I217E: Functional Programming

Nao Hirokawa

JAIST

Term 2-1, 2022

`http://www.jaist.ac.jp/~hirokawa/lectures/fp/`

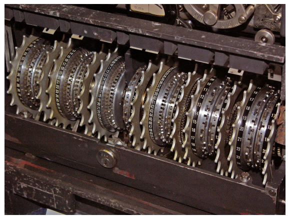
Schedule

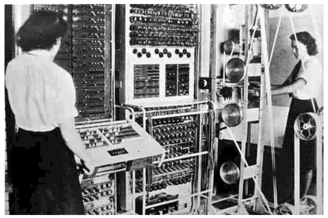
10/12	introduction	11/9	interpreters
10/14	algebraic data types I	11/11	compilers
10/19	algebraic data types II	11/16	termination
10/21	applications	11/18	confluence
10/26	program reasoning	11/25	verification
10/28	data structures I	11/30	review
11/2	data structures II	12/5	exam
11/4	computational models		

Evaluation

exam (60) + reports (40)

Lorenz and Colossus





Colossus (UK, 1943)

Mission

codebreak enemies' communications

encrypted text → MACHINE → decrypted text

Scientific Solution

- linguistic methods reduce task to combinatorial problems
- developed super-efficient device for Boolean operations
- Boolean operations are enough for computation

Computational Models

how to formalize execution of program?

- **Turing machine**
- **register machine**
- **μ-recursive functions**
- **lambda calculus**
- **combinatory logic**
- **term rewriting**

Regarding Computation as Rewriting

program in Haskell

```
data Nat = Z | S Nat

add Z y     = y
add (S x) y = S (add x y)
```

(applicative) term rewrite system

$$\mathcal{R} = \left\{ \begin{array}{ll} \mathtt{add}\ \mathsf{Z}\ y & \to y \\ \mathtt{add}\ (\mathsf{S}\ x)\ y & \to \mathsf{S}\ (\mathtt{add}\ x\ y) \end{array} \right\}$$

computation

$$\begin{aligned} & \underline{\mathtt{add}\ (\mathsf{S}\ (\mathsf{S}\ \mathsf{Z}))\ (\mathsf{S}\ \mathsf{Z})} \\ = {} & \mathsf{S}\ (\underline{\mathtt{add}\ (\mathsf{S}\ \mathsf{Z})\ (\mathsf{S}\ \mathsf{Z})}) \\ = {} & \mathsf{S}\ (\mathsf{S}\ (\underline{\mathtt{add}\ \mathsf{Z}\ (\mathsf{S}\ \mathsf{Z})})) \\ = {} & \mathsf{S}\ (\mathsf{S}\ (\mathsf{S}\ \mathsf{Z})) \end{aligned}$$

rewriting

$$\begin{aligned} & \underline{\mathtt{add}\ (\mathsf{S}\ (\mathsf{S}\ \mathsf{Z}))\ (\mathsf{S}\ \mathsf{Z})} \\ \to_{\mathcal{R}} {} & \mathsf{S}\ (\underline{\mathtt{add}\ (\mathsf{S}\ \mathsf{Z})\ (\mathsf{S}\ \mathsf{Z})}) \\ \to_{\mathcal{R}} {} & \mathsf{S}\ (\mathsf{S}\ (\underline{\mathtt{add}\ \mathsf{Z}\ (\mathsf{S}\ \mathsf{Z})})) \\ \to_{\mathcal{R}} {} & \mathsf{S}\ (\mathsf{S}\ (\mathsf{S}\ \mathsf{Z})) \end{aligned}$$

Terms

Definitions

- signature $\mathcal{F}$ — function symbols with arities
- variables $\mathcal{V}$ — $\mathcal{F} \cap \mathcal{V} = \varnothing$ infinitely many
- terms $\mathcal{T}(\mathcal{F},\mathcal{V})$ — smallest set such that

$$\frac{x \in \mathcal{V}}{x \in \mathcal{T}(\mathcal{F},\mathcal{V})} \qquad \frac{f^{(n)} \in \mathcal{F} \quad t_1,\ldots,t_n \in \mathcal{T}(\mathcal{F},\mathcal{V})}{f(t_1,\ldots,t_n) \in \mathcal{T}(\mathcal{F},\mathcal{V})}$$

- ground terms $\mathcal{T}(\mathcal{F}) = \mathcal{T}(\mathcal{F},\varnothing)$

Example

for $\mathcal{F} = \{0^{(0)}, \mathsf{s}^{(1)}, \mathsf{add}^{(2)}\}$ and $\mathcal{V} = \{x, y, \ldots\}$

$\mathsf{add}(\mathsf{s}(x), y) \in \mathcal{T}(\mathcal{F},\mathcal{V})$ and $\mathsf{add}(\mathsf{s}(0), 0) \in \mathcal{T}(\mathcal{F})$

Variables in Terms

Definition

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

Example

$$\begin{aligned} \mathcal{V}\mathsf{ar}(\mathsf{add}(\mathsf{s}(x), y)) &= \mathcal{V}\mathsf{ar}(\mathsf{s}(x)) \cup \mathcal{V}\mathsf{ar}(y) \\ &= \mathcal{V}\mathsf{ar}(x) \cup \mathcal{V}\mathsf{ar}(y) \\ &= \{x\} \cup \{y\} \\ &= \{x, y\} \end{aligned}$$

Substitutions

Definition

- $\sigma \colon \mathcal{V} \to \mathcal{T}(\mathcal{F},\mathcal{V})$ is substitution if $\mathcal{D}\mathrm{om}(\sigma) = \{x \in \mathcal{V} \mid \sigma(x) \neq x\}$ is finite
- $t\sigma = \begin{cases} \sigma(t) & \text{if } t \in \mathcal{V} \\ f(t_1\sigma,\ldots,t_n\sigma) & \text{if } t = f(t_1,\ldots,t_n) \end{cases}$
- $\{x_1 \mapsto t_1,\ldots,x_n \mapsto t_n\}$ maps variable x_i to t_i and other variable y to y

Example

let $\sigma = \{x \mapsto \mathsf{s}(y),\ y \mapsto \mathsf{s}(0)\}$

$$\begin{aligned} \mathsf{add}(\mathsf{s}(x),\mathsf{add}(y,z))\sigma &= \mathsf{add}(\mathsf{s}(\sigma(x)),\mathsf{add}(\sigma(y),\sigma(z))) \\ &= \mathsf{add}(\mathsf{s}(\mathsf{s}(y)),\mathsf{add}(\mathsf{s}(0),z)) \end{aligned}$$

Contexts

Definition

- context is term in $\mathcal{T}(\mathcal{F} \cup \{\square\},\mathcal{V})$ with one hole $\square$
- $C[t]$ denotes result of replacing $\square$ in C by t

Example

let $C = \mathsf{s}(\mathsf{add}(\square,y))$

$$C[\mathsf{s}(x)] = \mathsf{s}(\mathsf{add}(\mathsf{s}(x),y))$$

Term Rewrite Systems

Definition

- rewrite rule $\ell \to r$ is pair of terms with

  $\ell \notin \mathcal{V}$ and $\mathcal{V}\mathrm{ar}(r) \subseteq \mathcal{V}\mathrm{ar}(\ell)$
- term rewrite system (TRS) $\mathcal{R}$ is set of rewrite rules
- $s \to_{\mathcal{R}} t$ if $\begin{cases} s = C[\ell\sigma] \\ t = C[r\sigma] \end{cases}$ for some $\ell \to r \in \mathcal{R}$, C, and σ

Example

for TRS $\mathcal{R} = \left\{ \begin{array}{r@{\ }l} \mathsf{add}(0,y) &\to y \\ \mathsf{add}(\mathsf{s}(x),y) &\to \mathsf{s}(\mathsf{add}(x,y)) \end{array} \right\}$

$$\mathsf{add}(\mathsf{s}(0),\mathsf{s}(0)) \to_{\mathcal{R}} \mathsf{s}(\mathsf{add}(0,\mathsf{s}(0))) \to_{\mathcal{R}} \mathsf{s}(\mathsf{s}(0))$$

More Notations

Definition

- $s \to_{\mathcal{R}}^{*} t$ if $s \to_{\mathcal{R}}^{n} t$ for some $n \geqslant 0$
- $\mathsf{NF}(\mathcal{R}) = \{t \in \mathcal{T}(\mathcal{F},\mathcal{V}) \mid t \to_{\mathcal{R}} u \text{ for no } u \in \mathcal{T}(\mathcal{F},\mathcal{V})\}$
- u is **normal form** of t if $t \to_{\mathcal{R}}^{*} u \in \mathsf{NF}(\mathcal{R})$

Example

for TRS $\mathcal{R} = \left\{ \begin{array}{r@{\ }l} \mathsf{add}(0,y) &\to y \\ \mathsf{add}(\mathsf{s}(x),y) &\to \mathsf{s}(\mathsf{add}(x,y)) \end{array} \right\}$

$$\mathsf{add}(\mathsf{s}(0),\mathsf{s}(0)) \to_{\mathcal{R}} \mathsf{s}(\mathsf{add}(0,\mathsf{s}(0))) \to_{\mathcal{R}} \mathsf{s}(\mathsf{s}(0)) \in \mathsf{NF}(\mathcal{R})$$

Pattern Matching Algorithm

Definition (Matching Problem)

instance: terms s, t

question: $s\sigma = t$ for some σ?

Matching Algorithm

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

$$
\begin{aligned}
\{f(s_1,\ldots,s_n) \mapsto f(t_1,\ldots,t_n)\} \cup S &\implies \{s_1 \mapsto t_1,\ldots,s_n \mapsto t_n\} \cup S \\
\{f(s_1,\ldots,s_n) \mapsto g(t_1,\ldots,t_n)\} \cup S &\implies \bot \text{ if } f \neq g \\
\{f(s_1,\ldots,s_n) \mapsto x\} \cup S &\implies \bot \\
\{x \mapsto t\} \cup S &\implies \bot \text{ if } x \mapsto t' \in S \text{ with } t \neq t'
\end{aligned}
$$

Example for Pattern Matching I

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

$$\implies \left\{ \begin{array}{c} x \mapsto \mathsf{s}(y) \\ \mathsf{s}(\mathsf{add}(y,z)) \mapsto \mathsf{s}(\mathsf{add}(\mathsf{add}(x,0),z)) \end{array} \right\}$$

$$\implies \left\{ \begin{array}{c} x \mapsto \mathsf{s}(y) \\ \mathsf{add}(y,z) \mapsto \mathsf{add}(\mathsf{add}(x,0),z) \end{array} \right\}$$

$$\implies \left\{ \begin{array}{l} x \mapsto \mathsf{s}(y) \\ y \mapsto \mathsf{add}(x,0) \\ z \mapsto z \end{array} \right\} \quad \cdots \text{ matching substitution}$$

Example for Pattern Matching II

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

$$\implies \left\{ \begin{array}{c} \mathsf{s}(x) \mapsto \mathsf{s}(0+x) \\ x+y \mapsto (0+0)+x \end{array} \right\}$$

$$\implies \left\{ \begin{array}{c} x \mapsto 0+x \\ x+y \mapsto (0+0)+x \end{array} \right\}$$

$$\implies \left\{ \begin{array}{l} x \mapsto 0+x \\ x \mapsto 0+0 \\ y \mapsto x \end{array} \right\}$$

$$\implies \bot \quad \cdots \text{ no matching substitution}$$

To Handle Higher-Order Functions...

Program in Haskell

$$
\begin{aligned}
\mathtt{map}\ f\ [\,] &= [\,] \\
\mathtt{map}\ f\ (x : xs) &= f\ x : \mathtt{map}\ f\ xs
\end{aligned}
$$

Term Rewrite System (really?)

for $\mathcal{F} = \{\mathtt{map}^{(2)}, [\,]^{(0)}, :^{(2)}\}$ and $\mathcal{V} = \{x, xs, f, \ldots\}$

$$
\begin{aligned}
\mathtt{map}(f, [\,]) &\to [\,] \\
\mathtt{map}(f, x : xs) &\to f(x) : \mathtt{map}(f, xs)
\end{aligned}
$$

$f(x)$ is not valid term

Applicative Terms

Definition

1. **applicative signature** consists of
   constant symbols and one binary application symbol $\circ$
2. **applicative terms** are terms over applicative signature

Juxtaposition Notation

$t_1\ t_2\ t_3\ \cdots\ t_n$ denotes $((t_1 \circ t_2) \circ t_3) \circ \cdots \circ t_n$

Example (Applicative Term)

let $\mathcal{F} = \{\circ^{(2)}, \mathsf{map}^{(0)}, [\,]^{(0)}, :^{(0)}\}$ and $\mathcal{V} = \{f, x, xs, \ldots\}$

$$(f\ x) : \mathsf{map}\ f\ xs = ((:) \circ (f \circ x)) \circ ((\mathsf{map} \circ f) \circ xs)$$

Applicative Term Rewrite Systems

Definition

applicative TRSs are TRSs over applicative signatures

Example

consider applicative TRS over $\{\circ^{(2)}, \mathsf{map}^{(0)}, [\,]^{(0)}, :^{(0)}\}$

$$\mathcal{R} = \left\{ \begin{array}{c} \mathsf{id}\ x \to x \\ \mathsf{map}\ f\ [\,] \to [\,] \\ \mathsf{map}\ f\ (x : xs) \to f\ x : \mathsf{map}\ f\ xs \end{array} \right\}$$

where $s : t$ stands for $(:)\ s\ t$

$$\underline{\mathsf{map}\ \mathsf{id}\ (0 : [\,])} \to_{\mathcal{R}} \underline{\mathsf{id}\ 0} : \mathsf{map}\ \mathsf{id}\ [\,] \to_{\mathcal{R}} 0 : \underline{\mathsf{map}\ \mathsf{id}\ [\,]} \to_{\mathcal{R}} 0 : [\,]$$

Homework 1/3

1. Compute $C[t]$ or $t\sigma$.
   1. $C = \mathsf{s}(\Box) + x$ and $t = x + y$.
   2. $C = \Box$ and $t = \mathsf{s}(x)$.
   3. $t = \mathsf{f}(x, x)$ and $\sigma = \{x \mapsto \mathsf{s}(x),\ y \mapsto 0\}$
   4. $t = \mathsf{g}(x, y, z)$ and $\sigma = \{x \mapsto y,\ y \mapsto x\}$
2. Is there a substitution σ such that $t\sigma = u$?
   1. $t = \mathsf{f}(x, \mathsf{s}(y))$ and $u = \mathsf{f}(\mathsf{s}(x), \mathsf{s}(\mathsf{s}(0)))$
   2. $t = \mathsf{f}(x, \mathsf{s}(y))$ and $u = \mathsf{f}(\mathsf{s}(x), \mathsf{g}(y))$
   3. $t = \mathsf{f}(x, \mathsf{s}(x))$ and $u = \mathsf{f}(\mathsf{s}(x), \mathsf{s}(\mathsf{s}(0)))$

Homework 2/3

3. Consider the TRS $\mathcal{R}$ over the signature $\{\texttt{++}^{(2)}, [\,]^{(0)}, :^{(2)}\}$:

$$3\underset{1}{\texttt{++}}:\quad [\,] \texttt{++}\, ys \to ys \qquad \underset{2}{\texttt{++}}:\quad (x : xs) \texttt{++}\, ys \to x : (xs \texttt{++}\, ys)$$

Indicate the rewrite rule $\ell \to r$, context C, and substitution σ of each rewrite step $C[\ell\sigma] \to_{\mathcal{R}} C[r\sigma]$:

$$(1 : (2 : [\,])) \texttt{++} (3 : [\,])$$

$$\downarrow_{\mathcal{R}} \quad (\ell \to r) = \underset{2}{\texttt{++}},\ C = \Box, \text{ and } \sigma = \left\{ \begin{array}{l} x \mapsto 1 \\ xs \mapsto 2 : [\,] \\ ys \mapsto 3 : [\,] \end{array} \right\}$$

$$1 : ((2 : [\,]) \texttt{++} (3 : [\,]))$$

$$\downarrow_{\mathcal{R}} \quad \boxed{?}$$

$$1 : (2 : ([\,] \texttt{++} (3 : [\,])))$$

$$\downarrow_{\mathcal{R}} \quad \boxed{?}$$

$$1 : (2 : (3 : [\,]))$$

Homework 3/3

4 Let $\mathcal{R} = \{\mathsf{b}(\mathsf{a}(x)) \to \mathsf{a}(\mathsf{b}(x))\}$ and $s = \mathsf{b}(\mathsf{a}(\mathsf{b}(\mathsf{a}(x))))$. Compute the following sets.

1. $\{t \mid s \to_{\mathcal{R}} t\}$
2. $\{t \mid s \to_{\mathcal{R}}^{*} t\}$
3. $\{t \mid s \to_{\mathcal{R}}^{*} t \in \mathsf{NF}(\mathcal{R})\}$

5 Without using juxtaposition notation, represent the following applicative TRS:

$$\begin{aligned} \mathsf{S}\ x\ y\ z &\to x\ z\ (y\ z) \\ \mathsf{K}\ x\ y &\to x \\ \mathsf{I}\ x &\to x \end{aligned}$$

For instance, the third rule is represented by $\mathsf{I} \circ x \to x$. Note that this system is known as combinatory logic.