



Term Rewriting

Philipp Dablander and **Aart Middeldorp**

Outline

- 1. Summary of Lecture 3**
- 2. Term Rewriting**
- 3. Undecidability**
- 4. Post Correspondence Problem**
- 5. Exercises**
- 6. Further Reading**

Definitions

- ▶ finite **multiset** M over A is function from A to \mathbb{N} such that $M(a) \neq 0$ for finitely many $a \in A$
- ▶ set of all finite multisets over A is denoted by $\mathcal{M}(A)$
- ▶ **multiset extension** of proper order $>$ on A is relation $>_{\text{mul}}$ defined on $\mathcal{M}(A)$ as follows:
 $M_1 >_{\text{mul}} M_2$ if there exist $X, Y \in \mathcal{M}(A)$ such that $M_2 = (M_1 - X) \uplus Y$, $\emptyset \neq X \subseteq M_1$ and $\forall y \in Y \exists x \in X \ x > y$

Theorem

multiset extension of well-founded order is well-founded order

Definition

equational system (ES) is pair $(\mathcal{F}, \mathcal{E})$ consisting of signature \mathcal{F} and set \mathcal{E} of equations between terms in $\mathcal{T}(\mathcal{F}, \mathcal{V})$

Inference Rules of Equational Logic

r reflexivity

$$\frac{}{t \approx t}$$

$\forall t$

s symmetry

$$\frac{s \approx t}{t \approx s}$$

$\forall s, t$

t transitivity

$$\frac{s \approx t \quad t \approx u}{s \approx u}$$

$\forall s, t, u$

a application

$$\frac{}{l\sigma \approx r\sigma}$$

$\forall l \approx r \in \mathcal{E} \quad \forall \sigma$

c congruence

$$\frac{s_1 \approx t_1 \quad \dots \quad s_n \approx t_n}{f(s_1, \dots, s_n) \approx f(t_1, \dots, t_n)}$$

$\forall n\text{-ary } f \quad \forall s_1, t_1, \dots, s_n, t_n$

Definition

$s \approx_{\mathcal{E}} t$ if equation $s \approx t$ is derivable from equations in \mathcal{E}

Definitions

▶ **\mathcal{F} -algebra** $\mathcal{A} = (A, \{f_{\mathcal{A}}\}_{f \in \mathcal{F}})$ consists of

▶ **carrier** A

▶ **interpretations** $f_{\mathcal{A}}: \underbrace{A \times \cdots \times A}_n \rightarrow A$ if $f \in \mathcal{F}$ has arity n

▶ **assignment** $\alpha: \mathcal{V} \rightarrow A$

▶ interpretation function $[\alpha]_{\mathcal{A}}(\cdot): \mathcal{T}(\mathcal{F}, \mathcal{V}) \rightarrow A$

$$[\alpha]_{\mathcal{A}}(t) = \begin{cases} \alpha(t) & \text{if } t \in \mathcal{V} \\ f_{\mathcal{A}}([\alpha]_{\mathcal{A}}(t_1), \dots, [\alpha]_{\mathcal{A}}(t_n)) & \text{if } t = f(t_1, \dots, t_n) \end{cases}$$

▶ equation $s \approx t$ is **valid** in algebra \mathcal{A} ($s =_{\mathcal{A}} t$) if $[\alpha]_{\mathcal{A}}(s) = [\alpha]_{\mathcal{A}}(t)$ for all assignments α

▶ algebra \mathcal{A} is **model** of ES \mathcal{E} if $s =_{\mathcal{A}} t$ for all equations $s \approx t \in \mathcal{E}$

▶ $s =_{\mathcal{E}} t$ if equation $s \approx t$ is valid in all models of \mathcal{E}

Definition

equational theory of \mathcal{E} consists of all equations $s \approx t$ such that $s =_{\mathcal{E}} t$

Theorem (Birkhoff)

equational reasoning is **sound** and **complete**: $\forall ES \mathcal{E} \quad \approx_{\mathcal{E}} = =_{\mathcal{E}}$

Validity Problem

instance: $ES (\mathcal{F}, \mathcal{E})$ terms $s, t \in \mathcal{T}(\mathcal{F}, \mathcal{V})$

question: $s =_{\mathcal{E}} t?$

Theorem

validity problem is **undecidable**

Outline

1. Summary of Lecture 3
- 2. Term Rewriting**
3. Undecidability
4. Post Correspondence Problem
5. Exercises
6. Further Reading

Definition (Positions)

- $\mathcal{P}os(\cdot)$ positions are strings of positive integers

$$\mathcal{P}os(t) = \begin{cases} \{\epsilon\} & \text{if } t \in \mathcal{V} \\ \{\epsilon\} \cup \{ip \mid 1 \leq i \leq n \text{ and } p \in \mathcal{P}os(t_i)\} & \text{if } t = f(t_1, \dots, t_n) \end{cases}$$

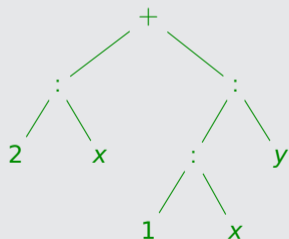
Definition (Positions)

► $\mathcal{P}os(\cdot)$ positions are strings of positive integers

$$\mathcal{P}os(t) = \begin{cases} \{\epsilon\} & \text{if } t \in \mathcal{V} \\ \{\epsilon\} \cup \{ip \mid 1 \leq i \leq n \text{ and } p \in \mathcal{P}os(t_i)\} & \text{if } t = f(t_1, \dots, t_n) \end{cases}$$

Example

$$t = (2 : x) + ((1 : x) : y)$$



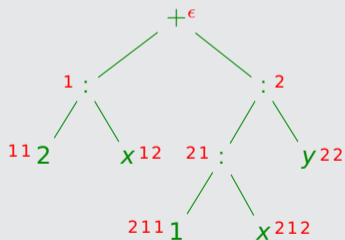
Definition (Positions)

► $\mathcal{P}os(\cdot)$ positions are strings of positive integers

$$\mathcal{P}os(t) = \begin{cases} \{\epsilon\} & \text{if } t \in \mathcal{V} \\ \{\epsilon\} \cup \{ip \mid 1 \leq i \leq n \text{ and } p \in \mathcal{P}os(t_i)\} & \text{if } t = f(t_1, \dots, t_n) \end{cases}$$

Example

$$t = (2 : x) + ((1 : x) : y)$$



$$\mathcal{P}os(t) = \{\epsilon, 1, 11, 12, 2, 21, 211, 212, 22\}$$

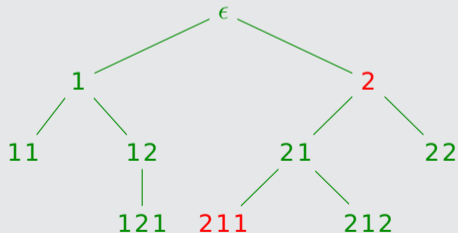
Definitions (Relations on Positions)

► $p < q$ if $pr = q$ for some $r \neq \epsilon$ " p is strictly above q " " q is strictly below p "

Definitions (Relations on Positions)

► $p < q$ if $pr = q$ for some $r \neq \epsilon$ " p is strictly above q " " q is strictly below p "

Example

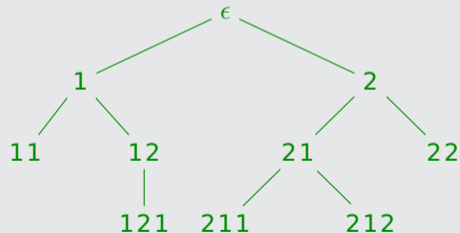


► $2 < 211$

Definitions (Relations on Positions)

- ▶ $p < q$ if $pr = q$ for some $r \neq \epsilon$ " p is strictly above q " " q is strictly below p "
- ▶ $p \leq q$ if $pr = q$ for some r " p is above q " " q is below p "

Example

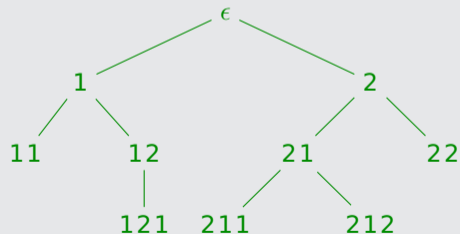


- ▶ $2 < 211$

Definitions (Relations on Positions)

- ▶ $p < q$ if $pr = q$ for some $r \neq \epsilon$ " p is strictly above q " " q is strictly below p "
- ▶ $p \leq q$ if $pr = q$ for some r " p is above q " " q is below p "
- ▶ $p \parallel q$ if $p \not\leq q$ and $q \not\leq p$ " p and q are **parallel**"

Example

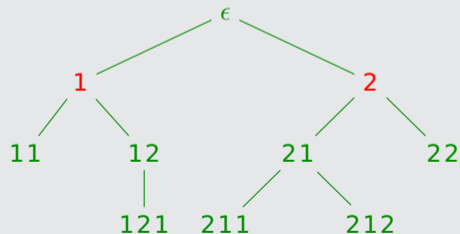


- ▶ $2 < 211$

Definitions (Relations on Positions)

- ▶ $p < q$ if $pr = q$ for some $r \neq \epsilon$ " p is strictly above q " " q is strictly below p "
- ▶ $p \leq q$ if $pr = q$ for some r " p is above q " " q is below p "
- ▶ $p \parallel q$ if $p \not\leq q$ and $q \not\leq p$ " p and q are **parallel**"

Example



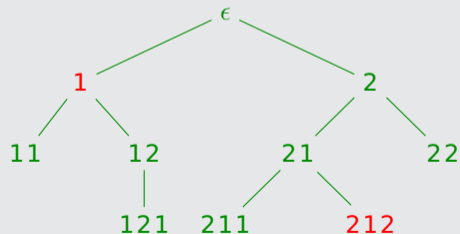
▶ $2 < 211$

▶ $1 \parallel 2$

Definitions (Relations on Positions)

- ▶ $p < q$ if $pr = q$ for some $r \neq \epsilon$ " p is strictly above q " " q is strictly below p "
- ▶ $p \leq q$ if $pr = q$ for some r " p is above q " " q is below p "
- ▶ $p \parallel q$ if $p \not\leq q$ and $q \not\leq p$ " p and q are **parallel**"

Example

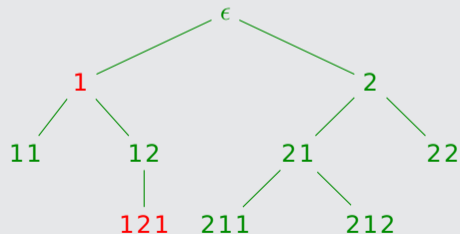


- ▶ $2 < 211$
- ▶ $1 \parallel 2$
- ▶ $1 \parallel 212$

Definitions (Relations on Positions)

- ▶ $p < q$ if $pr = q$ for some $r \neq \epsilon$ " p is strictly above q " " q is strictly below p "
- ▶ $p \leq q$ if $pr = q$ for some r " p is above q " " q is below p "
- ▶ $p \parallel q$ if $p \not\leq q$ and $q \not\leq p$ " p and q are parallel"

Example



- ▶ $2 < 211$
- ▶ $1 \parallel 2$
- ▶ $1 \parallel 212$
- ▶ $1 \not\parallel 121$

Definitions (Operations on Terms and Positions)

► $t|_p$ subterm of t at position p

$$t|_p = \begin{cases} t & \text{if } p = \epsilon \\ t_i|_q & \text{if } t = f(t_1, \dots, t_n) \text{ and } p = iq \end{cases}$$

Definitions (Operations on Terms and Positions)

▶ $t|_p$ subterm of t at position p

$$t|_p = \begin{cases} t & \text{if } p = \epsilon \\ t_i|_q & \text{if } t = f(t_1, \dots, t_n) \text{ and } p = iq \end{cases}$$

▶ $t(p) = \text{root}(t|_p)$ symbol in t at position p

Definitions (Operations on Terms and Positions)

▶ $t|_p$ subterm of t at position p

$$t|_p = \begin{cases} t & \text{if } p = \epsilon \\ t_i|_q & \text{if } t = f(t_1, \dots, t_n) \text{ and } p = iq \end{cases}$$

▶ $t(p) = \text{root}(t|_p)$ symbol in t at position p

▶ $t[]_p$ context obtained by removing subterm in t at position p

$$t[]_p = \begin{cases} \square & \text{if } p = \epsilon \\ f(t_1, \dots, t_i[]_q, \dots, t_n) & \text{if } t = f(t_1, \dots, t_n) \text{ and } p = iq \end{cases}$$

Definitions (Operations on Terms and Positions)

- ▶ $t|_p$ subterm of t at position p

$$t|_p = \begin{cases} t & \text{if } p = \epsilon \\ t_i|_q & \text{if } t = f(t_1, \dots, t_n) \text{ and } p = iq \end{cases}$$

- ▶ $t(p) = \text{root}(t|_p)$ symbol in t at position p

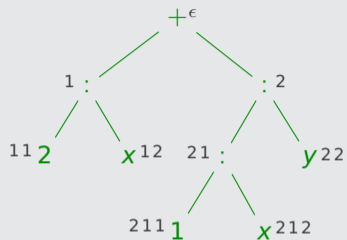
- ▶ $t[]_p$ context obtained by removing subterm in t at position p

$$t[]_p = \begin{cases} \square & \text{if } p = \epsilon \\ f(t_1, \dots, t_i[]_q, \dots, t_n) & \text{if } t = f(t_1, \dots, t_n) \text{ and } p = iq \end{cases}$$

- ▶ $t[s]_p = (t[]_p)[s]$ replace subterm in t at position p by s

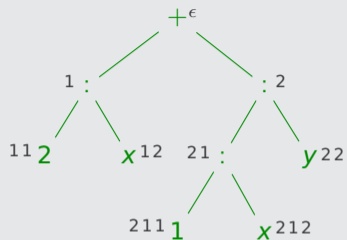
Example

$$t = (2 : x) + ((1 : x) : y)$$



Example

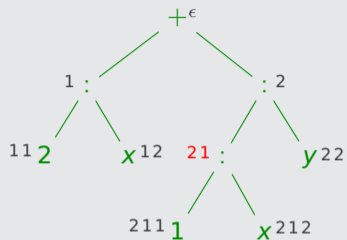
$$t = (2 : x) + ((1 : x) : y)$$



► $t|_{21} = ?$

Example

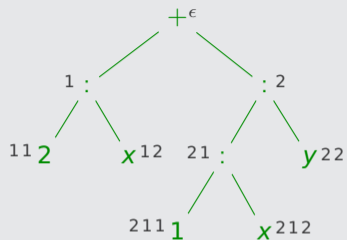
$$t = (2 : x) + ((1 : x) : y)$$



► $t|_{21} = 1 : x$

Example

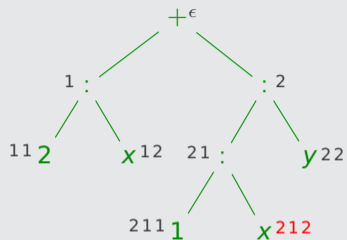
$$t = (2 : x) + ((1 : x) : y)$$



- ▶ $t|_{21} = 1 : x$
- ▶ $t(212) = ?$

Example

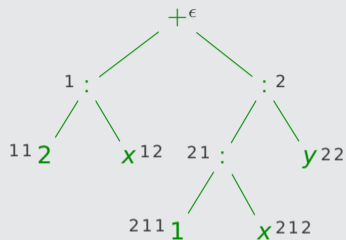
$$t = (2 : x) + ((1 : x) : y)$$



- ▶ $t|_{21} = 1 : x$
- ▶ $t(212) = x$

Example

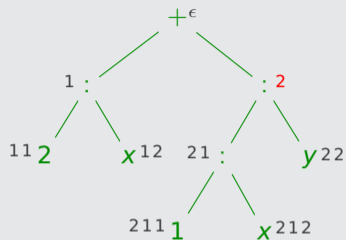
$$t = (2 : x) + ((1 : x) : y)$$



- ▶ $t|_{21} = 1 : x$
- ▶ $t(212) = x$
- ▶ $t[x + 3]_2 = ?$

Example

$$t = (2 : x) + ((1 : x) : y)$$



▶ $t|_{21} = 1 : x$

▶ $t(212) = x$

▶ $t[x + 3]_2 = (2 : x) + (x + 3)$

Definition (Rewriting)

binary relation $\rightarrow_{\mathcal{E}}$ on $\mathcal{T}(\mathcal{F}, \mathcal{V})$ for every ES $(\mathcal{F}, \mathcal{E})$:

$$s \rightarrow_{\mathcal{E}} t \iff \begin{array}{l} \exists p \in \text{Pos}(s) \\ \exists l \approx r \in \mathcal{E} \\ \exists \text{ substitution } \sigma \end{array} \quad \text{with} \quad \begin{array}{l} s|_p = l\sigma \\ t = s[r\sigma]_p \end{array}$$

Definition (Rewriting)

binary relation $\rightarrow_{\mathcal{E}}$ on $\mathcal{T}(\mathcal{F}, \mathcal{V})$ for every ES $(\mathcal{F}, \mathcal{E})$:

$$s \rightarrow_{\mathcal{E}} t \iff \begin{array}{l} \exists p \in \text{Pos}(s) \\ \exists l \approx r \in \mathcal{E} \\ \exists \text{ substitution } \sigma \end{array} \quad \text{with} \quad \begin{array}{l} s|_p = l\sigma \quad \text{redex} \\ t = s[r\sigma]_p \end{array}$$

Definition (Rewriting)

binary relation $\rightarrow_{\mathcal{E}}$ on $\mathcal{T}(\mathcal{F}, \mathcal{V})$ for every ES $(\mathcal{F}, \mathcal{E})$:

$$s \rightarrow_{\mathcal{E}} t \iff \begin{array}{l} \exists p \in \text{Pos}(s) \\ \exists l \approx r \in \mathcal{E} \\ \exists \text{ substitution } \sigma \end{array} \quad \text{with} \quad \begin{array}{l} s|_p = l\sigma \quad \text{redex} \\ t = s[r\sigma]_p \end{array}$$

Example

$$\text{ES } \mathcal{E} = \{0 + y \approx y, s(x) + y \approx s(x + y)\}$$

Definition (Rewriting)

binary relation $\rightarrow_{\mathcal{E}}$ on $\mathcal{T}(\mathcal{F}, \mathcal{V})$ for every ES $(\mathcal{F}, \mathcal{E})$:

$$s \rightarrow_{\mathcal{E}} t \iff \begin{array}{l} \exists p \in \text{Pos}(s) \\ \exists l \approx r \in \mathcal{E} \\ \exists \text{ substitution } \sigma \end{array} \quad \text{with} \quad \begin{array}{l} s|_p = l\sigma \quad \text{redex} \\ t = s[r\sigma]_p \end{array}$$

Example

ES $\mathcal{E} = \{0 + y \approx y, s(x) + y \approx s(x + y)\}$

$s(s(0) + s(0))$

$\downarrow_{\mathcal{E}}$

$s(s(0 + s(0)))$

position

1

equation

$s(x) + y \approx s(x + y)$

substitution

$\{x \mapsto 0, y \mapsto s(0)\}$

Definition (Rewriting)

binary relation $\rightarrow_{\mathcal{E}}$ on $\mathcal{T}(\mathcal{F}, \mathcal{V})$ for every ES $(\mathcal{F}, \mathcal{E})$:

$$s \rightarrow_{\mathcal{E}} t \iff \begin{array}{l} \exists p \in \text{Pos}(s) \\ \exists l \approx r \in \mathcal{E} \\ \exists \text{ substitution } \sigma \end{array} \quad \text{with} \quad \begin{array}{l} s|_p = l\sigma \quad \text{redex} \\ t = s[r\sigma]_p \end{array}$$

Example

ES $\mathcal{E} = \{0 + y \approx y, s(x) + y \approx s(x + y)\}$

	position	equation	substitution
$s(s(0) + s(0))$			
$\downarrow_{\mathcal{E}}$	1	$s(x) + y \approx s(x + y)$	$\{x \mapsto 0, y \mapsto s(0)\}$
$s(s(0 + s(0)))$			
$\downarrow_{\mathcal{E}}$	11	$0 + y \approx y$	$\{y \mapsto s(0)\}$
$s(s(s(0)))$			

Lemma

$\rightarrow_{\mathcal{E}}$ is **smallest** relation that contains \mathcal{E} and is closed under contexts and substitutions

Lemma

$\rightarrow_{\mathcal{E}}$ is smallest relation that contains \mathcal{E} and is closed under contexts and substitutions

Remark

with every ES $(\mathcal{F}, \mathcal{E})$ we associate ARS $\langle \mathcal{T}(\mathcal{F}, \mathcal{V}), \rightarrow_{\mathcal{E}} \rangle$

Lemma

$\rightarrow_{\mathcal{E}}$ is smallest relation that contains \mathcal{E} and is closed under contexts and substitutions

Remark

with every ES $(\mathcal{F}, \mathcal{E})$ we associate ARS $\langle \mathcal{T}(\mathcal{F}, \mathcal{V}), \rightarrow_{\mathcal{E}} \rangle$

▶ notation $(\rightarrow_{\mathcal{E}}^*, \leftrightarrow_{\mathcal{E}}^*, \text{NF}(\mathcal{E}), \dots)$

▶ properties (SN, CR, ...)

are obtained via associated ARS

Lemma

$\rightarrow_{\mathcal{E}}$ is smallest relation that contains \mathcal{E} and is closed under contexts and substitutions

Remark

with every ES $(\mathcal{F}, \mathcal{E})$ we associate ARS $\langle \mathcal{T}(\mathcal{F}, \mathcal{V}), \rightarrow_{\mathcal{E}} \rangle$

▶ notation $(\rightarrow_{\mathcal{E}}^*, \leftrightarrow_{\mathcal{E}}^*, \text{NF}(\mathcal{E}), \dots)$

▶ properties (SN, CR, ...)

are obtained via associated ARS

Theorem

$\forall \text{ ES } \mathcal{E} \quad \leftrightarrow_{\mathcal{E}}^* = \approx_{\mathcal{E}} = =_{\mathcal{E}}$

Example

ES $\mathcal{E} = \{0 + y \approx y, s(x) + y \approx s(x + y)\}$

► $s(s(0) + s(0)) \approx_{\mathcal{E}} s(s(s(0)))$:

$$\frac{\frac{\frac{s(0) + s(0) \approx s(0 + s(0))}{a} \quad \frac{\frac{0 + s(0) \approx s(0)}{a} \quad s(0 + s(0)) \approx s(s(0))}{c}}{t}}{\frac{s(0) + s(0) \approx s(s(0))}{c}}{s(s(0) + s(0)) \approx s(s(s(0)))} c$$

Example

ES $\mathcal{E} = \{0 + y \approx y, s(x) + y \approx s(x + y)\}$

► $s(s(0) + s(0)) \approx_{\mathcal{E}} s(s(s(0)))$:

$$\frac{\frac{\frac{}{s(0) + s(0) \approx s(0 + s(0))} \text{a}}{\frac{}{s(0) + s(0) \approx s(s(0))} \text{c}} \text{t}}{\frac{}{s(s(0) + s(0)) \approx s(s(s(0)))} \text{c}} \text{c}}$$

► $s(s(0) + s(0)) \leftrightarrow_{\mathcal{E}}^* s(s(s(0)))$: $s(s(0) + s(0)) \rightarrow_{\mathcal{E}} s(s(0 + s(0))) \rightarrow_{\mathcal{E}} s(s(s(0)))$

Definitions (Term Rewrite System)

- ▶ **rewrite rule** $l \rightarrow r$ is equation $l \approx r$ such that
 - ▶ $l \notin \mathcal{V}$
 - ▶ $\text{Var}(r) \subseteq \text{Var}(l)$

Definitions (Term Rewrite System)

- ▶ rewrite rule $l \rightarrow r$ is equation $l \approx r$ such that
 - ▶ $l \notin \mathcal{V}$
 - ▶ $\text{Var}(r) \subseteq \text{Var}(l)$
- ▶ **term rewrite system (TRS)** is pair $(\mathcal{F}, \mathcal{R})$ consisting of
 - ▶ \mathcal{F} signature
 - ▶ \mathcal{R} set of rewrite rules between terms in $\mathcal{T}(\mathcal{F}, \mathcal{V})$

Definitions (Term Rewrite System)

- ▶ rewrite rule $l \rightarrow r$ is equation $l \approx r$ such that
 - ▶ $l \notin \mathcal{V}$
 - ▶ $\text{Var}(r) \subseteq \text{Var}(l)$
- ▶ term rewrite system (TRS) is pair $(\mathcal{F}, \mathcal{R})$ consisting of
 - ▶ \mathcal{F} signature
 - ▶ \mathcal{R} set of rewrite rules between terms in $\mathcal{T}(\mathcal{F}, \mathcal{V})$

Lemma

TRSs are ESs

Validity Problem

instance: ES \mathcal{E} terms s, t

question: $s =_{\mathcal{E}} t$?

Theorem

validity problem is **undecidable**

Validity Problem

instance: ES \mathcal{E} terms s, t

question: $s =_{\mathcal{E}} t$?

Theorem

validity problem is **decidable** for ES \mathcal{E} if there exists finite TRS \mathcal{R} such that

① \mathcal{R} is **complete** (**confluent** and **terminating**)

Validity Problem

instance: ES \mathcal{E} terms s, t

question: $s =_{\mathcal{E}} t$?

Theorem

validity problem is **decidable** for ES \mathcal{E} if there exists finite TRS \mathcal{R} such that

- 1 \mathcal{R} is complete (confluent and terminating)
- 2 $\leftrightarrow_{\mathcal{E}}^* = \leftrightarrow_{\mathcal{R}}^*$

Example (Group Theory)

▶ signature e (constant) $^-$ (unary, postfix) \cdot (binary, infix)

▶ ES \mathcal{E} $e \cdot x \approx x$ $x^- \cdot x \approx e$ $(x \cdot y) \cdot z \approx x \cdot (y \cdot z)$

▶ theorems $e^- \approx_{\mathcal{E}} e$ $(x \cdot y)^- \approx_{\mathcal{E}} y^- \cdot x^-$

▶ TRS \mathcal{R}

$$e \cdot x \rightarrow x \qquad x \cdot e \rightarrow x$$

$$x^- \cdot x \rightarrow e \qquad x \cdot x^- \rightarrow e$$

$$(x \cdot y) \cdot z \rightarrow x \cdot (y \cdot z) \qquad x^{--} \rightarrow x$$

$$e^- \rightarrow e \qquad (x \cdot y)^- \rightarrow y^- \cdot x^-$$

$$x^- \cdot (x \cdot y) \rightarrow y \qquad x \cdot (x^- \cdot y) \rightarrow y$$

Example (Group Theory)

▶ signature e (constant) $^-$ (unary, postfix) \cdot (binary, infix)

▶ ES \mathcal{E} $e \cdot x \approx x$ $x^- \cdot x \approx e$ $(x \cdot y) \cdot z \approx x \cdot (y \cdot z)$

▶ theorems $e^- \approx_{\mathcal{E}} e$ $(x \cdot y)^- \approx_{\mathcal{E}} y^- \cdot x^-$

▶ TRS \mathcal{R}

$e \cdot x \rightarrow x$	$x \cdot e \rightarrow x$
$x^- \cdot x \rightarrow e$	$x \cdot x^- \rightarrow e$
$(x \cdot y) \cdot z \rightarrow x \cdot (y \cdot z)$	$x^{--} \rightarrow x$
$e^- \rightarrow e$	$(x \cdot y)^- \rightarrow y^- \cdot x^-$
$x^- \cdot (x \cdot y) \rightarrow y$	$x \cdot (x^- \cdot y) \rightarrow y$

▶ \mathcal{R} is complete and $\leftrightarrow_{\mathcal{E}}^* = \leftrightarrow_{\mathcal{R}}^* \implies \mathcal{E}$ has decidable validity problem

Example (Group Theory)

▶ signature e (constant) $^-$ (unary, postfix) \cdot (binary, infix)

▶ ES \mathcal{E} $e \cdot x \approx x$ $x^- \cdot x \approx e$ $(x \cdot y) \cdot z \approx x \cdot (y \cdot z)$

▶ theorems $e^- \downarrow_{\mathcal{R}} e$ $(x \cdot y)^- \downarrow_{\mathcal{R}} y^- \cdot x^-$

▶ TRS \mathcal{R}

$$e \cdot x \rightarrow x \qquad x \cdot e \rightarrow x$$

$$x^- \cdot x \rightarrow e \qquad x \cdot x^- \rightarrow e$$

$$(x \cdot y) \cdot z \rightarrow x \cdot (y \cdot z) \qquad x^{--} \rightarrow x$$

$$e^- \rightarrow e \qquad (x \cdot y)^- \rightarrow y^- \cdot x^-$$

$$x^- \cdot (x \cdot y) \rightarrow y \qquad x \cdot (x^- \cdot y) \rightarrow y$$

▶ \mathcal{R} is complete and $\leftrightarrow_{\mathcal{E}}^* = \leftrightarrow_{\mathcal{R}}^* \implies \mathcal{E}$ has decidable validity problem

Example (Group Theory)

▶ signature e (constant) $^-$ (unary, postfix) \cdot (binary, infix)

▶ ES \mathcal{E} $e \cdot x \approx x$ $x^- \cdot x \approx e$ $(x \cdot y) \cdot z \approx x \cdot (y \cdot z)$

▶ theorems $e^- \downarrow_{\mathcal{R}} e$ $(x \cdot y)^- \downarrow_{\mathcal{R}} y^- \cdot x^-$

▶ TRS \mathcal{R}

$e \cdot x \rightarrow x$	$x \cdot e \rightarrow x$
$x^- \cdot x \rightarrow e$	$x \cdot x^- \rightarrow e$
$(x \cdot y) \cdot z \rightarrow x \cdot (y \cdot z)$	$x^{--} \rightarrow x$
$e^- \rightarrow e$	$(x \cdot y)^- \rightarrow y^- \cdot x^-$
$x^- \cdot (x \cdot y) \rightarrow y$	$x \cdot (x^- \cdot y) \rightarrow y$

▶ \mathcal{R} is complete and $\leftrightarrow_{\mathcal{E}}^* = \leftrightarrow_{\mathcal{R}}^* \implies \mathcal{E}$ has decidable validity problem

▶ how to compute \mathcal{R} ?

Example (Group Theory)

- ▶ signature e (constant) $^-$ (unary, postfix) \cdot (binary, infix)
- ▶ ES \mathcal{E} $e \cdot x \approx x$ $x^- \cdot x \approx e$ $(x \cdot y) \cdot z \approx x \cdot (y \cdot z)$
- ▶ theorems $e^- \downarrow_{\mathcal{R}} e$ $(x \cdot y)^- \downarrow_{\mathcal{R}} y^- \cdot x^-$
- ▶ TRS \mathcal{R}

$e \cdot x \rightarrow x$	$x \cdot e \rightarrow x$
$x^- \cdot x \rightarrow e$	$x \cdot x^- \rightarrow e$
$(x \cdot y) \cdot z \rightarrow x \cdot (y \cdot z)$	$x^{--} \rightarrow x$
$e^- \rightarrow e$	$(x \cdot y)^- \rightarrow y^- \cdot x^-$
$x^- \cdot (x \cdot y) \rightarrow y$	$x \cdot (x^- \cdot y) \rightarrow y$
- ▶ \mathcal{R} is complete and $\leftrightarrow_{\mathcal{E}}^* = \leftrightarrow_{\mathcal{R}}^* \implies \mathcal{E}$ has decidable validity problem
- ▶ how to compute \mathcal{R} ? **completion** (lectures 6, 7, 8)

Example (Group Theory)

- ▶ signature e (constant) $^-$ (unary, postfix) \cdot (binary, infix)
- ▶ ES \mathcal{E} $e \cdot x \approx x$ $x^- \cdot x \approx e$ $(x \cdot y) \cdot z \approx x \cdot (y \cdot z)$
- ▶ theorems $e^- \downarrow_{\mathcal{R}} e$ $(x \cdot y)^- \downarrow_{\mathcal{R}} y^- \cdot x^-$
- ▶ TRS \mathcal{R}

$e \cdot x \rightarrow x$	$x \cdot e \rightarrow x$
$x^- \cdot x \rightarrow e$	$x \cdot x^- \rightarrow e$
$(x \cdot y) \cdot z \rightarrow x \cdot (y \cdot z)$	$x^{--} \rightarrow x$
$e^- \rightarrow e$	$(x \cdot y)^- \rightarrow y^- \cdot x^-$
$x^- \cdot (x \cdot y) \rightarrow y$	$x \cdot (x^- \cdot y) \rightarrow y$
- ▶ \mathcal{R} is complete and $\leftrightarrow_{\mathcal{E}}^* = \leftrightarrow_{\mathcal{R}}^* \implies \mathcal{E}$ has decidable validity problem
- ▶ how to compute \mathcal{R} ? **completion** (lectures 6, 7, 8)
- ▶ how to prove termination of \mathcal{R} ?

Example (Group Theory)

▶ signature e (constant) $^-$ (unary, postfix) \cdot (binary, infix)

▶ ES \mathcal{E} $e \cdot x \approx x$ $x^- \cdot x \approx e$ $(x \cdot y) \cdot z \approx x \cdot (y \cdot z)$

▶ theorems $e^- \downarrow_{\mathcal{R}} e$ $(x \cdot y)^- \downarrow_{\mathcal{R}} y^- \cdot x^-$

▶ TRS \mathcal{R}

$$e \cdot x \rightarrow x \qquad x \cdot e \rightarrow x$$

$$x^- \cdot x \rightarrow e \qquad x \cdot x^- \rightarrow e$$

$$(x \cdot y) \cdot z \rightarrow x \cdot (y \cdot z) \qquad x^{--} \rightarrow x$$

$$e^- \rightarrow e \qquad (x \cdot y)^- \rightarrow y^- \cdot x^-$$

$$x^- \cdot (x \cdot y) \rightarrow y \qquad x \cdot (x^- \cdot y) \rightarrow y$$

▶ \mathcal{R} is complete and $\leftrightarrow_{\mathcal{E}}^* = \leftrightarrow_{\mathcal{R}}^* \implies \mathcal{E}$ has decidable validity problem

▶ how to compute \mathcal{R} ? **completion** (lectures 6, 7, 8)

▶ how to prove termination of \mathcal{R} ? **LPO** or **KBO** (lectures 5, 6, 8, 11, 12, 13)

Definitions (Canonicity)

► TRS \mathcal{R} is **reduced** if for all $l \rightarrow r \in \mathcal{R}$

① r is normal form of \mathcal{R}

Definitions (Canonicity)

- ▶ TRS \mathcal{R} is **reduced** if for all $l \rightarrow r \in \mathcal{R}$
 - ① r is normal form of \mathcal{R}
 - ② l is normal form of $\mathcal{R} \setminus \{l \rightarrow r\}$

Definitions (Canonicity)

- ▶ TRS \mathcal{R} is reduced if for all $l \rightarrow r \in \mathcal{R}$
 - ① r is normal form of \mathcal{R}
 - ② l is normal form of $\mathcal{R} \setminus \{l \rightarrow r\}$
- ▶ reduced complete TRS is **canonical**

Definitions (Canonicity)

- ▶ TRS \mathcal{R} is reduced if for all $l \rightarrow r \in \mathcal{R}$
 - ① r is normal form of \mathcal{R}
 - ② l is normal form of $\mathcal{R} \setminus \{l \rightarrow r\}$
- ▶ reduced complete TRS is canonical

Example

$$\begin{array}{ll} e \cdot x \rightarrow x & x \cdot e \rightarrow x \\ x^- \cdot x \rightarrow e & x \cdot x^- \rightarrow e \\ (x \cdot y) \cdot z \rightarrow x \cdot (y \cdot z) & x^{-^-} \rightarrow x \\ e^- \rightarrow e & (x \cdot y)^- \rightarrow y^- \cdot x^- \\ x^- \cdot (x \cdot y) \rightarrow y & x \cdot (x^- \cdot y) \rightarrow y \end{array} \quad \text{canonical}$$

Example

TRS \mathcal{R} modeling **Sieve of Eratosthenes** for generating list of prime numbers

$\text{primes} \rightarrow \text{sieve}(\text{from}(\text{s}(\text{s}(0))))$

$\text{sieve}(0 : y) \rightarrow \text{sieve}(y)$

$\text{from}(x) \rightarrow x : \text{from}(\text{s}(x))$

$\text{sieve}(\text{s}(x) : y) \rightarrow \text{s}(x) : \text{sieve}(\text{filter}(x, y, x))$

$\text{head}(x : y) \rightarrow x$

$\text{filter}(0, y : z, w) \rightarrow 0 : \text{filter}(w, z, w)$

$\text{tail}(x : y) \rightarrow y$

$\text{filter}(\text{s}(x), y : z, w) \rightarrow y : \text{filter}(x, z, w)$

Example

TRS \mathcal{R} modeling Sieve of Eratosthenes for generating list of prime numbers

$\text{primes} \rightarrow \text{sieve}(\text{from}(\text{s}(\text{s}(0))))$

$\text{sieve}(0 : y) \rightarrow \text{sieve}(y)$

$\text{from}(x) \rightarrow x : \text{from}(\text{s}(x))$

$\text{sieve}(\text{s}(x) : y) \rightarrow \text{s}(x) : \text{sieve}(\text{filter}(x, y, x))$

$\text{head}(x : y) \rightarrow x$

$\text{filter}(0, y : z, w) \rightarrow 0 : \text{filter}(w, z, w)$

$\text{tail}(x : y) \rightarrow y$

$\text{filter}(\text{s}(x), y : z, w) \rightarrow y : \text{filter}(x, z, w)$

► \mathcal{R} is confluent but not terminating:

$\text{from}(0) \rightarrow 0 : \text{from}(\text{s}(0)) \rightarrow 0 : (\text{s}(0) : \text{from}(\text{s}(\text{s}(0)))) \rightarrow \dots$

Example

TRS \mathcal{R} modeling Sieve of Eratosthenes for generating list of prime numbers

$\text{primes} \rightarrow \text{sieve}(\text{from}(\text{s}(\text{s}(0))))$

$\text{sieve}(0 : y) \rightarrow \text{sieve}(y)$

$\text{from}(x) \rightarrow x : \text{from}(\text{s}(x))$

$\text{sieve}(\text{s}(x) : y) \rightarrow \text{s}(x) : \text{sieve}(\text{filter}(x, y, x))$

$\text{head}(x : y) \rightarrow x$

$\text{filter}(0, y : z, w) \rightarrow 0 : \text{filter}(w, z, w)$

$\text{tail}(x : y) \rightarrow y$

$\text{filter}(\text{s}(x), y : z, w) \rightarrow y : \text{filter}(x, z, w)$

- ▶ \mathcal{R} is confluent but not terminating:

$\text{from}(0) \rightarrow 0 : \text{from}(\text{s}(0)) \rightarrow 0 : (\text{s}(0) : \text{from}(\text{s}(\text{s}(0)))) \rightarrow \dots$

- ▶ how to prove confluence of \mathcal{R} ?

Example

TRS \mathcal{R} modeling Sieve of Eratosthenes for generating list of prime numbers

$\text{primes} \rightarrow \text{sieve}(\text{from}(\text{s}(\text{s}(0))))$

$\text{sieve}(0 : y) \rightarrow \text{sieve}(y)$

$\text{from}(x) \rightarrow x : \text{from}(\text{s}(x))$

$\text{sieve}(\text{s}(x) : y) \rightarrow \text{s}(x) : \text{sieve}(\text{filter}(x, y, x))$

$\text{head}(x : y) \rightarrow x$

$\text{filter}(0, y : z, w) \rightarrow 0 : \text{filter}(w, z, w)$

$\text{tail}(x : y) \rightarrow y$

$\text{filter}(\text{s}(x), y : z, w) \rightarrow y : \text{filter}(x, z, w)$

- ▶ \mathcal{R} is confluent but not terminating:

$\text{from}(0) \rightarrow 0 : \text{from}(\text{s}(0)) \rightarrow 0 : (\text{s}(0) : \text{from}(\text{s}(\text{s}(0)))) \rightarrow \dots$

- ▶ how to prove confluence of \mathcal{R} ? **orthogonality** (lectures 9, 12)

Example

TRS \mathcal{R} modeling Sieve of Eratosthenes for generating list of prime numbers

$\text{primes} \rightarrow \text{sieve}(\text{from}(\text{s}(\text{s}(0))))$

$\text{sieve}(0 : y) \rightarrow \text{sieve}(y)$

$\text{from}(x) \rightarrow x : \text{from}(\text{s}(x))$

$\text{sieve}(\text{s}(x) : y) \rightarrow \text{s}(x) : \text{sieve}(\text{filter}(x, y, x))$

$\text{head}(x : y) \rightarrow x$

$\text{filter}(0, y : z, w) \rightarrow 0 : \text{filter}(w, z, w)$

$\text{tail}(x : y) \rightarrow y$

$\text{filter}(\text{s}(x), y : z, w) \rightarrow y : \text{filter}(x, z, w)$

- ▶ \mathcal{R} is confluent but not terminating:

$\text{from}(0) \rightarrow 0 : \text{from}(\text{s}(0)) \rightarrow 0 : (\text{s}(0) : \text{from}(\text{s}(\text{s}(0)))) \rightarrow \dots$

- ▶ how to prove confluence of \mathcal{R} ? orthogonality (lectures 9, 12)
- ▶ \exists non-terminating terms with (unique) normal form

$\text{head}(\text{tail}(\text{tail}(\text{primes}))) \rightarrow^! \text{s}(\text{s}(\text{s}(\text{s}(\text{s}(0))))))$

Example

TRS \mathcal{R} modeling Sieve of Eratosthenes for generating list of prime numbers

$\text{primes} \rightarrow \text{sieve}(\text{from}(\text{s}(\text{s}(0))))$

$\text{sieve}(0 : y) \rightarrow \text{sieve}(y)$

$\text{from}(x) \rightarrow x : \text{from}(\text{s}(x))$

$\text{sieve}(\text{s}(x) : y) \rightarrow \text{s}(x) : \text{sieve}(\text{filter}(x, y, x))$

$\text{head}(x : y) \rightarrow x$

$\text{filter}(0, y : z, w) \rightarrow 0 : \text{filter}(w, z, w)$

$\text{tail}(x : y) \rightarrow y$

$\text{filter}(\text{s}(x), y : z, w) \rightarrow y : \text{filter}(x, z, w)$

- ▶ \mathcal{R} is confluent but not terminating:

$\text{from}(0) \rightarrow 0 : \text{from}(\text{s}(0)) \rightarrow 0 : (\text{s}(0) : \text{from}(\text{s}(\text{s}(0)))) \rightarrow \dots$

- ▶ how to prove confluence of \mathcal{R} ? orthogonality (lectures 9, 12)
- ▶ \exists non-terminating terms with (unique) normal form

$\text{head}(\text{tail}(\text{tail}(\text{primes}))) \rightarrow^! \text{s}(\text{s}(\text{s}(\text{s}(\text{s}(0))))))$

- ▶ how to compute normal forms in \mathcal{R} ?

Example

TRS \mathcal{R} modeling Sieve of Eratosthenes for generating list of prime numbers

$\text{primes} \rightarrow \text{sieve}(\text{from}(\text{s}(\text{s}(0))))$

$\text{sieve}(0 : y) \rightarrow \text{sieve}(y)$

$\text{from}(x) \rightarrow x : \text{from}(\text{s}(x))$

$\text{sieve}(\text{s}(x) : y) \rightarrow \text{s}(x) : \text{sieve}(\text{filter}(x, y, x))$

$\text{head}(x : y) \rightarrow x$

$\text{filter}(0, y : z, w) \rightarrow 0 : \text{filter}(w, z, w)$

$\text{tail}(x : y) \rightarrow y$

$\text{filter}(\text{s}(x), y : z, w) \rightarrow y : \text{filter}(x, z, w)$

- ▶ \mathcal{R} is confluent but not terminating:

$\text{from}(0) \rightarrow 0 : \text{from}(\text{s}(0)) \rightarrow 0 : (\text{s}(0) : \text{from}(\text{s}(\text{s}(0)))) \rightarrow \dots$

- ▶ how to prove confluence of \mathcal{R} ? orthogonality (lectures 9, 12)
- ▶ \exists non-terminating terms with (unique) normal form

$\text{head}(\text{tail}(\text{tail}(\text{primes}))) \rightarrow^! \text{s}(\text{s}(\text{s}(\text{s}(0))))$

- ▶ how to compute normal forms in \mathcal{R} ? **strategy** (lectures 10, 11)

Definitions (Constructor System)

- ▶ TRS \mathcal{R} over signature \mathcal{F}
 - ▶ **defined symbols** $\mathcal{F}_{\mathcal{D}} = \{\text{root}(\ell) \mid \ell \rightarrow r \in \mathcal{R}\}$

Definitions (Constructor System)

- ▶ TRS \mathcal{R} over signature \mathcal{F}
 - ▶ defined symbols $\mathcal{F}_D = \{\text{root}(\ell) \mid \ell \rightarrow r \in \mathcal{R}\}$
 - ▶ **constructors** $\mathcal{F}_C = \mathcal{F} \setminus \mathcal{F}_D$

Definitions (Constructor System)

- ▶ TRS \mathcal{R} over signature \mathcal{F}
 - ▶ defined symbols $\mathcal{F}_D = \{\text{root}(\ell) \mid \ell \rightarrow r \in \mathcal{R}\}$
 - ▶ constructors $\mathcal{F}_C = \mathcal{F} \setminus \mathcal{F}_D$
- ▶ TRS is **constructor system** (CS) if proper subterms of left-hand sides of rewrite rules do not contain defined symbols

Definitions (Constructor System)

- ▶ TRS \mathcal{R} over signature \mathcal{F}
 - ▶ defined symbols $\mathcal{F}_D = \{\text{root}(\ell) \mid \ell \rightarrow r \in \mathcal{R}\}$
 - ▶ constructors $\mathcal{F}_C = \mathcal{F} \setminus \mathcal{F}_D$
- ▶ TRS is constructor system (CS) if proper subterms of left-hand sides of rewrite rules do not contain defined symbols

Example

primes \rightarrow sieve(from(s(s(0))))

sieve(0 : y) \rightarrow sieve(y)

from(x) \rightarrow x : from(s(x))

sieve(s(x) : y) \rightarrow s(x) : sieve(filter(x, y, x))

head(x : y) \rightarrow x

filter(0, y : z, w) \rightarrow 0 : filter(w, z, w)

tail(x : y) \rightarrow y

filter(s(x), y : z, w) \rightarrow y : filter(x, z, w)

- ▶ defined symbols primes from head tail sieve filter

Definitions (Constructor System)

- ▶ TRS \mathcal{R} over signature \mathcal{F}
 - ▶ defined symbols $\mathcal{F}_D = \{\text{root}(\ell) \mid \ell \rightarrow r \in \mathcal{R}\}$
 - ▶ constructors $\mathcal{F}_C = \mathcal{F} \setminus \mathcal{F}_D$
- ▶ TRS is constructor system (CS) if proper subterms of left-hand sides of rewrite rules do not contain defined symbols

Example

primes \rightarrow sieve(from(s(s(0))))

sieve(0 : y) \rightarrow sieve(y)

from(x) \rightarrow x : from(s(x))

sieve(s(x) : y) \rightarrow s(x) : sieve(filter(x, y, x))

head(x : y) \rightarrow x

filter(0, y : z, w) \rightarrow 0 : filter(w, z, w)

tail(x : y) \rightarrow y

filter(s(x), y : z, w) \rightarrow y : filter(x, z, w)

▶ defined symbols primes from head tail sieve filter

▶ constructors 0 s :

Definitions (Constructor System)

- ▶ TRS \mathcal{R} over signature \mathcal{F}
 - ▶ defined symbols $\mathcal{F}_D = \{\text{root}(\ell) \mid \ell \rightarrow r \in \mathcal{R}\}$
 - ▶ constructors $\mathcal{F}_C = \mathcal{F} \setminus \mathcal{F}_D$
- ▶ TRS is constructor system (CS) if proper subterms of left-hand sides of rewrite rules do not contain defined symbols

Example

primes \rightarrow sieve(from(s(s(0))))

sieve(0 : y) \rightarrow sieve(y)

from(x) \rightarrow x : from(s(x))

sieve(s(x) : y) \rightarrow s(x) : sieve(filter(x, y, x))

head(x : y) \rightarrow x

filter(0, y : z, w) \rightarrow 0 : filter(w, z, w)

tail(x : y) \rightarrow y

filter(s(x), y : z, w) \rightarrow y : filter(x, z, w)

CS

▶ defined symbols primes from head tail sieve filter

▶ constructors 0 s :

Example (Combinatory Logic)

$$I \cdot x \approx x$$

$$(K \cdot x) \cdot y \approx x$$

$$((S \cdot x) \cdot y) \cdot z \approx (x \cdot z) \cdot (y \cdot z)$$

Example (Combinatory Logic)

$$I \cdot x \rightarrow x$$

$$(K \cdot x) \cdot y \rightarrow x$$

$$((S \cdot x) \cdot y) \cdot z \rightarrow (x \cdot z) \cdot (y \cdot z)$$

Example (Combinatory Logic)

$$I \cdot x \rightarrow x$$

$$(K \cdot x) \cdot y \rightarrow x$$

$$((S \cdot x) \cdot y) \cdot z \rightarrow (x \cdot z) \cdot (y \cdot z)$$

$$Ix \rightarrow x$$

$$(Kx)y \rightarrow x$$

$$((Sx)y)z \rightarrow (xz)(yz)$$

► **applicative notation**: suppress \cdot

Example (Combinatory Logic)

$$I \cdot x \rightarrow x$$

$$(K \cdot x) \cdot y \rightarrow x$$

$$((S \cdot x) \cdot y) \cdot z \rightarrow (x \cdot z) \cdot (y \cdot z)$$

$$Ix \rightarrow x$$

$$(Kx)y \rightarrow x$$

$$((Sx)y)z \rightarrow (xz)(yz)$$

$$Ix \rightarrow x$$

$$Kxy \rightarrow x$$

$$Sxyz \rightarrow xz(yz)$$

- **applicative notation**: suppress \cdot and adopt **left-association**

Example (Combinatory Logic)

$$I \cdot x \rightarrow x$$

$$(K \cdot x) \cdot y \rightarrow x$$

$$((S \cdot x) \cdot y) \cdot z \rightarrow (x \cdot z) \cdot (y \cdot z)$$

$$Ix \rightarrow x$$

$$(Kx)y \rightarrow x$$

$$((Sx)y)z \rightarrow (xz)(yz)$$

$$Ix \rightarrow x$$

$$Kxy \rightarrow x$$

$$Sxyz \rightarrow xz(yz)$$

- ▶ applicative notation: suppress \cdot and adopt left-association
- ▶ CL is confluent but not terminating:

$$SII(SII) \rightarrow I(SII)(I(SII)) \rightarrow SII(I(SII)) \rightarrow SII(SII)$$

Example (Combinatory Logic)

$$I \cdot x \rightarrow x$$

$$Ix \rightarrow x$$

$$Ix \rightarrow x$$

$$(K \cdot x) \cdot y \rightarrow x$$

$$(Kx)y \rightarrow x$$

$$Kxy \rightarrow x$$

$$((S \cdot x) \cdot y) \cdot z \rightarrow (x \cdot z) \cdot (y \cdot z)$$

$$((Sx)y)z \rightarrow (xz)(yz)$$

$$Sxyz \rightarrow xz(yz)$$

- ▶ applicative notation: suppress \cdot and adopt left-association
- ▶ CL is confluent but not terminating:

$$SII(SII) \rightarrow I(SII)(I(SII)) \rightarrow SII(I(SII)) \rightarrow SII(SII)$$

- ▶ CL is **consistent**: $S \leftrightarrow^* K$ does not hold

Example (Combinatory Logic)

$$I \cdot x \rightarrow x$$

$$Ix \rightarrow x$$

$$Ix \rightarrow x$$

$$(K \cdot x) \cdot y \rightarrow x$$

$$(Kx)y \rightarrow x$$

$$Kxy \rightarrow x$$

$$((S \cdot x) \cdot y) \cdot z \rightarrow (x \cdot z) \cdot (y \cdot z)$$

$$((Sx)y)z \rightarrow (xz)(yz)$$

$$Sxyz \rightarrow xz(yz)$$

- ▶ applicative notation: suppress \cdot and adopt left-association
- ▶ CL is confluent but not terminating:

$$SII(SII) \rightarrow I(SII)(I(SII)) \rightarrow SII(I(SII)) \rightarrow SII(SII)$$

- ▶ CL is consistent: $S \leftrightarrow^* K$ does not hold

Remark

CL is **Turing-complete**

Definition (String Rewrite System)

string rewrite system (SRS) is TRS over signature consisting of unary function symbols

Definition (String Rewrite System)

string rewrite system (SRS) is TRS over signature consisting of unary function symbols

Example

$$\text{orange}(\text{green}(x)) \rightarrow \text{blue}(\text{blue}(x))$$

$$\text{blue}(\text{orange}(x)) \rightarrow \text{green}(\text{green}(x))$$

$$\text{green}(\text{blue}(x)) \rightarrow \text{orange}(\text{orange}(x))$$

$$\text{green}(\text{orange}(x)) \rightarrow \text{blue}(\text{blue}(x))$$

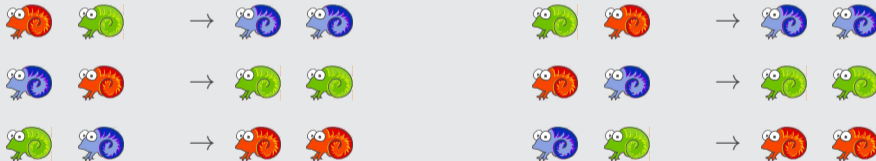
$$\text{orange}(\text{blue}(x)) \rightarrow \text{green}(\text{green}(x))$$

$$\text{blue}(\text{green}(x)) \rightarrow \text{orange}(\text{orange}(x))$$

Definition (String Rewrite System)

string rewrite system (SRS) is TRS over signature consisting of unary function symbols

Example



Example (Cola Gene Puzzle)

- ▶ signature $ACGT$ (unary)
- ▶ ES \mathcal{E} $TCAT \approx T$ $GAG \approx AG$ $CTC \approx TC$ $AGTA \approx A$ $TAT \approx CT$
- ▶ theorem $TAGCTAGCTAGCT \approx_{\mathcal{E}} CTGACTGACT$
- ▶ SRS \mathcal{R} $GA \rightarrow A$ $AGT \rightarrow AT$ $ATA \rightarrow A$ $CT \rightarrow T$ $TAT \rightarrow T$ $TCA \rightarrow TA$

Example (Cola Gene Puzzle)

- ▶ signature $ACGT$ (unary)
- ▶ ES \mathcal{E} $TCAT \approx T$ $GAG \approx AG$ $CTC \approx TC$ $AGTA \approx A$ $TAT \approx CT$
- ▶ theorem $TAGCTAGCTAGCT \approx_{\mathcal{E}} CTGACTGACT$
- ▶ SRS \mathcal{R} $GA \rightarrow A$ $AGT \rightarrow AT$ $ATA \rightarrow A$ $CT \rightarrow T$ $TAT \rightarrow T$ $TCA \rightarrow TA$
- ▶ \mathcal{R} is canonical and $\leftrightarrow_{\mathcal{E}}^* = \leftrightarrow_{\mathcal{R}}^* \implies \mathcal{E}$ has decidable validity problem

Example (Cola Gene Puzzle)

- ▶ signature $ACGT$ (unary)
- ▶ ES \mathcal{E} $TCAT \approx T$ $GAG \approx AG$ $CTC \approx TC$ $AGTA \approx A$ $TAT \approx CT$
- ▶ theorem $TAGCTAGCTAGCT \downarrow_{\mathcal{R}} CTGACTGACT$
- ▶ SRS \mathcal{R} $GA \rightarrow A$ $AGT \rightarrow AT$ $ATA \rightarrow A$ $CT \rightarrow T$ $TAT \rightarrow T$ $TCA \rightarrow TA$
- ▶ \mathcal{R} is canonical and $\leftrightarrow_{\mathcal{E}}^* = \leftrightarrow_{\mathcal{R}}^* \implies \mathcal{E}$ has decidable validity problem

Outline

1. Summary of Lecture 3
2. Term Rewriting
- 3. Undecidability**
4. Post Correspondence Problem
5. Exercises
6. Further Reading

Theorem

term rewriting is **Turing-complete** hence all non-trivial questions are undecidable

Theorem

term rewriting is Turing-complete hence all non-trivial questions are undecidable

Undecidable Problems

instance: (finite) TRS \mathcal{R}

question: is \mathcal{R} terminating?

Theorem

term rewriting is Turing-complete hence all non-trivial questions are undecidable

Undecidable Problems

instance: (finite) TRS \mathcal{R}

question: is \mathcal{R} terminating?

instance: TRS \mathcal{R}

question: is \mathcal{R} confluent?

Theorem

term rewriting is Turing-complete hence all non-trivial questions are undecidable

Undecidable Problems

instance: (finite) TRS \mathcal{R}

question: is \mathcal{R} terminating?

instance: TRS \mathcal{R} , term t

question: is t terminating?

instance: TRS \mathcal{R}

question: is \mathcal{R} confluent?

Theorem

term rewriting is Turing-complete hence all non-trivial questions are undecidable

Undecidable Problems

instance: (finite) TRS \mathcal{R}

question: is \mathcal{R} terminating?

instance: TRS \mathcal{R} , term t

question: is t terminating?

instance: TRS \mathcal{R}

question: is \mathcal{R} confluent?

...

Theorem

term rewriting is Turing-complete hence all non-trivial questions are undecidable

Undecidable Problems

instance: (finite) TRS \mathcal{R}

question: is \mathcal{R} terminating?

instance: TRS \mathcal{R} , term t

question: is t terminating?

instance: TRS \mathcal{R}

question: is \mathcal{R} confluent?

...

Theorem

► confluence is decidable for terminating TRSs

Theorem

term rewriting is Turing-complete hence all non-trivial questions are undecidable

Undecidable Problems

instance: (finite) TRS \mathcal{R}

question: is \mathcal{R} terminating?

instance: TRS \mathcal{R} , term t

question: is t terminating?

instance: TRS \mathcal{R}

question: is \mathcal{R} confluent?

...

Theorem

- ▶ confluence is decidable for terminating TRSs
- ▶ termination is undecidable for confluent TRSs

Theorem

term rewriting is Turing-complete hence all non-trivial questions are undecidable

Undecidable Problems

instance: (finite) TRS \mathcal{R}

question: is \mathcal{R} terminating ?

instance: TRS \mathcal{R} , term t

question: is t terminating ?

instance: TRS \mathcal{R}

question: is \mathcal{R} confluent ?

...

Theorem

- ▶ confluence is decidable for terminating TRSs (lecture 6)
- ▶ termination is undecidable for confluent TRSs

Outline

1. Summary of Lecture 3
2. Term Rewriting
3. Undecidability
- 4. Post Correspondence Problem**
5. Exercises
6. Further Reading

Post Correspondence Problem (PCP)

instance: finite subset P of $\Gamma^+ \times \Gamma^+$ over some alphabet Γ

question: $\exists (\alpha_1, \beta_1), \dots, (\alpha_n, \beta_n) \in P$ with $n > 0$ such that $\alpha_1 \cdots \alpha_n = \beta_1 \cdots \beta_n$?

Post Correspondence Problem (PCP)

instance: finite subset P of $\Gamma^+ \times \Gamma^+$ over some alphabet Γ

question: $\exists (\alpha_1, \beta_1), \dots, (\alpha_n, \beta_n) \in P$ with $n > 0$ such that $\alpha_1 \cdots \alpha_n = \beta_1 \cdots \beta_n$?

Examples

①

1 2 3

$\alpha_i = 1 \quad 10111 \quad 10$

$\beta_i = 11 \quad 101 \quad 01$

Post Correspondence Problem (PCP)

instance: finite subset P of $\Gamma^+ \times \Gamma^+$ over some alphabet Γ

question: $\exists (\alpha_1, \beta_1), \dots, (\alpha_n, \beta_n) \in P$ with $n > 0$ such that $\alpha_1 \cdots \alpha_n = \beta_1 \cdots \beta_n$?

Examples

①

$\alpha_i =$ 1 2 3
1 10111 10

$\beta_i =$ 11 101 01

solution 2 1 1
 α 10111 1 1 = 1011111
 β 101 11 11 = 1011111

Post Correspondence Problem (PCP)

instance: finite subset P of $\Gamma^+ \times \Gamma^+$ over some alphabet Γ

question: $\exists (\alpha_1, \beta_1), \dots, (\alpha_n, \beta_n) \in P$ with $n > 0$ such that $\alpha_1 \cdots \alpha_n = \beta_1 \cdots \beta_n$?

Examples

1

$\alpha_i =$ 1 2 3
1 10111 10

$\beta_i =$ 11 101 01

solution 2 1 1
 α 10111 1 1 = 1011111

β 101 11 11 = 1011111

2

$\alpha_i =$ 1 2 3
10 011 101

$\beta_i =$ 101 11 011

Post Correspondence Problem (PCP)

instance: finite subset P of $\Gamma^+ \times \Gamma^+$ over some alphabet Γ

question: $\exists (\alpha_1, \beta_1), \dots, (\alpha_n, \beta_n) \in P$ with $n > 0$ such that $\alpha_1 \cdots \alpha_n = \beta_1 \cdots \beta_n$?

Examples

1

$\alpha_i =$ 1 2 3
1 10111 10

$\beta_i =$ 11 101 01

solution 2 1 1
 α 10111 1 1 = 1011111

β 101 11 11 = 1011111

2

$\alpha_i =$ 1 2 3
10 011 101

$\beta_i =$ 101 11 011

no solution

solution must start with 1

α 10

β 101

Post Correspondence Problem (PCP)

instance: finite subset P of $\Gamma^+ \times \Gamma^+$ over some alphabet Γ

question: $\exists (\alpha_1, \beta_1), \dots, (\alpha_n, \beta_n) \in P$ with $n > 0$ such that $\alpha_1 \cdots \alpha_n = \beta_1 \cdots \beta_n$?

Examples

①

$\alpha_i =$ 1 2 3
1 10111 10
 $\beta_i =$ 11 101 01

solution 2 1 1
 α 10111 1 1 = 1011111
 β 101 11 11 = 1011111

②

$\alpha_i =$ 1 2 3
10 011 101
 $\beta_i =$ 101 11 011

no solution
solution must start with 1, followed by 3
 α 10101
 β 101011

Post Correspondence Problem (PCP)

instance: finite subset P of $\Gamma^+ \times \Gamma^+$ over some alphabet Γ

question: $\exists (\alpha_1, \beta_1), \dots, (\alpha_n, \beta_n) \in P$ with $n > 0$ such that $\alpha_1 \cdots \alpha_n = \beta_1 \cdots \beta_n$?

Examples

①

$\alpha_i =$ 1 2 3
1 10111 10
 $\beta_i =$ 11 101 01

solution 2 1 1
 α 10111 1 1 = 1011111
 β 101 11 11 = 1011111

②

$\alpha_i =$ 1 2 3
10 011 101
 $\beta_i =$ 101 11 011

no solution

solution must start with 1, followed by 3, followed by 3, ...

α 10101101
 β 101011011

Theorem (Post)

PCP is undecidable

Theorem (Post)

PCP is undecidable

Definition (PCP Encoding 1, Signature)

PCP instance $P \subseteq \Gamma^+ \times \Gamma^+$

- ▶ signature \mathcal{F}_P consists of
 - ▶ binary function symbol f
 - ▶ unary function symbol a for every $a \in \Gamma$
 - ▶ constants $start, stop, c$

Theorem (Post)

PCP is undecidable

Definition (PCP Encoding 1, Signature)

PCP instance $P \subseteq \Gamma^+ \times \Gamma^+$

- ▶ signature \mathcal{F}_P consists of
 - ▶ binary function symbol f
 - ▶ unary function symbol a for every $a \in \Gamma$
 - ▶ constants start , stop , c
- ▶ mapping $\varphi: \Gamma^* \times \mathcal{T}(\mathcal{F}_P, \mathcal{V}) \rightarrow \mathcal{T}(\mathcal{F}_P, \mathcal{V})$

$$\varphi(x, t) = \begin{cases} t & \text{if } x = \epsilon \\ a(\varphi(y, t)) & \text{if } x = ay \end{cases}$$

Notation

$$x(t) = \varphi(x, t)$$

Notation

$$x(t) = \varphi(x, t)$$

Definition (PCP Encoding 1, Rewrite Rules)

PCP instance $P \subseteq \Gamma^+ \times \Gamma^+$

► TRS \mathcal{R}_P over signature \mathcal{F}_P consists of

$$\text{start} \rightarrow f(\alpha(\mathbf{c}), \beta(\mathbf{c})) \quad \forall (\alpha, \beta) \in P$$

Notation

$$x(t) = \varphi(x, t)$$

Definition (PCP Encoding 1, Rewrite Rules)

PCP instance $P \subseteq \Gamma^+ \times \Gamma^+$

► TRS \mathcal{R}_P over signature \mathcal{F}_P consists of

$$\text{start} \rightarrow f(\alpha(\mathbf{c}), \beta(\mathbf{c})) \quad \forall (\alpha, \beta) \in P$$

$$f(\mathbf{x}, \mathbf{y}) \rightarrow f(\alpha(\mathbf{x}), \beta(\mathbf{y})) \quad \forall (\alpha, \beta) \in P$$

Notation

$$x(t) = \varphi(x, t)$$

Definition (PCP Encoding 1, Rewrite Rules)

PCP instance $P \subseteq \Gamma^+ \times \Gamma^+$

► TRS \mathcal{R}_P over signature \mathcal{F}_P consists of

$$\begin{array}{ll} \text{start} \rightarrow f(\alpha(\mathbf{c}), \beta(\mathbf{c})) & \forall (\alpha, \beta) \in P \\ f(\mathbf{x}, \mathbf{y}) \rightarrow f(\alpha(\mathbf{x}), \beta(\mathbf{y})) & \forall (\alpha, \beta) \in P \end{array} \qquad f(\mathbf{x}, \mathbf{x}) \rightarrow \text{stop}$$

Notation

$$x(t) = \varphi(x, t)$$

Definition (PCP Encoding 1, Rewrite Rules)

PCP instance $P \subseteq \Gamma^+ \times \Gamma^+$

► TRS \mathcal{R}_P over signature \mathcal{F}_P consists of

$$\begin{array}{lll} \text{start} \rightarrow f(\alpha(\mathbf{c}), \beta(\mathbf{c})) & \forall (\alpha, \beta) \in P & f(\mathbf{x}, \mathbf{x}) \rightarrow \text{stop} \\ f(\mathbf{x}, \mathbf{y}) \rightarrow f(\alpha(\mathbf{x}), \beta(\mathbf{y})) & \forall (\alpha, \beta) \in P & f(\mathbf{x}, \mathbf{y}) \rightarrow \text{start} \end{array}$$

Example

► PCP instance P

	1	2	3
α	1	10111	10
β	11	101	01

► TRS \mathcal{R}_P

start $\rightarrow f(1(c), 11(c))$

$f(x, y) \rightarrow f(1(x), 11(y))$

$f(x, x) \rightarrow \text{stop}$

start $\rightarrow f(10111(c), 101(c))$

$f(x, y) \rightarrow f(10111(x), 101(y))$

$f(x, y) \rightarrow \text{start}$

start $\rightarrow f(10(c), 01(c))$

$f(x, y) \rightarrow f(10(x), 01(y))$

Example

► PCP instance P

	1	2	3
α	1	10111	10
β	11	101	01

► TRS \mathcal{R}_P

start $\rightarrow f(1(c), 11(c))$

$f(x, y) \rightarrow f(1(x), 11(y))$

$f(x, x) \rightarrow \text{stop}$

start $\rightarrow f(10111(c), 101(c))$

$f(x, y) \rightarrow f(10111(x), 101(y))$

$f(x, y) \rightarrow \text{start}$

start $\rightarrow f(10(c), 01(c))$

$f(x, y) \rightarrow f(10(x), 01(y))$

► start $\rightarrow_{\mathcal{R}_P}^* \text{stop}$

Example

► PCP instance P

	1	2	3
α	1	10111	10
β	11	101	01

► TRS \mathcal{R}_P

start $\rightarrow f(1(c), 11(c))$

$f(x, y) \rightarrow f(1(x), 11(y))$

$f(x, x) \rightarrow \text{stop}$

start $\rightarrow f(10111(c), 101(c))$

$f(x, y) \rightarrow f(10111(x), 101(y))$

$f(x, y) \rightarrow \text{start}$

start $\rightarrow f(10(c), 01(c))$

$f(x, y) \rightarrow f(10(x), 01(y))$

► start $\xrightarrow{*}_{\mathcal{R}_P}$ stop:

start $\xrightarrow{1}$ $f(1(c), 11(c))$

Example

► PCP instance P

	1	2	3
α	1	10111	10
β	11	101	01

► TRS \mathcal{R}_P

start $\rightarrow f(1(c), 11(c))$

$f(x, y) \rightarrow f(1(x), 11(y))$

$f(x, x) \rightarrow \text{stop}$

start $\rightarrow f(10111(c), 101(c))$

$f(x, y) \rightarrow f(10111(x), 101(y))$

$f(x, y) \rightarrow \text{start}$

start $\rightarrow f(10(c), 01(c))$

$f(x, y) \rightarrow f(10(x), 01(y))$

► start $\xrightarrow{*}_{\mathcal{R}_P}$ stop:

start $\xrightarrow{1} f(1(c), 11(c)) \xrightarrow{1} f(11(c), 1111(c))$

Example

► PCP instance P

	1	2	3
α	1	10111	10
β	11	101	01

► TRS \mathcal{R}_P

start $\rightarrow f(1(c), 11(c))$

$f(x, y) \rightarrow f(1(x), 11(y))$

$f(x, x) \rightarrow \text{stop}$

start $\rightarrow f(10111(c), 101(c))$

$f(x, y) \rightarrow f(10111(x), 101(y))$

$f(x, y) \rightarrow \text{start}$

start $\rightarrow f(10(c), 01(c))$

$f(x, y) \rightarrow f(10(x), 01(y))$

► start $\xrightarrow{*}_{\mathcal{R}_P}$ stop:

start $\xrightarrow{1} f(1(c), 11(c)) \xrightarrow{1} f(11(c), 1111(c)) \xrightarrow{2} f(1011111(c), 1011111(c))$

Example

► PCP instance P

	1	2	3
α	1	10111	10
β	11	101	01

► TRS \mathcal{R}_P

start $\rightarrow f(1(c), 11(c))$

$f(x, y) \rightarrow f(1(x), 11(y))$

$f(x, x) \rightarrow \text{stop}$

start $\rightarrow f(10111(c), 101(c))$

$f(x, y) \rightarrow f(10111(x), 101(y))$

$f(x, y) \rightarrow \text{start}$

start $\rightarrow f(10(c), 01(c))$

$f(x, y) \rightarrow f(10(x), 01(y))$

► start $\xrightarrow{*}_{\mathcal{R}_P}$ stop:

start $\xrightarrow{1} f(1(c), 11(c)) \xrightarrow{1} f(11(c), 1111(c)) \xrightarrow{2} f(1011111(c), 1011111(c)) \rightarrow \text{stop}$

Notation

$$x(t) = \varphi(x, t)$$

Definition (PCP Encoding 1, Rewrite Rules)

PCP instance $P \subseteq \Gamma^+ \times \Gamma^+$

► TRS \mathcal{R}_P over signature \mathcal{F}_P consists of

$$\begin{array}{llll} \text{start} \rightarrow f(\alpha(\mathbf{c}), \beta(\mathbf{c})) & \forall (\alpha, \beta) \in P & & f(x, x) \rightarrow \text{stop} \\ f(x, y) \rightarrow f(\alpha(x), \beta(y)) & \forall (\alpha, \beta) \in P & & f(x, y) \rightarrow \text{start} \end{array}$$

Lemma

$$\begin{array}{llll} \forall \text{ PCP instance } P & \mathcal{R}_P \text{ is CR} & \iff & f(\mathbf{c}, \mathbf{c}) \text{ is CR} & \iff & \mathcal{R}_P \text{ is WN} \\ & & \iff & \text{start is WN} & \iff & \mathcal{R}_P \text{ is WCR} \\ & & \iff & f(\mathbf{c}, \mathbf{c}) \text{ is WCR} & \iff & P \text{ has solution} \end{array}$$

Corollary

CR, WCR, WN are undecidable properties of (terms in) TRSs

Corollary

CR, WCR, WN are undecidable properties of (terms in) TRSs

Definition (PCP Encoding 2)

PCP instance $P \subseteq \Gamma^+ \times \Gamma^+$

► TRS \mathcal{S}_P consists of

$$f(\alpha(\mathbf{x}), \beta(\mathbf{y}), \mathbf{z}) \rightarrow f(\mathbf{x}, \mathbf{y}, \mathbf{z})$$

$$f(\alpha(\mathbf{c}), \beta(\mathbf{c}), \mathbf{z}) \rightarrow f(\mathbf{z}, \mathbf{z}, \mathbf{z})$$

$$\forall (\alpha, \beta) \in P$$

Corollary

CR, WCR, WN are undecidable properties of (terms in) TRSs

Definition (PCP Encoding 2)

PCP instance $P \subseteq \Gamma^+ \times \Gamma^+$

► TRS \mathcal{S}_P consists of

$$f(\alpha(\mathbf{x}), \beta(\mathbf{y}), \mathbf{z}) \rightarrow f(\mathbf{x}, \mathbf{y}, \mathbf{z})$$

$$f(\alpha(\mathbf{c}), \beta(\mathbf{c}), \mathbf{z}) \rightarrow f(\mathbf{z}, \mathbf{z}, \mathbf{z})$$

$$\forall (\alpha, \beta) \in P$$

Lemma

\forall PCP instance P \mathcal{S}_P is SN \iff P has no solution

Outline

1. Summary of Lecture 3
2. Term Rewriting
3. Undecidability
4. Post Correspondence Problem
- 5. Exercises**
6. Further Reading

Homework Exercises for April 13

① Exercise 2.6.

1

② Exercise 3.3.

2

③ Exercise 3.4.

3

④ Exercise 3.12.

1

⑤ Exercise 2.36.

★ ★ ★

Outline

1. Summary of Lecture 3
2. Term Rewriting
3. Undecidability
4. Post Correspondence Problem
5. Exercises
- 6. Further Reading**

Lecture Notes

- ▶ Section 2.1 (from Definition 2.1.14)
- ▶ Section 3.1
- ▶ Section 3.2 (until Theorem 3.2.4)
- ▶ Section 3.3 (from Definition 3.3.14)

Lecture Notes

- ▶ Section 2.1 (from Definition 2.1.14)
- ▶ Section 3.1
- ▶ Section 3.2 (until Theorem 3.2.4)
- ▶ Section 3.3 (from Definition 3.3.14)

Important Concepts

- ▶ canonical
- ▶ combinatory logic (CL)
- ▶ constructor
- ▶ constructor system (CS)
- ▶ defined symbol
- ▶ position
- ▶ rewrite rule
- ▶ string rewrite system (SRS)
- ▶ term rewrite system (TRS)
- ▶ term rewriting
- ▶ reduced