# Rewriting-Based Deduction. Completion

Temur Kutsia

RISC, JKU Linz

# Motivation

- Unrestricted use of the paramodulation rule can be very inefficient.
- Various methods have been proposed to restrict it without compromising the completeness.
- Term rewriting contributed essential techniques for refining paramodulation into a practical inference system.

# Rewriting-Based Deduction for Unit Equalities

- We assume that the given set of clauses consists of unit equalities and one ground inequality.
- Goal: Design a calculus which works on such sets, restricts applications of the paramodulation rule, and is complete.
- Later this calculus can be extended to general clauses.

# Equational Theory

- $E$: A set of equations.
- $A$: The set of equality axioms for $E$.
- $E \vDash s \approx t$ iff $I \vDash s \approx t$ for all interpretations $I$ which is a model of $E \cup A$.
- Equational theory of $E$:

$$\approx_E := \{(s,t) \mid E \vDash s \approx t\}$$

- Notation: $s \approx_E t$ iff $(s,t) \in \approx_E$.

# Basic Concepts in Term Rewriting

- A rewrite rule is an ordered pair of terms, written $l \to r$.
- Term rewriting system (TRS): a set of rewrite rules.

# Problem

Given: A set of equations $E$ and two terms $s$ and $t$.

Decide: $s \approx_E t$ holds or not.

# Problem

Given: A set of equations $E$ and two terms $s$ and $t$.

Decide: $s \approx_E t$ holds or not.

The problem is undecidable for an arbitrary $E$.

# Problem

Given: A set of equations $E$ and two terms $s$ and $t$.

Decide: $s \approx_E t$ holds or not.

The problem is undecidable for an arbitrary $E$.

When $E$ is finite and induces a (ground) convergent TRS, the problem is decidable.

# Problem

Given: A set of equations $E$ and two terms $s$ and $t$.

Decide: $s \approx_E t$ holds or not.

The problem is undecidable for an arbitrary $E$.

When $E$ is finite and induces a (ground) convergent TRS, the problem is decidable.

What's this?

# Problem and Solving Idea

Given: A set of equations $E$ and two terms $s$ and $t$.

Decide: $s \approx_E t$ holds or not.

Solving Idea:

# Problem and Solving Idea

Given: A set of equations $E$ and two terms $s$ and $t$.

Decide: $s \approx_E t$ holds or not.

Solving Idea:

- Refute and skolemize the goal, obtaining the ground disequation $s' \not\approx_E t'$.

# Problem and Solving Idea

Given: A set of equations $E$ and two terms $s$ and $t$.

Decide: $s \approx_E t$ holds or not.

Solving Idea:

- Refute and skolemize the goal, obtaining the ground disequation $s' \not\approx_E t'$.
- Try to construct from $E$ a ground convergent set of equations and rewrite rules, with the procedure called completion.

# Problem and Solving Idea

Given: A set of equations $E$ and two terms $s$ and $t$.

Decide: $s \approx_E t$ holds or not.

Solving Idea:

- Refute and skolemize the goal, obtaining the ground disequation $s' \not\approx_E t'$.
- Try to construct from $E$ a ground convergent set of equations and rewrite rules, with the procedure called completion.
- In the course of completion, from time to time check whether $s'$ and $t'$ can be rewritten to the same term with the equations and rules constructed so far.

# Problem and Solving Idea

Given: A set of equations $E$ and two terms $s$ and $t$.

Decide: $s \approx_E t$ holds or not.

Solving Idea:

- Refute and skolemize the goal, obtaining the ground disequation $s' \not\approx_E t'$.
- Try to construct from $E$ a ground convergent set of equations and rewrite rules, with the procedure called completion.
- In the course of completion, from time to time check whether $s'$ and $t'$ can be rewritten to the same term with the equations and rules constructed so far.
- If yes, stop. You obtained a contradiction, which proves $s \approx_E t$.

# Problem and Solving Idea

Given: A set of equations $E$ and two terms $s$ and $t$.

Decide: $s \approx_E t$ holds or not.

Solving Idea:

- Refute and skolemize the goal, obtaining the ground disequation $s' \not\approx_E t'$.
- Try to construct from $E$ a ground convergent set of equations and rewrite rules, with the procedure called completion.
- In the course of completion, from time to time check whether $s'$ and $t'$ can be rewritten to the same term with the equations and rules constructed so far.
- If yes, stop. You obtained a contradiction, which proves $s \approx_E t$.
- If not, continue with completion. If this is not possible, then report: $s \approx_E t$ does not hold.

# What We Need To Know

- What is rewriting?
- What is a ground convergent set of equations and rewrite rules?
- What is completion?

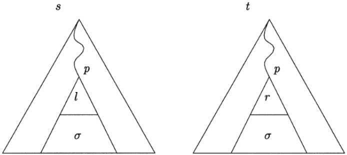
# Basic Concepts in Term Rewriting

$R$: A term rewriting system.

- The rewrite relation induced by $R$, denoted $\to_R$, is a binary relation on terms defined as:

$s \to_R t$ iff

there exist $l \to r \in R$, a position $p$ in $s$, a substitution $\sigma$ such that $s|_p = l\sigma$ and $t = s[r\sigma]_p$.



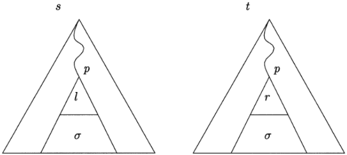
## Basic Concepts in Term Rewriting

$R$: A term rewriting system.

- The rewrite relation induced by $R$, denoted $\to_R$, is a binary relation on terms defined as:

  $s \to_R t$ iff

  there exist $l \to r \in R$, a position $p$ in $s$, a substitution $\sigma$ such that $s|_p = l\sigma$ and $t = s[r\sigma]_p$.




- Obviously $R \subseteq \to_R$.
- We may omit $R$ when it is obvious from the context.

# Basic Concepts in Term Rewriting

- $s$ rewrites to $t$ by $R$ iff $s \to_R t$.
- $\leftarrow_R$ stands for the inverse and $\to_R^*$ for reflexive-transitive closure of $\to_R$.
- $s$ is irreducible by $R$ iff there is no $t$ such that $s \to_R t$.
- $t$ is a normal form of $s$ by $R$ iff $s \to_R^* t$ and $t$ is irreducible by $R$.
- $R$ is terminating iff $\to_R$ is well-founded, i.e., there is no infinite sequence of rewrite steps $s_1 \to_R s_2 \to_R s_3 \to_R \cdots$.

# Basic Concepts in Term Rewriting

- $R$ is confluent iff for all terms $s, t_1, t_2$, if

  $$s \to_R^* t_1 \quad \text{and} \quad s \to_R^* t_2,$$

  then there exists a term $r$ such that

  $$t_1 \to_R^* r \quad \text{and} \quad t_2 \to_R^* r.$$

# Basic Concepts in Term Rewriting

- $R$ is confluent iff for all terms $s, t_1, t_2$, if

  $$s \to_R^* t_1 \quad \text{and} \quad s \to_R^* t_2,$$

  then there exists a term $r$ such that

  $$t_1 \to_R^* r \quad \text{and} \quad t_2 \to_R^* r.$$

  Graphically:

$$\begin{tikzcd} s \arrow[r, "*"] \arrow[d, "*"'] & t_1 \arrow[d, dashed, "*"] \\ t_2 \arrow[r, dashed, "*"'] & r \end{tikzcd}$$

# Basic Concepts in Term Rewriting

- $t_1$ and $t_2$ are joinable by $R$ if there exists a term $r$ such that

$$t_1 \to_R^* r \quad \text{and} \quad t_2 \to_R^* r.$$

- Notation: $t_1 \downarrow_R t_2$.

# Basic Concepts in Term Rewriting

## Example 1

Let $+$ be a binary (infix) function symbol, $s$ a unary function symbol, $0$ a constant.

$$R := \{0 + x \to x, \quad s(x) + y \to s(x + y)\}.$$

Then:

- $s(0) + s(s(0)) \to_R s(0 + s(s(0))) \to_R s(s(s(0)))$.
- $s(0) + s(s(0)) \to_R^* s(s(s(0)))$.
- $s(s(s(0)))$ is irreducible by $R$ and, hence, is a normal form of $s(0) + s(s(0))$, of $s(0 + s(s(0)))$, and of $s(s(s(0)))$.

# Basic Concepts in Term Rewriting

- A TRS $R$ is convergent iff it is confluent and terminating.
- A convergent TRS provides a decision procedure for the underlying equational theory: Two terms are equivalent iff they reduce to the same normal form.
- Computation of normal forms by repeated reduction is a don't care non-deterministic process for convergent TRSs.

# Basic Concepts in Term Rewriting

A strict order $>$ on terms is called a reduction order iff it is

1. monotonic: If $s > t$, then $r[s] > r[t]$ for all terms $s, t, r$;
2. stable: If $s > t$, then $s\sigma > t\sigma$ for all terms $s, t$ and a substitution $\sigma$;
3. well-founded.

# Basic Concepts in Term Rewriting

A strict order $>$ on terms is called a reduction order iff it is

1. monotonic: If $s > t$, then $r[s] > r[t]$ for all terms $s, t, r$;
2. stable: If $s > t$, then $s\sigma > t\sigma$ for all terms $s, t$ and a substitution $\sigma$;
3. well-founded.

Why are reduction orders interesting?

# Basic Concepts in Term Rewriting

A strict order $>$ on terms is called a reduction order iff it is

1. monotonic: If $s > t$, then $r[s] > r[t]$ for all terms $s, t, r$;
2. stable: If $s > t$, then $s\sigma > t\sigma$ for all terms $s, t$ and a substitution $\sigma$;
3. well-founded.

Why are reduction orders interesting?

**Theorem 1**
*A TRS $R$ terminates iff there exists a reduction order $>$ that satisfies $l > r$ for all $l \to r \in R$.*

# Reduction Orders

## Example 2

- $|t|$: The size of the term $t$.
- The order $>_1$: $s >_1 t$ iff $|s| > |t|$.

# Reduction Orders

## Example 2

- $|t|$: The size of the term $t$.
- The order $>_1$: $s >_1 t$ iff $|s| > |t|$.
- $>_1$ is monotonic and well-founded.

# Reduction Orders

## Example 2

- $|t|$: The size of the term $t$.
- The order $>_1$: $s >_1 t$ iff $|s| > |t|$.
- $>_1$ is monotonic and well-founded.
- However, $>_1$ is not a reduction order because it is not stable:

$$|f(f(x,x),y)| = 5 > 3 = |f(y,y)|$$

For $\sigma = \{y \mapsto f(x,x)\}$:

$$\begin{aligned} |\sigma(f(f(x,x),y))| &= |f(f(x,x),f(x,x))| = 7, \\ |\sigma(f(y,y)| &= |f(f(x,x),f(x,x))| = 7. \end{aligned}$$

# Reduction Orders

**Example 2 (Cont.)**

- $|t|_x$: The number of occurrences of $x$ in $t$.
- The order $>_2$: $s >_2 t$ iff $|s| > |t|$ and $|s|_x \geq |t|_x$ for all $x$.

# Reduction Orders

Example 2 (Cont.)

- $|t|_x$: The number of occurrences of $x$ in $t$.
- The order $>_2$: $s >_2 t$ iff $|s| > |t|$ and $|s|_x \geq |t|_x$ for all $x$.
- $>_2$ is a reduction order.

# Methods for Construction Reduction Orders

- Polynomial orders
- Simplification orders:
  - Recursive path orders
  - Knuth-Bendix orders

# Methods for Construction Reduction Orders

- Polynomial orders
- Simplification orders:
  - Recursive path orders
  - Knuth-Bendix orders

Goal: Provide a variety of different reduction orders that can be used to show termination; not only by hand, but also automatically.

# Lexicographic Path Order

Main idea behind recursive path orders:

- Two terms are compared by first comparing their root symbols.
- Then recursively comparing the collections of their immediate subterms.

# Lexicographic Path Order

Main idea behind recursive path orders:

- Two terms are compared by first comparing their root symbols.
- Then recursively comparing the collections of their immediate subterms.
- Collections seen as multisets yields the multiset path order. (Not considered in this course.)

# Lexicographic Path Order

Main idea behind recursive path orders:

- Two terms are compared by first comparing their root symbols.
- Then recursively comparing the collections of their immediate subterms.
- Collections seen as multisets yields the multiset path order. (Not considered in this course.)
- Collections seen as tuples yields the lexicographic path order.

# Lexicographic Path Order

Main idea behind recursive path orders:

- Two terms are compared by first comparing their root symbols.
- Then recursively comparing the collections of their immediate subterms.
- Collections seen as multisets yields the multiset path order. (Not considered in this course.)
- Collections seen as tuples yields the lexicographic path order.
- Combination of multisets and tuples yields the recursive path order with status. (Not considered in this course.)

# Lexicographic Path Order

### Definition 1

Let $\mathcal{F}$ be a finite signature and $>$ be a strict order on $\mathcal{F}$ (called the precedence). The lexicographic path order $>_{lpo}$on $T(\mathcal{F}, \mathcal{V})$ induced by $>$ is defined as follows:

$s >_{lpo} t$ iff

- (LPO1) $t \in \mathcal{V}ar(s)$ and $t \neq s$, or
- (LPO2) $s = f(s_1, \ldots, s_m)$, $t = g(t_1, \ldots, t_n)$, and
  - (LPO2a) $s_i \geq_{lpo} t$ for some $i$, $1 \leq i \leq m$, or
  - (LPO2b) $f > g$ and $s >_{lpo} t_j$ for all $j$, $1 \leq j \leq n$, or
  - (LPO2c) $f = g$, $s >_{lpo} t_j$ for all $j$, $1 \leq j \leq n$, and there exists $i$, $1 \leq i \leq m$ such that $s_1 = t_1, \ldots s_{i-1} = t_{i-1}$ and $s_i >_{lpo} t_i$.

$\geq_{lpo}$ stands for the reflexive closure of $>_{lpo}$.

# Lexicographic Path Order

$s >_{lpo} t$ iff

(LPO1) $t \in \mathcal{V}ar(s)$ and $t \neq s$, or

(LPO2) $s = f(s_1, \ldots, s_m)$, $t = g(t_1, \ldots, t_n)$, and

(LPO2a) $s_i \geq_{lpo} t$ for some $i$, $1 \leq i \leq m$, or

(LPO2b) $f > g$ and $s >_{lpo} t_j$ for all $j$, $1 \leq j \leq n$, or

(LPO2c) $f = g$, $s >_{lpo} t_j$ for all $j$, $1 \leq j \leq n$, and there exists $i$, $1 \leq i \leq m$ such that $s_1 = t_1, \ldots s_{i-1} = t_{i-1}$ and $s_i >_{lpo} t_i$.

## Example 3

$\mathcal{F} = \{f, i, e\}$, $f$ is binary, $i$ is unary, $e$ is constant, with $i > f > e$.

# Lexicographic Path Order

$s >_{lpo} t$ iff

(LPO1) $t \in \mathcal{V}ar(s)$ and $t \neq s$, or

(LPO2) $s = f(s_1, \ldots, s_m)$, $t = g(t_1, \ldots, t_n)$, and

(LPO2a) $s_i \geq_{lpo} t$ for some $i$, $1 \leq i \leq m$, or

(LPO2b) $f > g$ and $s >_{lpo} t_j$ for all $j$, $1 \leq j \leq n$, or

(LPO2c) $f = g$, $s >_{lpo} t_j$ for all $j$, $1 \leq j \leq n$, and there exists $i$, $1 \leq i \leq m$ such that $s_1 = t_1, \ldots s_{i-1} = t_{i-1}$ and $s_i >_{lpo} t_i$.

## Example 3

$\mathcal{F} = \{f, i, e\}$, $f$ is binary, $i$ is unary, $e$ is constant, with $i > f > e$.

- $f(x, e) >_{lpo} x$ by (LPO1)

# Lexicographic Path Order

$s >_{lpo} t$ iff

(LPO1) $t \in \mathcal{V}ar(s)$ and $t \neq s$, or

(LPO2) $s = f(s_1, \ldots, s_m)$, $t = g(t_1, \ldots, t_n)$, and

(LPO2a) $s_i \geq_{lpo} t$ for some $i$, $1 \leq i \leq m$, or

(LPO2b) $f > g$ and $s >_{lpo} t_j$ for all $j$, $1 \leq j \leq n$, or

(LPO2c) $f = g$, $s >_{lpo} t_j$ for all $j$, $1 \leq j \leq n$, and there exists $i$, $1 \leq i \leq m$ such that $s_1 = t_1, \ldots s_{i-1} = t_{i-1}$ and $s_i >_{lpo} t_i$.

## Example 3

$\mathcal{F} = \{f, i, e\}$, $f$ is binary, $i$ is unary, $e$ is constant, with $i > f > e$.

- $f(x, e) >_{lpo} x$ by (LPO1)
- $i(e) >_{lpo} e$ by (LPO2a), because $e \geq_{lpo} e$.

# Lexicographic Path Order

$s >_{lpo} t$ iff

(LPO1) $t \in \mathcal{V}ar(s)$ and $t \neq s$, or

(LPO2) $s = f(s_1, \ldots, s_m)$, $t = g(t_1, \ldots, t_n)$, and

- (LPO2a) $s_i \geq_{lpo} t$ for some $i$, $1 \leq i \leq m$, or
- (LPO2b) $f > g$ and $s >_{lpo} t_j$ for all $j$, $1 \leq j \leq n$, or
- (LPO2c) $f = g$, $s >_{lpo} t_j$ for all $j$, $1 \leq j \leq n$, and there exists $i$, $1 \leq i \leq m$ such that $s_1 = t_1, \ldots s_{i-1} = t_{i-1}$ and $s_i >_{lpo} t_i$.

## Example 3 (Cont.)

$\mathcal{F} = \{f, i, e\}$, $f$ is binary, $i$ is unary, $e$ is constant, with $i > f > e$.

# Lexicographic Path Order

$s >_{lpo} t$ iff

(LPO1) $t \in \mathcal{V}ar(s)$ and $t \neq s$, or

(LPO2) $s = f(s_1, \ldots, s_m)$, $t = g(t_1, \ldots, t_n)$, and

(LPO2a) $s_i \geq_{lpo} t$ for some $i$, $1 \leq i \leq m$, or

(LPO2b) $f > g$ and $s >_{lpo} t_j$ for all $j$, $1 \leq j \leq n$, or

(LPO2c) $f = g$, $s >_{lpo} t_j$ for all $j$, $1 \leq j \leq n$, and there exists $i$, $1 \leq i \leq m$ such that $s_1 = t_1, \ldots s_{i-1} = t_{i-1}$ and $s_i >_{lpo} t_i$.

## Example 3 (Cont.)

$\mathcal{F} = \{f, i, e\}$, $f$ is binary, $i$ is unary, $e$ is constant, with $i > f > e$.

- $i(f(x, y)) >^{?}_{lpo} f(i(x), i(y))$:

# Lexicographic Path Order

$s >_{lpo} t$ iff

(LPO1) $t \in \mathcal{V}ar(s)$ and $t \neq s$, or

(LPO2) $s = f(s_1, \ldots, s_m)$, $t = g(t_1, \ldots, t_n)$, and

(LPO2a) $s_i \geq_{lpo} t$ for some $i$, $1 \leq i \leq m$, or

(LPO2b) $f > g$ and $s >_{lpo} t_j$ for all $j$, $1 \leq j \leq n$, or

(LPO2c) $f = g$, $s >_{lpo} t_j$ for all $j$, $1 \leq j \leq n$, and there exists $i$, $1 \leq i \leq m$ such that $s_1 = t_1, \ldots s_{i-1} = t_{i-1}$ and $s_i >_{lpo} t_i$.

## Example 3 (Cont.)

$\mathcal{F} = \{f, i, e\}$, $f$ is binary, $i$ is unary, $e$ is constant, with $i > f > e$.

- $i(f(x,y)) >^?_{lpo} f(i(x), i(y))$:
  - Since $i > f$, (LPO2b) reduces it to the problems: $i(f(x,y)) >^?_{lpo} i(x)$ and $i(f(x,y)) >^?_{lpo} i(y)$.

# Lexicographic Path Order

$s >_{lpo} t$ iff

(LPO1) $t \in \mathcal{V}ar(s)$ and $t \neq s$, or

(LPO2) $s = f(s_1, \ldots, s_m)$, $t = g(t_1, \ldots, t_n)$, and

(LPO2a) $s_i \geq_{lpo} t$ for some $i$, $1 \leq i \leq m$, or

(LPO2b) $f > g$ and $s >_{lpo} t_j$ for all $j$, $1 \leq j \leq n$, or

(LPO2c) $f = g$, $s >_{lpo} t_j$ for all $j$, $1 \leq j \leq n$, and there exists $i$, $1 \leq i \leq m$ such that $s_1 = t_1, \ldots s_{i-1} = t_{i-1}$ and $s_i >_{lpo} t_i$.

## Example 3 (Cont.)

$\mathcal{F} = \{f, i, e\}$, $f$ is binary, $i$ is unary, $e$ is constant, with $i > f > e$.

- $i(f(x,y)) >^?_{lpo} f(i(x), i(y))$:
  - Since $i > f$, (LPO2b) reduces it to the problems: $i(f(x,y)) >^?_{lpo} i(x)$ and $i(f(x,y)) >^?_{lpo} i(y)$.
  - $i(f(x,y)) >^?_{lpo} i(x)$ is reduced by (LPO2c) to $i(f(x,y)) >^?_{lpo} x$ and $f(x,y) >^?_{lpo} x$, which hold by (LPO1).

# Lexicographic Path Order

$s >_{lpo} t$ iff

(LPO1) $t \in \mathcal{V}ar(s)$ and $t \neq s$, or

(LPO2) $s = f(s_1, \ldots, s_m)$, $t = g(t_1, \ldots, t_n)$, and

(LPO2a) $s_i \geq_{lpo} t$ for some $i$, $1 \leq i \leq m$, or

(LPO2b) $f > g$ and $s >_{lpo} t_j$ for all $j$, $1 \leq j \leq n$, or

(LPO2c) $f = g$, $s >_{lpo} t_j$ for all $j$, $1 \leq j \leq n$, and there exists $i$, $1 \leq i \leq m$ such that $s_1 = t_1, \ldots s_{i-1} = t_{i-1}$ and $s_i >_{lpo} t_i$.

## Example 3 (Cont.)

$\mathcal{F} = \{f, i, e\}$, $f$ is binary, $i$ is unary, $e$ is constant, with $i > f > e$.

- $i(f(x,y)) >^?_{lpo} f(i(x), i(y))$:
  - Since $i > f$, (LPO2b) reduces it to the problems: $i(f(x,y)) >^?_{lpo} i(x)$ and $i(f(x,y)) >^?_{lpo} i(y)$.
  - $i(f(x,y)) >^?_{lpo} i(x)$ is reduced by (LPO2c) to $i(f(x,y)) >^?_{lpo} x$ and $f(x,y) >^?_{lpo} x$, which hold by (LPO1).
  - $i(f(x,y)) >_{lpo} i(y)$ is shown similarly.

# Lexicographic Path Order

$s >_{lpo} t$ iff

(LPO1) $t \in \mathcal{V}ar(s)$ and $t \neq s$, or

(LPO2) $s = f(s_1, \ldots, s_m)$, $t = g(t_1, \ldots, t_n)$, and

- (LPO2a) $s_i \geq_{lpo} t$ for some $i$, $1 \leq i \leq m$, or
- (LPO2b) $f > g$ and $s >_{lpo} t_j$ for all $j$, $1 \leq j \leq n$, or
- (LPO2c) $f = g$, $s >_{lpo} t_j$ for all $j$, $1 \leq j \leq n$, and there exists $i$, $1 \leq i \leq m$ such that $s_1 = t_1, \ldots s_{i-1} = t_{i-1}$ and $s_i >_{lpo} t_i$.

## Example 3 (Cont.)

$\mathcal{F} = \{f, i, e\}$, $f$ is binary, $i$ is unary, $e$ is constant, with $i > f > e$.

# Lexicographic Path Order

$s >_{lpo} t$ iff

(LPO1) $t \in \mathcal{V}ar(s)$ and $t \neq s$, or

(LPO2) $s = f(s_1, \ldots, s_m)$, $t = g(t_1, \ldots, t_n)$, and

- (LPO2a) $s_i \geq_{lpo} t$ for some $i$, $1 \leq i \leq m$, or
- (LPO2b) $f > g$ and $s >_{lpo} t_j$ for all $j$, $1 \leq j \leq n$, or
- (LPO2c) $f = g$, $s >_{lpo} t_j$ for all $j$, $1 \leq j \leq n$, and there exists $i$, $1 \leq i \leq m$ such that $s_1 = t_1, \ldots s_{i-1} = t_{i-1}$ and $s_i >_{lpo} t_i$.

## Example 3 (Cont.)

$\mathcal{F} = \{f, i, e\}$, $f$ is binary, $i$ is unary, $e$ is constant, with $i > f > e$.

- $f(f(x, y), z) >^{?}_{lpo} f(x, f(y, z)))$. By (LPO2c) with $i = 1$:

# Lexicographic Path Order

$s >_{lpo} t$ iff

(LPO1) $t \in \mathcal{V}ar(s)$ and $t \neq s$, or

(LPO2) $s = f(s_1, \ldots, s_m)$, $t = g(t_1, \ldots, t_n)$, and

(LPO2a) $s_i \geq_{lpo} t$ for some $i$, $1 \leq i \leq m$, or

(LPO2b) $f > g$ and $s >_{lpo} t_j$ for all $j$, $1 \leq j \leq n$, or

(LPO2c) $f = g$, $s >_{lpo} t_j$ for all $j$, $1 \leq j \leq n$, and there exists $i$, $1 \leq i \leq m$ such that $s_1 = t_1, \ldots s_{i-1} = t_{i-1}$ and $s_i >_{lpo} t_i$.

## Example 3 (Cont.)

$\mathcal{F} = \{f, i, e\}$, $f$ is binary, $i$ is unary, $e$ is constant, with $i > f > e$.

- $f(f(x, y), z) >^?_{lpo} f(x, f(y, z)))$. By (LPO2c) with $i = 1$:
  - $f(f(x, y), z) >_{lpo} x$ because of (LPO1).

# Lexicographic Path Order

$s >_{lpo} t$ iff

(LPO1) $t \in \mathcal{V}ar(s)$ and $t \neq s$, or

(LPO2) $s = f(s_1, \ldots, s_m)$, $t = g(t_1, \ldots, t_n)$, and

- (LPO2a) $s_i \geq_{lpo} t$ for some $i$, $1 \leq i \leq m$, or
- (LPO2b) $f > g$ and $s >_{lpo} t_j$ for all $j$, $1 \leq j \leq n$, or
- (LPO2c) $f = g$, $s >_{lpo} t_j$ for all $j$, $1 \leq j \leq n$, and there exists $i$, $1 \leq i \leq m$ such that $s_1 = t_1, \ldots s_{i-1} = t_{i-1}$ and $s_i >_{lpo} t_i$.

## Example 3 (Cont.)

$\mathcal{F} = \{f, i, e\}$, $f$ is binary, $i$ is unary, $e$ is constant, with $i > f > e$.

- $f(f(x, y), z) >^{?}_{lpo} f(x, f(y, z)))$. By (LPO2c) with $i = 1$:
  - $f(f(x, y), z) >_{lpo} x$ because of (LPO1).
  - $f(f(x, y), z) >^{?}_{lpo} f(y, z)$: By (LPO2c) with $i = 1$:
    - $f(f(x, y), z) >_{lpo} y$ and $f(f(x, y), z) >_{lpo} z$ by (LPO1).
    - $f(x, y) >_{lpo} y$ by (LPO1).

# Lexicographic Path Order

$s >_{lpo} t$ iff

(LPO1) $t \in \mathcal{V}ar(s)$ and $t \neq s$, or

(LPO2) $s = f(s_1, \ldots, s_m)$, $t = g(t_1, \ldots, t_n)$, and

(LPO2a) $s_i \geq_{lpo} t$ for some $i$, $1 \leq i \leq m$, or

(LPO2b) $f > g$ and $s >_{lpo} t_j$ for all $j$, $1 \leq j \leq n$, or

(LPO2c) $f = g$, $s >_{lpo} t_j$ for all $j$, $1 \leq j \leq n$, and there exists $i$, $1 \leq i \leq m$ such that $s_1 = t_1, \ldots s_{i-1} = t_{i-1}$ and $s_i >_{lpo} t_i$.

## Example 3 (Cont.)

$\mathcal{F} = \{f, i, e\}$, $f$ is binary, $i$ is unary, $e$ is constant, with $i > f > e$.

- $f(f(x,y),z) >^?_{lpo} f(x, f(y,z)))$. By (LPO2c) with $i = 1$:
  - $f(f(x,y),z) >_{lpo} x$ because of (LPO1).
  - $f(f(x,y),z) >^?_{lpo} f(y,z)$: By (LPO2c) with $i = 1$:
    - $f(f(x,y),z) >_{lpo} y$ and $f(f(x,y),z) >_{lpo} z$ by (LPO1).
    - $f(x,y) >_{lpo} y$ by (LPO1).
  - $f(x,y) >_{lpo} x$ by (LPO1).

# Reduction Orders

- Reduction orders are not total for terms with variables.
- For instance, $f(x)$ and $f(y)$ can not be ordered.
- $f(x, y)$ and $f(y, x)$ can not be ordered either.
- However, many reduction orders are total on ground terms.
- Fortunately, in theorem proving applications one can often reason about non-ground formulas by considering the corresponding ground instances.
- In such situations, ordered rewriting techniques can be applied.

# Ordered Rewriting

- Given: A reduction order $>$ and a set of equations $E$.
- The rewrite system $E^>$ is defined as

$$E^> := \{s\sigma \to r\sigma \mid (s \approx t \in E \text{ or } t \approx s \in E) \text{ and } s\sigma > t\sigma\}$$

- The rewrite relation $\to_{E^>}$ induced by $E^>$ represents ordered rewriting with respect to $E$ and $>$.

# Ordered Rewriting

## Example 4

- If $>$ is a lexicographic path ordering with precedence $+ > a > b > c$, then $b + c > c + b > c$.
- Let $E := \{x + y \approx y + x\}$.
- We may use the commutativity equation for ordered rewriting.
- $(b + c) + c \rightarrow_{E^>} (c + b) + c \rightarrow_{E^>} c + (c + b)$.

# Ordered Rewriting

- If $>$ is a reduction ordering total on ground terms, then $E^>$ contains all (non-trivial) ground instances of an equation $s \approx t \in E$, either as a rule $s\sigma \to t\sigma$ or a rule $t\sigma \to s\sigma$.
- A rewrite system $R$ is called ground convergent if the induced ground rewrite relation (that is, the rewrite relation $\to_R$ restricted to pairs of ground terms) is terminating and confluent.
- A set of equations $E$ is called ground convergent with respect to $>$ if $E^>$ is ground convergent.

# Critical Pairs

Ordered rewriting leads to the inference rule, called superposition:

$$\frac{s \approx t \qquad r[u] \approx v}{(r[t] \approx v)\sigma},$$

where $\sigma = mgu(s, u)$, $t\sigma \not\succeq s\sigma$, $v\sigma \not\succeq r\sigma$, and $u$ is not a variable.

The equation $(r[t] \approx v)\sigma$ is called an ordered critical pair (with overlapped term $r[u]\sigma$) between $s \approx t$ and $r[u] \approx v$.

# Critical Pairs

Ordered rewriting leads to the inference rule, called superposition:

$$\frac{s \approx t \qquad r[u] \approx v}{(r[t] \approx v)\sigma},$$

where $\sigma = mgu(s, u)$, $t\sigma \not\succeq s\sigma$, $v\sigma \not\succeq r\sigma$, and $u$ is not a variable.

The equation $(r[t] \approx v)\sigma$ is called an ordered critical pair (with overlapped term $r[u]\sigma$) between $s \approx t$ and $r[u] \approx v$.

**Lemma 1**

*Let $>$ be a ground total reduction ordering. A set $E$ of equations is ground convergent with respect to $>$ iff for all ordered critical pairs $(r[t] \approx v)\sigma$ (with overlapped term $r[u]\sigma$) between equations in $E$ and for all ground substitutions $\varphi$, if $r[u]\sigma\varphi > r[t]\sigma\varphi$ and $r[u]\sigma\varphi > v\sigma\varphi$, then $r[t]\sigma\varphi \downarrow_{E^>} v\sigma\varphi$.*

# Critical Pairs

## Example 5

- Let $E := \{f(f(x)) \approx g(x)\}$ and $>$ be the LPO with $f > g$.
- Take a critical pair between the equation and its renamed copy, $f(f(x)) \approx g(x)$ and $f(f(y)) \approx g(y)$.

$$\begin{tikzcd}
 & f(f(f(x))) \arrow[dl] \arrow[dr] & \\
f(g(x)) & & g(f(x))
\end{tikzcd}$$

# Critical Pairs

## Example 5

- Let $E := \{f(f(x)) \approx g(x)\}$ and $>$ be the LPO with $f > g$.
- Take a critical pair between the equation and its renamed copy, $f(f(x)) \approx g(x)$ and $f(f(y)) \approx g(y)$.

$$\begin{tikzcd}
 & f(f(f(x))) \arrow[dl] \arrow[dr] & \\
f(g(x)) & & g(f(x))
\end{tikzcd}$$

- $f(f(f(x))) > f(g(x))$ and $f(f(f(x))) > g(f(x))$, but $f(g(x)) \not\downarrow_{E>} g(f(x))$.

# Critical Pairs

## Example 5

- Let $E := \{f(f(x)) \approx g(x)\}$ and $>$ be the LPO with $f > g$.
- Take a critical pair between the equation and its renamed copy, $f(f(x)) \approx g(x)$ and $f(f(y)) \approx g(y)$.

$$
\begin{tikzcd}
 & f(f(f(x))) \arrow[dl] \arrow[dr] & \\
f(g(x)) & & g(f(x))
\end{tikzcd}
$$

- $f(f(f(x))) > f(g(x))$ and $f(f(f(x))) > g(f(x))$, but $f(g(x)) \not\downarrow_{E^{>}} g(f(x))$.
- $E$ is not ground convergent with respect to $>$.

# Adding Critical Pairs to Equations

- Since critical pairs are equational consequences, adding a critical pair to the set of equations does not change the induced equational theory.
- If $E'$ is obtained from $E$ by adding a critical pair, then $\approx_E = \approx_{E'}$.
- The idea of adding a critical pair as a new equation is called "completion".

# Convergence

## Example 6

- Let $E' := \{f(f(x)) \approx g(x), f(g(x)) \approx g(f(x))\}$
- Let $>$ be the LPO with $f > g$.

# Convergence

## Example 6

- Let $E' := \{f(f(x)) \approx g(x), f(g(x)) \approx g(f(x))\}$
- Let $>$ be the LPO with $f > g$.
- $E'$ has two critical pairs. Both are joinable:

$$\begin{tikzcd}
 & f(f(f(x))) \arrow[dl] \arrow[dr] & \\
f(g(x)) \arrow[rr] & & g(f(x))
\end{tikzcd}
\qquad
\begin{tikzcd}
 & f(f(g(x))) \arrow[dl] \arrow[ddrr] & & \\
f(g(f(x))) \arrow[dr] & & & \\
 & g(f(f(x))) \arrow[rr] & & g(g(x))
\end{tikzcd}$$

# Convergence

## Example 6

- Let $E' := \{f(f(x)) \approx g(x), f(g(x)) \approx g(f(x))\}$
- Let $>$ be the LPO with $f > g$.
- $E'$ has two critical pairs. Both are joinable:

$$
\begin{tikzcd}
 & f(f(f(x))) \arrow[dl] \arrow[dr] & \\
f(g(x)) \arrow[rr] & & g(f(x))
\end{tikzcd}
\qquad
\begin{tikzcd}
 & f(f(g(x))) \arrow[dl] \arrow[ddr] & \\
f(g(f(x))) \arrow[dr] & & \\
 & g(f(f(x))) \arrow[r] & g(g(x))
\end{tikzcd}
$$

- $E'$ is (ground) convergent.

# Ordered Completion

- Described as a set of inference rules.
- Parametrized by a reduction ordering $>$.
- Works on pairs $(E, R)$, where $E$ is a set of equations and $R$ is a set of rewrite rules.
- $E; R \vdash E'; R'$ means that $E'; R'$ can be obtained from $E; R$ by applying a completion inference.

# Ordered Completion: Notions

- Derivation: A (finite or countably infinite) sequence $(E_0; R_0) \vdash (E_1; R_1) \cdots$.
- Usually, $E_0$ is the set of initial equations and $R_0 = \emptyset$.
- The limit of a derivation: the pair $E_\omega; R_\omega$, where

$$E_\omega := \bigcup_{i \geq 0} \bigcap_{j \geq i} E_j \text{ and } R_\omega := \bigcup_{i \geq 0} \bigcap_{j \geq i} R_j.$$

- Goal: to obtain a limit system that is ground convergent.

# Ordered Completion: Notation

- $\uplus$: Disjoint union
- $s \rhd t$: Strict encompassment relation. An instance of $t$ is a subterm of $s$, but not vice versa.
- $s \approxeq t$ stands for $s \approx t$ or $t \approx s$.
- $CP_>(E \cup R)$: The set of all ordered critical pairs, with the ordering $>$, generated by equations in $E$ and rewrite rules in $R$ treated as equations.

# Ordered Completion: Rules

| | |
|---|---|
| DEDUCTION: | $E; R \vdash E \cup \{s \approx t\}; R$ <br> if $s \approx t \in CP_{>}(E \cup R)$. |
| ORIENTATION: | $E \uplus \{s \,\dot{\approx}\, t\}; R \vdash E; R \cup \{s \to t\}$, if $s > t$. |
| DELETION: | $E \uplus \{s \approx s\}; R \vdash E; R$. |
| COMPOSITION: | $E; R \uplus \{s \to t\} \vdash E; R \cup \{s \to r\}$, <br> if $t \to_{R \cup E^{>}} r$. |

# Ordered Completion: Rules

SIMPLIFICATION: $E \cup \{s \approx t\}; R \vdash E \cup \{u \approx t\}; R,$
if $s \to_R u$ or $s \to_{E^>} u$ with $l\sigma \to r\sigma$
for $l \approx r \in E, s \rhd l$.

COLLAPSE: $E; R \uplus \{s \to t\} \vdash E \cup \{u \approx t\}; R,$
if $s \to_R u$ or $s \to_{E^>} u$ with $l\sigma \to r\sigma$
for $l \approx r \in E, s \rhd l$.

# Ordered Completion: Properties

### Theorem 2

*Let* $(E_0; R_0), (E_1; R_1), \ldots$ *be an ordered completion derivation where all critical pairs are eventually generated (a fair derivation). Then these three properties are equivalent for all ground terms* $s$ *and* $t$*:*

(1) $E_0 \models s \approx t$.

(2) $s \downarrow_{E_\omega^> \cup R_\omega} t$.

(3) $s \downarrow_{E_i^> \cup R_i} t$ *for some* $i \geq 0$.

This theorem, in particular, asserts the refutational completeness of ordered completion.

# Proving by Ordered Completion: Example

Given:

1. $(x \cdot y) \cdot z \approx x \cdot (y \cdot z)$.
2. $x \cdot e \approx x$.
3. $x \cdot i(x) \approx e$.
4. $x \cdot x \approx e$.

Prove

Goal: $x \cdot y \approx y \cdot x$.

# Proving by Ordered Completion: Example

Proof by ordered completion:

- Skolemize the goal: $a \cdot b \approx b \cdot a$.
- Take LPO as the reduction ordering with the precedence $i > f > e > a > b$
- $E_0 := \{(x \cdot y) \cdot z \approx x \cdot (y \cdot z),\ x \cdot e \approx x,\ x \cdot i(x) \approx e,\ x \cdot x \approx e\}$
- $R_0 := \emptyset$
- Start applying the rules.

# Proving by Ordered Completion: Example

$$E_0 = \{(x \cdot y) \cdot z \approx x \cdot (y \cdot z),\ x \cdot e \approx x,\ x \cdot i(x) \approx e,\ x \cdot x \approx e\}$$
$$R_0 = \emptyset$$

Apply Orient 4 times:

$$E_4 = \emptyset$$
$$R_4 = \{(x \cdot y) \cdot z \to x \cdot (y \cdot z),\ x \cdot e \to x,\ x \cdot i(x) \to e,\ x \cdot x \to e\}$$

# Proving by Ordered Completion: Example

$$E_0 = \{(x \cdot y) \cdot z \approx x \cdot (y \cdot z),\ x \cdot e \approx x,\ x \cdot i(x) \approx e,\ x \cdot x \approx e\}$$
$$R_0 = \emptyset$$

Apply Orient 4 times:

$$E_4 = \emptyset$$
$$R_4 = \{(x \cdot y) \cdot z \to x \cdot (y \cdot z),\ x \cdot e \to x,\ x \cdot i(x) \to e,\ x \cdot x \to e\}$$

Apply Deduce with the rules $(x \cdot y) \cdot z \to x \cdot (y \cdot z)$ and $x \cdot e \to x$ to the overlapping term $(x \cdot e) \cdot z$, and then Orient:

$$E_6 = \emptyset$$
$$R_6 = \{(x \cdot y) \cdot z \to x \cdot (y \cdot z),\ x \cdot e \to x,\ x \cdot i(x) \to e,\ x \cdot x \to e,\ x_1 \cdot (e \cdot x_2) \to x_1 \cdot x_2\}$$

# Proving by Ordered Completion: Example

$$
\begin{aligned}
E_6 &= \emptyset \\
R_6 &= \{(x \cdot y) \cdot z \to x \cdot (y \cdot z),\ x \cdot e \to x,\ x \cdot i(x) \to e,\ x \cdot x \to e, \\
&\qquad x_1 \cdot (e \cdot x_2) \to x_1 \cdot x_2\}
\end{aligned}
$$

Apply $\textsc{Deduce}$ with the rules $x_1 \cdot (e \cdot x_2) \to x_1 \cdot x_2$ and $x \cdot i(x) \to e$ to the overlapping term $x_1 \cdot (e \cdot i(e))$:

$$
\begin{aligned}
E_7 &= \{x_1 \cdot i(e) \approx x_1 \cdot e\} \\
R_7 &= \{(x \cdot y) \cdot z \to x \cdot (y \cdot z),\ x \cdot e \to x,\ x \cdot i(x) \to e,\ x \cdot x \to e, \\
&\qquad x_1 \cdot (e \cdot x_2) \to x_1 \cdot x_2\}
\end{aligned}
$$

# Proving by Ordered Completion: Example

$$E_7 = \{x_1 \cdot i(e) \approx x_1 \cdot e\}$$
$$R_7 = \{(x \cdot y) \cdot z \to x \cdot (y \cdot z),\ x \cdot e \to x,\ x \cdot i(x) \to e,\ x \cdot x \to e,$$
$$x_1 \cdot (e \cdot x_2) \to x_1 \cdot x_2\}$$

Apply ORIENT to $x_1 \cdot i(e) \approx x_1 \cdot e$ and then COMPOSITION with the rule $x \cdot e \to x$:

$$E_9 = \emptyset$$
$$R_9 = \{(x \cdot y) \cdot z \to x \cdot (y \cdot z),\ x \cdot e \to x,\ x \cdot i(x) \to e,\ x \cdot x \to e,$$
$$x_1 \cdot (e \cdot x_2) \to x_1 \cdot x_2,\ x \cdot i(e) \to x\}$$

# Proving by Ordered Completion: Example

$$
\begin{aligned}
E_9 &= \emptyset \\
R_9 &= \{(x \cdot y) \cdot z \to x \cdot (y \cdot z),\ x \cdot e \to x,\ x \cdot i(x) \to e,\ x \cdot x \to e, \\
&\qquad x_1 \cdot (e \cdot x_2) \to x_1 \cdot x_2,\ x \cdot i(e) \to x\}
\end{aligned}
$$

Apply DEDUCE with the rules $x \cdot x \to e$ and $x \cdot i(e) \to x$ to the overlapping term $i(e) \cdot i(e)$, and then ORIENT:

$$
\begin{aligned}
E_{11} &= \emptyset \\
R_{11} &= \{(x \cdot y) \cdot z \to x \cdot (y \cdot z),\ x \cdot e \to x,\ x \cdot i(x) \to e,\ x \cdot x \to e, \\
&\qquad x_1 \cdot (e \cdot x_2) \to x_1 \cdot x_2,\ x \cdot i(e) \to x,\ i(e) \to e\}
\end{aligned}
$$

# Proving by Ordered Completion: Example

$$E_{11} = \emptyset$$

$$R_{11} = \{(x \cdot y) \cdot z \to x \cdot (y \cdot z),\ x \cdot e \to x,\ x \cdot i(x) \to e,\ x \cdot x \to e,\\ x_1 \cdot (e \cdot x_2) \to x_1 \cdot x_2,\ x \cdot i(e) \to x,\ i(e) \to e\}$$

Apply Collapse to $x \cdot i(e) \to x$ with $i(e) \to e$:

$$E_{12} = \{x \cdot e \approx x\}$$

$$R_{12} = \{(x \cdot y) \cdot z \to x \cdot (y \cdot z),\ x \cdot e \to x,\ x \cdot i(x) \to e,\ x \cdot x \to e,\\ x_1 \cdot (e \cdot x_2) \to x_1 \cdot x_2,\ i(e) \to e\}$$

# Proving by Ordered Completion: Example

$$E_{12} = \{x \cdot e \approx x\}$$

$$R_{12} = \{(x \cdot y) \cdot z \to x \cdot (y \cdot z),\ x \cdot e \to x,\ x \cdot i(x) \to e,\ x \cdot x \to e,$$
$$x_1 \cdot (e \cdot x_2) \to x_1 \cdot x_2,\ i(e) \to e\}$$

Apply SIMPLIFICATION to $x \cdot e \approx x$ with $x \cdot e \to x$ and then DELETE to the obtained $x \approx x$:

$$E_{14} = \emptyset$$

$$R_{14} = \{(x \cdot y) \cdot z \to x \cdot (y \cdot z),\ x \cdot e \to x,\ x \cdot i(x) \to e,\ x \cdot x \to e,$$
$$x_1 \cdot (e \cdot x_2) \to x_1 \cdot x_2,\ i(e) \to e\}$$

# Proving by Ordered Completion: Example

$$E_{14} = \emptyset$$

$$R_{14} = \{(x \cdot y) \cdot z \to x \cdot (y \cdot z),\ x \cdot e \to x,\ x \cdot i(x) \to e,\ x \cdot x \to e,\ x_1 \cdot (e \cdot x_2) \to x_1 \cdot x_2,\ i(e) \to e\}$$

Apply DEDUCE to $(x \cdot y) \cdot z \to x \cdot (y \cdot z)$ and $x \cdot i(x) \to e$ with the overlapping term $(x \cdot i(x)) \cdot z$ and then ORIENT:

$$E_{16} = \emptyset$$

$$R_{16} = \{(x \cdot y) \cdot z \to x \cdot (y \cdot z),\ x \cdot e \to x,\ x \cdot i(x) \to e,\ x \cdot x \to e,\ x_1 \cdot (e \cdot x_2) \to x_1 \cdot x_2,\ i(e) \to e,\ x_1 \cdot (i(x_1) \cdot x_2) \to e \cdot x_2\}$$

# Proving by Ordered Completion: Example

$$E_{16} = \emptyset$$
$$R_{16} = \{(x \cdot y) \cdot z \to x \cdot (y \cdot z),\ x \cdot e \to x,\ x \cdot i(x) \to e,\ x \cdot x \to e,\ x_1 \cdot (e \cdot x_2) \to x_1 \cdot x_2,\ i(e) \to e,\ x_1 \cdot (i(x_1) \cdot x_2) \to e \cdot x_2\}$$

Apply DEDUCE to $x_1 \cdot (i(x_1) \cdot x_2) \to e \cdot x_2$ and $x \cdot x \to e$ with the overlapping term $x_1 \cdot (i(x_1) \cdot i(x_1))$:

$$E_{17} = \{e \cdot i(x) \approx x \cdot e\}$$
$$R_{17} = \{(x \cdot y) \cdot z \to x \cdot (y \cdot z),\ x \cdot e \to x,\ x \cdot i(x) \to e,\ x \cdot x \to e,\ x_1 \cdot (e \cdot x_2) \to x_1 \cdot x_2,\ i(e) \to e,\ x_1 \cdot (i(x_1) \cdot x_2) \to e \cdot x_2\}$$

# Proving by Ordered Completion: Example

$$E_{17} = \{e \cdot i(x) \approx x \cdot e\}$$

$$R_{17} = \{(x \cdot y) \cdot z \to x \cdot (y \cdot z),\ x \cdot e \to x,\ x \cdot i(x) \to e,\ x \cdot x \to e,$$
$$x_1 \cdot (e \cdot x_2) \to x_1 \cdot x_2,\ i(e) \to e,\ x_1 \cdot (i(x_1) \cdot x_2) \to e \cdot x_2\}$$

Apply Simplification to $e \cdot i(x) \approx x \cdot e$ with $x \cdot e \to x$ and then Orient:

$$E_{19} = \emptyset$$

$$R_{19} = \{(x \cdot y) \cdot z \to x \cdot (y \cdot z),\ x \cdot e \to x,\ x \cdot i(x) \to e,\ x \cdot x \to e,$$
$$x_1 \cdot (e \cdot x_2) \to x_1 \cdot x_2,\ i(e) \to e,\ x_1 \cdot (i(x_1) \cdot x_2) \to e \cdot x_2,$$
$$e \cdot i(x) \to x\}$$

# Proving by Ordered Completion: Example

$$E_{19} = \emptyset$$

$$\begin{aligned} R_{19} = \{&(x \cdot y) \cdot z \to x \cdot (y \cdot z),\ x \cdot e \to x,\ x \cdot i(x) \to e,\ x \cdot x \to e, \\ &x_1 \cdot (e \cdot x_2) \to x_1 \cdot x_2,\ i(e) \to e,\ x_1 \cdot (i(x_1) \cdot x_2) \to e \cdot x_2, \\ &e \cdot i(x) \to x\} \end{aligned}$$

Apply DEDUCE to $x_1 \cdot (e \cdot x_2) \to x_1 \cdot x_2$ and $e \cdot i(x) \to x$ with the overlapping term $x_1 \cdot (e \cdot i(x_2))$ and then ORIENT:

$$E_{21} = \emptyset$$

$$\begin{aligned} R_{21} = \{&(x \cdot y) \cdot z \to x \cdot (y \cdot z),\ x \cdot e \to x,\ x \cdot i(x) \to e,\ x \cdot x \to e, \\ &x_1 \cdot (e \cdot x_2) \to x_1 \cdot x_2,\ i(e) \to e,\ x_1 \cdot (i(x_1) \cdot x_2) \to e \cdot x_2, \\ &e \cdot i(x) \to x,\ x_1 \cdot i(x_2) \to x_1 \cdot x_2\} \end{aligned}$$

# Proving by Ordered Completion: Example

$$E_{21} = \emptyset$$

$$\begin{aligned} R_{21} = \{ & (x \cdot y) \cdot z \to x \cdot (y \cdot z),\ x \cdot e \to x,\ x \cdot i(x) \to e,\ x \cdot x \to e, \\ & x_1 \cdot (e \cdot x_2) \to x_1 \cdot x_2,\ i(e) \to e,\ x_1 \cdot (i(x_1) \cdot x_2) \to e \cdot x_2, \\ & e \cdot i(x) \to x,\ x_1 \cdot i(x_2) \to x_1 \cdot x_2 \} \end{aligned}$$

Applying Collapse, Simplification, and Delete, we get rid of $x \cdot i(x) \to e$:

$$E_{24} = \emptyset$$

$$\begin{aligned} R_{24} = \{ & (x \cdot y) \cdot z \to x \cdot (y \cdot z),\ x \cdot e \to x,\ x \cdot x \to e, \\ & x_1 \cdot (e \cdot x_2) \to x_1 \cdot x_2,\ i(e) \to e,\ x_1 \cdot (i(x_1) \cdot x_2) \to e \cdot x_2, \\ & e \cdot i(x) \to x,\ x_1 \cdot i(x_2) \to x_1 \cdot x_2 \} \end{aligned}$$

# Proving by Ordered Completion: Example

$$E_{24} = \emptyset$$
$$R_{24} = \{(x \cdot y) \cdot z \to x \cdot (y \cdot z),\ x \cdot e \to x,\ x \cdot x \to e,$$
$$x_1 \cdot (e \cdot x_2) \to x_1 \cdot x_2,\ i(e) \to e,\ x_1 \cdot (i(x_1) \cdot x_2) \to e \cdot x_2,$$
$$e \cdot i(x) \to x,\ x_1 \cdot i(x_2) \to x_1 \cdot x_2\}$$

Applying COLLAPSE and ORIENT, we replace $e \cdot i(x) \to x$ with $e \cdot x \to x$:

$$E_{26} = \emptyset$$
$$R_{26} = \{(x \cdot y) \cdot z \to x \cdot (y \cdot z),\ x \cdot e \to x,\ x \cdot x \to e,$$
$$x_1 \cdot (e \cdot x_2) \to x_1 \cdot x_2,\ i(e) \to e,\ x_1 \cdot (i(x_1) \cdot x_2) \to e \cdot x_2,$$
$$e \cdot x \to x,\ x_1 \cdot i(x_2) \to x_1 \cdot x_2\}$$

# Proving by Ordered Completion: Example

$$E_{26} = \emptyset$$
$$R_{26} = \{(x \cdot y) \cdot z \to x \cdot (y \cdot z),\ x \cdot e \to x,\ x \cdot x \to e,\ x_1 \cdot (e \cdot x_2) \to x_1 \cdot x_2,\ i(e) \to e,\ x_1 \cdot (i(x_1) \cdot x_2) \to e \cdot x_2,\ e \cdot x \to x,\ x_1 \cdot i(x_2) \to x_1 \cdot x_2\}$$

Applying COLLAPSE and DELETE, we get rid of $x_1 \cdot (e \cdot x_2) \to x_1 \cdot x_2$:

$$E_{28} = \emptyset$$
$$R_{28} = \{(x \cdot y) \cdot z \to x \cdot (y \cdot z),\ x \cdot e \to x,\ x \cdot x \to e,\ i(e) \to e,\ x_1 \cdot (i(x_1) \cdot x_2) \to e \cdot x_2,\ e \cdot x \to x,\ x_1 \cdot i(x_2) \to x_1 \cdot x_2\}$$

# Proving by Ordered Completion: Example

$$
\begin{aligned}
E_{28} &= \emptyset \\
R_{28} &= \{(x \cdot y) \cdot z \to x \cdot (y \cdot z),\ x \cdot e \to x,\ x \cdot x \to e, \\
&\qquad i(e) \to e,\ x_1 \cdot (i(x_1) \cdot x_2) \to e \cdot x_2, \\
&\qquad e \cdot x \to x,\ x_1 \cdot i(x_2) \to x_1 \cdot x_2\}
\end{aligned}
$$

Apply $\textsc{Deduce}$ to $e \cdot x \to x$ and $x_1 \cdot i(x_2) \to x_1 \cdot x_2$ with the overlapping term $e \cdot i(x_2)$:

$$
\begin{aligned}
E_{29} &= \{i(x_1) \approx e \cdot x_2\} \\
R_{29} &= \{(x \cdot y) \cdot z \to x \cdot (y \cdot z),\ x \cdot e \to x,\ x \cdot x \to e, \\
&\qquad i(e) \to e,\ x_1 \cdot (i(x_1) \cdot x_2) \to e \cdot x_2, \\
&\qquad e \cdot x \to x,\ x_1 \cdot i(x_2) \to x_1 \cdot x_2\}
\end{aligned}
$$

# Proving by Ordered Completion: Example

$$
\begin{aligned}
E_{29} &= \{i(x_2) \approx e \cdot x_2\} \\
R_{29} &= \{(x \cdot y) \cdot z \to x \cdot (y \cdot z),\ x \cdot e \to x,\ x \cdot x \to e, \\
&\qquad i(e) \to e,\ x_1 \cdot (i(x_1) \cdot x_2) \to e \cdot x_2, \\
&\qquad e \cdot x \to x,\ x_1 \cdot i(x_2) \to x_1 \cdot x_2\}
\end{aligned}
$$

Apply SIMPLIFICATION to $i(x_1) \approx e \cdot x_2$ with $e \cdot x \to x$ and then ORIENT:

$$
\begin{aligned}
E_{31} &= \emptyset \\
R_{31} &= \{(x \cdot y) \cdot z \to x \cdot (y \cdot z),\ x \cdot e \to x,\ x \cdot x \to e, \\
&\qquad i(e) \to e,\ x_1 \cdot (i(x_1) \cdot x_2) \to e \cdot x_2, \\
&\qquad e \cdot x \to x,\ x_1 \cdot i(x_2) \to x_1 \cdot x_2,\ i(x) \to x\}
\end{aligned}
$$

# Proving by Ordered Completion: Example

$$
\begin{aligned}
E_{31} &= \emptyset \\
R_{31} &= \{(x \cdot y) \cdot z \to x \cdot (y \cdot z),\ x \cdot e \to x,\ x \cdot x \to e, \\
&\qquad i(e) \to e,\ x_1 \cdot (i(x_1) \cdot x_2) \to e \cdot x_2, \\
&\qquad e \cdot x \to x,\ x_1 \cdot i(x_2) \to x_1 \cdot x_2,\ i(x) \to x\}
\end{aligned}
$$

Apply Collapse and Delete, we get rid of $i(e) \to e$:

$$
\begin{aligned}
E_{33} &= \emptyset \\
R_{33} &= \{(x \cdot y) \cdot z \to x \cdot (y \cdot z),\ x \cdot e \to x,\ x \cdot x \to e, \\
&\qquad x_1 \cdot (i(x_1) \cdot x_2) \to e \cdot x_2,\ e \cdot x \to x, \\
&\qquad x_1 \cdot i(x_2) \to x_1 \cdot x_2,\ i(x) \to x\}
\end{aligned}
$$

# Proving by Ordered Completion: Example

$$
\begin{aligned}
E_{33} &= \emptyset \\
R_{33} &= \{(x \cdot y) \cdot z \to x \cdot (y \cdot z),\ x \cdot e \to x,\ x \cdot x \to e, \\
&\qquad x_1 \cdot (i(x_1) \cdot x_2) \to e \cdot x_2,\ e \cdot x \to x, \\
&\qquad x_1 \cdot i(x_2) \to x_1 \cdot x_2,\ i(x) \to x\}
\end{aligned}
$$

Applying COMPOSITION, we replace $x_1 \cdot (i(x_1) \cdot x_2) \to e \cdot x_2$ by $x_1 \cdot (i(x_1) \cdot x_2) \to x_2$:

$$
\begin{aligned}
E_{34} &= \emptyset \\
R_{34} &= \{(x \cdot y) \cdot z \to x \cdot (y \cdot z),\ x \cdot e \to x,\ x \cdot x \to e, \\
&\qquad x_1 \cdot (i(x_1) \cdot x_2) \to x_2,\ e \cdot x \to x, \\
&\qquad x_1 \cdot i(x_2) \to x_1 \cdot x_2,\ i(x) \to x\}
\end{aligned}
$$

# Proving by Ordered Completion: Example

$$
\begin{aligned}
E_{34} &= \emptyset \\
R_{34} &= \{(x \cdot y) \cdot z \to x \cdot (y \cdot z),\ x \cdot e \to x,\ x \cdot x \to e, \\
&\qquad x_1 \cdot (i(x_1) \cdot x_2) \to x_2,\ e \cdot x \to x, \\
&\qquad x_1 \cdot i(x_2) \to x_1 \cdot x_2,\ i(x) \to x\}
\end{aligned}
$$

Applying Simplification and Orient, we replace $x_1 \cdot (i(x_1) \cdot x_2) \to x_2$ by $x_1 \cdot (x_1 \cdot x_2) \to x_2$:

$$
\begin{aligned}
E_{36} &= \emptyset \\
R_{36} &= \{(x \cdot y) \cdot z \to x \cdot (y \cdot z),\ x \cdot e \to x,\ x \cdot x \to e, \\
&\qquad x_1 \cdot (x_1 \cdot x_2) \to x_2,\ e \cdot x \to x, \\
&\qquad x_1 \cdot i(x_2) \to x_1 \cdot x_2,\ i(x) \to x\}
\end{aligned}
$$

# Proving by Ordered Completion: Example

$$
\begin{aligned}
E_{36} &= \emptyset \\
R_{36} &= \{(x \cdot y) \cdot z \to x \cdot (y \cdot z),\ x \cdot e \to x,\ x \cdot x \to e, \\
&\qquad x_1 \cdot (i(x_1) \cdot x_2) \to x_2,\ e \cdot x \to x, \\
&\qquad x_1 \cdot i(x_2) \to x_1 \cdot x_2,\ i(x) \to x\}
\end{aligned}
$$

Apply Deduce to $(x \cdot y) \cdot z \to x \cdot (y \cdot z)$ and $x \cdot x \to e$ with the overlapping term $(x_1 \cdot x_2) \cdot (x_1 \cdot x_2)$, then Orient:

$$
\begin{aligned}
E_{37} &= \emptyset \\
R_{37} &= \{(x \cdot y) \cdot z \to x \cdot (y \cdot z),\ x \cdot e \to x,\ x \cdot x \to e, \\
&\qquad x_1 \cdot (x_1 \cdot x_2) \to x_2,\ e \cdot x \to x,\ x_1 \cdot i(x_2) \to x_1 \cdot x_2, \\
&\qquad i(x) \to x,\ x_1 \cdot (x_2 \cdot (x_1 \cdot x_2)) \to e\}
\end{aligned}
$$

# Proving by Ordered Completion: Example

$$E_{37} = \emptyset$$

$$\begin{aligned} R_{37} = \{ & (x \cdot y) \cdot z \to x \cdot (y \cdot z),\ x \cdot e \to x,\ x \cdot x \to e, \\ & x_1 \cdot (x_1 \cdot x_2) \to x_2,\ e \cdot x \to x,\ x_1 \cdot i(x_2) \to x_1 \cdot x_2, \\ & i(x) \to x,\ x_1 \cdot (x_2 \cdot (x_1 \cdot x_2)) \to e \} \end{aligned}$$

Apply DEDUCE to $x_1 \cdot (x_1 \cdot x_2) \to x_2$ and $x_1 \cdot (x_2 \cdot (x_1 \cdot x_2)) \to e$ with the overlapping term $x_1 \cdot (x_1 \cdot (x_2 \cdot (x_1 \cdot x_2)))$, then ORIENT:

$$E_{39} = \emptyset$$

$$\begin{aligned} R_{39} = \{ & (x \cdot y) \cdot z \to x \cdot (y \cdot z),\ x \cdot e \to x,\ x \cdot x \to e, \\ & x_1 \cdot (x_1 \cdot x_2) \to x_2,\ e \cdot x \to x,\ x_1 \cdot i(x_2) \to x_1 \cdot x_2, \\ & i(x) \to x,\ x_1 \cdot (x_2 \cdot (x_1 \cdot x_2)) \to e,\ x_2 \cdot (x_1 \cdot x_2) \to x_1 \cdot e \} \end{aligned}$$

# Proving by Ordered Completion: Example

$$E_{39} = \emptyset$$

$$\begin{aligned}R_{39} = \{&(x \cdot y) \cdot z \to x \cdot (y \cdot z),\ x \cdot e \to x,\ x \cdot x \to e,\\ &x_1 \cdot (x_1 \cdot x_2) \to x_2,\ e \cdot x \to x,\ x_1 \cdot i(x_2) \to x_1 \cdot x_2,\\ &i(x) \to x,\ x_1 \cdot (x_2 \cdot (x_1 \cdot x_2)) \to e,\ x_2 \cdot (x_1 \cdot x_2) \to x_1 \cdot e\}\end{aligned}$$

Apply Composition to $x_2 \cdot (x_1 \cdot x_2) \to x_1 \cdot e$ with $x \cdot e \to x$:

$$E_{40} = \emptyset$$

$$\begin{aligned}R_{40} = \{&(x \cdot y) \cdot z \to x \cdot (y \cdot z),\ x \cdot e \to x,\ x \cdot x \to e,\\ &x_1 \cdot (x_1 \cdot x_2) \to x_2,\ e \cdot x \to x,\ x_1 \cdot i(x_2) \to x_1 \cdot x_2,\\ &i(x) \to x,\ x_1 \cdot (x_2 \cdot (x_1 \cdot x_2)) \to e,\ x_2 \cdot (x_1 \cdot x_2) \to x_1\}\end{aligned}$$

# Proving by Ordered Completion: Example

$$
\begin{aligned}
E_{41} &= \emptyset \\
R_{41} &= \{(x \cdot y) \cdot z \to x \cdot (y \cdot z),\ x \cdot e \to x,\ x \cdot x \to e, \\
&\qquad x_1 \cdot (x_1 \cdot x_2) \to x_2,\ e \cdot x \to x,\ x_1 \cdot i(x_2) \to x_1 \cdot x_2, \\
&\qquad i(x) \to x,\ x_1 \cdot (x_2 \cdot (x_1 \cdot x_2)) \to e,\ x_2 \cdot (x_1 \cdot x_2) \to x_1\}
\end{aligned}
$$

Apply DEDUCE to $x_1 \cdot (x_1 \cdot x_2) \to x_2$ and $x_2 \cdot (x_1 \cdot x_2) \to x_1$ with the overlapping term $x_2 \cdot (x_2 \cdot (x_1 \cdot x_2))$:

$$
\begin{aligned}
E_{42} &= \{x_1 \cdot x_2 \approx x_2 \cdot x_1\} \\
R_{42} &= \{(x \cdot y) \cdot z \to x \cdot (y \cdot z),\ x \cdot e \to x,\ x \cdot x \to e, \\
&\qquad x_1 \cdot (x_1 \cdot x_2) \to x_2,\ e \cdot x \to x,\ x_1 \cdot i(x_2) \to x_1 \cdot x_2, \\
&\qquad i(x) \to x,\ x_1 \cdot (x_2 \cdot (x_1 \cdot x_2)) \to e,\ x_2 \cdot (x_1 \cdot x_2) \to x_1 \cdot e\}
\end{aligned}
$$

# Proving by Ordered Completion: Example

$$\begin{aligned}
E_{42} &= \{x_1 \cdot x_2 \approx x_2 \cdot x_1\} \\
R_{42} &= \{(x \cdot y) \cdot z \to x \cdot (y \cdot z),\ x \cdot e \to x,\ x \cdot x \to e, \\
&\qquad x_1 \cdot (x_1 \cdot x_2) \to x_2,\ e \cdot x \to x,\ x_1 \cdot i(x_2) \to x_1 \cdot x_2, \\
&\qquad i(x) \to x,\ x_1 \cdot (x_2 \cdot (x_1 \cdot x_2)) \to e,\ x_2 \cdot (x_1 \cdot x_2) \to x_1 \cdot e\}
\end{aligned}$$

The equation $x_1 \cdot x_2 \approx x_2 \cdot x_1$ joins the goal $a \cdot b \approx b \cdot a$. Hence, the goal is proved.