$$\begin{array}{rcl} \mathtt{f}(y, \mathtt{h}(F)) & \to & F \cdot \mathtt{g}(y) \\ \mathtt{g}(x) & \to & x \end{array}$$

$$\begin{array}{rcl} \mathtt{f} & :: & \mathsf{c} \Rightarrow \mathsf{a} \\ \mathtt{g} & :: & \mathsf{a} \Rightarrow \mathsf{c} \\ \mathtt{h} & :: & (\mathsf{a} \Rightarrow \mathsf{b}) \Rightarrow \mathsf{c} \\ \mathtt{k} & :: & \mathsf{a} \Rightarrow \mathsf{c} \Rightarrow \mathsf{b} \end{array}$$

$$\mathtt{k}(\mathtt{f}(\mathtt{h}(F)), \mathtt{g}(y)) \to F \cdot y$$

11

Bonus exercises

Construct a (general) self-loop for the following HTRSs:

$$\begin{array}{rcl} \mathtt{f} & :: & \mathsf{a} \Rightarrow (\mathsf{a} \Rightarrow \mathsf{a}) \\ \mathtt{g} & :: & (\mathsf{a} \Rightarrow \mathsf{a}) \Rightarrow \mathsf{a} \end{array}$$

$$\mathtt{f}(\mathtt{g}(x)) \to x$$

$$\begin{array}{rcl} \mathtt{f} & :: & (\mathsf{b} \Rightarrow \mathsf{a} \Rightarrow \mathsf{b} \Rightarrow \mathsf{a}) \Rightarrow \mathsf{c} \\ \mathtt{g} & :: & \mathsf{b} \Rightarrow \mathsf{c} \\ \mathtt{h} & :: & \mathsf{c} \Rightarrow \mathsf{b} \\ \mathtt{k} & :: & \mathsf{c} \Rightarrow \mathsf{b} \Rightarrow \mathsf{b} \Rightarrow \mathsf{a} \Rightarrow \mathsf{a} \end{array}$$

$$\mathtt{k}(\mathtt{g}(x), y, \mathtt{h}(\mathtt{f}(F)), z) \to F \cdot \mathtt{h}(\mathtt{g}(y)) \cdot z \cdot x$$

12

Nasty example

$$\begin{array}{rcl} \mathtt{map}(F, []) & \to & [] \\ \mathtt{map}(F, \mathtt{cons}(x, y)) & \to & \mathtt{cons}(F \cdot x, \mathtt{map}(F, y)) \end{array}$$

Not terminating if:

$$\begin{array}{rcl} [] & :: & \mathsf{o} \\ \mathtt{cons} & :: & (\mathsf{o} \Rightarrow \mathsf{o}) \Rightarrow \mathsf{o} \Rightarrow \mathsf{o} \\ \mathtt{map} & :: & ((\mathsf{o} \Rightarrow \mathsf{o}) \Rightarrow \mathsf{o} \Rightarrow \mathsf{o}) \Rightarrow \mathsf{o} \Rightarrow \mathsf{o} \end{array}$$

Proof: choose $\omega := \mathtt{cons}(\lambda x_{\mathsf{o}}.\mathtt{map}(\lambda y_{\mathsf{o} \Rightarrow \mathsf{o}}.\lambda z_{\mathsf{o}}.y_{\mathsf{o} \Rightarrow \mathsf{o}} \cdot z_{\mathsf{o}}, []))$. Then:

$$\begin{array}{ll} & \mathtt{map}(\lambda y_{\mathsf{o} \Rightarrow \mathsf{o}}.\lambda z_{\mathsf{o}}.y_{\mathsf{o} \Rightarrow \mathsf{o}} \cdot \omega, \omega) \\ \to & \mathtt{cons}(\ (\lambda y.\lambda z.y \cdot \omega)\ \langle \lambda x.\mathtt{map}(\lambda y.\lambda z.y \cdot x, x) \rangle\ ,\ \mathtt{map}(\dots)) \\ = & \mathtt{cons}(\ \lambda z.\langle \lambda x.\mathtt{map}(\lambda y.\lambda z'.y \cdot x, x) \rangle \cdot \omega, \mathtt{map}(\dots)) \\ \to_\beta & \mathtt{cons}(\ \lambda z.\underline{\mathtt{map}(\lambda y.\lambda z'.y \cdot \omega, \omega)}\ , \mathtt{map}(\dots)) \end{array}$$

(But *is* terminating if $\mathtt{cons} :: (\mathsf{a} \Rightarrow \mathsf{a}) \Rightarrow \mathsf{o} \Rightarrow \mathsf{o}$.)

The actual names of the base types matter! And there are indeed termination methods that exploit this.

1

Handout for part 2:

Termination: non-termination

2

Termination

Definition: there is no infinite reduction sequence $s_1 \to_{\mathcal{R}} s_2 \to_{\mathcal{R}} s_3 \to_{\mathcal{R}} \dots$

Put differently:

- a **term** s is terminating if every reduction sequence starting in s is finite; i.e., there is no infinite reduction sequence $s \to_{\mathcal{R}} t_1 \to_{\mathcal{R}} t_2 \to_{\mathcal{R}} \dots$
- a **HTRS** is terminating if all its terms are

Definition: a HTRS is **non-terminating** if it has a non-terminating term.

Example:

$$\begin{array}{rcl} \mathtt{a} & \to & \mathtt{a} \\ \mathtt{a} & \to & \mathtt{b} \end{array}$$

This system is clearly non-terminating, as there is an infinite reduction sequence $a \to_{\mathcal{R}} a \to_{\mathcal{R}} a \to_{\mathcal{R}} \dots$.

It is also **weakly normalising**; that is, for every term there *exists* a reduction that ends in a normal form. This property is also sometimes studied, but is not the question we consider here.

3

Proving non-termination

Some ways to prove non-termination:

- Obvious self-loop: $s \to_{\mathcal{R}}^* s$

  $$\mathtt{f}(x, F) \to \mathtt{f}(F \cdot \mathtt{0}, \lambda y.x)$$

  In this system, we have $\mathtt{f}(x, \lambda y.x) \to_{\mathcal{R}} \mathtt{f}(\ (\lambda y.x) \cdot \mathtt{0}\ ,\ \lambda y.x) \to_\beta \mathtt{f}(x, \lambda y.x)$.

- Instantiated self-loop: $s \to_{\mathcal{R}}^* s\gamma$

  $$\begin{array}{rcl} \mathtt{f}(x, y) & \to & \mathtt{g}(y, \mathtt{s}(x)) \\ \mathtt{g}(\mathtt{s}(x), y) & \to & \mathtt{f}(x, \mathtt{s}(y)) \end{array}$$

  In this system, we have $\mathtt{f}(x, \mathtt{s}(y)) \to_{\mathcal{R}} \mathtt{g}(\mathtt{s}(y), \mathtt{s}(x)) \to_{\mathcal{R}} \mathtt{f}(y, \mathtt{s}(\mathtt{s}(x))) = \mathtt{f}(x, \mathtt{s}(y))[x := y, y := \mathtt{s}(x)]$.

- General self-loop: $s \to_{\mathcal{R}}^* C[s\gamma]$

  $$\mathtt{f}(x, F) \to \mathtt{s}(\mathtt{f}(\mathtt{s}(x), \lambda y.\mathtt{g}(F, y, x))$$

  In this system, we have $\mathtt{f}(x, F) \to_{\mathcal{R}} C[\mathtt{f}(x, F)\gamma]$ where $C[] = \mathtt{s}(\ \ )$ and $\gamma = [x := \mathtt{s}(x), F := \lambda y.\mathtt{g}(F, y, x)]$.

- Specialised methods: note the shape of an infinite reduction

  $$\begin{array}{rcl} \mathtt{f}(\mathtt{s}(\mathtt{0}), F) & \to & \mathtt{f}(\mathtt{0}, \lambda y.\mathtt{s}(F \cdot y)) \\ \mathtt{f}(\mathtt{0}, F) & \to & \mathtt{f}(F \cdot \mathtt{s}(\mathtt{0}), F) \end{array}$$

  In this system, we have $\mathtt{f}(\mathtt{0}, \lambda x.\mathtt{s}^n(x)) \to_{\mathcal{R}}^* \mathtt{f}(\mathtt{s}^{n+1}(\mathtt{0}), \lambda x.\mathtt{s}^n(x)) \to_{\mathcal{R}}^* \mathtt{f}(\mathtt{0}, \lambda x.\mathtt{s}^{2n+1}(x))$

4

Finding self-loops

How would you **automatically** detect that the following rule admits a self-loop?

$$\mathtt{f}(x, F) \quad \to \quad \mathtt{s}(\mathtt{f}(\mathtt{s}(x), \lambda y.\mathtt{g}(F, y, x)))$$

Idea: for a rule $\ell \to C[r]$ show that $\ell\gamma\delta = r\gamma$

If this is the case, then $\ell\gamma\delta \to_{\mathcal{R}} C\gamma\delta[r\gamma\delta] = D[(\ell\gamma\delta)\delta]$

Note: this is a first-order idea!

The primary higher-order difficulty is extending semi-unification techniques.

But instead of extending first-order non-termination techniques, let us focus on particularly higher-order approach. Recall the first lecture. Without types, we often run into nasty counterexamples for termination. But even with types, we can often reproduce such examples!

5

Non-termination of the untyped λ-calculus

Recall:

$$(\lambda x.s) \cdot t \to_\beta s[x := t]$$

Self-loop:

- Let $\omega := \lambda x.x \cdot x$.
- Then: $\omega \cdot \omega \to_\beta \omega \cdot \omega$!

As a (simply-typed) HTRS:

$$\begin{array}{lcl} \Lambda & : & [\mathsf{term} \Rightarrow \mathsf{term}] \Rightarrow \mathsf{term} \\ @ & : & [\mathsf{term} \times \mathsf{term}] \Rightarrow \mathsf{term} \end{array}$$

$$@(\Lambda(F), x) \quad \to \quad F \cdot x$$

Self-loop: Let $\omega := \Lambda(\lambda x.@(x, x))$.

$$@(\omega, \omega) \to_{\mathcal{R}} (\lambda x.@(x, x)) \cdot \omega \to_\beta @(\omega, \omega)$$

6

The $\omega\omega$ self-loop

$$@(\underbrace{\Lambda(\underbrace{F}_{\mathsf{term} \Rightarrow \mathsf{term}})}_{\mathsf{term}}, \underbrace{x}_{\mathsf{term}}) \to F \cdot x$$

The key danger is that a term of higher type, $F :: \mathsf{term} \Rightarrow \mathsf{term}$, is hidden inside a strictly smaller type, $\mathtt{Lambda}(\dots) :: \mathsf{term}$. The rule takes the function out of the constructor, and then applies it.

7

Finding $\omega\omega$ elsewhere

A different example:

$$\begin{array}{lcl} \mathtt{f} & :: & (\mathsf{A} \Rightarrow \mathsf{B} \Rightarrow \mathsf{C}) \Rightarrow \mathsf{A} \\ \mathtt{g} & :: & \mathsf{A} \Rightarrow \mathsf{B} \Rightarrow \mathsf{A} \Rightarrow \mathsf{C} \\ \mathtt{h} & :: & \mathsf{C} \Rightarrow \mathsf{C} \end{array}$$

$$\mathtt{g}(\underbrace{\mathtt{f}(\underbrace{F}_{\mathsf{A} \Rightarrow ? \Rightarrow \mathsf{C}})}_{\mathsf{A}}, y, \underbrace{z}_{\mathsf{A}}) \to \mathtt{h}(F \cdot z \cdot y)$$

$$\omega = \mathtt{f}(\lambda xy.\mathtt{g}(x, z, x))$$

$$\mathtt{g}(\omega, z, \omega) \to_{\mathcal{R}}^{*} \mathtt{h}(\mathtt{g}(\omega, z, \omega))$$

8

Not examples

$$\begin{array}{lcl} \Lambda & :: & (\mathtt{term} \Rightarrow \mathtt{term}) \Rightarrow \mathtt{term} \\ @ & :: & \mathtt{term} \Rightarrow \mathtt{term} \Rightarrow \mathtt{term} \\ \mathtt{c} & :: & \mathtt{term} \Rightarrow \mathtt{term} \end{array}$$

$$@(\Lambda(F), x) \to F \cdot \mathtt{c}(x)$$

$$\begin{array}{lcl} \Lambda & :: & (\mathsf{a} \Rightarrow \mathsf{b}) \Rightarrow \mathsf{b} \\ @ & :: & \mathsf{b} \Rightarrow \mathsf{a} \Rightarrow \mathsf{b} \end{array}$$

$$@(\Lambda(F), x) \to F \cdot x$$

9

The general shape of $\omega\omega$ occurrences

- Reduction: $C[D[F], x] \to^* E[F \cdot s_1 \cdots x \cdots s_k]$
- Variables: $F : \sigma_1 \Rightarrow \dots \Rightarrow \sigma_i \Rightarrow \dots \Rightarrow \sigma_k \Rightarrow \tau$ and $x : \sigma_i$
- $C[D[F], x] : \tau$ and $D[F] : \sigma_i$
- F and x do not appear at other positions in C or D
- Then let $\omega := D[\lambda x_1 \dots x_k.C[x_i, x_i]]$
- We have: $C[\omega, \omega] \to^* E[\ (\lambda x_1 \dots x_k.C[x_i, x_i]) \cdot \omega\ ] \to_\beta^* E[\ C[\omega, \omega]\ ]$

10

Exercises

Construct a (general) self-loop for the follow HTRSs:

$$\begin{array}{lcl} \mathtt{f} & :: & \mathsf{o} \Rightarrow \mathsf{o} \Rightarrow \mathsf{o} \\ \mathtt{g} & :: & \mathsf{o} \Rightarrow \mathsf{o} \\ \mathtt{h} & :: & (\mathsf{o} \Rightarrow \mathsf{o}) \Rightarrow \mathsf{o} \end{array}$$