S$ normalizes term $t$ if there are no infinite $S$ rewrite sequences starting from $t$
- $S$ is normalizing if it normalizes every term that has a normal form
- $S$ is perpetual if every maximal $S$ rewrite sequence starting from any non-terminating term is infinite

Definition

A rewrite sequence is maximal if it is infinite, or it ends in a normal form.
**Definition**

A rewrite strategy $S$ is mapping that assigns to every reducible term $t$ a nonempty set of finite nonempty rewrite sequences starting from $t$.

- $S$ is **deterministic** if $|S(t)| = 1$ for every reducible term $t$.
- $S$ normalizes term $t$ if there are no infinite $S$ rewrite sequences starting from $t$.
- $S$ is **normalizing** if it normalizes every term that has a normal form.
- $S$ is **perpetual** if every maximal $S$ rewrite sequence starting from any non-terminating term is infinite.

**Definition**

A rewrite sequence is **maximal** if it is infinite, or it ends in a normal form.

**Lemma**

*For terminating TRSs every strategy is normalizing and perpetual.*
Definition

Let $\rho = t_0 \rightarrow t_1 \rightarrow \ldots$ be a finite or infinite rewrite sequence.
Strategies

**Definition**

Let \( \rho = t_0 \rightarrow t_1 \rightarrow \ldots \) be a finite or infinite rewrite sequence.

- Consider a redex occurrence \( s \) is some term \( t_n \) of \( \rho \).

Definition

A strategy \( S \) is fair if every maximal \( S \) rewrite sequence is fair.
Definition

Let $\rho = t_0 \rightarrow t_1 \rightarrow \ldots$ be a finite or infinite rewrite sequence.

- Consider a redex occurrence $s$ is some term $t_n$ of $\rho$.

  Then $s$ is secured if eventually there are no residuals of $s$ left.
Definition

Let $\rho = t_0 \rightarrow t_1 \rightarrow \ldots$ be a finite or infinite rewrite sequence.

- Consider a redex occurrence $s$ is some term $t_n$ of $\rho$.
  
  Then $s$ is secured if eventually there are no residuals of $s$ left.
  
  That is, there exits $m > n$ such that $t_m$ contains no residuals of $s$. 

Definition

A strategy $S$ is fair if every maximal $S$ rewrite sequence is fair.
Strategies

Let $\rho = t_0 \rightarrow t_1 \rightarrow \ldots$ be a finite or infinite rewrite sequence.

- Consider a redex occurrence $s$ is some term $t_n$ of $\rho$.
  
  Then $s$ is secured if eventually there are no residuals of $s$ left.
  
  That is, there exits $m > n$ such that $t_m$ contains no residuals of $s$.

- The reduction $\rho$ is fair if every redex occurring in $\rho$ is secured.

Definition

A strategy $S$ is fair if every maximal $S$ rewrite sequence is fair.
Strategies

**Definition**

Let $\rho = t_0 \rightarrow t_1 \rightarrow \ldots$ be a finite or infinite rewrite sequence.

- Consider a redex occurrence $s$ is some term $t_n$ of $\rho$.
  
  Then $s$ is **secured** if eventually there are no residuals of $s$ left.
  
  That is, there exits $m > n$ such that $t_m$ contains no residuals of $s$.

- The reduction $\rho$ is **fair** if every redex occurring in $\rho$ is secured.

**Definition**

A strategy $S$ is **fair** if every every maximal $S$ rewrite sequence is fair.
Definition
A one-step strategy maps every reducible term to a set of one-step reductions.

Example
There exists no fair one-step strategy for $R = \{I(x) \to I(x)\}$.

For the term:
\[ t = I(I(x)) \]
there are only 3 possible mappings:
• $S(t) = \{I(I(x)) \to \varepsilon, I(I(x)) \to 1\}$
• $S(t) = \{I(I(x)) \to \varepsilon, I(I(x)) \to I(I(x)) \to 1\}$
• $S(t) = \{I(I(x)) \to \varepsilon, I(I(x)) \to 1\}$.
None of these is fair as we can always continue to reduce the same occurrence of $I$.
Definition

A **one-step** strategy maps every reducible term to a set of one-step reductions.

Example

There exists no fair one-step strategy for $\mathcal{R} = \{I(x) \rightarrow I(x)\}$.

For the term:

$$t = I(I(x))$$

there are only 3 possible mappings:
**Definition**

A **one-step** strategy maps every reducible term to a set of one-step reductions.

**Example**

There exists no fair one-step strategy for \( \mathcal{R} = \{ l(x) \rightarrow l(x) \} \).

For the term:

\[
t = l(l(x))
\]

there are only 3 possible mappings:

- \( S(t) = \{ l(l(x)) \rightarrow_\varepsilon l(l(x)) \} \),
**Definition**

A **one-step** strategy maps every reducible term to a set of one-step reductions.

**Example**

There exists no fair one-step strategy for $R = \{l(x) \rightarrow l(x)\}$.

For the term:

$$t = l(l(x))$$

there are only 3 possible mappings:

- $S(t) = \{l(l(x)) \rightarrow_\epsilon l(l(x))\}$,
- $S(t) = \{l(l(x)) \rightarrow_1 l(l(x))\}$, or
Definition

A one-step strategy maps every reducible term to a set of one-step reductions.

Example

There exists no fair one-step strategy for \( R = \{ l(x) \rightarrow l(x) \} \).

For the term:

\[
 t = l(l(x))
\]

there are only 3 possible mappings:

- \( S(t) = \{ l(l(x)) \rightarrow_\varepsilon l(l(x)) \} \),
- \( S(t) = \{ l(l(x)) \rightarrow_1 l(l(x)) \} \), or
- \( S(t) = \{ l(l(x)) \rightarrow_\varepsilon l(l(x)), l(l(x)) \rightarrow_1 l(l(x)) \} \).
Definition

A one-step strategy maps every reducible term to a set of one-step reductions.

Example

There exists no fair one-step strategy for \( R = \{ I(x) \to I(x) \} \).

For the term:

\[ t = I(I(x)) \]

there are only 3 possible mappings:

- \( S(t) = \{ I(I(x)) \to_\varepsilon I(I(x)) \} \),
- \( S(t) = \{ I(I(x)) \to_1 I(I(x)) \} \), or
- \( S(t) = \{ I(I(x)) \to_\varepsilon I(I(x)), I(I(x)) \to_1 I(I(x)) \} \).

None of these is fair as we can always continue to reduce the same occurrence of \( I \).
Example

- **rewrite rules**

\[
\begin{align*}
0 + 0 & \rightarrow 0 & 1 + 0 & \rightarrow 1 & \cdots & 9 + 0 & \rightarrow 9 \\
0 + 1 & \rightarrow 1 & 1 + 1 & \rightarrow 2 & \cdots & 9 + 1 & \rightarrow 1 : 0 \\
0 + 2 & \rightarrow 2 & 1 + 2 & \rightarrow 3 & \cdots & 9 + 2 & \rightarrow 1 : 1 \\
0 + 3 & \rightarrow 3 & 1 + 3 & \rightarrow 4 & \cdots & 9 + 3 & \rightarrow 1 : 2 \\
0 + 4 & \rightarrow 4 & 1 + 4 & \rightarrow 5 & \cdots & 9 + 4 & \rightarrow 1 : 3 \\
0 + 5 & \rightarrow 5 & 1 + 5 & \rightarrow 6 & \cdots & 9 + 5 & \rightarrow 1 : 4 \\
0 + 6 & \rightarrow 6 & 1 + 6 & \rightarrow 7 & \cdots & 9 + 6 & \rightarrow 1 : 5 \\
0 + 7 & \rightarrow 7 & 1 + 7 & \rightarrow 8 & \cdots & 9 + 7 & \rightarrow 1 : 6 \\
0 + 8 & \rightarrow 8 & 1 + 8 & \rightarrow 9 & \cdots & 9 + 8 & \rightarrow 1 : 7 \\
0 + 9 & \rightarrow 9 & 1 + 9 & \rightarrow 1 : 0 & \cdots & 9 + 9 & \rightarrow 1 : 8 \\
x + (y : z) & \rightarrow y : (x + z) & 0 : x & \rightarrow x \\
(x : y) + z & \rightarrow x : (y + z) & x : (y : z) & \rightarrow (x + y) : z
\end{align*}
\]

- **term**

\[((0 : (1 + 2)) + (3 + 4)) + (5 + 6)\]
Example (cont’d)

term

\[ 0 : 1 + 2 + 3 + 4 + 5 + 6 \]

tree representation
Example (cont’d)

```
0 : 1 + 2 + 3 + 4 + 5 + 6
```

tree representation

```
0 + 1 + 2 + 3 + 4 + 5 + 6
```

outermost redexes
Example (cont’d)

term

\[ 0 : \boxed{1 + 2} + \boxed{3 + 4} + \boxed{5 + 6} \]

tree representation

innermost redexes
Example (cont’d)

term

\[ 0 : (1 + 2) + (3 + 4) + (5 + 6) \]

tree representation

leftmost outermost strategy
Example (cont’d)

term

\[
0 : \boxed{1 + 2} + \boxed{3 + 4} + \boxed{5 + 6}
\]

tree representation

leftmost innermost strategy
Example (cont’d)

term

\[
0 : [1 + 2] + [3 + 4] + [5 + 6]
\]

tree representation

parallel outermost strategy
Example (cont’d)

Term

\[ 0 : (1 + 2) + (3 + 4) + (5 + 6) \]

Tree representation

parallel innermost strategy
Definition

Leftmost outermost strategy always reduces the leftmost or the outermost redexes.
Definition

Leftmost outermost strategy always reduces the leftmost or the outermost redexes.

Example (cont’d)

\((0 : (1 + 2)) + (3 + 4)) + (5 + 6)\)
Definition

**Leftmost outermost** strategy always reduces the leftmost or the outermost redexes.

Example (cont’d)

\[
\frac{(0 : (1 + 2)) + (3 + 4)) + (5 + 6)}{(0 : ((1 + 2) + (3 + 4))) + (5 + 6)}
\]
Definition

Leftmost outermost strategy always reduces the leftmost or the outermost redexes.

Example (cont’d)

\[
((0 : (1 + 2)) + (3 + 4)) + (5 + 6) \rightarrow (0 : ((1 + 2) + (3 + 4))) + (5 + 6)
\]

\[
\rightarrow 0 : (((1 + 2) + (3 + 4)) + (5 + 6))
\]
Definition

Leftmost outermost strategy always reduces the leftmost or the outermost redexes.

Example (cont’d)

\[(0 : (1 + 2)) + (3 + 4)) + (5 + 6) \rightarrow (0 : ((1 + 2) + (3 + 4))) + (5 + 6)\]
\[\rightarrow 0 : (((1 + 2) + (3 + 4)) + (5 + 6))\]
\[\rightarrow ((1 + 2) + (3 + 4)) + (5 + 6)\]
**Definition**

*Leftmost outermost* strategy always reduces the leftmost or the outermost redexes.

**Example (cont’d)**

\[
((0 : (1 + 2)) + (3 + 4)) + (5 + 6) \rightarrow (0 : ((1 + 2) + (3 + 4))) + (5 + 6)
\]
\[
\rightarrow 0 : (((1 + 2) + (3 + 4)) + (5 + 6))
\]
\[
\rightarrow ((1 + 2) + (3 + 4)) + (5 + 6)
\]
\[
\rightarrow (3 + (3 + 4)) + (5 + 6)
\]
\[
\rightarrow 2 : (0 + 1)
\]
\[
\rightarrow 2 : 1
\]
**Definition**

Leftmost outermost strategy always reduces the leftmost or the outermost redexes.

**Example (cont’d)**

\[
((0 : (1 + 2)) + (3 + 4)) + (5 + 6) \rightarrow (0 : ((1 + 2) + (3 + 4))) + (5 + 6)
\]

\[
\rightarrow 0 : (((1 + 2) + (3 + 4)) + (5 + 6))
\]

\[
\rightarrow ((1 + 2) + (3 + 4)) + (5 + 6)
\]

\[
\rightarrow (3 + (3 + 4)) + (5 + 6)
\]

\[
\rightarrow (3 + 7) + (5 + 6)
\]
Definition

Leftmost outermost strategy always reduces the leftmost or the outermost redexes.

Example (cont’d)

\[ ((0 : (1 + 2)) + (3 + 4)) + (5 + 6) \rightarrow (0 : ((1 + 2) + (3 + 4))) + (5 + 6) \]
\[ \rightarrow 0 : (((1 + 2) + (3 + 4)) + (5 + 6)) \]
\[ \rightarrow ((1 + 2) + (3 + 4)) + (5 + 6) \]
\[ \rightarrow (3 + (3 + 4)) + (5 + 6) \]
\[ \rightarrow (3 + 7) + (5 + 6) \]
\[ \rightarrow (1 : 0) + (5 + 6) \]
**Definition**

_leftmost outermost_ strategy always reduces the leftmost or the outermost redexes.

**Example (cont’d)**

\[
((0 : (1 + 2)) + (3 + 4)) + (5 + 6) \rightarrow (0 : ((1 + 2) + (3 + 4))) + (5 + 6) \\
\rightarrow 0 : (((1 + 2) + (3 + 4)) + (5 + 6)) \\
\rightarrow ((1 + 2) + (3 + 4)) + (5 + 6) \\
\rightarrow (3 + (3 + 4)) + (5 + 6) \\
\rightarrow (3 + 7) + (5 + 6) \\
\rightarrow (1 : 0) + (5 + 6) \\
\rightarrow 1 : (0 + (5 + 6))
\]
### Definition

**Leftmost outermost** strategy always reduces the leftmost or the outermost redexes.

### Example (cont’d)

\[
((0 : (1 + 2)) + (3 + 4)) + (5 + 6) \rightarrow (0 : ((1 + 2) + (3 + 4))) + (5 + 6) \\
\rightarrow 0 : (((1 + 2) + (3 + 4)) + (5 + 6)) \\
\rightarrow ((1 + 2) + (3 + 4)) + (5 + 6) \\
\rightarrow (3 + (3 + 4)) + (5 + 6) \\
\rightarrow (3 + 7) + (5 + 6) \\
\rightarrow (1 : 0) + (5 + 6) \\
\rightarrow 1 : (0 + (5 + 6)) \\
\rightarrow 1 : (0 + (1 : 1))
\]
Definition

**Leftmost outermost** strategy always reduces the leftmost or the outermost redexes.

Example (cont’d)

\[((0 : (1 + 2)) + (3 + 4)) + (5 + 6)\] → \((0 : ((1 + 2) + (3 + 4))) + (5 + 6)\)

→ \(0 : (((1 + 2) + (3 + 4)) + (5 + 6))\)

→ \(((1 + 2) + (3 + 4)) + (5 + 6)\)

→ \((3 + (3 + 4)) + (5 + 6)\)

→ \((3 + 7) + (5 + 6)\)

→ \((1 : 0) + (5 + 6)\)

→ \(1 : (0 + (5 + 6))\)

→ \(1 : (0 + (1 : 1))\)

→ \(1 : (1 : (0 + 1))\)
**Definition**

*Leftmost outermost* strategy always reduces the leftmost or the outermost redexes.

**Example (cont’d)**

\[ ((0 : (1 + 2)) + (3 + 4)) + (5 + 6) \rightarrow (0 : ((1 + 2) + (3 + 4))) + (5 + 6) \]
\[ \rightarrow 0 : (((1 + 2) + (3 + 4)) + (5 + 6)) \]
\[ \rightarrow ((1 + 2) + (3 + 4)) + (5 + 6) \]
\[ \rightarrow (3 + (3 + 4)) + (5 + 6) \]
\[ \rightarrow (3 + 7) + (5 + 6) \]
\[ \rightarrow (1 : 0) + (5 + 6) \]
\[ \rightarrow 1 : (0 + (5 + 6)) \]
\[ \rightarrow 1 : (0 + (1 : 1)) \]
\[ \rightarrow 1 : (1 : (0 + 1)) \]
\[ \rightarrow (1 + 1) : (0 + 1) \]
**Definition**

Leftmost outermost strategy always reduces the leftmost or the outermost redexes.

**Example (cont’d)**

\[ ((0 : (1 + 2)) + (3 + 4)) + (5 + 6) \rightarrow (0 : ((1 + 2) + (3 + 4))) + (5 + 6) \]
\[ \rightarrow 0 : (((1 + 2) + (3 + 4)) + (5 + 6)) \]
\[ \rightarrow ((1 + 2) + (3 + 4)) + (5 + 6) \]
\[ \rightarrow (3 + (3 + 4)) + (5 + 6) \]
\[ \rightarrow (3 + 7) + (5 + 6) \]
\[ \rightarrow (1 : 0) + (5 + 6) \]
\[ \rightarrow 1 : (0 + (5 + 6)) \]
\[ \rightarrow 1 : (0 + (1 : 1)) \]
\[ \rightarrow 1 : (1 : (0 + 1)) \]
\[ \rightarrow (1 + 1) : (0 + 1) \]
\[ \rightarrow 2 : (0 + 1) \]
**Definition**

Leftmost outermost strategy always reduces the leftmost or the outermost redexes.

**Example (cont’d)**

\[
((0 : (1 + 2)) + (3 + 4)) + (5 + 6) \rightarrow (0 : ((1 + 2) + (3 + 4))) + (5 + 6) \\
\rightarrow 0 : (((1 + 2) + (3 + 4)) + (5 + 6)) \\
\rightarrow ((1 + 2) + (3 + 4)) + (5 + 6) \\
\rightarrow (3 + (3 + 4)) + (5 + 6) \\
\rightarrow (3 + 7) + (5 + 6) \\
\rightarrow (1 + 0) + (5 + 6) \\
\rightarrow 1 : (0 + (5 + 6)) \\
\rightarrow 1 : (0 + (1 : 1)) \\
\rightarrow 1 : (1 : (0 + 1)) \\
\rightarrow (1 + 1) : (0 + 1) \\
\rightarrow 2 : (0 + 1) \\
\rightarrow 2 : 1
\]
Definition

Leftmost innermost strategy always reduces the leftmost of the innermost redexes.
**Definition**

*Leftmost innermost* strategy always reduces the leftmost of the innermost redexes.

**Example (cont’d)**

\[
(0 : (1 + 2)) + (3 + 4) + (5 + 6)
\]
**Definition**

**Leftmost innermost** strategy always reduces the leftmost of the innermost redexes.

**Example (cont’d)**

\[(0 : (1 + 2)) + (3 + 4) + (5 + 6) \rightarrow ((0 : 3) + (3 + 4)) + (5 + 6)\]
**Definition**

Leftmost innermost strategy always reduces the leftmost of the innermost redexes.

**Example (cont’d)**

\[(0 : (1 + 2)) + (3 + 4) + (5 + 6) \rightarrow ((0 : 3) + (3 + 4)) + (5 + 6)\]
\[\rightarrow (3 + (3 + 4)) + (5 + 6)\]
**Definition**

*Leftmost innermost* strategy always reduces the leftmost of the innermost redexes.

**Example (cont’d)**

\[
(0 : (1 + 2)) + (3 + 4) + (5 + 6) \rightarrow (0 : 3) + (3 + 4)) + (5 + 6) \\
\rightarrow (3 + (3 + 4)) + (5 + 6) \\
\rightarrow (3 + 7) + (5 + 6)
\]
**Definition**

**Leftmost innermost** strategy always reduces the leftmost of the innermost redexes.

**Example (cont’d)**

\[
(0 : (1 + 2)) + (3 + 4) + (5 + 6) \rightarrow (0 : 3) + (3 + 4) + (5 + 6) \\
\rightarrow 3 + (3 + 4) + (5 + 6) \\
\rightarrow (3 + 7) + (5 + 6) \\
\rightarrow (1 : 0) + (5 + 6) \\
\rightarrow (1 + 1) : 1 \\
\rightarrow 2 : 1
\]
**Definition**

*Leftmost innermost* strategy always reduces the leftmost of the innermost redexes.

**Example (cont’d)**

\[
(0 : (1 + 2)) + (3 + 4) + (5 + 6) \rightarrow (0 : (3 + 4)) + (5 + 6) \\
\rightarrow (3 + (3 + 4)) + (5 + 6) \\
\rightarrow (3 + 7) + (5 + 6) \\
\rightarrow (1 : 0) + (5 + 6) \\
\rightarrow (1 : 0) + (1 : 1)
\]
Definition

Leftmost innermost strategy always reduces the leftmost of the innermost redexes.

Example (cont’d)

\[
(0 : (1 + 2)) + (3 + 4) + (5 + 6) \rightarrow (0 : 3) + (3 + 4) + (5 + 6) \\
\rightarrow (3 + (3 + 4)) + (5 + 6) \\
\rightarrow (3 + 7) + (5 + 6) \\
\rightarrow (1 : 0) + (5 + 6) \\
\rightarrow (1 : 0) + (1 : 1) \\
\rightarrow 1 : (0 + (1 : 1))
\]
**Definition**

Leftmost innermost strategy always reduces the leftmost of the innermost redexes.

**Example (cont’d)**

\[
(0 : (1 + 2)) + (3 + 4) + (5 + 6) \rightarrow ((0 : 3) + (3 + 4)) + (5 + 6)
\]
\[
\rightarrow (3 + (3 + 4)) + (5 + 6)
\]
\[
\rightarrow (3 + 7) + (5 + 6)
\]
\[
\rightarrow (1 : 0) + (5 + 6)
\]
\[
\rightarrow (1 : 0) + (1 : 1)
\]
\[
\rightarrow 1 : (0 + (1 : 1))
\]
\[
\rightarrow 1 : (1 : (0 + 1))
\]
**Definition**

*Leftmost innermost* strategy always reduces the leftmost of the innermost redexes.

**Example (cont’d)**

\[
(0 : (1 + 2)) + (3 + 4) + (5 + 6) \rightarrow ((0 : 3) + (3 + 4)) + (5 + 6) \\
\quad \rightarrow (3 + (3 + 4)) + (5 + 6) \\
\quad \rightarrow (3 + 7) + (5 + 6) \\
\quad \rightarrow (1 : 0) + (5 + 6) \\
\quad \rightarrow (1 : 0) + (1 : 1) \\
\quad \rightarrow 1 : (0 + (1 : 1)) \\
\quad \rightarrow 1 : (1 : (0 + 1)) \\
\quad \rightarrow 1 : (1 : 1)
\]
**Definition**

*Leftmost innermost* strategy always reduces the leftmost of the innermost redexes.

**Example (cont’d)**

\[
(0 : (1 + 2)) + (3 + 4) + (5 + 6) \rightarrow (0 : 3) + (3 + 4) + (5 + 6) \\
\rightarrow (3 + (3 + 4)) + (5 + 6) \\
\rightarrow (3 + 7) + (5 + 6) \\
\rightarrow (1 : 0) + (5 + 6) \\
\rightarrow (1 : 0) + (1 : 1) \\
\rightarrow 1 : (0 + (1 : 1)) \\
\rightarrow 1 : (1 : (0 + 1)) \\
\rightarrow 1 : (1 : 1) \\
\rightarrow (1 + 1) : 1
\]
### Definition

**Leftmost innermost** strategy always reduces the leftmost of the innermost redexes.

### Example (cont’d)

\[(0 : (1 + 2)) + (3 + 4) + (5 + 6) \rightarrow ((0 : 3) + (3 + 4)) + (5 + 6)\]
\[\rightarrow (3 + (3 + 4)) + (5 + 6)\]
\[\rightarrow (3 + 7) + (5 + 6)\]
\[\rightarrow (1 : 0) + (5 + 6)\]
\[\rightarrow (1 : 0) + (1 : 1)\]
\[\rightarrow 1 : (0 + (1 : 1))\]
\[\rightarrow 1 : (1 : (0 + 1))\]
\[\rightarrow 1 : (1 : 1)\]
\[\rightarrow (1 + 1) : 1\]
\[\rightarrow 2 : 1\]
**Definition**

Parallel outermost strategy always reduces all outermost redexes in parallel.

```
((0 : (1 + 2)) + (3 + 4)) + (5 + 6)
∥ → (0 : ((1 + 2) + (3 + 4))) + (1 : 1)
→ 0 : (((1 + 2) + (3 + 4)) + (1 : 1))
→ ((1 + 2) + (3 + 4)) + (1 : 1)
∥ → 1 : ((1 + 2) + (3 + 4)) + 1
→ 1 : (3 + 7) + 1
→ 1 : ((1 : 0) + 1)
→ 1 : (1 : (0 + 1))
∥ → (1 + 1) : (0 + 1)
∥ → 2 : 1
```
**Definition**

Parallel outermost strategy always reduces all outermost redexes in parallel.

**Example (cont’d)**

\[
((0 : (1 + 2)) + (3 + 4)) + (5 + 6)
\]
Definition

Parallel outermost strategy always reduces all outermost redexes in parallel.

Example (cont’d)

\[(0 : (1 + 2)) + (3 + 4)) + (5 + 6) \mapsto (0 : ((1 + 2) + (3 + 4))) + (1 : 1)\]
Definition

Parallel outermost strategy always reduces all outermost redexes in parallel.

Example (cont’d)

\[
((0 : (1 + 2)) + (3 + 4)) + (5 + 6) \Rightarrow (0 : ((1 + 2) + (3 + 4))) + (1 : 1)
\]

\[
\Rightarrow 0 : (((1 + 2) + (3 + 4)) + (1 : 1))
\]
Definition

Parallel outermost strategy always reduces all outermost redexes in parallel.

Example (cont’d)

\[ ((0 : (1 + 2)) + (3 + 4)) + (5 + 6) \Rightarrow (0 : ((1 + 2) + (3 + 4))) + (1 : 1) \]
\[ \rightarrow 0 : (((1 + 2) + (3 + 4)) + (1 : 1)) \]
\[ \rightarrow ((1 + 2) + (3 + 4)) + (1 : 1) \]
**Definition**

**Parallel outermost** strategy always reduces all outermost redexes in parallel.

**Example (cont’d)**

\[
((0 : (1 + 2)) + (3 + 4)) + (5 + 6) \Rightarrow (0 : ((1 + 2) + (3 + 4))) + (1 : 1) \\
\rightarrow 0 : (((1 + 2) + (3 + 4)) + (1 : 1)) \\
\rightarrow ((1 + 2) + (3 + 4)) + (1 : 1) \\
\rightarrow 1 : (((1 + 2) + (3 + 4)) + 1)
\]
Definition

Parallel outermost strategy always reduces all outermost redexes in parallel.

Example (cont’d)

\[
((0 : (1 + 2)) + (3 + 4)) + (5 + 6) \quad \Rightarrow \quad (0 : ((1 + 2) + (3 + 4))) + (1 : 1)
\]

\[
\Rightarrow 0 : (((1 + 2) + (3 + 4)) + (1 : 1))
\]

\[
\Rightarrow ((1 + 2) + (3 + 4)) + (1 : 1)
\]

\[
\Rightarrow 1 : (((1 + 2) + (3 + 4)) + 1)
\]

\[
\Rightarrow 1 : ((3 + 7) + 1)
\]
Definition

Parallel outermost strategy always reduces all outermost redexes in parallel.

Example (cont’d)

$$
((0 : (1 + 2)) + (3 + 4)) + (5 + 6) \Rightarrow (0 : ((1 + 2) + (3 + 4))) + (1 : 1)
\rightarrow 0 : (((1 + 2) + (3 + 4)) + (1 : 1))
\rightarrow ((1 + 2) + (3 + 4)) + (1 : 1)
\rightarrow 1 : (((1 + 2) + (3 + 4)) + 1)
\Rightarrow 1 : ((3 + 7) + 1)
\rightarrow 1 : ((1 : 0) + 1)
$$
Definition
Parallel outermost strategy always reduces all outermost redexes in parallel.

Example (cont’d)

\[(0 : (1 + 2)) + (3 + 4)) + (5 + 6) \Rightarrow (0 : ((1 + 2) + (3 + 4))) + (1 : 1)\]
\[\Rightarrow 0 : (((1 + 2) + (3 + 4)) + (1 : 1))\]
\[\Rightarrow ((1 + 2) + (3 + 4)) + (1 : 1)\]
\[\Rightarrow 1 : (((1 + 2) + (3 + 4)) + 1)\]
\[\Rightarrow 1 : ((3 + 7) + 1)\]
\[\Rightarrow 1 : ((1 : 0) + 1)\]
\[\Rightarrow 1 : (1 : (0 + 1))\]
**Definition**

Parallel outermost strategy always reduces all outermost redexes in parallel.

**Example (cont’d)**

\[
((0 : (1 + 2)) + (3 + 4)) + (5 + 6) \Rightarrow (0 : ((1 + 2) + (3 + 4))) + (1 : 1)
\]

\[
\Rightarrow 0 : (((1 + 2) + (3 + 4)) + (1 : 1))
\]

\[
\Rightarrow ((1 + 2) + (3 + 4)) + (1 : 1)
\]

\[
\Rightarrow 1 : (((1 + 2) + (3 + 4)) + 1)
\]

\[
\Rightarrow 1 : ((1 + 2) + (3 + 7)) + 1
\]

\[
\Rightarrow 1 : ((3 + 7) + 1)
\]

\[
\Rightarrow 1 : ((1 : 0) + 1)
\]

\[
\Rightarrow 1 : (1 : (0 + 1))
\]

\[
\Rightarrow (1 + 1) : (0 + 1)
\]
Definition

Parallel outermost strategy always reduces all outermost redexes in parallel.

Example (cont’d)

\[ ((0 : (1 + 2)) + (3 + 4)) + (5 + 6) \rightarrow (0 : ((1 + 2) + (3 + 4))) + (1 : 1) \]
\[ \rightarrow 0 : (((1 + 2) + (3 + 4)) + (1 : 1)) \]
\[ \rightarrow ((1 + 2) + (3 + 4)) + (1 : 1) \]
\[ \rightarrow 1 : (((1 + 2) + (3 + 4)) + 1) \]
\[ \rightarrow 1 : ((3 + 7) + 1) \]
\[ \rightarrow 1 : ((1 : 0) + 1) \]
\[ \rightarrow 1 : (1 : (0 + 1)) \]
\[ \rightarrow (1 + 1) : (0 + 1) \]
\[ \rightarrow 2 : 1 \]
Strategies

**Definition**

Parallel innermost strategy always reduces all innermost redexes in parallel.
**Definition**

Parallel innermost strategy always reduces all innermost redexes in parallel.

**Example (cont’d)**

\[(0 : (1 + 2)) + (3 + 4)) + (5 + 6)\]
Definition

Parallel innermost strategy always reduces all innermost redexes in parallel.

Example (cont’d)

\[
((0 : (1 + 2)) + (3 + 4)) + (5 + 6) \Rightarrow ((0 : 3) + 7) + (1 : 1)
\]
**Definition**

Parallel innermost strategy always reduces all innermost redexes in parallel.

**Example (cont’d)**

\[
((0 : (1 + 2)) + (3 + 4)) + (5 + 6) \Rightarrow ((0 : 3) + 7) + (1 : 1) \\
\Rightarrow (3 + 7) + (1 : 1)
\]
**Definition**

Parallel innermost strategy always reduces all innermost redexes in parallel.

**Example (cont’d)**

\[
((0 : (1 + 2)) + (3 + 4)) + (5 + 6) \rightarrow ((0 : 3) + 7) + (1 : 1)
\]

\[
\rightarrow (3 + 7) + (1 : 1)
\]

\[
\rightarrow (1 : 0) + (1 : 1)
\]
Definition

**Parallel innermost** strategy always reduces all innermost redexes in parallel.

Example (cont’d)

\[
((0 : (1 + 2)) + (3 + 4)) + (5 + 6) \Rightarrow ((0 : 3) + 7) + (1 : 1)
\]

\[
\Rightarrow (3 + 7) + (1 : 1)
\]

\[
\Rightarrow (1 : 0) + (1 : 1)
\]

\[
\Rightarrow 1 : (0 + (1 : 1))
\]
Definition

Parallel innermost strategy always reduces all innermost redexes in parallel.

Example (cont’d)

\[(0 : (1 + 2)) + (3 + 4) + (5 + 6) \Rightarrow (0 : 3) + 7 + (1 : 1) \]
\[\Rightarrow (3 + 7) + (1 : 1) \]
\[\Rightarrow (1 : 0) + (1 : 1) \]
\[\Rightarrow 1 : (0 + (1 : 1)) \]
\[\Rightarrow 1 : (1 : (0 + 1)) \]
**Definition**

Parallel innermost strategy always reduces all innermost redexes in parallel.

**Example (cont’d)**

\[
((0 : (1 + 2)) + (3 + 4)) + (5 + 6) \quad \Rightarrow \quad ((0 : 3) + 7) + (1 : 1)
\]

\[
\Rightarrow (3 + 7) + (1 : 1)
\]

\[
\Rightarrow (1 : 0) + (1 : 1)
\]

\[
\Rightarrow 1 : (0 + (1 : 1))
\]

\[
\Rightarrow 1 : (1 : (0 + 1))
\]

\[
\Rightarrow 1 : (1 : 1)
\]
Definition

Parallel innermost strategy always reduces all innermost redexes in parallel.

Example (cont’d)

\[(0 : (1 + 2)) + (3 + 4)) + (5 + 6) \rightarrow ((0 : 3) + 7) + (1 : 1)\]
\[\rightarrow (3 + 7) + (1 : 1)\]
\[\rightarrow (1 : 0) + (1 : 1)\]
\[\rightarrow 1 : (0 + (1 : 1))\]
\[\rightarrow 1 : (1 : (0 + 1))\]
\[\rightarrow 1 : (1 : 1)\]
\[\rightarrow (1 + 1) : 1\]
**Definition**

Parallel innermost strategy always reduces all innermost redexes in parallel.

**Example (cont’d)**

\[((0 : (1 + 2)) + (3 + 4)) + (5 + 6)\] \[\Rightarrow\] \[((0 : 3) + 7) + (1 : 1)\]

\[\rightarrow (3 + 7) + (1 : 1)\]

\[\rightarrow (1 : 0) + (1 : 1)\]

\[\rightarrow 1 : (0 + (1 : 1))\]

\[\rightarrow 1 : (1 : (0 + 1))\]

\[\rightarrow 1 : (1 : 1)\]

\[\rightarrow (1 + 1) : 1\]

\[\rightarrow 2 : 1\]
**Definition**

A development of set of redex positions $Q$ in term $t$ is rewrite sequence starting from $t$ in which all contracted redex positions descend from position in $Q$.

**Example**

- **rewrite rules**
  
  $0 + y \rightarrow y$  
  $0 \times y \rightarrow 0$  
  $s(x) + y \rightarrow s(x + y)$  
  $s(x) \times y \rightarrow (x \times y) + y$

- **rewrite sequences**
  
  $s(0) \times (0 \times 0) \rightarrow (0 \times (0 \times 0)) + (0 \times 0) \rightarrow (0 \times 0) + (0 \times 0)$
Definition

A development of set of redex positions \( Q \) in term \( t \) is rewrite sequence starting from \( t \) in which all contracted redex positions descend from position in \( Q \).

Example

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

- rewrite sequences
  
  \[
  s(0) \times (0 \times 0) \rightarrow (0 \times (0 \times 0)) + (0 \times 0) \rightarrow (0 \times 0) + (0 \times 0)
  \]

  \[
  s(0) \times (0 \times 0) \rightarrow (0 \times (0 \times 0)) + (0 \times 0) \rightarrow (0 \times 0) + (0 \times 0)
  \]
**Definition**

A development of set of redex positions \( Q \) in term \( t \) is rewrite sequence starting from \( t \) in which all contracted redex positions descend from position in \( Q \).

**Example**

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

- **rewrite sequences**
  
  \[
  s(0) \times (0 \times 0) \rightarrow (0 \times (0 \times 0)) + (0 \times 0) \rightarrow (0 \times 0) + (0 \times 0) \\
  s(0) \times (0 \times 0) \rightarrow (0 \times (0 \times 0)) + (0 \times 0) \rightarrow (0 \times 0) + (0 \times 0) \\
  s(0) \times (0 \times 0) \rightarrow s(0) \times 0 \rightarrow (0 \times 0) + 0
  \]
Definition (Overlining)

For a TRS \( \mathcal{R} = \langle \Sigma, R \rangle \) we define the overlined TRS \( \overline{\mathcal{R}} = \langle \overline{\Sigma}, \overline{R} \rangle \):

- \( \overline{\Sigma} = \Sigma \cup \{ \overline{f} \mid f \in \Sigma \} \),
- \( \overline{R} = \{ \overline{\rho} \mid \rho \in R \} \)

where \( \overline{\rho} \) is obtained from \( \rho \) by overlining the head symbol of the left-hand side.

\[ \rho : f(s_1, \ldots, s_n) \rightarrow r \quad \text{yields} \quad \overline{\rho} : \overline{f}(s_1, \ldots, s_n) \rightarrow r \]
Definition (Overlining)

For a TRS $\mathcal{R} = \langle \Sigma, R \rangle$ we define the overlined TRS $\overline{\mathcal{R}} = \langle \overline{\Sigma}, \overline{R} \rangle$:

- $\overline{\Sigma} = \Sigma \cup \{ \overline{f} \mid f \in \Sigma \}$,
- $\overline{R} = \{ \overline{\rho} \mid \rho \in R \}$

where $\overline{\rho}$ is obtained from $\rho$ by overlining the head symbol of the left-hand side.

$$\rho : f(s_1, \ldots, s_n) \rightarrow r \quad \text{yields} \quad \overline{\rho} : \overline{f}(s_1, \ldots, s_n) \rightarrow r$$

Example

The overlined version of Combinatory Logic:

$$\overline{Ap}(Ap(Ap(S, x), y), z) \rightarrow Ap(Ap(x, z), Ap(y, z))$$
$$\overline{Ap}(Ap(K, x), y) \rightarrow x$$
$$\overline{Ap}(l, x) \rightarrow x$$
We write $t \geq s$ if $t$ can be obtained from $s$ by overlining some redex positions.

**Definition (Lifting)**

A $\mathcal{R}$ rewrite sequence $A : s_1 \rightarrow s_2 \rightarrow \ldots \rightarrow s_n$ can be lifted if:

$$
\begin{align*}
t_1 & \overset{\langle \rho_1, p_1 \rangle}{\rightarrow} t_2 & \overset{\langle \rho_2, p_2 \rangle}{\rightarrow} \ldots & \overset{\langle \rho_{n-1}, p_{n-1} \rangle}{\rightarrow} t_n \\
s_1 & \overset{\langle \rho_1, p_1 \rangle}{\rightarrow} s_2 & \overset{\langle \rho_2, p_2 \rangle}{\rightarrow} \ldots & \overset{\langle \rho_{n-1}, p_{n-1} \rangle}{\rightarrow} s_n
\end{align*}
$$

for some $\overline{\mathcal{R}}$ rewrite sequence $B : t_1 \rightarrow \ldots t_n$.

Lemma

For orthogonal TRSs: a reduction is a development $\iff$ it can be lifted.
We write $t \geq s$ if $t$ can be obtained from $s$ by overlining some redex positions.

**Definition (Lifting)**

A $\mathcal{R}$ rewrite sequence $A : s_1 \rightarrow s_2 \rightarrow \ldots \rightarrow s_n$ can be lifted if:

$$\langle \rho_1, p_1 \rangle \rightarrow \langle \rho_2, p_2 \rangle \rightarrow \ldots \rightarrow \langle \rho_{n-1}, p_{n-1} \rangle$$

for some $\mathcal{R}$ rewrite sequence $B : t_1 \rightarrow \ldots t_n$.

**Lemma**

For orthogonal TRSs: a reduction is a development $\iff$ it can be lifted.
**Theorem**

Properties of $\overline{R}$ for orthogonal TRSs $R$:

- $\overline{R}$ is orthogonal.
- $\overline{R}$ is SN.
- $\overline{R}$ is CR.

Proof. The orthogonality of $\overline{R}$ is immediate. Hence $\overline{R}$ is CR.

For SN we show that $\overline{R}$ is ILPO terminating where $\overline{f} > g$ for every $f, g \in \Sigma$.

Let $\ell = \overline{f}(\ell_1, \ldots, \ell_n)$ and $r \in T(\Sigma, \mathcal{X})$ with $\text{Var}(r) \subseteq \text{Var}\ell$.

Then $\ell^* \rightarrow_{ilpo}^* r$ by induction on $r$:

- If $r \in \text{Var}(\ell)$, then we use $\rightarrow_{\text{put}}$ and $\rightarrow_{\text{select}}$.
- If $r = g(r_1, \ldots, r_m)$, then we use $\ell^* \rightarrow_{\text{copy}} g(\ell^*, \ldots, \ell^*)$.

Moreover by the induction hypothesis $\ell^* \rightarrow_{ilpo}^* r_i$ for every $i$. 

Theorem

Properties of $\overline{R}$ for orthogonal TRSs $R$:

• $\overline{R}$ is orthogonal.
• $\overline{R}$ is SN.
• $\overline{R}$ is CR.

Proof.

The orthogonality of $\overline{R}$ is immediate. Hence $\overline{R}$ is CR.
Theorem

Properties of $\overline{R}$ for orthogonal TRSs $R$:

- $\overline{R}$ is orthogonal.
- $\overline{R}$ is SN.
- $\overline{R}$ is CR.

Proof.

The orthogonality of $\overline{R}$ is immediate. Hence $\overline{R}$ is CR.

For SN we show that $\overline{R}$ is ILPO terminating where $\overline{f} > g$ for every $f, g \in \Sigma$. 
Theorem

Properties of $\overline{R}$ for orthogonal TRSs $R$:

- $\overline{R}$ is orthogonal.
- $\overline{R}$ is SN.
- $\overline{R}$ is CR.

Proof.

The orthogonality of $\overline{R}$ is immediate. Hence $\overline{R}$ is CR.

For SN we show that $\overline{R}$ is ILPO terminating where $\overline{f} > g$ for every $f, g \in \Sigma$. Let $\ell = \overline{f}(\ell_1, \ldots, \ell_n)$ and $r \in T(\Sigma, \mathcal{X})$ with $\text{Var}(r) \subseteq \text{Var}\ell$. 
Theorem

Properties of \( \overline{\mathcal{R}} \) for orthogonal TRSs \( \mathcal{R} \):

- \( \overline{\mathcal{R}} \) is orthogonal.
- \( \overline{\mathcal{R}} \) is SN.
- \( \overline{\mathcal{R}} \) is CR.

Proof.

The orthogonality of \( \overline{\mathcal{R}} \) is immediate. Hence \( \overline{\mathcal{R}} \) is CR.

For SN we show that \( \overline{\mathcal{R}} \) is ILPO terminating where \( \bar{f} > g \) for every \( f, g \in \Sigma \).

Let \( \ell = \bar{f}(\ell_1, \ldots, \ell_n) \) and \( r \in T(\Sigma, \mathcal{X}) \) with \( \text{Var}(r) \subseteq \text{Var}\ell \).

Then \( \ell^* \rightarrow_{ilpo}^* r \) by induction on \( r \):
**Theorem**

Properties of $\mathcal{R}$ for orthogonal TRSs $\mathcal{R}$:

- $\mathcal{R}$ is orthogonal.
- $\mathcal{R}$ is SN.
- $\mathcal{R}$ is CR.

**Proof.**

The orthogonality of $\mathcal{R}$ is immediate. Hence $\mathcal{R}$ is CR.

For SN we show that $\mathcal{R}$ is ILPO terminating where $\bar{f} > g$ for every $f, g \in \Sigma$.

Let $\ell = \bar{f}(\ell_1, \ldots, \ell_n)$ and $r \in T(\Sigma, \mathcal{X})$ with $\text{Var}(r) \subseteq \text{Var}\ell$.

Then $\ell^* \rightarrow_{ilpo}^* r$ by induction on $r$:

- If $r \in \text{Var}(\ell)$, then we use $\rightarrow_{put}$ and $\rightarrow_{select}$.
Theorem

Properties of $\overline{R}$ for orthogonal TRSs $R$:

- $\overline{R}$ is orthogonal.
- $\overline{R}$ is SN.
- $\overline{R}$ is CR.

Proof.

The orthogonality of $\overline{R}$ is immediate. Hence $\overline{R}$ is CR.

For SN we show that $\overline{R}$ is ILPO terminating where $\overline{f} > g$ for every $f, g \in \Sigma$.

Let $\ell = \overline{f}(\ell_1, \ldots, \ell_n)$ and $r \in T(\Sigma, \mathcal{X})$ with $\text{Var}(r) \subseteq \text{Var}\ell$.

Then $\ell^* \rightarrow_{\text{ilpo}}^* r$ by induction on $r$:

- If $r \in \text{Var}(\ell)$, then we use $\rightarrow_{\text{put}}$ and $\rightarrow_{\text{select}}$.
- If $r = g(\ell_1, \ldots, \ell_m)$, then we use $\ell^* \rightarrow_{\text{copy}} g(\ell^*, \ldots, \ell^*)$. 
Theorem

Properties of $\overline{\mathcal{R}}$ for orthogonal TRSs $\mathcal{R}$:

- $\overline{\mathcal{R}}$ is orthogonal.
- $\overline{\mathcal{R}}$ is SN.
- $\overline{\mathcal{R}}$ is CR.

Proof.

The orthogonality of $\overline{\mathcal{R}}$ is immediate. Hence $\overline{\mathcal{R}}$ is CR.

For SN we show that $\overline{\mathcal{R}}$ is ILPO terminating where $\overline{f} > g$ for every $f, g \in \Sigma$. Let $\ell = \overline{f}(\ell_1, \ldots, \ell_n)$ and $r \in T(\Sigma, X)$ with $\text{Var}(r) \subseteq \text{Var}\ell$.

Then $\ell^* \rightarrow^*_{ilpo} r$ by induction on $r$:

- If $r \in \text{Var}(\ell)$, then we use $\rightarrow_{\text{put}}$ and $\rightarrow_{\text{select}}$.
- If $r = g(r_1, \ldots, r_m)$, then we use $\ell^* \rightarrow_{\text{copy}} g(\ell^*, \ldots, \ell^*)$.

Moreover by the induction hypothesis $\ell^* \rightarrow^*_{ilpo} r_i$ for every $i$. 
Theorem

*Developments are finite.*
Theorem

*Developments are finite.*

Definition

A development $A: s \rightarrow^* t$ of $Q \subseteq \mathcal{P}os(s)$ is **complete** if $Q/A = \emptyset$.

We write $s \rightarrow t$ (**called multi-step**) if there is a complete development $s \rightarrow^* t$. 
Theorem

Developments are finite.

Definition

A development $A: s \rightarrow^* t$ of $Q \subseteq \mathcal{P} \text{os}(s)$ is complete if $Q/A = \emptyset$.

We write $s \Leftrightarrow t$ (called multi-step) if there is a complete development $s \rightarrow^* t$.

Example

\[
\begin{align*}
\underline{s(0) \times (0 \times 0)} & \rightarrow (0 \times (0 \times 0)) + (0 \times 0) \rightarrow (0 \times 0) + (0 \times 0) & \text{🙂} \\
\underline{s(0) \times (0 \times 0)} & \rightarrow \underline{s(0) \times 0} \rightarrow (0 \times 0) + 0 & \text{🥱}
\end{align*}
\]
**Theorem**

*Developments are finite.*

**Definition**

A development $A: s \rightarrow^* t$ of $Q \subseteq \mathcal{Pos}(s)$ is complete if $Q/A = \emptyset$. We write $s \leftrightarrow t$ (called \textit{multi-step}) if there is a complete development $s \rightarrow^* t$.

**Example**

$$
\begin{align*}
\text{s(0) \times (0 \times 0)} & \rightarrow (0 \times (0 \times 0)) + (0 \times 0) \rightarrow (0 \times 0) + (0 \times 0) & \text{☹}
\end{align*}
$$

$$
\begin{align*}
\text{s(0) \times (0 \times 0)} & \rightarrow \text{s(0) \times 0} \rightarrow (0 \times 0) + 0 & \text{☺}
\end{align*}
$$

**Theorem**

*All complete developments of Q are permutation equivalent.*
**Definition**

For orthogonal TRSs the **full substitution** strategy performs **complete development** of all redexes.
### Definition

For orthogonal TRSs the **full substitution strategy** performs **complete development** of all redexes.

### Example

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

- **full substitution strategy**
  
  \[
  s(s(0)) \times (s(0) + s(s(0)))
  \]
Strategies

Definition

For orthogonal TRSs the full substitution strategy performs complete development of all redexes.

Example

- rewrite rules

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

- full substitution strategy

\[
s(s(0)) \times (s(0) + s(s(0))) \\
\Rightarrow (s(0) \times s(0 + s(s(0)))) + s(0 + s(s(0)))
\]
**Definition**

For orthogonal TRSs the **full substitution strategy** performs **complete development** of all redexes.

**Example**

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

- **full substitution strategy**

  \[
  s(s(0)) \times (s(0) + s(s(0)))
  \]

  \[
  \Rightarrow (s(0) \times s(0 + s(s(0)))) + s(0 + s(s(0)))
  \]

  \[
  \Rightarrow ((0 \times s(s(0))) + s(s(0))) + s(s(s(0)))
  \]
Definition

For orthogonal TRSs the **full substitution strategy** performs complete development of all redexes.

Example

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

- full substitution strategy
  
  \[
  s(s(0)) \times (s(0) + s(s(0)))
  \Rightarrow (s(0) \times s(0 + s(s(0)))) + s(0 + s(s(0)))
  \Rightarrow ((0 \times s(s(0)))) + s(s(s(0))) + s(s(s(0)))
  \rightarrow (0 + s(s(s(0)))) + s(s(s(0)))
  \]

\[\rightarrow s(s(s(s(s(s(0))))))\]
Definition

For orthogonal TRSs the full substitution strategy performs complete development of all redexes.

Example

- rewrite rules

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

- full substitution strategy

\[
\begin{align*}
s(s(0)) \times (s(0) + s(s(0))) & \\
\Rightarrow & (s(0) \times s(0 + s(s(0)))) + s(0 + s(s(0))) \\
\Rightarrow & ((0 \times s(s(0)))) + s(s(s(0)))) + s(s(s(0))) \\
\rightarrow & (0 + s(s(s(0)))) + s(s(s(0))) \\
\rightarrow & s(s(s(0))) + s(s(s(0)))
\end{align*}
\]
**Definition**

For orthogonal TRSs the full substitution strategy performs complete development of all redexes.

**Example**

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

- **full substitution strategy**

  \[
  s(s(0)) \times (s(0) + s(s(0))) \\
  \Rightarrow (s(0) \times s(0 + s(s(0)))) + s(0 + s(s(0)))) \\
  \Rightarrow ((0 \times s(s(0)))) + s(s(s(0)))) + s(s(s(0)))) \\
  \rightarrow (0 + s(s(s(0)))) + s(s(s(0)))) \\
  \rightarrow s(s(s(0)))) + s(s(s(0)))) \\
  \rightarrow \ldots \rightarrow s(s(s(s(s(s(0)))))))
  \]
Outline

- Overview

- Modularity
  - Definitions
  - Results
Theorem

For orthogonal TRSs

- full substitution and parallel outermost strategies are normalizing
Theorem

For orthogonal TRSs

- **full substitution** and **parallel outermost strategies are normalizing**
- **innermost strategies are perpetual**
Theorem

For orthogonal TRSs

- full substitution and parallel outermost strategies are normalizing
- innermost strategies are perpetual
- leftmost outermost strategy is not normalizing
Theorem

For orthogonal TRSs

- full substitution and parallel outermost strategies are normalizing
- innermost strategies are perpetual
- leftmost outermost strategy is not normalizing
- full substitution is fair

Example

\[
\begin{align*}
a & \rightarrow b \\
c & \rightarrow c \\
f(x, b) & \rightarrow b
\end{align*}
\]

- leftmost outermost: \( f(c, a) \)
- leftmost innermost: \( f(c, a) \)
- parallel outermost: \( f(c, a) \)
- parallel innermost: \( f(c, a) \)
- full substitution: \( f(c, a) \)
Theorem

For orthogonal TRSs

- full substitution and parallel outermost strategies are normalizing
- innermost strategies are perpetual
- leftmost outermost strategy is not normalizing
- full substitution is fair

Example

\[
\begin{align*}
  a & \rightarrow b \\
  c & \rightarrow c \\
  f(x, b) & \rightarrow b \\
\end{align*}
\]

- leftmost outermost: \( f(c, a) \rightarrow f(c, a) \)
- leftmost innermost: \( f(c, a) \)
- parallel outermost: \( f(c, a) \)
- parallel innermost: \( f(c, a) \)
- full substitution: \( f(c, a) \)
Theorem

For orthogonal TRSs

- full substitution and parallel outermost strategies are normalizing
- innermost strategies are perpetual
- leftmost outermost strategy is not normalizing
- full substitution is fair

Example

\[ a \rightarrow b \quad c \rightarrow c \quad f(x, b) \rightarrow b \]

- leftmost outermost \( f(c, a) \rightarrow f(c, a) \rightarrow f(c, a) \)
- leftmost innermost \( f(c, a) \)
- parallel outermost \( f(c, a) \)
- parallel innermost \( f(c, a) \)
- full substitution \( f(c, a) \)
Theorem

For orthogonal TRSs

- full substitution and parallel outermost strategies are normalizing
- innermost strategies are perpetual
- leftmost outermost strategy is not normalizing
- full substitution is fair

Example

\[ a \rightarrow b \quad c \rightarrow c \quad f(x, b) \rightarrow b \]

- leftmost outermost \[ f(c, a) \rightarrow f(c, a) \rightarrow f(c, a) \rightarrow \cdots \]
- leftmost innermost \[ f(c, a) \]
- parallel outermost \[ f(c, a) \]
- parallel innermost \[ f(c, a) \]
- full substitution \[ f(c, a) \]
Theorem

For orthogonal TRSs

- full substitution and parallel outermost strategies are normalizing
- innermost strategies are perpetual
- leftmost outermost strategy is not normalizing
- full substitution is fair

Example

\[ a \rightarrow b \quad c \rightarrow c \quad f(x, b) \rightarrow b \]

- leftmost outermost \( f(c, a) \rightarrow f(c, a) \rightarrow f(c, a) \rightarrow \cdots \)
- leftmost innermost \( f(c, a) \rightarrow f(c, a) \)
- parallel outermost \( f(c, a) \)
- parallel innermost \( f(c, a) \)
- full substitution \( f(c, a) \)
Theorem

For orthogonal TRSs

- **full substitution and parallel outermost strategies are normalizing**
- **innermost strategies are perpetual**
- **leftmost outermost strategy is not normalizing**
- **full substitution is fair**

Example

\[
\begin{align*}
  a & \rightarrow b \\
  c & \rightarrow c \\
  f(x, b) & \rightarrow b
\end{align*}
\]

- **leftmost outermost**
  \[f(c, a) \rightarrow f(c, a) \rightarrow f(c, a) \rightarrow \cdots\]
- **leftmost innermost**
  \[f(c, a) \rightarrow f(c, a) \rightarrow f(c, a)\]
- **parallel outermost**
  \[f(c, a)\]
- **parallel innermost**
  \[f(c, a)\]
- **full substitution**
  \[f(c, a)\]
Theorem

For orthogonal TRSs

- **full substitution** and **parallel outermost strategies** are **normalizing**
- **innermost strategies** are **perpetual**
- **leftmost outermost strategy** is **not normalizing**
- **full substitution** is **fair**

Example

\[
\begin{align*}
  a & \rightarrow b & c & \rightarrow c & f(x, b) & \rightarrow b \\
  \text{• leftmost outermost} & \quad f(c, a) & \rightarrow f(c, a) & \rightarrow f(c, a) & \rightarrow \cdots \\
  \text{• leftmost innermost} & \quad f(c, a) & \rightarrow f(c, a) & \rightarrow f(c, a) & \rightarrow \cdots \\
  \text{• parallel outermost} & \quad f(c, a) \\
  \text{• parallel innermost} & \quad f(c, a) \\
  \text{• full substitution} & \quad f(c, a)
\end{align*}
\]
Theorem

For orthogonal TRSs

- full substitution and parallel outermost strategies are normalizing
- innermost strategies are perpetual
- leftmost outermost strategy is not normalizing
- full substitution is fair

Example

- leftmost outermost $f(c, a) \rightarrow f(c, a) \rightarrow f(c, a) \rightarrow \cdots$
- leftmost innermost $f(c, a) \rightarrow f(c, a) \rightarrow f(c, a) \rightarrow \cdots$
- parallel outermost $f(c, a) \parallel \rightarrow f(c, b)$
- parallel innermost $f(c, a)$
- full substitution $f(c, a)$
Theorem

For orthogonal TRSs

- **full substitution** and parallel outermost strategies are **normalizing**
- innermost strategies are **perpetual**
- leftmost outermost strategy is **not normalizing**
- full substitution is **fair**

Example

\[
\begin{align*}
a & \to b \\
c & \to c \\
f(x, b) & \to b
\end{align*}
\]

- leftmost outermost: \( f(c, a) \to f(c, a) \to f(c, a) \to \cdots \)
- leftmost innermost: \( f(c, a) \to f(c, a) \to f(c, a) \to \cdots \)
- parallel outermost: \( f(c, a) \parallel \to f(c, b) \to b \)
- parallel innermost: \( f(c, a) \)
- full substitution: \( f(c, a) \)
**Theorem**

For orthogonal TRSs

- **full substitution and parallel outermost strategies are normalizing**
- **innermost strategies are perpetual**
- **leftmost outermost strategy is not normalizing**
- **full substitution is fair**

**Example**

<table>
<thead>
<tr>
<th>Rule</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$a \rightarrow b$</td>
<td></td>
</tr>
<tr>
<td>$c \rightarrow c$</td>
<td></td>
</tr>
<tr>
<td>$f(x, b) \rightarrow b$</td>
<td></td>
</tr>
<tr>
<td><strong>leftmost outermost</strong></td>
<td>$f(c, a) \rightarrow f(c, a) \rightarrow f(c, a) \rightarrow \cdots$</td>
</tr>
<tr>
<td><strong>leftmost innermost</strong></td>
<td>$f(c, a) \rightarrow f(c, a) \rightarrow f(c, a) \rightarrow \cdots$</td>
</tr>
<tr>
<td><strong>parallel outermost</strong></td>
<td>$f(c, a) \parallel f(c, b) \rightarrow b$</td>
</tr>
<tr>
<td><strong>parallel innermost</strong></td>
<td>$f(c, a) \parallel f(c, b)$</td>
</tr>
<tr>
<td><strong>full substitution</strong></td>
<td>$f(c, a)$</td>
</tr>
</tbody>
</table>
Theorem

For orthogonal TRSs

- full substitution and parallel outermost strategies are normalizing
- innermost strategies are perpetual
- leftmost outermost strategy is not normalizing
- full substitution is fair

Example

\[
\begin{align*}
\text{a} & \rightarrow \text{b} \\
\text{c} & \rightarrow \text{c} \\
\text{f(x, b)} & \rightarrow \text{b}
\end{align*}
\]

- leftmost outermost: \( \text{f(c, a)} \rightarrow \text{f(c, a)} \rightarrow \text{f(c, a)} \rightarrow \cdots \)
- leftmost innermost: \( \text{f(c, a)} \rightarrow \text{f(c, a)} \rightarrow \text{f(c, a)} \rightarrow \cdots \)
- parallel outermost: \( \text{f(c, a)} \parallel \rightarrow \text{f(c, b)} \rightarrow \text{b} \)
- parallel innermost: \( \text{f(c, a)} \parallel \rightarrow \text{f(c, b)} \rightarrow \text{f(c, b)} \)
- full substitution: \( \text{f(c, a)} \)
Theorem

For orthogonal TRSs

- full substitution and parallel outermost strategies are normalizing
- innermost strategies are perpetual
- leftmost outermost strategy is not normalizing
- full substitution is fair

Example

\[
\begin{align*}
  a & \rightarrow b \\
  c & \rightarrow c \\
  f(x, b) & \rightarrow b
\end{align*}
\]

- leftmost outermost: \(f(c, a) \rightarrow f(c, a) \rightarrow f(c, a) \rightarrow \cdots\)
- leftmost innermost: \(f(c, a) \rightarrow f(c, a) \rightarrow f(c, a) \rightarrow \cdots\)
- parallel outermost: \(f(c, a) \parallel \rightarrow f(c, b) \rightarrow b\)
- parallel innermost: \(f(c, a) \parallel \rightarrow f(c, b) \rightarrow f(c, b) \rightarrow \cdots\)
- full substitution: \(f(c, a)\)
Theorem

For orthogonal TRSs

- full substitution and parallel outermost strategies are normalizing
- innermost strategies are perpetual
- leftmost outermost strategy is not normalizing
- full substitution is fair

Example

\[ a \rightarrow b \quad c \rightarrow c \quad f(x, b) \rightarrow b \]

- leftmost outermost: \( f(c, a) \rightarrow f(c, a) \rightarrow f(c, a) \rightarrow \cdots \)
- leftmost innermost: \( f(c, a) \rightarrow f(c, a) \rightarrow f(c, a) \rightarrow \cdots \)
- parallel outermost: \( f(c, a) \parallel \rightarrow f(c, b) \rightarrow b \)
- parallel innermost: \( f(c, a) \parallel \rightarrow f(c, b) \rightarrow f(c, b) \rightarrow \cdots \)
- full substitution: \( f(c, a) \rightarrow f(c, b) \)
Theorem

For orthogonal TRSs

- full substitution and parallel outermost strategies are normalizing
- innermost strategies are perpetual
- leftmost outermost strategy is not normalizing
- full substitution is fair

Example

\[ a \rightarrow b \]
\[ c \rightarrow c \]
\[ f(x, b) \rightarrow b \]

- leftmost outermost
  \[ f(c, a) \rightarrow f(c, a) \rightarrow f(c, a) \rightarrow \cdots \]

- leftmost innermost
  \[ f(c, a) \rightarrow f(c, a) \rightarrow f(c, a) \rightarrow \cdots \]

- parallel outermost
  \[ f(c, a) \parallel \rightarrow f(c, b) \rightarrow b \]

- parallel innermost
  \[ f(c, a) \parallel \rightarrow f(c, b) \rightarrow f(c, b) \rightarrow \cdots \]

- full substitution
  \[ f(c, a) \lnot\rightarrow f(c, b) \lnot\rightarrow b \]
Strategies

\textbf{Definition}

A reduction $\rho = t_0 \to t_1 \to \ldots$ is \textbf{cofinal} if for every $t_0 \to^* s$ there exists $t_n$ in $\rho$ such that $s \to^* t_n$.

\textbf{Definition}

A strategy $S$ is cofinal if every maximal $S$ rewrite sequence is cofinal.

\textbf{Theorem}

Cofinal strategies are normalizing.

\textbf{Proof.}

Let $S$ be a cofinal strategy. Let $t_0$ be a term that has a normal form $u$ ($t_0 \to^* u$).

Consider a maximal $S$ rewrite sequence $\rho : t_0 \to t_1 \to \ldots$ starting from $t_0$.

By cofinality there must be $t_n$ in $\rho$ such that $u \to^* t_n$.

Hence $t_n = u$ since $u$ is a normal form.
**Definition**

A reduction \( \rho = t_0 \rightarrow t_1 \rightarrow \ldots \) is **cofinal** if for every \( t_0 \rightarrow^* s \) there exists \( t_n \) in \( \rho \) such that \( s \rightarrow^* t_n \).

**Definition**

A strategy \( S \) is **cofinal** if every maximal \( S \) rewrite sequence is cofinal.

---

Let \( t_0 \) be a term that has a normal form \( u \) (\( t_0 \rightarrow^* u \)).

Consider a maximal \( S \) rewrite sequence \( \rho : t_0 \rightarrow t_1 \rightarrow \ldots \) starting from \( t_0 \).

By cofinality there must be \( t_n \) in \( \rho \) such that \( u \rightarrow^* t_n \).

Hence \( t_n = u \) since \( u \) is a normal form.
Strategies

Definition

A reduction $\rho = t_0 \rightarrow t_1 \rightarrow \ldots$ is **cofinal** if for every $t_0 \rightarrow^* s$ there exists $t_n$ in $\rho$ such that $s \rightarrow^* t_n$.

Definition

A strategy $S$ is **cofinal** if every maximal $S$ rewrite sequence is cofinal.

Theorem

*Cofinal strategies are normalizing.*

Let $t_0$ be a term that has a normal form $u$ ($t_0 \rightarrow^* u$).

Consider a maximal $S$ rewrite sequence $\rho : t_0 \rightarrow t_1 \rightarrow \ldots$ starting from $t_0$.

By cofinality there must be $t_n$ in $\rho$ such that $u \rightarrow^* t_n$.

Hence $t_n = u$ since $u$ is a normal form.
**Definition**

A reduction $\rho = t_0 \rightarrow t_1 \rightarrow \ldots$ is **cofinal** if for every $t_0 \rightarrow^* s$ there exists $t_n$ in $\rho$ such that $s \rightarrow^* t_n$.

**Definition**

A strategy $S$ is **cofinal** if every maximal $S$ rewrite sequence is cofinal.

**Theorem**

*Cofinal strategies are normalizing.*

**Proof.**

Let $S$ be a cofinal strategy.
**Definition**

A reduction $\rho = t_0 \rightarrow t_1 \rightarrow \ldots$ is **cofinal** if for every $t_0 \rightarrow^* s$ there exists $t_n$ in $\rho$ such that $s \rightarrow^* t_n$.

**Definition**

A strategy $S$ is **cofinal** if every maximal $S$ rewrite sequence is cofinal.

**Theorem**

*Cofinal strategies are normalizing.*

**Proof.**

Let $S$ be a cofinal strategy. Let $t_0$ be a term that has a normal form $u$ ($t_0 \rightarrow^* u$).
Definition

A reduction $\rho = t_0 \to t_1 \to \ldots$ is cofinal if for every $t_0 \to^* s$ there exists $t_n$ in $\rho$ such that $s \to^* t_n$.

Definition

A strategy $S$ is cofinal if every every maximal $S$ rewrite sequence is cofinal.

Theorem

Cofinal strategies are normalizing.

Proof.

Let $S$ be a cofinal strategy. Let $t_0$ be a term that has a normal form $u$ ($t_0 \to^* u$).

Consider a maximal $S$ rewrite sequence $\rho : t_0 \to t_1 \to \ldots$ starting from $t_0$. 
**Definition**

A reduction $\rho = t_0 \to t_1 \to \ldots$ is **cofinal** if for every $t_0 \to^* s$ there exists $t_n$ in $\rho$ such that $s \to^* t_n$.

**Definition**

A strategy $S$ is **cofinal** if every maximal $S$ rewrite sequence is cofinal.

**Theorem**

*Cofinal strategies are normalizing.*

**Proof.**

Let $S$ be a cofinal strategy. Let $t_0$ be a term that has a normal form $u$ ($t_0 \to^* u$).

Consider a maximal $S$ rewrite sequence $\rho : t_0 \to t_1 \to \ldots$ starting from $t_0$.

By cofinality there must be $t_n$ in $\rho$ such that $u \to^* t_n$. 
Definition
A reduction $\rho = t_0 \rightarrow t_1 \rightarrow \ldots$ is cofinal if for every $t_0 \rightarrow^* s$ there exists $t_n$ in $\rho$ such that $s \rightarrow^* t_n$.

Definition
A strategy $S$ is cofinal if every maximal $S$ rewrite sequence is cofinal.

Theorem
Cofinal strategies are normalizing.

Proof.
Let $S$ be a cofinal strategy. Let $t_0$ be a term that has a normal form $u$ ($t_0 \rightarrow^* u$).

Consider a maximal $S$ rewrite sequence $\rho : t_0 \rightarrow t_1 \rightarrow \ldots$ starting from $t_0$.

By cofinality there must be $t_n$ in $\rho$ such that $u \rightarrow^* t_n$.

Hence $t_n = u$ since $u$ is a normal form.
Theorem

For orthogonal TRSs, every fair strategy is cofinal.

Let $\tau : t_0 \rightarrow u_0$. We show that $u_0 \rightarrow^* t_n$ for some $t_n$ in $\rho$.

The reduction $u_0 \rightarrow u_1 \rightarrow \ldots \rightarrow u_n \rightarrow t_{n+1} \rightarrow t_{n+2}$ is fair again.

By induction over the length of $t_0 \rightarrow^* u_0$ we get $u_0 \rightarrow^* t_n$ for some $t_n$ in $\rho$. 

Here $\rho_i$ consists of the first $i$ steps of $\rho$.

By fairness of $\rho$ there exists $n$ such that $\tau/\rho_n = \emptyset$. Hence $u_n = t_n$ and $u_0 \rightarrow^* t_n$. 

The reduction $u_0 \rightarrow u_1 \rightarrow \ldots \rightarrow u_n \rightarrow t_{n+1} \rightarrow t_{n+2}$ is fair again.

(every redex occurrence in $t_n$ is eventually secured)
Theorem

For orthogonal TRSs, every fair strategy is cofinal.

Proof.

Let $\rho : t_0 \rightarrow t_1 \rightarrow \ldots$ be a fair rewrite sequence.
Theorem

For orthogonal TRSs, every fair strategy is cofinal.

Proof.

Let \( \rho : t_0 \rightarrow t_1 \rightarrow \ldots \) be a fair rewrite sequence.

Let \( \tau : t_0 \rightarrow u_0 \). We show that \( u_0 \rightarrow^* t_n \) for some \( t_n \) in \( \rho \).
**Theorem**

*For orthogonal TRSs, every fair strategy is cofinal.*

**Proof.**

Let $\rho : t_0 \rightarrow t_1 \rightarrow \ldots$ be a fair rewrite sequence.

Let $\tau : t_0 \rightarrow u_0$. We show that $u_0 \rightarrow^* t_n$ for some $t_n$ in $\rho$.

\[
\begin{array}{c}
t_0 \quad t_1 \quad \ldots \quad t_n \\
\downarrow \tau \quad \downarrow \tau/\rho_1 \quad \downarrow \tau/\rho_2 \quad | \emptyset \\
u_0 \quad u_1 \quad u_2 \quad \ldots \quad u_n
\end{array}
\]

Here $\rho_i$ consists of the first $i$ steps of $\rho$. 
Theorem

For orthogonal TRSs, every fair strategy is cofinal.

Proof.

Let \( \rho : t_0 \rightarrow t_1 \rightarrow \ldots \) be a fair rewrite sequence.

Let \( \tau : t_0 \rightarrow u_0 \). We show that \( u_0 \rightarrow^* t_n \) for some \( t_n \) in \( \rho \).

\[
\begin{array}{ccccccc}
  t_0 & \rightarrow & t_1 & \rightarrow & t_2 & \rightarrow & \cdots & \rightarrow & t_n & \rightarrow & \cdots \\
  \downarrow \tau & & \downarrow \tau/\rho_1 & & \downarrow \tau/\rho_2 & & & & & & \downarrow \emptyset \\
  u_0 & \rightarrow & u_1 & \rightarrow & u_2 & \rightarrow & \cdots & \rightarrow & u_n &
\end{array}
\]

Here \( \rho_i \) consists of the first \( i \) steps of \( \rho \).

By fairness of \( \rho \) there exists \( n \) such that \( \tau/\rho_n = \emptyset \).
Theorem

For orthogonal TRSs, every fair strategy is cofinal.

Proof.

Let \( \rho : t_0 \rightarrow t_1 \rightarrow \ldots \) be a fair rewrite sequence.

Let \( \tau : t_0 \rightarrow u_0 \). We show that \( u_0 \rightarrow^* t_n \) for some \( t_n \) in \( \rho \).

\[
\begin{array}{ccccccc}
  t_0 & \rightarrow & t_1 & \rightarrow & t_2 & \rightarrow & \ldots & \rightarrow & t_n & \rightarrow & \ldots \\
  \downarrow & \tau & \downarrow & \tau/\rho_1 & \downarrow & \tau/\rho_2 & \downarrow & \ldots & \downarrow & \emptyset \\
  u_0 & \rightarrow & u_1 & \rightarrow & u_2 & \rightarrow & \ldots & \rightarrow & u_n \\
\end{array}
\]

Here \( \rho_i \) consists of the first \( i \) steps of \( \rho \).

By fairness of \( \rho \) there exists \( n \) such that \( \tau/\rho_n = \emptyset \). Hence \( u_n = t_n \) and \( u_0 \rightarrow^* t_n \).
Strategies

**Theorem**

*For orthogonal TRSs, every fair strategy is cofinal.*

**Proof.**

Let \( ρ : t_0 \rightarrow t_1 \rightarrow \ldots \) be a fair rewrite sequence.

Let \( τ : t_0 \rightarrow u_0 \). We show that \( u_0 \rightarrow^* t_n \) for some \( t_n \) in \( ρ \).

\[
\begin{array}{cccccccc}
  t_0 & \rightarrow & t_1 & \rightarrow & t_2 & \rightarrow & \ldots & \rightarrow & t_n & \rightarrow & \ldots \\
  \downarrow τ & & \downarrow τ/ρ_1 & & \downarrow τ/ρ_2 & & \downarrow & & \downarrow \emptyset \\
  u_0 & \rightarrow & u_1 & \rightarrow & u_2 & \rightarrow & \ldots & \rightarrow & u_n
\end{array}
\]

Here \( ρ_i \) consists of the first \( i \) steps of \( ρ \).

By fairness of \( ρ \) there exists \( n \) such that \( τ/ρ_n = \emptyset \). Hence \( u_n = t_n \) and \( u_0 \rightarrow^* t_n \).

The reduction \( u_0 \rightarrow u_1 \rightarrow \ldots \rightarrow u_n \rightarrow t_{n+1} \rightarrow t_{n+2} \) is fair again.

(every redex occurrence in \( t_n \) is eventually secured)
Theorem

For orthogonal TRSs, every fair strategy is cofinal.

Proof.

Let $\rho : t_0 \rightarrow t_1 \rightarrow \ldots$ be a fair rewrite sequence.

Let $\tau : t_0 \rightarrow u_0$. We show that $u_0 \rightarrow^* t_n$ for some $t_n$ in $\rho$.

\[
\begin{array}{cccccccccc}
t_0 & \rightarrow & t_1 & \rightarrow & t_2 & \rightarrow & \ldots & \rightarrow & t_n & \rightarrow & \ldots \\
\downarrow \tau & & \downarrow \tau/\rho_1 & & \downarrow \tau/\rho_2 & & & & \downarrow \emptyset \\
u_0 & \rightarrow & u_1 & \rightarrow & u_2 & \rightarrow & \ldots & \rightarrow & u_n & \rightarrow & \ldots \\
\end{array}
\]

Here $\rho_i$ consists of the first $i$ steps of $\rho$.

By fairness of $\rho$ there exists $n$ such that $\tau/\rho_n = \emptyset$. Hence $u_n = t_n$ and $u_0 \rightarrow^* t_n$.

The reduction $u_0 \rightarrow u_1 \rightarrow \ldots \rightarrow u_n \rightarrow t_{n+1} \rightarrow t_{n+2}$ is fair again.

(every redex occurrence in $t_n$ is eventually secured)

By induction over the length of $t_0 \rightarrow^* u_0$ we get $u_0 \rightarrow^* t_n$ for some $t_n$ in $\rho$. 

$\blacksquare$
A TRS is **left-normal** if variables do not precede function symbols in left-hand sides (where the left-hand sides are written in prefix notation).
Definition

A TRS is **left-normal** if variables do not precede function symbols in left-hand sides (where the left-hand sides are written in prefix notation).

Example

- \( f(x, g(y, z)) \rightarrow g(y, f(x, z)) \)
Definition

A TRS is **left-normal** if variables do not precede function symbols in left-hand sides (where the left-hand sides are written in prefix notation).

Example

- \( f(x, g(y, z)) \rightarrow g(y, f(x, z)) \) 😞
Definition

A TRS is **left-normal** if variables do not precede function symbols in left-hand sides (where the left-hand sides are written in prefix notation).

Example

- \( f(x, g(y, z)) \rightarrow g(y, f(x, z)) \) 😞
- \( f(g(x, y), z) \rightarrow g(x, g(y, z)) \)
Strategies

Definition

A TRS is **left-normal** if variables do not precede function symbols in left-hand sides (where the left-hand sides are written in prefix notation).

Example

- \( f(x, g(y, z)) \rightarrow g(y, f(x, z)) \)
- \( f(g(x, y), z) \rightarrow g(x, g(y, z)) \)
**Definition**

A TRS is **left-normal** if variables do not precede function symbols in left-hand sides (where the left-hand sides are written in prefix notation).

**Example**

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

**Theorem**

*Leftmost outermost strategy is normalizing for orthogonal left-normal TRSs.*
Definition

A TRS is left-normal if variables do not precede function symbols in left-hand sides (where the left-hand sides are written in prefix notation).

Example

- \( f(x, g(y, z)) \rightarrow g(y, f(x, z)) \)
- \( f(g(x, y), z) \rightarrow g(x, g(y, z)) \)

Theorem

Leftmost outermost strategy is normalizing for orthogonal left-normal TRSs.

Remark

Combinatory Logic is left-normal

\[
I x \rightarrow x \\
K x y \rightarrow x \\
S x y z \rightarrow x z (y z)
\]