Term Rewriting

Basic Concepts, Tools, and Applications

Sarah Winkler

Logical Perspectives Summer School
June 14–16 2021, Moscow

Outline

Term Rewriting

- abstract, Turing complete model of computation
- can model programs, simplification systems, other change processes
- rewriting techniques can serve as toolbox to analyze properties

Outline

Term Rewriting

- abstract, Turing complete model of computation
- can model programs, simplification systems, other change processes
- rewriting techniques can serve as toolbox to analyze properties

Crash course: Term rewriting in a nutshell

- analysis of termination, confluence, completion, complexity
- applications: fun/profit examples

Outline

Term Rewriting

- abstract, Turing complete model of computation
- can model programs, simplification systems, other change processes
- rewriting techniques can serve as toolbox to analyze properties

Crash course: Term rewriting in a nutshell

- analysis of termination, confluence, completion, complexity
- applications: fun/profit examples

Rewrite Tools Developed @ CL Group Innsbruck

$\mathsf{T_TT_2}$, $\mathsf{T_CT}$, CSI, mkbtt, KBCV, mædmax, FORT, ProTeM, CeTA, ConCon, MiniSmt, AutoStrat, Ctrl, . . .

Teaching Example 1: Virus Genes

A COVID medication research team wants to develop a medicine that transforms the DNA of SARS-CoV-2:

$$\text{TAGCTAGCTAGCT} \tag{1}$$

into the DNA of a known and relatively benign influenza virus:

$$\text{CTGACTGACT} \tag{2}$$

Teaching Example 1: Virus Genes

A COVID medication research team wants to develop a medicine that transforms the DNA of SARS-CoV-2:

$$\text{TAGCTAGCTAGCT} \tag{1}$$

into the DNA of a known and relatively benign influenza virus:

$$\text{CTGACTGACT} \tag{2}$$

Techniques exist to perform the following DNA transformations:

$$\text{TCAT} \leftrightarrow \text{T} \quad \text{GAG} \leftrightarrow \text{AG} \quad \text{CTC} \leftrightarrow \text{TC} \quad \text{AGTA} \leftrightarrow \text{A} \quad \text{TAT} \leftrightarrow \text{CT}$$

Teaching Example 1: Virus Genes

A COVID medication research team wants to develop a medicine that transforms the DNA of SARS-CoV-2:

$$\text{TAGCTAGCTAGCT} \tag{1}$$

into the DNA of a known and relatively benign influenza virus:

$$\text{CTGACTGACT} \tag{2}$$

Techniques exist to perform the following DNA transformations:

$$\text{TCAT} \leftrightarrow \text{T} \qquad \text{GAG} \leftrightarrow \text{AG} \qquad \text{CTC} \leftrightarrow \text{TC} \qquad \text{AGTA} \leftrightarrow \text{A} \qquad \text{TAT} \leftrightarrow \text{CT}$$

Recently it has been discovered that the mad cow disease is caused by a retrovirus with the following DNA sequence

$$\text{CTGCTACTGACT}$$

Teaching Example 1: Virus Genes

A COVID medication research team wants to develop a medicine that transforms the DNA of SARS-CoV-2:

$$\text{TAGCTAGCTAGCT} \tag{1}$$

into the DNA of a known and relatively benign influenza virus:

$$\text{CTGACTGACT} \tag{2}$$

Techniques exist to perform the following DNA transformations:

$$\text{TCAT} \leftrightarrow \text{T} \quad \text{GAG} \leftrightarrow \text{AG} \quad \text{CTC} \leftrightarrow \text{TC} \quad \text{AGTA} \leftrightarrow \text{A} \quad \text{TAT} \leftrightarrow \text{CT}$$

Recently it has been discovered that the mad cow disease is caused by a retrovirus with the following DNA sequence

$$\text{CTGCTACTGACT}$$

Questions

- are the known transformations sufficient to transform (1) into (2)?

Teaching Example 1: Virus Genes

A COVID medication research team wants to develop a medicine that transforms the DNA of SARS-CoV-2:

$$\text{TAGCTAGCTAGCT} \tag{1}$$

into the DNA of a known and relatively benign influenza virus:

$$\text{CTGACTGACT} \tag{2}$$

Techniques exist to perform the following DNA transformations:

$$\text{TCAT} \leftrightarrow \text{T} \quad \text{GAG} \leftrightarrow \text{AG} \quad \text{CTC} \leftrightarrow \text{TC} \quad \text{AGTA} \leftrightarrow \text{A} \quad \text{TAT} \leftrightarrow \text{CT}$$

Recently it has been discovered that the mad cow disease is caused by a retrovirus with the following DNA sequence

$$\text{CTGCTACTGACT}$$

Questions

- are the known transformations sufficient to transform (1) into (2)?
- is it possible that the mad cow disease DNA is created in this process?

Teaching Example 2: Bean Game

- two-player game where state is sequence of black and white stones

Teaching Example 2: Bean Game

- two-player game where state is sequence of black and white stones
- allowed moves are

$$\bullet\ \bullet \to \circ \qquad \circ\ \circ \to \circ \qquad \bullet\ \circ \to \bullet \qquad \circ\ \bullet \to \bullet$$

Teaching Example 2: Bean Game

- two-player game where state is sequence of black and white stones
- allowed moves are

$$\bullet\ \bullet \rightarrow \circ \qquad \circ\ \circ \rightarrow \circ \qquad \bullet\ \circ \rightarrow \bullet \qquad \circ\ \bullet \rightarrow \bullet$$

- player who puts last white wins

Teaching Example 2: Bean Game

- two-player game where state is sequence of black and white stones
- allowed moves are

$$\bullet\,\bullet \to \circ \qquad \circ\,\circ \to \circ \qquad \bullet\,\circ \to \bullet \qquad \circ\,\bullet \to \bullet$$

- player who puts last white wins
- initial state

$$\bullet\,\bullet\,\circ\,\bullet\,\circ\,\bullet\,\bullet\,\circ\,\circ\,\bullet\,\circ\,\circ\,\bullet\,\bullet\,\circ$$

Teaching Example 2: Bean Game

- two-player game where state is sequence of black and white stones
- allowed moves are

● ● → ○   ○ ○ → ○   ● ○ → ●   ○ ● → ●

- player who puts last white wins
- initial state

● ● ○ ● ○ ● ● ○ ○ ● ○ ○ ● ● ○

player 1   ○ ○ ● ○ ● ● ○ ○ ● ○ ○ ● ● ○

Teaching Example 2: Bean Game

- two-player game where state is sequence of black and white stones
- allowed moves are

  ● ● → ○   ○ ○ → ○   ● ○ → ●   ○ ● → ●

- player who puts last white wins
- initial state

  ● ● ○ ● ○ ● ● ○ ○ ● ○ ○ ● ● ○

  player 1   ○ ○ ● ○ ● ● ○ ○ ● ○ ○ ● ● ○

Teaching Example 2: Bean Game

- two-player game where state is sequence of black and white stones
- allowed moves are

$$\bullet\ \bullet \longrightarrow \circ \qquad \circ\ \circ \longrightarrow \circ \qquad \bullet\ \circ \longrightarrow \bullet \qquad \circ\ \bullet \longrightarrow \bullet$$

- player who puts last white wins
- initial state

	● ● ○ ● ○ ● ● ○ ○ ● ○ ○ ● ● ○
player 1	○ ○ ● ○ ● ● ○ ○ ● ○ ○ ● ● ○
player 2	● ○ ● ○ ● ● ○ ○ ● ○ ○ ● ○

Teaching Example 2: Bean Game

- two-player game where state is sequence of black and white stones
- allowed moves are

$$\bullet\ \bullet \longrightarrow \circ \qquad \circ\ \circ \longrightarrow \circ \qquad \bullet\ \circ \longrightarrow \bullet \qquad \circ\ \bullet \longrightarrow \bullet$$

- player who puts last white wins
- initial state

	● ● ○ ● ○ ● ● ○ ○ ● ○ ○ ● ● ○
player 1	○ ○ ● ○ ● ● ○ ○ ● ○ ○ ● ● ○
player 2	● ○ ● ○ ● ● ○ ○ ● ○ ○ ● ○

Teaching Example 2: Bean Game

- two-player game where state is sequence of black and white stones
- allowed moves are

  ● ● → ○ ○ ○ → ○ ● ○ → ● ○ ● → ●

- player who puts last white wins
- initial state

	● ● ○ ● ○ ● ● ○ ○ ● ○ ○ ● ● ○
player 1	○ ○ ● ○ ● ● ○ ○ ● ○ ○ ● ● ○
player 2	● ○ ● ○ ● ● ○ ○ ● ○ ○ ● ○
player 1	● ○ ● ○ ● ● ○ ● ○ ○ ● ○

Teaching Example 2: Bean Game

- two-player game where state is sequence of black and white stones
- allowed moves are

● ● $\rightarrow$ ○ ○ ○ $\rightarrow$ ○ ● ○ $\rightarrow$ ● ○ ● $\rightarrow$ ●

- player who puts last white wins
- initial state

	● ● ○ ● ○ ● ● ○ ○ ● ○ ○ ● ● ○
player 1	○ ○ ● ○ ● ● ○ ○ ● ○ ○ ● ● ○
player 2	● ○ ● ○ ● ● ○ ○ ● ○ ○ ● ○
player 1	● ○ ● ○ ● ● ○ ● ○ ○ ● ○

Teaching Example 2: Bean Game

- two-player game where state is sequence of black and white stones
- allowed moves are

  ● ● $\rightarrow$ ○   ○ ○ $\rightarrow$ ○   ● ○ $\rightarrow$ ●   ○ ● $\rightarrow$ ●

- player who puts last white wins
- initial state

	● ● ○ ● ○ ● ● ○ ○ ● ○ ○ ● ● ○
player 1	○ ○ ● ○ ● ● ○ ○ ● ○ ○ ● ● ○
player 2	● ○ ● ○ ● ● ○ ○ ● ○ ○ ● ○
player 1	● ○ ● ○ ● ● ○ ● ○ ○ ● ○
player 2	● ○ ● ○ ● ● ○ ● ○ ● ○

Teaching Example 2: Bean Game

- two-player game where state is sequence of black and white stones
- allowed moves are

$$\bullet\;\bullet \longrightarrow \circ \qquad \circ\;\circ \longrightarrow \circ \qquad \bullet\;\circ \longrightarrow \bullet \qquad \circ\;\bullet \longrightarrow \bullet$$

- player who puts last white wins
- initial state

	● ● ○ ● ○ ● ● ○ ○ ● ○ ○ ● ● ○
player 1	○ ○ ● ○ ● ● ○ ○ ● ○ ○ ● ● ○
player 2	● ○ ● ○ ● ● ○ ○ ● ○ ○ ● ○
player 1	● ○ ● ○ ● ● ○ ● ○ ○ ● ○
player 2	● ○ ● ○ ● ● ○ ● ○ ● ○

Teaching Example 2: Bean Game

- two-player game where state is sequence of black and white stones
- allowed moves are

$$\bullet\ \bullet \longrightarrow \circ \qquad \circ\ \circ \longrightarrow \circ \qquad \bullet\ \circ \longrightarrow \bullet \qquad \circ\ \bullet \longrightarrow \bullet$$

- player who puts last white wins
- initial state

	$\bullet\ \bullet\ \circ\ \bullet\ \circ\ \bullet\ \bullet\ \circ\ \circ\ \bullet\ \circ\ \circ\ \bullet\ \bullet\ \circ$
player 1	$\circ\ \circ\ \bullet\ \circ\ \bullet\ \bullet\ \circ\ \circ\ \bullet\ \circ\ \circ\ \bullet\ \bullet\ \circ$
player 2	$\bullet\ \circ\ \bullet\ \circ\ \bullet\ \bullet\ \circ\ \circ\ \bullet\ \circ\ \circ\ \bullet\ \circ$
player 1	$\bullet\ \circ\ \bullet\ \circ\ \bullet\ \bullet\ \circ\ \bullet\ \circ\ \circ\ \bullet\ \circ$
player 2	$\bullet\ \circ\ \bullet\ \circ\ \bullet\ \bullet\ \circ\ \bullet\ \circ\ \bullet\ \circ$
player 1	$\bullet\ \bullet\ \circ\ \bullet\ \bullet\ \circ\ \bullet\ \circ\ \bullet\ \circ$

Teaching Example 2: Bean Game

- two-player game where state is sequence of black and white stones
- allowed moves are

$$\bullet\ \bullet \to \circ \qquad \circ\ \circ \to \circ \qquad \bullet\ \circ \to \bullet \qquad \circ\ \bullet \to \bullet$$

- player who puts last white wins
- initial state

	$\bullet\ \bullet\ \circ\ \bullet\ \circ\ \bullet\ \bullet\ \circ\ \circ\ \bullet\ \circ\ \circ\ \bullet\ \bullet\ \circ$
player 1	$\circ\ \circ\ \bullet\ \circ\ \bullet\ \bullet\ \circ\ \circ\ \bullet\ \circ\ \circ\ \bullet\ \bullet\ \circ$
player 2	$\bullet\ \circ\ \bullet\ \circ\ \bullet\ \bullet\ \circ\ \circ\ \bullet\ \circ\ \circ\ \bullet\ \circ$
player 1	$\bullet\ \circ\ \bullet\ \circ\ \bullet\ \bullet\ \circ\ \bullet\ \circ\ \circ\ \bullet\ \circ$
player 2	$\bullet\ \circ\ \bullet\ \circ\ \bullet\ \bullet\ \circ\ \bullet\ \circ\ \bullet\ \circ$
player 1	$\bullet\ \bullet\ \circ\ \bullet\ \bullet\ \circ\ \bullet\ \circ\ \bullet\ \circ$
player 2	$\bullet\ \bullet\ \circ\ \bullet\ \bullet\ \circ\ \bullet\ \circ\ \bullet$

Teaching Example 2: Bean Game

- two-player game where state is sequence of black and white stones
- allowed moves are

$$\bullet\ \bullet \to \circ \qquad \circ\ \circ \to \circ \qquad \bullet\ \circ \to \bullet \qquad \circ\ \bullet \to \bullet$$

- player who puts last white wins
- initial state

	● ● ○ ● ○ ● ● ○ ○ ● ○ ○ ● ● ○
player 1	○ ○ ● ○ ● ● ○ ○ ● ○ ○ ● ● ○
player 2	● ○ ● ○ ● ● ○ ○ ● ○ ○ ● ○
player 1	● ○ ● ○ ● ● ○ ● ○ ○ ● ○
player 2	● ○ ● ○ ● ● ○ ● ○ ● ○
player 1	● ● ○ ● ● ○ ● ○ ● ○
player 2	● ● ○ ● ● ○ ● ○ ●
player 1	● ● ○ ● ● ○ ● ●

Teaching Example 2: Bean Game

- two-player game where state is sequence of black and white stones
- allowed moves are

$$\bullet\ \bullet \rightarrow \circ \qquad \circ\ \circ \rightarrow \circ \qquad \bullet\ \circ \rightarrow \bullet \qquad \circ\ \bullet \rightarrow \bullet$$

- player who puts last white wins
- initial state

	● ● ○ ● ○ ● ● ○ ○ ● ○ ○ ● ● ○
player 1	○ ○ ● ○ ● ● ○ ○ ● ○ ○ ● ● ○
player 2	● ○ ● ○ ● ● ○ ○ ● ○ ○ ● ○
player 1	● ○ ● ○ ● ● ○ ● ○ ○ ● ○
player 2	● ○ ● ○ ● ● ○ ● ○ ● ○
player 1	● ● ○ ● ● ○ ● ○ ● ○
player 2	● ● ○ ● ● ○ ● ○ ●
player 1	● ● ○ ● ● ○ ● ●
player 2	● ● ○ ● ● ○ ○

Teaching Example 2: Bean Game

- two-player game where state is sequence of black and white stones
- allowed moves are

● ● → ○ ○ ○ → ○ ● ○ → ● ○ ● → ●

- player who puts last white wins
- initial state

	● ● ○ ● ○ ● ● ○ ○ ● ○ ○ ● ● ○
player 1	○ ○ ● ○ ● ● ○ ○ ● ○ ○ ● ● ○
player 2	● ○ ● ○ ● ● ○ ○ ● ○ ○ ● ○
player 1	● ○ ● ○ ● ● ○ ● ○ ○ ● ○
player 2	● ○ ● ○ ● ● ○ ● ○ ● ○
player 1	● ● ○ ● ● ○ ● ○ ● ○
player 2	● ● ○ ● ● ○ ● ○ ●
player 1	● ● ○ ● ● ○ ● ●
player 2	● ● ○ ● ● ○ ○
player 1	● ● ○ ● ● ○

Teaching Example 2: Bean Game

- two-player game where state is sequence of black and white stones
- allowed moves are

● ● → ○ ○ ○ → ○ ● ○ → ● ○ ● → ●

- player who puts last white wins
- initial state

	● ● ○ ● ○ ● ● ○ ○ ● ○ ○ ● ● ○
player 1	○ ○ ● ○ ● ● ○ ○ ● ○ ○ ● ● ○
player 2	● ○ ● ○ ● ● ○ ○ ● ○ ○ ● ○
player 1	● ○ ● ○ ● ● ○ ● ○ ○ ● ○
player 2	● ○ ● ○ ● ● ○ ● ○ ● ○
player 1	● ● ○ ● ● ○ ● ○ ● ○
player 2	● ● ○ ● ● ○ ● ○ ●
player 1	● ● ○ ● ● ○ ● ●
player 2	● ● ○ ● ● ○ ○
player 1	● ● ○ ● ● ○
player 2	● ● ○ ● ●

Teaching Example 2: Bean Game

- two-player game where state is sequence of black and white stones
- allowed moves are

● ● → ○ ○ ○ → ○ ● ○ → ● ○ ● → ●

- player who puts last white wins
- initial state

	● ● ○ ● ○ ● ● ○ ○ ● ○ ○ ● ● ○
player 1	○ ○ ● ○ ● ● ○ ○ ● ○ ○ ● ● ○
player 2	● ○ ● ○ ● ● ○ ○ ● ○ ○ ● ○
player 1	● ○ ● ○ ● ● ○ ● ○ ○ ● ○
player 2	● ○ ● ○ ● ● ○ ● ○ ● ○
player 1	● ● ○ ● ● ○ ● ○ ● ○
player 2	● ● ○ ● ● ○ ● ○ ●
player 1	● ● ○ ● ● ○ ● ●
player 2	● ● ○ ● ● ○ ○
player 1	● ● ○ ● ● ○
player 2	● ● ○ ● ●
player 1	● ● ○ ○

Teaching Example 2: Bean Game

- two-player game where state is sequence of black and white stones
- allowed moves are

● ● → ○ ○ ○ → ○ ● ○ → ● ○ ● → ●

- player who puts last white wins
- initial state

	● ● ○ ● ○ ● ● ○ ○ ● ○ ○ ● ● ○
player 1	○ ○ ● ○ ● ● ○ ○ ● ○ ○ ● ● ○
player 2	● ○ ● ○ ● ● ○ ○ ● ○ ○ ● ○
player 1	● ○ ● ○ ● ● ○ ● ○ ○ ● ○
player 2	● ○ ● ○ ● ● ○ ● ○ ● ○
player 1	● ● ○ ● ● ○ ● ○ ● ○
player 2	● ● ○ ● ● ○ ● ○ ●
player 1	● ● ○ ● ● ○ ● ●
player 2	● ● ○ ● ● ○ ○
player 1	● ● ○ ● ● ○
player 2	● ● ○ ● ●
player 1	● ● ○ ○
player 2	● ● ○

Teaching Example 2: Bean Game

- two-player game where state is sequence of black and white stones
- allowed moves are

  ● ● → ○ ○ ○ → ○ ● ○ → ● ○ ● → ●

- player who puts last white wins
- initial state

	● ● ○ ● ○ ● ● ○ ○ ● ○ ○ ● ● ○
player 1	○ ○ ● ○ ● ● ○ ○ ● ○ ○ ● ● ○
player 2	● ○ ● ○ ● ● ○ ○ ● ○ ○ ● ○
player 1	● ○ ● ○ ● ● ○ ● ○ ○ ● ○
player 2	● ○ ● ○ ● ● ○ ● ○ ● ○
player 1	● ● ○ ● ● ○ ● ○ ● ○
player 2	● ● ○ ● ● ○ ● ○ ●
player 1	● ● ○ ● ● ○ ● ●
player 2	● ● ○ ● ● ○ ○
player 1	● ● ○ ● ● ○
player 2	● ● ○ ● ●
player 1	● ● ○ ○
player 2	● ● ○
player 1	● ●

Teaching Example 2: Bean Game

- two-player game where state is sequence of black and white stones
- allowed moves are

● ● → ○   ○ ○ → ○   ● ○ → ●   ○ ● → ●

- player who puts last white wins
- initial state

	● ● ○ ● ○ ● ● ○ ○ ● ○ ○ ● ● ○
player 1	○ ○ ● ○ ● ● ○ ○ ● ○ ○ ● ● ○
player 2	● ○ ● ○ ● ● ○ ○ ● ○ ○ ● ○
player 1	● ○ ● ○ ● ● ○ ● ○ ○ ● ○
player 2	● ○ ● ○ ● ● ○ ● ○ ● ○
player 1	● ● ○ ● ● ○ ● ○ ● ○
player 2	● ● ○ ● ● ○ ● ○ ●
player 1	● ● ○ ● ● ○ ● ●
player 2	● ● ○ ● ● ○ ○
player 1	● ● ○ ● ● ○
player 2	● ● ○ ● ●
player 1	● ● ○ ○
player 2	● ● ○
player 1	● ●
player 2	○

Teaching Example 2: Bean Game

- two-player game where state is sequence of black and white stones
- allowed moves are

$$\bullet\ \bullet \to \circ \qquad \circ\ \circ \to \circ \qquad \bullet\ \circ \to \bullet \qquad \circ\ \bullet \to \bullet$$

- player who puts last white wins
- initial state

$$\bullet\,\bullet\,\circ\,\bullet\,\circ\,\bullet\,\bullet\,\circ\,\circ\,\bullet\,\circ\,\circ\,\bullet\,\bullet\,\circ$$

Questions

- does the game terminate for every initial state?

Teaching Example 2: Bean Game

- two-player game where state is sequence of black and white stones
- allowed moves are

$$\bullet\,\bullet \to \circ \qquad \circ\,\circ \to \circ \qquad \bullet\,\circ \to \bullet \qquad \circ\,\bullet \to \bullet$$

- player who puts last white wins
- initial state

$$\bullet\,\bullet\,\circ\,\bullet\,\circ\,\bullet\,\bullet\,\circ\,\circ\,\bullet\,\circ\,\circ\,\bullet\,\bullet\,\circ$$

Questions

- does the game terminate for every initial state?
- which strategies are winning strategies for player 2?

Research Example 3: Functional Programs

TRS $\mathcal{R}$ models sieve of Eratosthenes to enumerate prime numbers:

$$
\begin{aligned}
\mathsf{primes} &\to \mathsf{sieve}(\mathsf{from}(\mathsf{s}(\mathsf{s}(0)))) & \mathsf{sieve}(0 : y) &\to \mathsf{sieve}(y)\\
\mathsf{from}(x) &\to x : \mathsf{from}(\mathsf{s}(x)) & \mathsf{sieve}(\mathsf{s}(x) : y) &\to \mathsf{s}(x) : \mathsf{sieve}(\mathsf{filter}(x, y, x))\\
\mathsf{hd}(x : y) &\to x & \mathsf{filter}(0, y : z, w) &\to 0 : \mathsf{filter}(w, z, w)\\
\mathsf{tl}(x : y) &\to y & \mathsf{filter}(\mathsf{s}(x), y : z, w) &\to y : \mathsf{filter}(x, z, w)
\end{aligned}
$$

Research Example 3: Functional Programs

TRS $\mathcal{R}$ models sieve of Eratosthenes to enumerate prime numbers:

$$\begin{aligned}
\mathsf{primes} &\to \mathsf{sieve}(\mathsf{from}(\mathsf{s}(\mathsf{s}(0)))) & \mathsf{sieve}(0 : y) &\to \mathsf{sieve}(y)\\
\mathsf{from}(x) &\to x : \mathsf{from}(\mathsf{s}(x)) & \mathsf{sieve}(\mathsf{s}(x) : y) &\to \mathsf{s}(x) : \mathsf{sieve}(\mathsf{filter}(x, y, x))\\
\mathsf{hd}(x : y) &\to x & \mathsf{filter}(0, y : z, w) &\to 0 : \mathsf{filter}(w, z, w)\\
\mathsf{tl}(x : y) &\to y & \mathsf{filter}(\mathsf{s}(x), y : z, w) &\to y : \mathsf{filter}(x, z, w)
\end{aligned}$$

Questions About (Functional) Programs

- is the given program terminating?

Research Example 3: Functional Programs

TRS $\mathcal{R}$ models sieve of Eratosthenes to enumerate prime numbers:

$$\mathsf{primes} \to \mathsf{sieve}(\mathsf{from}(\mathsf{s}(\mathsf{s}(0)))) \qquad \mathsf{sieve}(0 : y) \to \mathsf{sieve}(y)$$

$$\mathsf{from}(x) \to x : \mathsf{from}(\mathsf{s}(x)) \qquad \mathsf{sieve}(\mathsf{s}(x) : y) \to \mathsf{s}(x) : \mathsf{sieve}(\mathsf{filter}(x, y, x))$$

$$\mathsf{hd}(x : y) \to x \qquad \mathsf{filter}(0, y : z, w) \to 0 : \mathsf{filter}(w, z, w)$$

$$\mathsf{tl}(x : y) \to y \qquad \mathsf{filter}(\mathsf{s}(x), y : z, w) \to y : \mathsf{filter}(x, z, w)$$

Questions About (Functional) Programs

- is the given program terminating?
- if yes, what is its worst-case computational complexity?

Research Example 3: Functional Programs

TRS $\mathcal{R}$ models sieve of Eratosthenes to enumerate prime numbers:

$$\begin{aligned}
\mathsf{primes} &\to \mathsf{sieve}(\mathsf{from}(\mathsf{s}(\mathsf{s}(0)))) & \mathsf{sieve}(0 : y) &\to \mathsf{sieve}(y)\\
\mathsf{from}(x) &\to x : \mathsf{from}(\mathsf{s}(x)) & \mathsf{sieve}(\mathsf{s}(x) : y) &\to \mathsf{s}(x) : \mathsf{sieve}(\mathsf{filter}(x, y, x))\\
\mathsf{hd}(x : y) &\to x & \mathsf{filter}(0, y : z, w) &\to 0 : \mathsf{filter}(w, z, w)\\
\mathsf{tl}(x : y) &\to y & \mathsf{filter}(\mathsf{s}(x), y : z, w) &\to y : \mathsf{filter}(x, z, w)
\end{aligned}$$

Questions About (Functional) Programs

- is the given program terminating?
- if yes, what is its worst-case computational complexity?
- are results unique?

Practice Example 4: LLVM Expression Simplification

```
int foo(int z) {
  int x = 4*(z|101);
  return -256^x;
}
```

$\overset{*}{\dashrightarrow}$

```
01011111010101011011010
11111111101110100001 0
01001110010101100101 0
11111011101110100101 0
```

- LLVM provides widely used compilation toolchain

Practice Example 4: LLVM Expression Simplification

```
int foo(int z) {
  int x = 4*(z|101);
  return -256^x;
}
```

*

```
define i32 @foo(i32) #0 {
  %2 = or i32 %0, 101
  %3 = mul nsw i32 4, %2
  %4 = xor i32 -256, %3
  ret i32 %4
}
```

Instcombine

```
define i32 @foo(i32) #0 {
  %2 = shl i32 %0, 2
  %3 = or i32 %2, 404
  %4 = xor i32 %3, -256
  ret i32 %4
}
```

*

```
01011111010101011011010
11111111101110100001
0100111001010110010 10
1111101110111010010 10
```

- LLVM provides widely used compilation toolchain

Practice Example 4: LLVM Expression Simplification

- LLVM provides widely used compilation toolchain
- Instcombine pass: >1000 algebraic simplifications of expressions:

Practice Example 4: LLVM Expression Simplification

```
int foo(int z) {
  int x = 4*(z|101);
  return -256^x;
}
```

$\overset{*}{\dashrightarrow}$

```
define i32 @foo(i32) #0 {
  %2 = or i32 %0, 101
  %3 = mul nsw i32 4, %2
  %4 = xor i32 -256, %3
  ret i32 %4
}
```

$\xrightarrow{\text{Instcombine}}$

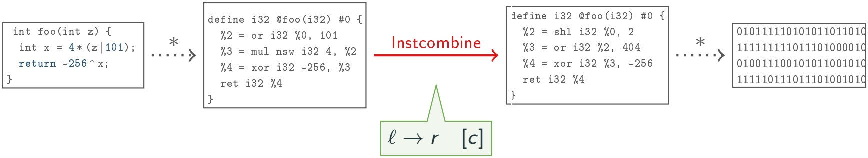
```
define i32 @foo(i32) #0 {
  %2 = shl i32 %0, 2
  %3 = or i32 %2, 404
  %4 = xor i32 %3, -256
  ret i32 %4
}
```

$\overset{*}{\dashrightarrow}$

```
0101111101010110110101
1111111110111010000010
0100111001010110010101
1111101110111010010101
```

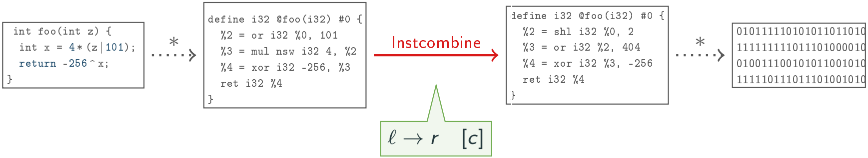
- LLVM provides widely used compilation toolchain
- Instcombine pass: >1000 algebraic simplifications of expressions: multiplications to shifts, reordering bitwise operations, . . .

Practice Example 4: LLVM Expression Simplification



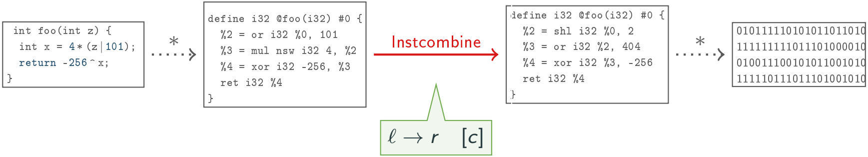
- LLVM provides widely used compilation toolchain
- Instcombine pass: >1000 algebraic simplifications of expressions: multiplications to shifts, reordering bitwise operations, ...
- applies "rewrite rules" with SMT side constraints

Practice Example 4: LLVM Expression Simplification



- LLVM provides widely used compilation toolchain
- Instcombine pass: >1000 algebraic simplifications of expressions: multiplications to shifts, reordering bitwise operations, ...
- applies "rewrite rules" with SMT side constraints
- optimization set is community maintained, interference unclear

Practice Example 4: LLVM Expression Simplification



- LLVM provides widely used compilation toolchain
- Instcombine pass: >1000 algebraic simplifications of expressions: multiplications to shifts, reordering bitwise operations, ...
- applies "rewrite rules" with SMT side constraints
- optimization set is community maintained, interference unclear
- termination is crucial

Course Content

Day 1

abstract rewriting, properties of abstract rewrite systems, Newman's Lemma, term rewriting

Day 2

termination, polynomial interpretations, lexicographic path order, Knuth-Bendix order, derivational complexity

Day 3

critical pairs, confluence, orthogonality, Knuth-Bendix completion

Course Content

Day 1

abstract rewriting, properties of abstract rewrite systems, Newman's Lemma, term rewriting

Day 2

termination, polynomial interpretations, lexicographic path order, Knuth-Bendix order, derivational complexity

Day 3

critical pairs, confluence, orthogonality, Knuth-Bendix completion

Outline

Motivating Examples

Abstract Rewriting

Term Rewriting

Definitions (Abstract Rewrite System)

- abstract rewrite system (ARS) consists of
  - carrier set A
  - binary relation $\to$ on A

Definitions (Abstract Rewrite System)

- abstract rewrite system (ARS) consists of
  - carrier set A
  - binary relation $\to$ on A

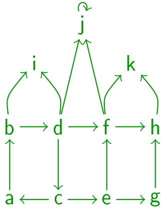
$$\text{ARS } \mathcal{A} = (A, \to)$$

- $A = \{\, \mathsf{a},\ \mathsf{b},\ \mathsf{c},\ \mathsf{d},\ \mathsf{e},\ \mathsf{f},\ \mathsf{g}\ \mathsf{h},\ \mathsf{i},\ \mathsf{j},\ \mathsf{k} \,\}$
- $\to\; = \left\{ \begin{array}{l} (\mathsf{a},\mathsf{b}),\ (\mathsf{b},\mathsf{d}),\ (\mathsf{b},\mathsf{i}),\ (\mathsf{c},\mathsf{a}),\ (\mathsf{c},\mathsf{e}), \\ (\mathsf{d},\mathsf{c}),\ (\mathsf{d},\mathsf{f}),\ (\mathsf{d},\mathsf{i}),\ (\mathsf{d},\mathsf{j}),\ (\mathsf{e},\mathsf{f}), \\ (\mathsf{e},\mathsf{g}),\ (\mathsf{f},\mathsf{h}),\ (\mathsf{f},\mathsf{j}),\ (\mathsf{f},\mathsf{k}),\ (\mathsf{g},\mathsf{h}), \\ (\mathsf{h},\mathsf{k}),\ (\mathsf{j},\mathsf{j}) \end{array} \right\}$

Definitions (Abstract Rewrite System)

- abstract rewrite system (ARS) consists of
  - carrier set A
  - binary relation $\to$ on A

ARS $\mathcal{A} = (A, \to)$

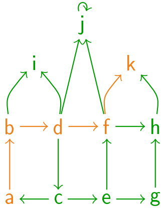


- $A = \{\, \mathsf{a},\ \mathsf{b},\ \mathsf{c},\ \mathsf{d},\ \mathsf{e},\ \mathsf{f},\ \mathsf{g}\ \mathsf{h},\ \mathsf{i},\ \mathsf{j},\ \mathsf{k} \,\}$
- $\to = \left\{ \begin{array}{l} (\mathsf{a},\mathsf{b}),\ (\mathsf{b},\mathsf{d}),\ (\mathsf{b},\mathsf{i}),\ (\mathsf{c},\mathsf{a}),\ (\mathsf{c},\mathsf{e}), \\ (\mathsf{d},\mathsf{c}),\ (\mathsf{d},\mathsf{f}),\ (\mathsf{d},\mathsf{i}),\ (\mathsf{d},\mathsf{j}),\ (\mathsf{e},\mathsf{f}), \\ (\mathsf{e},\mathsf{g}),\ (\mathsf{f},\mathsf{h}),\ (\mathsf{f},\mathsf{j}),\ (\mathsf{f},\mathsf{k}),\ (\mathsf{g},\mathsf{h}), \\ (\mathsf{h},\mathsf{k}),\ (\mathsf{j},\mathsf{j}) \end{array} \right\}$

Definitions (Abstract Rewrite System)

- abstract rewrite system (ARS) consists of
  - carrier set A
  - binary relation $\to$ on A

ARS $\mathcal{A} = (A, \to)$



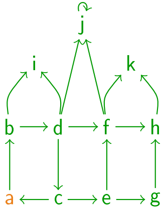
- $A = \{\, \mathsf{a},\ \mathsf{b},\ \mathsf{c},\ \mathsf{d},\ \mathsf{e},\ \mathsf{f},\ \mathsf{g}\ \mathsf{h},\ \mathsf{i},\ \mathsf{j},\ \mathsf{k} \,\}$
- $\to = \left\{ \begin{array}{l} (\mathsf{a},\mathsf{b}),\ (\mathsf{b},\mathsf{d}),\ (\mathsf{b},\mathsf{i}),\ (\mathsf{c},\mathsf{a}),\ (\mathsf{c},\mathsf{e}), \\ (\mathsf{d},\mathsf{c}),\ (\mathsf{d},\mathsf{f}),\ (\mathsf{d},\mathsf{i}),\ (\mathsf{d},\mathsf{j}),\ (\mathsf{e},\mathsf{f}), \\ (\mathsf{e},\mathsf{g}),\ (\mathsf{f},\mathsf{h}),\ (\mathsf{f},\mathsf{j}),\ (\mathsf{f},\mathsf{k}),\ (\mathsf{g},\mathsf{h}), \\ (\mathsf{h},\mathsf{k}),\ (\mathsf{j},\mathsf{j}) \end{array} \right\}$

- rewrite sequence
  - finite $\quad \mathsf{a} \to \mathsf{b} \to \mathsf{d} \to \mathsf{f} \to \mathsf{k}$

Definitions (Abstract Rewrite System)

- abstract rewrite system (ARS) consists of
  - carrier set A
  - binary relation $\to$ on A

ARS $\mathcal{A} = (A, \to)$



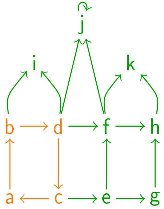
- $A = \{\, \mathsf{a},\ \mathsf{b},\ \mathsf{c},\ \mathsf{d},\ \mathsf{e},\ \mathsf{f},\ \mathsf{g}\ \mathsf{h},\ \mathsf{i},\ \mathsf{j},\ \mathsf{k} \,\}$
- $\to\; = \left\{ \begin{array}{l} (\mathsf{a},\mathsf{b}),\ (\mathsf{b},\mathsf{d}),\ (\mathsf{b},\mathsf{i}),\ (\mathsf{c},\mathsf{a}),\ (\mathsf{c},\mathsf{e}), \\ (\mathsf{d},\mathsf{c}),\ (\mathsf{d},\mathsf{f}),\ (\mathsf{d},\mathsf{i}),\ (\mathsf{d},\mathsf{j}),\ (\mathsf{e},\mathsf{f}), \\ (\mathsf{e},\mathsf{g}),\ (\mathsf{f},\mathsf{h}),\ (\mathsf{f},\mathsf{j}),\ (\mathsf{f},\mathsf{k}),\ (\mathsf{g},\mathsf{h}), \\ (\mathsf{h},\mathsf{k}),\ (\mathsf{j},\mathsf{j}) \end{array} \right\}$

- rewrite sequence
  - finite $\quad \mathsf{a} \to \mathsf{b} \to \mathsf{d} \to \mathsf{f} \to \mathsf{k}$
  - empty $\quad \mathsf{a}$

Definitions (Abstract Rewrite System)

- abstract rewrite system (ARS) consists of
  - carrier set A
  - binary relation $\to$ on A

$$\text{ARS } \mathcal{A} = (A, \to)$$



- $A = \{\, \mathsf{a},\ \mathsf{b},\ \mathsf{c},\ \mathsf{d},\ \mathsf{e},\ \mathsf{f},\ \mathsf{g}\ \mathsf{h},\ \mathsf{i},\ \mathsf{j},\ \mathsf{k} \,\}$
- $\to = \left\{ \begin{array}{l} (\mathsf{a},\mathsf{b}),\ (\mathsf{b},\mathsf{d}),\ (\mathsf{b},\mathsf{i}),\ (\mathsf{c},\mathsf{a}),\ (\mathsf{c},\mathsf{e}),\\ (\mathsf{d},\mathsf{c}),\ (\mathsf{d},\mathsf{f}),\ (\mathsf{d},\mathsf{i}),\ (\mathsf{d},\mathsf{j}),\ (\mathsf{e},\mathsf{f}),\\ (\mathsf{e},\mathsf{g}),\ (\mathsf{f},\mathsf{h}),\ (\mathsf{f},\mathsf{j}),\ (\mathsf{f},\mathsf{k}),\ (\mathsf{g},\mathsf{h}),\\ (\mathsf{h},\mathsf{k}),\ (\mathsf{j},\mathsf{j}) \end{array} \right\}$

- rewrite sequence
  - finite $\quad \mathsf{a} \to \mathsf{b} \to \mathsf{d} \to \mathsf{f} \to \mathsf{k}$
  - empty $\quad \mathsf{a}$
  - infinite $\quad \mathsf{a} \to \mathsf{b} \to \mathsf{d} \to \mathsf{c} \to \mathsf{a} \to \mathsf{b} \to \mathsf{d} \to \mathsf{c} \to \cdots$

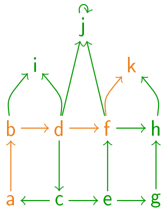
Definitions (Derived Relations)

- $\rightarrow^{+}$ transitive closure of $\rightarrow$

Definitions (Derived Relations)

- $\to^+$ transitive closure of $\to$

Example

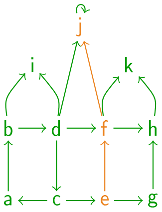


- $a \to^+ k$

Definitions (Derived Relations)

- $\to^+$ transitive closure of $\to$

Example

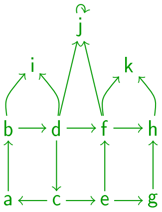


- $a \to^+ k$ $e \to^+ j$

Definitions (Derived Relations)

- $\to^+$ transitive closure of $\to$
- $\to^*$ transitive and reflexive closure of $\to$

Example

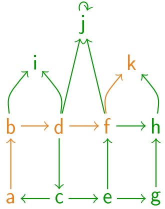


- $a \to^+ k \qquad e \to^+ j$

Definitions (Derived Relations)

- $\rightarrow^{+}$ transitive closure of $\rightarrow$
- $\rightarrow^{*}$ transitive and reflexive closure of $\rightarrow$

Example

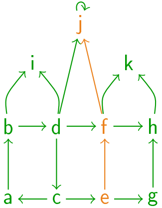


- $a \rightarrow^{+} k$ $e \rightarrow^{+} j$
- $a \rightarrow^{*} k$

Definitions (Derived Relations)

- $\rightarrow^+$ transitive closure of $\rightarrow$
- $\rightarrow^*$ transitive and reflexive closure of $\rightarrow$

Example

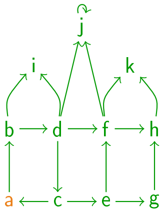


- $a \rightarrow^+ k$ $e \rightarrow^+ j$
- $a \rightarrow^* k$ $e \rightarrow^* j$

Definitions (Derived Relations)

- $\to^{+}$ transitive closure of $\to$
- $\to^{*}$ transitive and reflexive closure of $\to$

Example

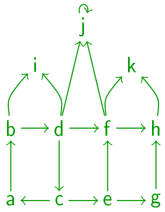


- $a \to^{+} k$ $e \to^{+} j$
- $a \to^{*} k$ $e \to^{*} j$ $a \to^{*} a$

Definitions (Derived Relations)

- $\to^+$ transitive closure of $\to$
- $\to^*$ transitive and reflexive closure of $\to$
- $\leftrightarrow^*$ conversion (equivalence relation generated by $\to$)

Example

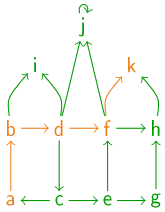


- $a \to^+ k$ $e \to^+ j$
- $a \to^* k$ $e \to^* j$ $a \to^* a$

Definitions (Derived Relations)

- $\to^+$ transitive closure of $\to$
- $\to^*$ transitive and reflexive closure of $\to$
- $\leftrightarrow^*$ conversion (equivalence relation generated by $\to$)

Example

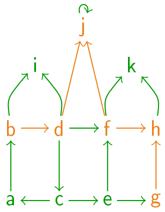


- $a \to^+ k$ $e \to^+ j$
- $a \to^* k$ $e \to^* j$ $a \to^* a$
- $a \leftrightarrow^* k$

Definitions (Derived Relations)

- $\rightarrow^{+}$ transitive closure of $\rightarrow$
- $\rightarrow^{*}$ transitive and reflexive closure of $\rightarrow$
- $\leftrightarrow^{*}$ conversion (equivalence relation generated by $\rightarrow$)

Example

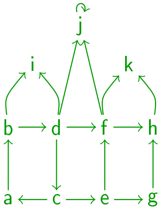


- $a \rightarrow^{+} k$ $e \rightarrow^{+} j$
- $a \rightarrow^{*} k$ $e \rightarrow^{*} j$ $a \rightarrow^{*} a$
- $a \leftrightarrow^{*} k$ $b \leftrightarrow^{*} g$

Definitions (Derived Relations)

- $\to^+$ transitive closure of $\to$
- $\to^*$ transitive and reflexive closure of $\to$
- $\leftrightarrow^*$ conversion (equivalence relation generated by $\to$)
- $\downarrow$ joinability $\downarrow = \to^* \cdot {}^*\!\leftarrow$

Example

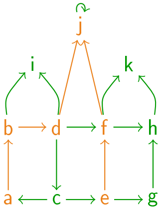


- $a \to^+ k$ $e \to^+ j$
- $a \to^* k$ $e \to^* j$ $a \to^* a$
- $a \leftrightarrow^* k$ $b \leftrightarrow^* g$

Definitions (Derived Relations)

- $\to^+$ transitive closure of $\to$
- $\to^*$ transitive and reflexive closure of $\to$
- $\leftrightarrow^*$ conversion (equivalence relation generated by $\to$)
- $\downarrow$ joinability $\downarrow = \to^* \cdot {}^*\!\leftarrow$

Example



- $a \to^+ k$ $e \to^+ j$
- $a \to^* k$ $e \to^* j$ $a \to^* a$
- $a \leftrightarrow^* k$ $b \leftrightarrow^* g$
- $a \downarrow e$

Definitions (Derived Relations)

- $\to^+$ transitive closure of $\to$
- $\to^*$ transitive and reflexive closure of $\to$
- $\leftrightarrow^*$ conversion (equivalence relation generated by $\to$)
- $\downarrow$ joinability $\downarrow = \to^* \cdot {}^*\!\leftarrow$

Example

$$
\begin{tikzcd}
& & & j \arrow[loop above] & & & \\
& i & & & & k & \\
b \arrow[rr] \arrow[ur, bend left] & & d \arrow[rr] \arrow[ul, bend right] \arrow[uur] \arrow[dd] & & f \arrow[rr] \arrow[uul] \arrow[ur, bend left] & & h \arrow[ul, bend right] \\
& & & & & & \\
a \arrow[uu] & & c \arrow[ll] \arrow[rr] & & e \arrow[uu] \arrow[rr] & & g \arrow[uu]
\end{tikzcd}
$$

- $a \to^+ k$ $e \to^+ j$
- $a \to^* k$ $e \to^* j$ $a \to^* a$
- $a \leftrightarrow^* k$ $b \leftrightarrow^* g$
- $a \downarrow e$ $e \downarrow g$

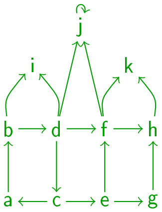
Definitions (Derived Relations)

- $\to^+$ transitive closure of $\to$
- $\to^*$ transitive and reflexive closure of $\to$
- $\leftrightarrow^*$ conversion (equivalence relation generated by $\to$)
- $\downarrow$ joinability $\downarrow = \to^* \cdot {}^*\!\leftarrow$

Definitions (Normal Forms)

- normal form is element x such that $x \not\to y$ for all y

Example



- $\mathsf{a} \to^+ \mathsf{k}$ $\mathsf{e} \to^+ \mathsf{j}$
- $\mathsf{a} \to^* \mathsf{k}$ $\mathsf{e} \to^* \mathsf{j}$ $\mathsf{a} \to^* \mathsf{a}$
- $\mathsf{a} \leftrightarrow^* \mathsf{k}$ $\mathsf{b} \leftrightarrow^* \mathsf{g}$
- $\mathsf{a} \downarrow \mathsf{e}$ $\mathsf{e} \downarrow \mathsf{g}$

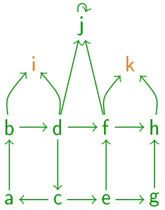
Definitions (Derived Relations)

- $\to^+$ transitive closure of $\to$
- $\to^*$ transitive and reflexive closure of $\to$
- $\leftrightarrow^*$ conversion (equivalence relation generated by $\to$)
- $\downarrow$ joinability $\downarrow = \to^* \cdot {}^*\!\leftarrow$

Definitions (Normal Forms)

- normal form is element x such that $x \not\to y$ for all y

Example



- a $\to^+$ k e $\to^+$ j
- a $\to^*$ k e $\to^*$ j a $\to^*$ a
- a $\leftrightarrow^*$ k b $\leftrightarrow^*$ g
- a $\downarrow$ e e $\downarrow$ g
- normal forms i, k

Definitions (Derived Relations)

- $\rightarrow^+$ transitive closure of $\rightarrow$
- $\rightarrow^*$ transitive and reflexive closure of $\rightarrow$
- $\leftrightarrow^*$ conversion (equivalence relation generated by $\rightarrow$)
- $\downarrow$ joinability $\downarrow = \rightarrow^* \cdot {}^*\!\leftarrow$

Definitions (Normal Forms)

- normal form is element x such that $x \not\rightarrow y$ for all y
- $x \rightarrow^! y$ if $x \rightarrow^* y$ for normal form y (x has normal form y)

Example

$$\begin{tikzcd}
 & & & \mathsf{j} \arrow[loop above] & & & \\
 & \mathsf{i} & & & & \mathsf{k} & \\
\mathsf{b} \arrow[ur, bend left] \arrow[rr] & & \mathsf{d} \arrow[ul] \arrow[uur] \arrow[rr] \arrow[d] & & \mathsf{f} \arrow[uul] \arrow[ur, bend left] \arrow[rr] & & \mathsf{h} \arrow[ul] \\
\mathsf{a} \arrow[u] & & \mathsf{c} \arrow[ll] \arrow[rr] & & \mathsf{e} \arrow[u] \arrow[rr] & & \mathsf{g} \arrow[u]
\end{tikzcd}$$

- $\mathsf{a} \rightarrow^+ \mathsf{k}$ $\mathsf{e} \rightarrow^+ \mathsf{j}$
- $\mathsf{a} \rightarrow^* \mathsf{k}$ $\mathsf{e} \rightarrow^* \mathsf{j}$ $\mathsf{a} \rightarrow^* \mathsf{a}$
- $\mathsf{a} \leftrightarrow^* \mathsf{k}$ $\mathsf{b} \leftrightarrow^* \mathsf{g}$
- $\mathsf{a} \downarrow \mathsf{e}$ $\mathsf{e} \downarrow \mathsf{g}$
- normal forms i, k

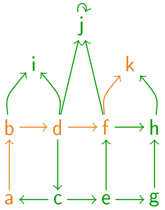
Definitions (Derived Relations)

- $\to^+$ transitive closure of $\to$
- $\to^*$ transitive and reflexive closure of $\to$
- $\leftrightarrow^*$ conversion (equivalence relation generated by $\to$)
- $\downarrow$ joinability $\downarrow = \to^* \cdot {}^*\!\!\leftarrow$

Definitions (Normal Forms)

- normal form is element x such that $x \not\to y$ for all y
- $x \to^! y$ if $x \to^* y$ for normal form y (x has normal form y)

Example



- a $\to^+$ k e $\to^+$ j
- a $\to^*$ k e $\to^*$ j a $\to^*$ a
- a $\leftrightarrow^*$ k b $\leftrightarrow^*$ g
- a $\downarrow$ e e $\downarrow$ g
- normal forms i, k
- a $\to^!$ k

Definitions (Derived Relations)

- $\to^+$ transitive closure of $\to$
- $\to^*$ transitive and reflexive closure of $\to$
- $\leftrightarrow^*$ conversion (equivalence relation generated by $\to$)
- $\downarrow$ joinability $\downarrow = \to^* \cdot {}^*\!\leftarrow$

Definitions (Normal Forms)

- normal form is element x such that $x \not\to y$ for all y
- $x \to^! y$ if $x \to^* y$ for normal form y (x has normal form y)

Terminology

- if $x \to^* y$ then x rewrites to y and y is reduct of x

Definitions (Derived Relations)

- $\to^+$ transitive closure of $\to$
- $\to^*$ transitive and reflexive closure of $\to$
- $\leftrightarrow^*$ conversion (equivalence relation generated by $\to$)
- $\downarrow$ joinability $\downarrow = \to^* \cdot {}^*\!\leftarrow$

Definitions (Normal Forms)

- normal form is element x such that $x \not\to y$ for all y
- $x \to^! y$ if $x \to^* y$ for normal form y (x has normal form y)

Terminology

- if $x \to^* y$ then x rewrites to y and y is reduct of x
- if $x \leftrightarrow^* y$ then x and y are convertible

Definitions

- SN strong normalization
  - no infinite rewrite sequences

Definitions

- SN strong normalization termination
  - no infinite rewrite sequences

Definitions

- SN strong normalization termination
  - no infinite rewrite sequences
- CR confluence
  - ${}^{*}\!\leftarrow \cdot \rightarrow^{*} \subseteq \downarrow$

Definitions

- SN strong normalization termination
  - no infinite rewrite sequences
- CR confluence
  - ${}^*\!\leftarrow \cdot \rightarrow^* \subseteq \downarrow$
  - $\forall\, a, b, c$

    $\exists\, d$

$$
\begin{tikzcd}
 & a \arrow[dl, "*"'] \arrow[dr, "*"] & \\
b \arrow[dr, dotted, "*"'] & & c \arrow[dl, dotted, "*"] \\
 & d &
\end{tikzcd}
$$

Definitions

- SN strong normalization termination
  - no infinite rewrite sequences
- CR confluence
  - $^*\leftarrow \cdot \rightarrow^* \subseteq \downarrow$
  - $\forall\, a,\ b,\ c$

    $\exists\, d$

$$\begin{tikzcd}[column sep=small] & a \arrow[dl, "*"'] \arrow[dr, "*"] & \\ b \arrow[dr, dotted, "*"'] & & c \arrow[dl, dotted, "*"] \\ & d & \end{tikzcd}$$

- WCR local confluence
  - $\leftarrow \cdot \rightarrow \subseteq \downarrow$

Definitions

- SN strong normalization termination
  - no infinite rewrite sequences
- CR confluence
  - ${}^{*}\!\leftarrow \cdot \rightarrow^{*} \subseteq \downarrow$
  - $\forall\, a, b, c$

$$\begin{tikzcd} & a \arrow[dl, "*"'] \arrow[dr, "*"] & \\ b \arrow[dr, dotted, "*"'] & & c \arrow[dl, dotted, "*"] \\ & d & \end{tikzcd}$$

    $\exists\, d$
- WCR local confluence
  - $\leftarrow \cdot \rightarrow \subseteq \downarrow$
  - $\forall\, a, b, c$

$$\begin{tikzcd} & a \arrow[dl] \arrow[dr] & \\ b \arrow[dr, dotted, "*"'] & & c \arrow[dl, dotted, "*"] \\ & d & \end{tikzcd}$$

    $\exists\, d$

Definitions

- SN strong normalization termination
  - no infinite rewrite sequences
- CR confluence
  - ${}^{*}\!\leftarrow \cdot \rightarrow^{*} \subseteq \downarrow$
  - $\forall\, a, b, c$

$$\begin{tikzcd} & a \arrow[dl, "*"'] \arrow[dr, "*"] & \\ b \arrow[dr, dotted, "*"'] & & c \arrow[dl, dotted, "*"] \\ & d & \end{tikzcd}$$

  $\exists\, d$

- WCR local confluence weak Church-Rosser property
  - $\leftarrow \cdot \rightarrow \subseteq \downarrow$
  - $\forall\, a, b, c$

$$\begin{tikzcd} & a \arrow[dl] \arrow[dr] & \\ b \arrow[dr, dotted, "*"'] & & c \arrow[dl, dotted, "*"] \\ & d & \end{tikzcd}$$

  $\exists\, d$

Definitions

- SN(x)   strong normalization   termination
  - no infinite rewrite sequences starting from $x \in A$
- CR(x)   confluence
  - ${}^*\!\leftarrow x \rightarrow^* \subseteq \downarrow$
  - $\forall$ b, c

$$\begin{tikzcd} & x \arrow[dl, "*"'] \arrow[dr, "*"] & \\ b \arrow[dr, dotted, "*"'] & & c \arrow[dl, dotted, "*"] \\ & d & \end{tikzcd}$$

$\exists\, d$

- WCR(x) local confluence   weak Church-Rosser property
  - $\leftarrow x \rightarrow \subseteq \downarrow$
  - $\forall$ b, c

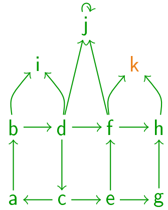
$$\begin{tikzcd} & x \arrow[dl] \arrow[dr] & \\ b \arrow[dr, dotted, "*"'] & & c \arrow[dl, dotted, "*"] \\ & d & \end{tikzcd}$$

$\exists\, d$

Example

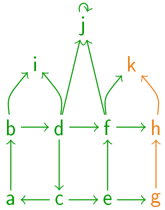
$$
\begin{tikzcd}[column sep=small]
 & & & j \arrow[loop above] & & & \\
 & i & & & & k & \\
b \arrow[ur, bend left] \arrow[rr] & & d \arrow[ul, bend right] \arrow[urr] \arrow[rr] \arrow[dd] & & f \arrow[ul] \arrow[ur, bend left] \arrow[rr] & & h \arrow[ul, bend right] \\
 & & & & & & \\
a \arrow[uu] & & c \arrow[ll] \arrow[rr] & & e \arrow[uu] \arrow[rr] & & g \arrow[uu]
\end{tikzcd}
$$

Example



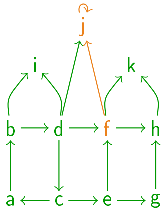
- SN(k)

Example



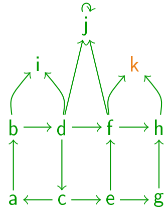
- SN(k) SN(g)

Example



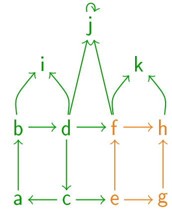
- $\mathrm{SN}(k)$ $\qquad$ $\mathrm{SN}(g)$ $\qquad$ $\neg\mathrm{SN}(f)$

Example



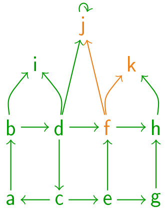
- $\mathsf{SN}(k)$ $\mathsf{SN}(g)$ $\neg\mathsf{SN}(f)$
- $\mathsf{WCR}(k)$

Example



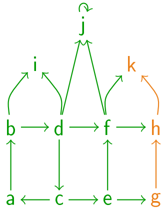
- $\mathsf{SN}(k)$ $\quad$ $\mathsf{SN}(g)$ $\quad$ $\neg\mathsf{SN}(f)$
- $\mathsf{WCR}(k)$ $\quad$ $\mathsf{WCR}(e)$

Example



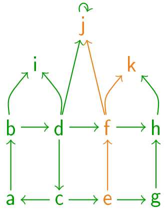
- $SN(k)$ $\quad SN(g)$ $\quad \neg SN(f)$
- $WCR(k)$ $\quad WCR(e)$ $\quad \neg WCR(f)$

Example



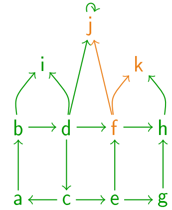
- SN(k) SN(g) ¬SN(f)
- WCR(k) WCR(e) ¬WCR(f)
- CR(g)

Example



- SN(k)   SN(g) $\neg$SN(f)
- WCR(k)   WCR(e) $\neg$WCR(f)
- CR(g) $\neg$CR(e)

Example



- $\mathsf{SN}(k)$ $\mathsf{SN}(g)$ $\neg\mathsf{SN}(f)$
- $\mathsf{WCR}(k)$ $\mathsf{WCR}(e)$ $\neg\mathsf{WCR}(f)$
- $\mathsf{CR}(g)$ $\neg\mathsf{CR}(e)$ $\neg\mathsf{CR}(f)$

Example

$$\begin{tikzcd}
 & & & \mathsf{j} \arrow[loop above] & & & \\
 & \mathsf{i} & & & & \mathsf{k} & \\
\mathsf{b} \arrow[ur] \arrow[rr] & & \mathsf{d} \arrow[ul] \arrow[uur] \arrow[rr] \arrow[d] & & \mathsf{f} \arrow[uul] \arrow[ur] \arrow[rr] & & \mathsf{h} \arrow[ul] \\
\mathsf{a} \arrow[u] & & \mathsf{c} \arrow[ll] \arrow[rr] & & \mathsf{e} \arrow[u] \arrow[rr] & & \mathsf{g} \arrow[u]
\end{tikzcd}$$

- $\mathsf{SN}(\mathsf{k})$ $\quad$ $\mathsf{SN}(\mathsf{g})$ $\quad$ $\neg\mathsf{SN}(\mathsf{f})$
- $\mathsf{WCR}(\mathsf{k})$ $\quad$ $\mathsf{WCR}(\mathsf{e})$ $\quad$ $\neg\mathsf{WCR}(\mathsf{f})$
- $\mathsf{CR}(\mathsf{g})$ $\quad$ $\neg\mathsf{CR}(\mathsf{e})$ $\quad$ $\neg\mathsf{CR}(\mathsf{f})$
- $\neg\mathsf{SN}(\mathcal{A})$ $\quad$ $\neg\mathsf{WCR}(\mathcal{A})$ $\quad$ $\neg\mathsf{CR}(\mathcal{A})$

Example

$$\begin{tikzcd}
 & \mathsf{i} & & \mathsf{j} \arrow[loop above] & & \mathsf{k} & \\
\mathsf{b} \arrow[ur, bend left] \arrow[rr] & & \mathsf{d} \arrow[ul, bend right] \arrow[ur] \arrow[rr] \arrow[d] & & \mathsf{f} \arrow[ul] \arrow[ur, bend left] \arrow[rr] & & \mathsf{h} \arrow[ul, bend right] \\
\mathsf{a} \arrow[u] & & \mathsf{c} \arrow[ll] \arrow[rr] & & \mathsf{e} \arrow[u] \arrow[rr] & & \mathsf{g} \arrow[u]
\end{tikzcd}$$

- $SN(k)$ $\quad$ $SN(g)$ $\quad$ $\neg SN(f)$
- $WCR(k)$ $\quad$ $WCR(e)$ $\quad$ $\neg WCR(f)$
- $CR(g)$ $\quad$ $\neg CR(e)$ $\quad$ $\neg CR(f)$
- $\neg SN(\mathcal{A})$ $\quad$ $\neg WCR(\mathcal{A})$ $\quad$ $\neg CR(\mathcal{A})$

Relationships between properties

- $CR \implies WCR$

Example

$$
\begin{tikzcd}
 & & & \mathsf{j} \arrow[loop above] & & & \\
 & \mathsf{i} & & & & \mathsf{k} & \\
\mathsf{b} \arrow[ur, bend left] \arrow[rr] & & \mathsf{d} \arrow[ul, bend right] \arrow[uur] \arrow[rr] \arrow[d] & & \mathsf{f} \arrow[uul] \arrow[ur, bend left] \arrow[rr] & & \mathsf{h} \arrow[ul, bend right] \\
\mathsf{a} \arrow[u] & & \mathsf{c} \arrow[ll] \arrow[rr] & & \mathsf{e} \arrow[u] \arrow[rr] & & \mathsf{g} \arrow[u]
\end{tikzcd}
$$

- SN(k) SN(g) ¬SN(f)
- WCR(k) WCR(e) ¬WCR(f)
- CR(g) ¬CR(e) ¬CR(f)
- $\neg\mathrm{SN}(\mathcal{A})$ $\neg\mathrm{WCR}(\mathcal{A})$ $\neg\mathrm{CR}(\mathcal{A})$

Relationships between properties

- CR $\Longrightarrow$ WCR
- WCR $\not\Longrightarrow$ CR

Example

$$
\begin{tikzcd}
 & & & j \arrow[loop above] & & & \\
 & i & & & & k & \\
b \arrow[ur] \arrow[rr] & & d \arrow[ul] \arrow[uur] \arrow[rr] \arrow[d] & & f \arrow[uul] \arrow[ur] \arrow[rr] & & h \arrow[ul] \\
a \arrow[u] & & c \arrow[ll] \arrow[rr] & & e \arrow[u] \arrow[rr] & & g \arrow[u]
\end{tikzcd}
$$

- $\mathsf{SN}(k)$ $\mathsf{SN}(g)$ $\neg\mathsf{SN}(f)$
- $\mathsf{WCR}(k)$ $\mathsf{WCR}(e)$ $\neg\mathsf{WCR}(f)$
- $\mathsf{CR}(g)$ $\neg\mathsf{CR}(e)$ $\neg\mathsf{CR}(f)$
- $\neg\mathsf{SN}(\mathcal{A})$ $\neg\mathsf{WCR}(\mathcal{A})$ $\neg\mathsf{CR}(\mathcal{A})$

Relationships between properties

- $\mathsf{CR} \implies \mathsf{WCR}$
- $\mathsf{WCR} \not\Longrightarrow \mathsf{CR}$

$$
\begin{tikzcd}
a & b \arrow[l] \arrow[r, bend left] & c \arrow[l, bend left] \arrow[r] & d
\end{tikzcd}
$$

Example

$$
\begin{tikzcd}
 & & & j \arrow[loop above] & & & \\
 & i & & & & k & \\
b \arrow[ur] \arrow[rr] & & d \arrow[ul] \arrow[uur] \arrow[rr] \arrow[d] & & f \arrow[uul] \arrow[ur] \arrow[rr] & & h \arrow[ul] \\
a \arrow[u] & & c \arrow[ll] \arrow[rr] & & e \arrow[u] \arrow[rr] & & g \arrow[u]
\end{tikzcd}
$$

- SN(k)   SN(g) $\neg$SN(f)
- WCR(k)   WCR(e) $\neg$WCR(f)
- CR(g) $\neg$CR(e) $\neg$CR(f)
- $\neg\mathrm{SN}(\mathcal{A})$ $\neg\mathrm{WCR}(\mathcal{A})$ $\neg\mathrm{CR}(\mathcal{A})$

Relationships between properties

- $\mathrm{CR} \implies \mathrm{WCR}$
- $\mathrm{WCR} \not\Longrightarrow \mathrm{CR}$
- $\mathrm{SN}\ \&\ \mathrm{WCR} \implies \mathrm{CR}$

$$
\begin{tikzcd}
a & b \arrow[l] \arrow[r, bend left] & c \arrow[l, bend left] \arrow[r] & d
\end{tikzcd}
$$

Example

$$
\begin{tikzcd}
 & & & j \arrow[loop above] & & & \\
 & i & & & & k & \\
b \arrow[rr] \arrow[ur, bend left] & & d \arrow[rr] \arrow[ul, bend right] \arrow[uur] \arrow[d] & & f \arrow[rr] \arrow[uul] \arrow[ur, bend left] & & h \arrow[ul, bend right] \\
a \arrow[u] & & c \arrow[ll] \arrow[rr] & & e \arrow[rr] \arrow[u] & & g \arrow[u]
\end{tikzcd}
$$

- SN(k)   SN(g) $\neg$SN(f)
- WCR(k)   WCR(e) $\neg$WCR(f)
- CR(g) $\neg$CR(e) $\neg$CR(f)
- $\neg\mathrm{SN}(\mathcal{A})$ $\neg\mathrm{WCR}(\mathcal{A})$ $\neg\mathrm{CR}(\mathcal{A})$

Relationships between properties

- $\mathrm{CR} \implies \mathrm{WCR}$
- $\mathrm{WCR} \not\Longrightarrow \mathrm{CR}$
- SN & $\mathrm{WCR} \implies \mathrm{CR}$

$$
\begin{tikzcd}
a & b \arrow[l] \arrow[r, bend left] & c \arrow[l, bend left] \arrow[r] & d
\end{tikzcd}
$$

Newman's Lemma

Definitions (Terms)

- signature $\mathcal{F}$ function symbols with arities

Definitions (Terms)

- signature $\mathcal{F}$ function symbols with arities

Example

- $\mathcal{F} = \{\,0, \mathsf{s}, +\,\}$ for constant 0 (arity 0), unary s, and binary $+$

Definitions (Terms)

- signature $\mathcal{F}$ function symbols with arities
- variables $\mathcal{V}$ $\mathcal{F} \cap \mathcal{V} = \varnothing$ and $\mathcal{V}$ is infinite

Definitions (Terms)

- signature $\mathcal{F}$ function symbols with arities
- variables $\mathcal{V}$ $\mathcal{F} \cap \mathcal{V} = \varnothing$ and $\mathcal{V}$ is infinite

Example

- $\mathcal{V} = \{ x, y, z, \ldots \}$

Definitions (Terms)

- signature $\mathcal{F}$ function symbols with arities
- variables $\mathcal{V}$ $\mathcal{F} \cap \mathcal{V} = \varnothing$ and $\mathcal{V}$ is infinite
- terms $\mathcal{T}(\mathcal{F}, \mathcal{V})$

Definitions (Terms)

- signature $\mathcal{F}$ function symbols with arities
- variables $\mathcal{V}$ $\mathcal{F} \cap \mathcal{V} = \varnothing$ and $\mathcal{V}$ is infinite
- terms $\mathcal{T}(\mathcal{F}, \mathcal{V})$ smallest set such that
  - $\mathcal{V} \subseteq \mathcal{T}(\mathcal{F}, \mathcal{V})$

Definitions (Terms)

- signature $\mathcal{F}$ function symbols with arities
- variables $\mathcal{V}$ $\mathcal{F} \cap \mathcal{V} = \varnothing$ and $\mathcal{V}$ is infinite
- terms $\mathcal{T}(\mathcal{F}, \mathcal{V})$ smallest set such that
  - $\mathcal{V} \subseteq \mathcal{T}(\mathcal{F}, \mathcal{V})$
  - if $\left.\begin{array}{l} f \in \mathcal{F} \text{ has arity } n \\ t_1, \ldots, t_n \in \mathcal{T}(\mathcal{F}, \mathcal{V}) \end{array}\right\}$

Definitions (Terms)

- signature $\mathcal{F}$ function symbols with arities
- variables $\mathcal{V}$ $\mathcal{F} \cap \mathcal{V} = \varnothing$ and $\mathcal{V}$ is infinite
- terms $\mathcal{T}(\mathcal{F},\mathcal{V})$ smallest set such that
  - $\mathcal{V} \subseteq \mathcal{T}(\mathcal{F},\mathcal{V})$
  - if $\left.\begin{array}{l} f \in \mathcal{F} \text{ has arity } n \\ t_1,\ldots,t_n \in \mathcal{T}(\mathcal{F},\mathcal{V}) \end{array}\right\}$ then $f(t_1,\ldots,t_n) \in \mathcal{T}(\mathcal{F},\mathcal{V})$

Definitions (Terms)

- signature $\mathcal{F}$ function symbols with arities
- variables $\mathcal{V}$ $\mathcal{F} \cap \mathcal{V} = \varnothing$ and $\mathcal{V}$ is infinite
- terms $\mathcal{T}(\mathcal{F}, \mathcal{V})$ smallest set such that
  - $\mathcal{V} \subseteq \mathcal{T}(\mathcal{F}, \mathcal{V})$
  - if $\left.\begin{array}{l} f \in \mathcal{F} \text{ has arity } n \\ t_1, \ldots, t_n \in \mathcal{T}(\mathcal{F}, \mathcal{V}) \end{array}\right\}$ then $f(t_1, \ldots, t_n) \in \mathcal{T}(\mathcal{F}, \mathcal{V})$

Example

- $\mathcal{T}(\mathcal{F}, \mathcal{V}) = \{\, x,\ y,\ z,\ 0,\ \mathsf{s}(0),\ \mathsf{s}(\mathsf{s}(0)),\ \mathsf{s}(0) + x,\ \mathsf{s}(x) + y,\ \ldots \,\}$

Definitions (Terms)

- signature $\mathcal{F}$ — function symbols with arities
- variables $\mathcal{V}$ — $\mathcal{F} \cap \mathcal{V} = \varnothing$ and $\mathcal{V}$ is infinite
- terms $\mathcal{T}(\mathcal{F}, \mathcal{V})$ — smallest set such that
  - $\mathcal{V} \subseteq \mathcal{T}(\mathcal{F}, \mathcal{V})$
  - if $\left.\begin{array}{l} f \in \mathcal{F} \text{ has arity } n \\ t_1, \ldots, t_n \in \mathcal{T}(\mathcal{F}, \mathcal{V}) \end{array}\right\}$ then $f(t_1, \ldots, t_n) \in \mathcal{T}(\mathcal{F}, \mathcal{V})$
- ground terms $\mathcal{T}(\mathcal{F})$ — smallest set such that
  - if $\left.\begin{array}{l} f \in \mathcal{F} \text{ has arity } n \\ t_1, \ldots, t_n \in \mathcal{T}(\mathcal{F}) \end{array}\right\}$ then $f(t_1, \ldots, t_n) \in \mathcal{T}(\mathcal{F})$

Definitions (Terms)

- signature $\mathcal{F}$ function symbols with arities
- variables $\mathcal{V}$ $\mathcal{F} \cap \mathcal{V} = \varnothing$ and $\mathcal{V}$ is infinite
- terms $\mathcal{T}(\mathcal{F}, \mathcal{V})$ smallest set such that
  - $\mathcal{V} \subseteq \mathcal{T}(\mathcal{F}, \mathcal{V})$
  - if $\left.\begin{array}{l} f \in \mathcal{F} \text{ has arity } n \\ t_1, \ldots, t_n \in \mathcal{T}(\mathcal{F}, \mathcal{V}) \end{array}\right\}$ then $f(t_1, \ldots, t_n) \in \mathcal{T}(\mathcal{F}, \mathcal{V})$
- ground terms $\mathcal{T}(\mathcal{F})$ smallest set such that
  - if $\left.\begin{array}{l} f \in \mathcal{F} \text{ has arity } n \\ t_1, \ldots, t_n \in \mathcal{T}(\mathcal{F}) \end{array}\right\}$ then $f(t_1, \ldots, t_n) \in \mathcal{T}(\mathcal{F})$

Example

- $\mathcal{T}(\mathcal{F}) = \{\, 0, \mathsf{s}(0), \mathsf{s}(\mathsf{s}(0)), \mathsf{s}(0) + 0, \mathsf{s}(0) + \mathsf{s}(0), \ldots \}$

Definitions (Context)

- context is term with one hole

Definitions (Context)

- context is term with one hole, i.e., element of $\mathcal{T}(\mathcal{F} \cup \{\Box\}, \mathcal{V})$ that contains exactly one occurrence of special constant $\Box$

Definitions (Context)

- context is term with one hole, i.e., element of $\mathcal{T}(\mathcal{F} \cup \{\Box\}, \mathcal{V})$ that contains exactly one occurrence of special constant $\Box$

Example (Examples)

- $\Box$ $\quad$ $\mathsf{s}(0) + \mathsf{s}(\mathsf{s}(\Box))$ $\quad$ $\Box + x$

Definitions (Context)

- context is term with one hole, i.e., element of $\mathcal{T}(\mathcal{F} \cup \{\Box\}, \mathcal{V})$ that contains exactly one occurrence of special constant $\Box$
- $C[t]$ denotes result of replacing hole in context C by term t

Example (Examples)

- $\Box$ $\qquad$ $\mathsf{s}(0) + \mathsf{s}(\mathsf{s}(\Box))$ $\qquad$ $\Box + x$

Definitions (Context)

- context is term with one hole, i.e., element of $\mathcal{T}(\mathcal{F} \cup \{\square\}, \mathcal{V})$ that contains exactly one occurrence of special constant $\square$
- $C[t]$ denotes result of replacing hole in context C by term t

Example (Examples)

- $\square \qquad \mathsf{s}(0) + \mathsf{s}(\mathsf{s}(\square)) \qquad \square + x$
- $\square[\,\mathsf{s}(0)\,] = \mathsf{s}(0) \qquad (\square + x)[\,0 + x\,] = (0 + x) + x$

Definitions

- substitution is mapping $\sigma \colon \mathcal{V} \to \mathcal{T}(\mathcal{F}, \mathcal{V})$ such that its domain

$$\mathcal{D}om(\sigma) = \{\, x \in \mathcal{V} \mid \sigma(x) \neq x \,\}$$

  is finite

Definitions

- substitution is mapping $\sigma\colon \mathcal{V} \to \mathcal{T}(\mathcal{F}, \mathcal{V})$ such that its domain

$$\mathcal{D}om(\sigma) = \{\, x \in \mathcal{V} \mid \sigma(x) \neq x \,\}$$

  is finite

Definitions

- substitution is mapping $\sigma\colon \mathcal{V} \to \mathcal{T}(\mathcal{F}, \mathcal{V})$ such that its domain

$$\mathcal{D}om(\sigma) = \{\, x \in \mathcal{V} \mid \sigma(x) \neq x \,\}$$

  is finite

- application of substitution σ to term t

$$t\sigma = \begin{cases} \sigma(t) & \text{if } t \in \mathcal{V} \\ f(t_1\sigma, \ldots, t_n\sigma) & \text{if } t = f(t_1, \ldots, t_n) \end{cases}$$

Definitions

- substitution is mapping $\sigma\colon \mathcal{V} \to \mathcal{T}(\mathcal{F},\mathcal{V})$ such that its domain

$$\mathcal{D}om(\sigma) = \{\, x \in \mathcal{V} \mid \sigma(x) \neq x \,\}$$

  is finite

- application of substitution σ to term t

$$t\sigma = \begin{cases} \sigma(t) & \text{if } t \in \mathcal{V} \\ f(t_1\sigma, \dots, t_n\sigma) & \text{if } t = f(t_1, \dots, t_n) \end{cases}$$

Example

$t = x + \mathsf{s}(y + z)$ $\qquad$ $\sigma = \{\, x \mapsto \mathsf{s}(y),\ y \mapsto x + \mathsf{s}(0) \,\}$

Definitions

- substitution is mapping $\sigma\colon \mathcal{V} \to \mathcal{T}(\mathcal{F}, \mathcal{V})$ such that its domain

$$\mathcal{D}om(\sigma) = \{\, x \in \mathcal{V} \mid \sigma(x) \neq x \,\}$$

  is finite

- application of substitution σ to term t

$$t\sigma = \begin{cases} \sigma(t) & \text{if } t \in \mathcal{V} \\ f(t_1\sigma, \dots, t_n\sigma) & \text{if } t = f(t_1, \dots, t_n) \end{cases}$$

Example

$t = x + \mathsf{s}(y + z)$ $\qquad \sigma = \{\, x \mapsto \mathsf{s}(y),\ y \mapsto x + \mathsf{s}(0) \,\}$

$t\sigma = \mathsf{s}(y) + \mathsf{s}((x + \mathsf{s}(0)) + z)$

Definition (Matching Problem)

t is instance of s

instance: terms s, t

question: $\exists$ substitution σ such that $s\sigma = t$?

Definition (Matching Problem)

t is instance of s

instance: terms s, t

question: $\exists$ substitution σ such that $s\sigma = t$?

Definition (Matching Problem)

t is instance of s

instance: terms s, t

question: $\exists$ substitution σ such that $s\sigma = t$?

Example

- $\mathsf{s}(0) + \mathsf{s}(\mathsf{s}(0))$ is instance of $x + \mathsf{s}(x)$

Definition (Matching Problem)

t is instance of s

instance: terms s, t

question: $\exists$ substitution σ such that $s\sigma = t$?

Example

- $\mathsf{s}(0) + \mathsf{s}(\mathsf{s}(0))$ is instance of $x + \mathsf{s}(x)$
- $\mathsf{s}(0) + (\mathsf{s}(0) + \mathsf{s}(0))$ is no instance of $x + (\mathsf{s}(x) + y)$

Definition (Matching Problem)

t is instance of s

instance: terms s, t

question: $\exists$ substitution σ such that $s\sigma = t$?

Example

- $\mathsf{s}(0) + \mathsf{s}(\mathsf{s}(0))$ is instance of $x + \mathsf{s}(x)$
- $\mathsf{s}(0) + (\mathsf{s}(0) + \mathsf{s}(0))$ is no instance of $x + (\mathsf{s}(x) + y)$

Matching Algorithm

1. start with $\{\, s \mapsto t \,\}$

Definition (Matching Problem)

t is instance of s

instance: terms s, t

question: $\exists$ substitution σ such that $s\sigma = t$?

Example

- $\mathsf{s}(0) + \mathsf{s}(\mathsf{s}(0))$ is instance of $x + \mathsf{s}(x)$
- $\mathsf{s}(0) + (\mathsf{s}(0) + \mathsf{s}(0))$ is no instance of $x + (\mathsf{s}(x) + y)$

Matching Algorithm

1. start with $\{\, s \mapsto t \,\}$
2. repeatedly apply following transformation rules

$$\{\, f(s_1, \ldots, s_n) \mapsto f(t_1, \ldots, t_n) \,\} \uplus S \implies \{\, s_1 \mapsto t_1, \ \ldots, \ s_n \mapsto t_n \,\} \cup S$$

Definition (Matching Problem)

t is instance of s

instance: terms s, t

question: $\exists$ substitution σ such that $s\sigma = t$?

Example

- $\mathsf{s}(0) + \mathsf{s}(\mathsf{s}(0))$ is instance of $x + \mathsf{s}(x)$
- $\mathsf{s}(0) + (\mathsf{s}(0) + \mathsf{s}(0))$ is no instance of $x + (\mathsf{s}(x) + y)$

Matching Algorithm

1. start with $\{\, s \mapsto t \,\}$
2. repeatedly apply following transformation rules

$$\{\, f(s_1, \ldots, s_n) \mapsto f(t_1, \ldots, t_n) \,\} \uplus S \implies \{\, s_1 \mapsto t_1, \ \ldots, \ s_n \mapsto t_n \,\} \cup S$$

$$\{\, f(s_1, \ldots, s_n) \mapsto g(t_1, \ldots, t_n) \,\} \uplus S \implies \bot \qquad \text{if } f \neq g$$

Definition (Matching Problem)

t is instance of s

instance: terms s, t

question: $\exists$ substitution σ such that $s\sigma = t$?

Example

- $\mathsf{s}(0) + \mathsf{s}(\mathsf{s}(0))$ is instance of $x + \mathsf{s}(x)$
- $\mathsf{s}(0) + (\mathsf{s}(0) + \mathsf{s}(0))$ is no instance of $x + (\mathsf{s}(x) + y)$

Matching Algorithm

1. start with $\{\, s \mapsto t \,\}$
2. repeatedly apply following transformation rules

$$\{\, f(s_1, \ldots, s_n) \mapsto f(t_1, \ldots, t_n) \,\} \uplus S \implies \{\, s_1 \mapsto t_1,\ \ldots,\ s_n \mapsto t_n \,\} \cup S$$

$$\{\, f(s_1, \ldots, s_n) \mapsto g(t_1, \ldots, t_n) \,\} \uplus S \implies \bot \qquad \text{if } f \neq g$$

$$\{\, f(s_1, \ldots, s_n) \mapsto x \,\} \uplus S \implies \bot$$

Definition (Matching Problem)

t is instance of s

instance: terms s, t

question: $\exists$ substitution σ such that $s\sigma = t$?

Example

- $\mathsf{s}(0) + \mathsf{s}(\mathsf{s}(0))$ is instance of $x + \mathsf{s}(x)$
- $\mathsf{s}(0) + (\mathsf{s}(0) + \mathsf{s}(0))$ is no instance of $x + (\mathsf{s}(x) + y)$

Matching Algorithm

1. start with $\{\, s \mapsto t \,\}$
2. repeatedly apply following transformation rules

$$\{\, f(s_1, \ldots, s_n) \mapsto f(t_1, \ldots, t_n) \,\} \uplus S \implies \{\, s_1 \mapsto t_1,\ \ldots,\ s_n \mapsto t_n \,\} \cup S$$

$$\{\, f(s_1, \ldots, s_n) \mapsto g(t_1, \ldots, t_n) \,\} \uplus S \implies \bot \qquad \text{if } f \neq g$$

$$\{\, f(s_1, \ldots, s_n) \mapsto x \,\} \uplus S \implies \bot$$

$$\{\, x \mapsto t \,\} \uplus S \implies \bot \qquad \text{if } x \mapsto t' \text{ in } S \text{ with } t \neq t'$$

Definitions (Term Rewrite System)

- rewrite rule $\ell \to r$ is pair of terms (ℓ, r) such that
  - $\ell \notin \mathcal{V}$
  - $\mathcal{V}ar(r) \subseteq \mathcal{V}ar(\ell)$

Definitions (Term Rewrite System)

- rewrite rule $\ell \to r$ is pair of terms (ℓ, r) such that
  - $\ell \notin \mathcal{V}$
  - $\mathcal{V}ar(r) \subseteq \mathcal{V}ar(\ell)$
- term rewrite system (TRS) consists of
  - $\mathcal{F}$ signature
  - $\mathcal{R}$ finite set of rewrite rules between terms in $\mathcal{T}(\mathcal{F}, \mathcal{V})$

Definitions (Term Rewrite System)

- rewrite rule $\ell \to r$ is pair of terms (ℓ, r) such that
  - $\ell \notin \mathcal{V}$
  - $\mathcal{V}ar(r) \subseteq \mathcal{V}ar(\ell)$
- term rewrite system (TRS) consists of
  - $\mathcal{F}$ signature
  - $\mathcal{R}$ finite set of rewrite rules between terms in $\mathcal{T}(\mathcal{F}, \mathcal{V})$

Example

- TRS $\mathcal{R}$ for $\mathcal{F} = \{0, \mathsf{s}, +\}$:

$$0 + x \to x \qquad\qquad \mathsf{s}(x) + y \to \mathsf{s}(x + y)$$

Definition (Rewriting)

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

Definition (Rewriting)

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

$s \rightarrow_{\mathcal{R}} t \quad \iff$

Definition (Rewriting)

binary relation $\to_{\mathcal{R}}$ on $\mathcal{T}(\mathcal{F},\mathcal{V})$ for every TRS $\mathcal{R}$:

$$s \to_{\mathcal{R}} t \quad\iff\quad \begin{array}{l} \exists \text{ context } C \\ \exists\ \ell \to r \in \mathcal{R} \\ \exists \text{ substitution } \sigma \end{array}$$

Definition (Rewriting)

binary relation $\to_{\mathcal{R}}$ on $\mathcal{T}(\mathcal{F},\mathcal{V})$ for every TRS $\mathcal{R}$:

$$s \to_{\mathcal{R}} t \quad \iff \quad \begin{array}{l} \exists \text{ context } C \\ \exists\, \ell \to r \in \mathcal{R} \\ \exists \text{ substitution } \sigma \end{array} \quad \text{with} \quad s = C[\ell\sigma]$$

Definition (Rewriting)

binary relation $\to_\mathcal{R}$ on $\mathcal{T}(\mathcal{F},\mathcal{V})$ for every TRS $\mathcal{R}$:

$$s \to_\mathcal{R} t \quad\Longleftrightarrow\quad \begin{array}{l} \exists \text{ context } C \\ \exists\, \ell \to r \in \mathcal{R} \\ \exists \text{ substitution } \sigma \end{array} \quad \text{with} \quad s = C[\ell\sigma] \quad \text{redex}$$

Definition (Rewriting)

binary relation $\to_{\mathcal{R}}$ on $\mathcal{T}(\mathcal{F},\mathcal{V})$ for every TRS $\mathcal{R}$:

$$s \to_{\mathcal{R}} t \quad\iff\quad \begin{array}{l} \exists \text{ context } C \\ \exists\, \ell \to r \in \mathcal{R} \\ \exists \text{ substitution } \sigma \end{array} \quad \text{with} \quad \begin{array}{rcll} s & = & C[\ell\sigma] & \text{redex} \\ t & = & C[r\sigma] & \end{array}$$

Definition (Rewriting)

binary relation $\to_{\mathcal{R}}$ on $\mathcal{T}(\mathcal{F},\mathcal{V})$ for every TRS $\mathcal{R}$:

$$s \to_{\mathcal{R}} t \quad \iff \quad \begin{array}{l} \exists\ \text{context}\ C \\ \exists\ \ell \to r \in \mathcal{R} \\ \exists\ \text{substitution}\ \sigma \end{array} \quad \text{with} \quad \begin{array}{rcll} s & = & C[\ell\sigma] & \text{redex} \\ t & = & C[r\sigma] & \end{array}$$

Example

Definition (Rewriting)

binary relation $\to_{\mathcal{R}}$ on $\mathcal{T}(\mathcal{F},\mathcal{V})$ for every TRS $\mathcal{R}$:

$$s \to_{\mathcal{R}} t \quad \Longleftrightarrow \quad \begin{array}{l} \exists \text{ context } C \\ \exists\, \ell \to r \in \mathcal{R} \\ \exists \text{ substitution } \sigma \end{array} \quad \text{with} \quad \begin{array}{rcl} s & = & C[\ell\sigma] \quad \text{redex} \\ t & = & C[r\sigma] \end{array}$$

Example

- TRS $\mathcal{R}$ for $\mathcal{F} = \{0, \mathsf{s}, +\}$:

$$0 + x \to x \qquad\qquad \mathsf{s}(x) + y \to \mathsf{s}(x + y)$$

Definition (Rewriting)

binary relation $\to_{\mathcal{R}}$ on $\mathcal{T}(\mathcal{F},\mathcal{V})$ for every TRS $\mathcal{R}$:

$$s \to_{\mathcal{R}} t \quad \Longleftrightarrow \quad \begin{array}{l} \exists \text{ context } C \\ \exists\, \ell \to r \in \mathcal{R} \\ \exists \text{ substitution } \sigma \end{array} \quad \text{with} \quad \begin{array}{rcll} s & = & C[\ell\sigma] & \text{redex} \\ t & = & C[r\sigma] & \end{array}$$

Example

- TRS $\mathcal{R}$ for $\mathcal{F} = \{0, \mathsf{s}, +\}$:

  $$0 + x \to x \qquad\qquad \mathsf{s}(x) + y \to \mathsf{s}(x + y)$$

- rewrite step

  $$\mathsf{s}(\mathsf{s}(0)) + \mathsf{s}(\mathsf{s}(\mathsf{s}(0)))$$

Definition (Rewriting)

binary relation $\to_{\mathcal{R}}$ on $\mathcal{T}(\mathcal{F},\mathcal{V})$ for every TRS $\mathcal{R}$:

$$s \to_{\mathcal{R}} t \quad \Longleftrightarrow \quad \begin{array}{l} \exists \text{ context } C \\ \exists\ \ell \to r \in \mathcal{R} \\ \exists \text{ substitution } \sigma \end{array} \quad \text{with} \quad \begin{array}{rcl} s & = & C[\ell\sigma] \quad \text{redex} \\ t & = & C[r\sigma] \end{array}$$

Example

- TRS $\mathcal{R}$ for $\mathcal{F} = \{0, \mathsf{s}, +\}$:

  $0 + x \ \to\ x \qquad\qquad \mathsf{s}(x) + y \ \to\ \mathsf{s}(x + y)$

- rewrite step

  $\mathsf{s}(\mathsf{s}(0)) + \mathsf{s}(\mathsf{s}(\mathsf{s}(0))) \ \to_{\mathcal{R}}\ \mathsf{s}(\ \mathsf{s}(0) + \mathsf{s}(\mathsf{s}(\mathsf{s}(0)))\ ) \qquad\qquad C = \square$

Definition (Rewriting)

binary relation $\to_{\mathcal{R}}$ on $\mathcal{T}(\mathcal{F}, \mathcal{V})$ for every TRS $\mathcal{R}$:

$$s \to_{\mathcal{R}} t \quad \iff \quad \begin{array}{l} \exists \text{ context } C \\ \exists\, \ell \to r \in \mathcal{R} \\ \exists \text{ substitution } \sigma \end{array} \quad \text{with} \quad \begin{array}{rcl} s & = & C[\ell\sigma] \quad \text{redex} \\ t & = & C[r\sigma] \end{array}$$

Example

- TRS $\mathcal{R}$ for $\mathcal{F} = \{0, \mathsf{s}, +\}$:

  $$0 + x \to x \qquad \mathsf{s}(x) + y \to \mathsf{s}(x + y)$$

- rewrite sequence

  $$\mathsf{s}(\mathsf{s}(0)) + \mathsf{s}(\mathsf{s}(\mathsf{s}(0))) \to_{\mathcal{R}} \mathsf{s}(\, \mathsf{s}(0) + \mathsf{s}(\mathsf{s}(\mathsf{s}(0))) \,) \qquad C = \square$$

Definition (Rewriting)

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

$$s \rightarrow_{\mathcal{R}} t \quad\Longleftrightarrow\quad \begin{array}{l} \exists \text{ context } C \\ \exists\, \ell \rightarrow r \in \mathcal{R} \\ \exists \text{ substitution } \sigma \end{array} \quad \text{with} \quad \begin{array}{rcll} s & = & C[\ell\sigma] & \text{redex} \\ t & = & C[r\sigma] & \end{array}$$

Example

- TRS $\mathcal{R}$ for $\mathcal{F} = \{0, \mathsf{s}, +\}$:

$$0 + x \rightarrow x \qquad\qquad \mathsf{s}(x) + y \rightarrow \mathsf{s}(x + y)$$

- rewrite sequence

$$\begin{array}{rll} \mathsf{s}(\mathsf{s}(0)) + \mathsf{s}(\mathsf{s}(\mathsf{s}(0))) & \rightarrow_{\mathcal{R}} \mathsf{s}(\,\mathsf{s}(0) + \mathsf{s}(\mathsf{s}(\mathsf{s}(0)))\,) & C = \Box \\ & \rightarrow_{\mathcal{R}} \mathsf{s}(\,\mathsf{s}(\,0 + \mathsf{s}(\mathsf{s}(\mathsf{s}(0)))\,)\,) & C = \mathsf{s}[\Box] \end{array}$$

Definition (Rewriting)

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

$$s \rightarrow_{\mathcal{R}} t \quad\iff\quad \begin{array}{l} \exists \text{ context } C \\ \exists\, \ell \to r \in \mathcal{R} \\ \exists \text{ substitution } \sigma \end{array} \quad \text{with} \quad \begin{array}{rcll} s & = & C[\ell\sigma] & \text{redex} \\ t & = & C[r\sigma] & \end{array}$$

Example

- TRS $\mathcal{R}$ for $\mathcal{F} = \{0, \mathsf{s}, +\}$:

  $$\boxed{0 + x} \to x \qquad\qquad \mathsf{s}(x) + y \to \mathsf{s}(x + y)$$

- rewrite sequence

$$\begin{array}{rll} \mathsf{s}(\mathsf{s}(0)) + \mathsf{s}(\mathsf{s}(\mathsf{s}(0))) & \rightarrow_{\mathcal{R}} \mathsf{s}(\,\mathsf{s}(0) + \mathsf{s}(\mathsf{s}(\mathsf{s}(0)))\,) & C = \square \\ & \rightarrow_{\mathcal{R}} \mathsf{s}(\,\mathsf{s}(\,\boxed{0 + \mathsf{s}(\mathsf{s}(\mathsf{s}(0)))}\,)\,) & C = \mathsf{s}[\square] \end{array}$$

Definition (Rewriting)

binary relation $\to_{\mathcal{R}}$ on $\mathcal{T}(\mathcal{F},\mathcal{V})$ for every TRS $\mathcal{R}$:

$$s \to_{\mathcal{R}} t \quad\Longleftrightarrow\quad \begin{array}{l} \exists \text{ context } C \\ \exists\ \ell \to r \in \mathcal{R} \\ \exists \text{ substitution } \sigma \end{array} \quad \text{with} \quad \begin{array}{rcll} s & = & C[\ell\sigma] & \text{redex} \\ t & = & C[r\sigma] & \end{array}$$

Example

- TRS $\mathcal{R}$ for $\mathcal{F} = \{0, \mathsf{s}, +\}$:
  $$0 + x \to x \qquad\qquad \mathsf{s}(x) + y \to \mathsf{s}(x+y)$$
- rewrite sequence

$$\begin{array}{rll} \mathsf{s}(\mathsf{s}(0)) + \mathsf{s}(\mathsf{s}(\mathsf{s}(0))) & \to_{\mathcal{R}} \mathsf{s}(\,\mathsf{s}(0) + \mathsf{s}(\mathsf{s}(\mathsf{s}(0)))\,) & C = \Box \\ & \to_{\mathcal{R}} \mathsf{s}(\,\mathsf{s}(\,0 + \mathsf{s}(\mathsf{s}(\mathsf{s}(0)))\,)\,) & C = \mathsf{s}[\Box] \\ & \to_{\mathcal{R}} \mathsf{s}(\mathsf{s}(\,\mathsf{s}(\mathsf{s}(\mathsf{s}(0)))\,)) & C = \mathsf{s}[\mathsf{s}[\Box]] \end{array}$$

Definition (Rewriting)

binary relation $\to_{\mathcal{R}}$ on $\mathcal{T}(\mathcal{F},\mathcal{V})$ for every TRS $\mathcal{R}$:

$$s \to_{\mathcal{R}} t \quad \Longleftrightarrow \quad \begin{array}{l} \exists \text{ context } C \\ \exists\, \ell \to r \in \mathcal{R} \\ \exists \text{ substitution } \sigma \end{array} \quad \text{with} \quad \begin{array}{rcll} s & = & C[\ell\sigma] & \text{redex} \\ t & = & C[r\sigma] & \end{array}$$

Example

- TRS $\mathcal{R}$ for $\mathcal{F} = \{0, \mathsf{s}, +\}$:

$$0 + x \to x \qquad\qquad \mathsf{s}(x) + y \to \mathsf{s}(x + y)$$

- rewrite sequence

$$\begin{array}{rll} \mathsf{s}(\mathsf{s}(0)) + \mathsf{s}(\mathsf{s}(\mathsf{s}(0))) & \to_{\mathcal{R}} \mathsf{s}(\,\mathsf{s}(0) + \mathsf{s}(\mathsf{s}(\mathsf{s}(0)))\,) & C = \Box \\ & \to_{\mathcal{R}} \mathsf{s}(\,\mathsf{s}(\,0 + \mathsf{s}(\mathsf{s}(\mathsf{s}(0)))\,)\,) & C = \mathsf{s}[\Box] \\ & \to_{\mathcal{R}} \mathsf{s}(\mathsf{s}(\,\mathsf{s}(\mathsf{s}(\mathsf{s}(0)))\,)) & C = \mathsf{s}[\mathsf{s}[\Box]] \end{array}$$

normal form

Theorem

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

Theorem

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

Undecidable Problems

instance: TRS $\mathcal{R}$	instance: TRS $\mathcal{R}$
question: is $\mathcal{R}$ terminating ?	question: is $\mathcal{R}$ confluent ?

Theorem

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

Undecidable Problems

instance:	TRS $\mathcal{R}$	instance:	TRS $\mathcal{R}$
question:	is $\mathcal{R}$ terminating ?	question:	is $\mathcal{R}$ confluent ?

Theorem

- *confluence is decidable for terminating TRSs*

Theorem

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

Undecidable Problems

instance:	TRS $\mathcal{R}$	instance:	TRS $\mathcal{R}$
question:	is $\mathcal{R}$ terminating ?	question:	is $\mathcal{R}$ confluent ?

Theorem

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

Exercises

1 Complete the following table:

	a	b	c	d	e	f	i	j
SN	✗							
WCR								
CR							✓	

2 Show that Newman's Lemma does not hold element-wise: find an ARS $(A, \to)$ such that $a \in A$, $SN(a)$ and $WCR(a)$ hold, but $\neg CR(a)$.

3 Which of the following matching problems have a solution?
$x + \mathsf{s}(y) \mapsto \mathsf{s}(x) + \mathsf{s}(\mathsf{s}(x))$ $\qquad$ $\mathsf{f}(x, \mathsf{g}(\mathsf{x}, \mathsf{y})) \mapsto \mathsf{f}(\mathsf{f}(z, z), \mathsf{g}(\mathsf{a}, z))$

4 Rewrite the term $\mathsf{s}(\mathsf{s}(0)) \times \mathsf{s}(\mathsf{s}(0))$ to normal form wrt the TRS
$0 + y \to y$ $\quad$ $\mathsf{s}(x) + y \to \mathsf{s}(x + y)$ $\quad$ $0 \times y \to 0$ $\quad$ $\mathsf{s}(x) \times y \to (x \times y) + y$

Exercises

1 Complete the following table:

	a	b	c	d	e	f	i	j
SN	✗							
WCR								
CR							✓	

2 Show that Newman's Lemma does not hold element-wise: find an ARS $(A, \rightarrow)$ such that $a \in A$, $SN(a)$ and $WCR(a)$ hold, but $\neg CR(a)$.

3 Which of the following matching problems have a solution?
$x + \mathsf{s}(y) \mapsto \mathsf{s}(x) + \mathsf{s}(\mathsf{s}(x))$ $\qquad$ $\mathsf{f}(x, \mathsf{g}(\mathsf{x}, \mathsf{y})) \mapsto \mathsf{f}(\mathsf{f}(z, z), \mathsf{g}(\mathsf{a}, z))$

4 Rewrite the term $\mathsf{s}(\mathsf{s}(0)) \times \mathsf{s}(\mathsf{s}(0))$ to normal form wrt the TRS
$0 + y \rightarrow y \quad \mathsf{s}(x) + y \rightarrow \mathsf{s}(x + y) \quad 0 \times y \rightarrow 0 \quad \mathsf{s}(x) \times y \rightarrow (x \times y) + y$

Exercises

1

$$\begin{tikzcd}
 & & & j \arrow[loop above] & & & \\
 & i & & & & k & \\
b \arrow[ur] \arrow[rr] & & d \arrow[ul] \arrow[uur] \arrow[rr] \arrow[d] & & f \arrow[uul] \arrow[ur] \arrow[rr] & & h \arrow[ul] \\
a \arrow[u] & & c \arrow[ll] \arrow[rr] & & e \arrow[u] \arrow[rr] & & g \arrow[u]
\end{tikzcd}$$

Complete the following table:

	a	b	c	d	e	f	i	j
SN	✗							
WCR								
CR							✓	

2 Show that Newman's Lemma does not hold element-wise: find an ARS $(A, \to)$ such that $a \in A$, $SN(a)$ and $WCR(a)$ hold, but $\neg CR(a)$.

3 Which of the following matching problems have a solution?

$x + \mathsf{s}(y) \mapsto \mathsf{s}(x) + \mathsf{s}(\mathsf{s}(x))$ $\qquad$ $\mathsf{f}(x, \mathsf{g}(\mathsf{x}, \mathsf{y})) \mapsto \mathsf{f}(\mathsf{f}(z, z), \mathsf{g}(\mathsf{a}, z))$

4 Rewrite the term $\mathsf{s}(\mathsf{s}(0)) \times \mathsf{s}(\mathsf{s}(0))$ to normal form wrt the TRS

$0 + y \to y \quad \mathsf{s}(x) + y \to \mathsf{s}(x + y) \quad 0 \times y \to 0 \quad \mathsf{s}(x) \times y \to (x \times y) + y$