Computability Theory IV

Recursive Function

Guoqiang Li

Shanghai Jiao Tong University

Oct. 17, 2014

Review Tips

Initial Function

1. The zero function
   - $\mathbf{0}$
   - $\mathbf{0}(\widetilde{x}) = 0$
2. The successor function
   - $s(x) = x + 1$
3. The projection function
   - $U_i^n(x_1, \ldots, x_n) = x_i$

Composition

Suppose $f(y_1, \ldots, y_k)$ is a k-ary function and $g_1(\widetilde{x}), \ldots, g_k(\widetilde{x})$ are n-ary functions, where $\widetilde{x}$ abbreviates $x_1, \ldots, x_n$.

The composition function $h(\widetilde{x})$ is defined by

$$h(\widetilde{x}) \quad = \quad f(g_1(\widetilde{x}), \ldots, g_k(\widetilde{x})),$$

Recursion

Suppose that $f(\tilde{x})$ is an n-ary function and $g(\tilde{x}, y, z)$ is an $(n+2)$-ary function.

The recursion function $h(\tilde{x}, y)$ is defined by

$$h(\tilde{x}, 0) = f(\tilde{x}), \tag{1}$$

$$h(\tilde{x}, y+1) = g(\tilde{x}, y, h(\tilde{x}, y)). \tag{2}$$

Clearly there is a unique function that satisfies (1) and (2).

Quiz

$$LCM(x, y)$$

Quiz

$$LCM(x, y)$$

Sol. $LCM(x, y) = \mu z < xy + 1(div(x, z)div(y, z) = 1)$.

Quiz

$$LCM(x, y)$$

Sol. $LCM(x, y) = \mu z < xy + 1(div(x, z)div(y, z) = 1)$.

$$HCF(x, y))$$

Quiz

$$LCM(x, y)$$

Sol. $LCM(x, y) = \mu z < xy + 1(div(x, z)div(y, z) = 1)$.

$$HCF(x, y))$$

Sol. $HCF(x, y) = \frac{xy}{LCM(x,y)}$.

Synopsis

1. Recursive Function
2. Ackermann Function
3. Definability in URM

Recursive Function

An Example

$$g(x) = \begin{cases} \sqrt{x} & \text{if } x \text{ is a pefect square.} \\ \text{undefined} & \text{otherwise.} \end{cases}$$

Minimization Operator, or Search Operator

Minimization function, or μ-function, or search function:

$$\mu y(f(\widetilde{x},y)=0) \quad \simeq \quad \begin{cases} \text{the least } y \text{ such that} \\ \quad f(\widetilde{x},z) \text{ is defined for all } z \leq y, \text{ and} \\ \quad f(\widetilde{x},y)=0, \\ \text{undefined if otherwise.} \end{cases}$$

Here $\simeq$ is the computational equality.

Minimization Operator, or Search Operator

Minimization function, or μ-function, or search function:

$$\mu y(f(\widetilde{x}, y) = 0) \simeq \begin{cases} \text{the least } y \text{ such that} \\ \quad f(\widetilde{x}, z) \text{ is defined for all } z \leq y, \text{ and} \\ \quad f(\widetilde{x}, y) = 0, \\ \text{undefined if otherwise.} \end{cases}$$

Here $\simeq$ is the computational equality.

- The recursion operation is a well-founded going-down procedure.

Minimization Operator, or Search Operator

Minimization function, or μ-function, or search function:

$$\mu y(f(\widetilde{x}, y) = 0) \quad \simeq \quad \begin{cases} \text{the least } y \text{ such that} \\ \quad f(\widetilde{x}, z) \text{ is defined for all } z \leq y, \text{ and} \\ \quad f(\widetilde{x}, y) = 0, \\ \text{undefined if otherwise.} \end{cases}$$

Here $\simeq$ is the computational equality.

- The recursion operation is a well-founded going-down procedure.
- The search operation is a possibly divergent going-up procedure.

An Example

$$g(x) = \begin{cases} \sqrt{x} & \text{if } x \text{ is a pefect square.} \\ \text{undefined} & \text{otherwise.} \end{cases}$$

An Example

$$g(x) = \begin{cases} \sqrt{x} & \text{if } x \text{ is a pefect square.} \\ \text{undefined} & \text{otherwise.} \end{cases}$$

$$f(x,y) = |x - y^2|$$

Recursive Function

The set of recursive functions is the least set generated from the initial functions, composition, recursion and minimization.

Decidable Predicate

A predicate $R(\widetilde{x})$ is decidable if its characteristic function

$$c_R(\widetilde{x}) \stackrel{\text{def}}{=} \begin{cases} 1, & \text{if } R(\widetilde{x}) \text{ is true,} \\ 0, & \text{otherwise.} \end{cases}$$

is a recursive function.

Decidable Predicate

A predicate $R(\widetilde{x})$ is decidable if its characteristic function

$$c_R(\widetilde{x}) \stackrel{\text{def}}{=} \begin{cases} 1, & \text{if } R(\widetilde{x}) \text{ is true,} \\ 0, & \text{otherwise.} \end{cases}$$

is a recursive function. The predicate $R(\widetilde{x})$ is partially decidable if its partial characteristic function

$$\chi_R(\widetilde{x}) \stackrel{\text{def}}{=} \begin{cases} 1, & \text{if } R(\widetilde{x}) \text{ is true,} \\ \uparrow, & \text{otherwise.} \end{cases}$$

is a recursive function.

Closure Property

The following statements are valid:

- If $R(\widetilde{x})$ is decidable, then so is $\neg R(\widetilde{x})$.
- If $R(\widetilde{x}), S(\widetilde{x})$ are (partially) decidable, then the following predicates are (partially) decidable:
  - $R(\widetilde{x}) \wedge S(\widetilde{x})$;
  - $R(\widetilde{x}) \vee S(\widetilde{x})$.
- If $R(\widetilde{x}, y)$ is (partially) decidable, then the following predicates are (partially) decidable:
  - $\forall z < y.R(\widetilde{x}, y)$;
  - $\exists z < y.R(\widetilde{x}, y)$.

Definition by Cases

Suppose $f_1(\widetilde{x}), \ldots, f_k(\widetilde{x})$ are recursive and $M_1(\widetilde{x}), \ldots, M_k(\widetilde{x})$ are partially decidable. For every $\widetilde{x}$ at most one of $M_1(\widetilde{x}), \ldots, M_k(\widetilde{x})$ holds. Then the function $g(\widetilde{x})$ given by

$$g(\widetilde{x}) \simeq \begin{cases} f_1(\widetilde{x}), & \text{if } M_1(\widetilde{x}) \text{ holds,} \\ f_2(\widetilde{x}), & \text{if } M_2(\widetilde{x}) \text{ holds,} \\ \vdots & \\ f_k(\widetilde{x}), & \text{if } M_k(\widetilde{x}) \text{ holds.} \end{cases}$$

is recursive.

Minimization via Decidable Predicate

Suppose $R(x, y)$ is a partially decidable predicate. The function

$$
\begin{aligned}
g(x) &= \mu y R(\tilde{x}, y) \\
&= \begin{cases} \text{the least } y \text{ such that } R(\tilde{x}, y) \text{ holds,} & \text{if there is such a } y \\ \text{undefined,} & \text{otherwise.} \end{cases}
\end{aligned}
$$

is recursive.

Minimization via Decidable Predicate

Suppose $R(x, y)$ is a partially decidable predicate. The function

$$
\begin{aligned}
g(x) &= \mu y R(\widetilde{x}, y) \\
&= \begin{cases} \text{the least } y \text{ such that } R(\widetilde{x}, y) \text{ holds,} & \text{if there is such a } y \\ \text{undefined,} & \text{otherwise.} \end{cases}
\end{aligned}
$$

is recursive.

Proof

$g(\widetilde{x}) = \mu y(\overline{\mathsf{sg}}(\chi_R(\widetilde{x}, y)) = 0)$.

Comment

The μ-operator allows one to define partial functions.

Comment

The μ-operator allows one to define partial functions.

The diagonalisation argument does not apply to the set $\mathfrak{F}$ of recursive functions.

Comment

The μ-operator allows one to define partial functions.

The diagonalisation argument does not apply to the set $\mathfrak{F}$ of recursive functions.

Using the μ-operator, one may define total functions that are not primitive recursive.

Minimization Operator is a Search Operator

It is clear from the above proof why the minimization operator is sometimes called a search operator.

Definable Function

A function is definable if there is a recursive function calculating it.

Ackermann function

Ackermann Function

The Ackermann function [1928] is defined as follows:

$$
\begin{aligned}
\psi(0, y) &\simeq y + 1, \\
\psi(x+1, 0) &\simeq \psi(x, 1), \\
\psi(x+1, y+1) &\simeq \psi(x, \psi(x+1, y)).
\end{aligned}
$$

The equations clearly define a total function.

Ackermann is not Primitive Recursive

Lemma 1.
$\psi(1, m) = m + 2$ and $\psi(2, m) = 2m + 3$

Ackermann is not Primitive Recursive

Lemma 1.
$\psi(1, m) = m + 2$ and $\psi(2, m) = 2m + 3$

Lemma 2.
$\psi(n, m) \geq m + 1$

Ackermann is not Primitive Recursive

Ackermann is not Primitive Recursive

Lemma 3.
The Ackermann function is monotone:

$$
\begin{aligned}
\psi(n, m) &< \psi(n, m+1), \\
\psi(n, m) &< \psi(n+1, m).
\end{aligned}
$$

Ackermann is not Primitive Recursive

Lemma 3.
The Ackermann function is monotone:

$$\begin{aligned}\psi(n,m) &< \psi(n,m+1),\\ \psi(n,m) &< \psi(n+1,m).\end{aligned}$$

Lemma 4.
The Ackermann function grows faster on the first parameter:

$$\psi(n,m+1) \leq \psi(n+1,m)$$

Ackermann is not Primitive Recursive

Lemma 5.
$\psi(n,m) + C$ is dominated by $\psi(J,m)$ for some large enough J:

$$\begin{aligned} \psi(n,m) + \psi(n',m) &< \psi(\max(n,n') + 4, m), \\ \psi(n,m) + m &< \psi(n+4, m). \end{aligned}$$

Ackermann is not Primitive Recursive

Lemma 6.
Let $f(\widetilde{x})$ be a k-ary primitive recursive function. Then there exists some J such that for all $n_1, \ldots, n_k$ we have that

$$f(n_1, \ldots, n_k) < \psi(J, \sum_{i=1}^{k} n_k).$$

Ackermann is not Primitive Recursive

Lemma 6.

Let $f(\widetilde{x})$ be a k-ary primitive recursive function. Then there exists some J such that for all $n_1, \ldots, n_k$ we have that

$$f(n_1, \ldots, n_k) < \psi(J, \sum_{i=1}^{k} n_k).$$

Proof. The proof is by structural induction.

Ackermann is not Primitive Recursive

Lemma 6.
Let $f(\widetilde{x})$ be a k-ary primitive recursive function. Then there exists some J such that for all $n_1, \ldots, n_k$ we have that

$$f(n_1, \ldots, n_k) < \psi(J, \sum_{i=1}^{k} n_k).$$

Proof. The proof is by structural induction.

(i) f is one of the initial functions. In this case take J to be 1.

Ackermann is not Primitive Recursive

(ii) f is the composition function $h(g_1(\tilde{x}), \ldots, g_m(\tilde{x}))$. Then

$$
\begin{aligned}
f(\tilde{n}) &= h(g_1(\tilde{n}), \ldots, g_m(\tilde{n})) \\
&< \psi(J_0, \sum_{i=1}^{m} g_i(\tilde{n})) \; < \; \psi(J_0, \sum_{i=1}^{m} \psi(J_i, \sum_{j=1}^{k} n_j)) \\
&< \psi(J_0, \psi(J^*, \sum_{j=1}^{k} n_j)) \; < \; \psi(J^*, \psi(J^* + 1, \sum_{j=1}^{k} n_j)) \\
&= \psi(J^* + 1, \sum_{j=1}^{k} n_j + 1) \; \leq \; \psi(J^* + 2, \sum_{j=1}^{k} n_j).
\end{aligned}
$$

Now set $J = J^* + 2$.

Ackermann is not Primitive Recursive

(iii) Suppose f is defined by the recursion:

$$\begin{aligned} f(\widetilde{x}, 0) &\simeq h(\widetilde{x}), \\ f(\widetilde{x}, y+1) &\simeq g(\widetilde{x}, y, f(\widetilde{x}, y)). \end{aligned}$$

Ackermann is not Primitive Recursive

(iii) Suppose f is defined by the recursion:

$$\begin{aligned} f(\widetilde{x}, 0) &\simeq h(\widetilde{x}), \\ f(\widetilde{x}, y+1) &\simeq g(\widetilde{x}, y, f(\widetilde{x}, y)). \end{aligned}$$

Then $h(\widetilde{n}) < \psi(J_h, \sum \widetilde{n})$ and $g(\widetilde{n}, m, p) < \psi(J_g, \sum \widetilde{n} + m + p)$.

Ackermann is not Primitive Recursive

(iii) Suppose f is defined by the recursion:

$$\begin{array}{rcl} f(\widetilde{x},0) & \simeq & h(\widetilde{x}), \\ f(\widetilde{x},y+1) & \simeq & g(\widetilde{x},y,f(\widetilde{x},y)). \end{array}$$

Then $h(\widetilde{n}) < \psi(J_h, \sum \widetilde{n})$ and $g(\widetilde{n},m,p) < \psi(J_g, \sum \widetilde{n} + m + p)$.

It is easy to prove

$$f(n_1,\ldots,n_k,m) < \psi(J, \sum_{i=1}^{k} n_k + m)$$

by induction on m.

Ackermann is not Primitive Recursive

Now suppose $\psi(x, y)$ was primitive recursive.
By composition $\psi(x, x)$ would be primitive recursive.

Ackermann is not Primitive Recursive

Now suppose $\psi(x, y)$ was primitive recursive.
By composition $\psi(x, x)$ would be primitive recursive.

According to the Lemma 6

$$\psi(n, n) < \psi(J, n)$$

for some J and all n,

Ackermann is not Primitive Recursive

Now suppose $\psi(x, y)$ was primitive recursive.
By composition $\psi(x, x)$ would be primitive recursive.

According to the Lemma 6

$$\psi(n, n) < \psi(J, n)$$

for some J and all n, which would lead to the contradiction

$$\psi(J, J) < \psi(J, J).$$

Ackermann is not Primitive Recursive

Theorem

The Ackermann function grows faster than every primitive recursive function.

Ackermann Function is Recursive

Theorem

The Ackermann function is recursive.

Ackermann Function is Recursive

A finite set S of triples is said to be suitable if the followings hold:
(i) if $(0, y, z) \in S$ then $z = y + 1$;
(ii) if $(x + 1, 0, z) \in S$ then $(x, 1, z) \in S$;
(iii) if $(x + 1, y + 1, z) \in S$ then $\exists u.((x + 1, y, u) \in S \wedge (x, u, z) \in S)$.

Ackermann Function is Recursive

A finite set S of triples is said to be suitable if the followings hold:
(i) if $(0, y, z) \in S$ then $z = y + 1$;
(ii) if $(x + 1, 0, z) \in S$ then $(x, 1, z) \in S$;
(iii) if $(x + 1, y + 1, z) \in S$ then $\exists u.((x + 1, y, u) \in S \wedge (x, u, z) \in S)$.

A triple (x, y, z) can be coded up by $2^x 3^y 5^z$.
A set $\{u_1, \ldots, u_k\}$ can be coded up by $p_{u_1} \cdots p_{u_k}$.

Ackermann Function is Recursive

A finite set S of triples is said to be suitable if the followings hold:
(i) if $(0, y, z) \in S$ then $z = y + 1$;
(ii) if $(x + 1, 0, z) \in S$ then $(x, 1, z) \in S$;
(iii) if $(x + 1, y + 1, z) \in S$ then $\exists u.((x + 1, y, u)\in S \wedge (x, u, z)\in S)$.

A triple (x, y, z) can be coded up by $2^x 3^y 5^z$.
A set $\{u_1, \ldots, u_k\}$ can be coded up by $p_{u_1} \cdots p_{u_k}$.

Let $R(x, y, v)$ be "v is a legal code and $\exists z < v.(x, y, z)\in S_v$".

Ackermann Function is Recursive

A finite set S of triples is said to be suitable if the followings hold:
(i) if $(0, y, z) \in S$ then $z = y + 1$;
(ii) if $(x + 1, 0, z) \in S$ then $(x, 1, z) \in S$;
(iii) if $(x + 1, y + 1, z) \in S$ then $\exists u.((x + 1, y, u) \in S \wedge (x, u, z) \in S)$.

A triple (x, y, z) can be coded up by $2^x 3^y 5^z$.
A set $\{u_1, \ldots, u_k\}$ can be coded up by $p_{u_1} \cdots p_{u_k}$.

Let $R(x, y, v)$ be "v is a legal code and $\exists z < v.(x, y, z) \in S_v$".

The Ackermann function $\psi(x, y) \simeq \mu z((x, y, z) \in S_{\mu v R(x,y,v)})$.

Definability in URM

Definability of Initial Function

Fact. The initial functions are URM-definable.

Definability of Composition

Fact. If $f(y_1, \ldots, y_k)$ and $g_1(\widetilde{x}), \ldots, g_k(\widetilde{x})$ are URM-definable, then the composition function $h(\widetilde{x})$ given by

$$h(\widetilde{x}) \simeq f(g_1(\widetilde{x}), \ldots, g_k(\widetilde{x}))$$

is URM-definable.

Some Notations

Suppose the program P computes f.

Let $\rho(P)$ be the least number i such that the register R_i is not used by the program P.

Some Notations

The notation $P[l_1, \ldots, l_n \to l]$ stands for the following program

$$
\begin{array}{rcl}
I_1 & : & T(l_1, 1) \\
 & \vdots & \\
I_n & : & T(l_n, n) \\
I_{n+1} & : & Z(n+1) \\
 & \vdots & \\
I_{\rho(P)} & : & Z(\rho(P)) \\
\_ & : & P \\
\_ & : & T(1, l)
\end{array}
$$

Definability of Composition

Let $F, G_1, \ldots, G_k$ be programs that compute $f, g_1, \ldots, g_k$.

Let m be $\max\{n, k, \rho(F), \rho(G_1), \ldots, \rho(G_k)\}$.

Definability of Composition

Let $F, G_1, \ldots, G_k$ be programs that compute $f, g_1, \ldots, g_k$.

Let m be $\max\{n, k, \rho(F), \rho(G_1), \ldots, \rho(G_k)\}$.

Registers:

$$[\ldots]_1^m [\widetilde{x}]_{m+1}^{m+n} [g_1(\widetilde{x})]_{m+n+1}^{m+n+1} \ldots [g_k(\widetilde{x})]_{m+n+k}^{m+n+k}$$

Definability of Composition

The program for h:

$$
\begin{array}{rcl}
I_1 & : & T(1, m+1) \\
 & \vdots & \\
I_n & : & T(n, m+n) \\
I_{n+1} & : & G_1[m+1, m+2, \ldots, m+n \;\to\; m+n+1] \\
 & \vdots & \\
I_{n+k} & : & G_k[m+1, m+2, \ldots, m+n \;\to\; m+n+k] \\
I_{n+k+1} & : & F[m+n+1 \ldots, m+n+k \;\to\; 1]
\end{array}
$$

Definability of Recursion

Fact. Suppose $f(\widetilde{x})$ and $g(\widetilde{x}, y, z)$ are URM-definable.
The recursion function $h(\widetilde{x}, y)$ defined by the following recursion

$$\begin{aligned} h(\widetilde{x}, 0) &\simeq f(\widetilde{x}), \\ h(\widetilde{x}, y+1) &\simeq g(\widetilde{x}, y, h(\widetilde{x}, y)) \end{aligned}$$

is URM-definable.

Definability of Recursion

Let F compute f and G compute g. Let m be $\max\{n, \rho(F), \rho(G)\}$.

Definability of Recursion

Let F compute f and G compute g. Let m be $\max\{n, \rho(F), \rho(G)\}$.

Registers: $[\ldots]_1^m [\widetilde{x}]_{m+1}^{m+n} [y]_{m+n+1}^{m+n+1} [k]_{m+n+2}^{m+n+2} [h(\widetilde{x}, k)]_{m+n+3}^{m+n+3}$.

Definability of Recursion

Let F compute f and G compute g. Let m be $\max\{n, \rho(F), \rho(G)\}$.

Registers: $[\ldots]_1^m[\widetilde{x}]_{m+1}^{m+n}[y]_{m+n+1}^{m+n+1}[k]_{m+n+2}^{m+n+2}[h(\widetilde{x}, k)]_{m+n+3}^{m+n+3}$.

Program:

$$
\begin{aligned}
I_1 &: T(1, m+1) \\
&\vdots \\
I_{n+1} &: T(n+1, m+n+1) \\
I_{n+2} &: F[1, 2, \ldots, n \ \rightarrow \ m+n+3] \\
I_{n+3} &: J(m+n+2, m+n+1, n+7) \\
I_{n+4} &: G[m+1, \ldots, m+n, m+n+2, m+n+3 \ \rightarrow \ m+n+3] \\
I_{n+5} &: S(m+n+2) \\
I_{n+6} &: J(1, 1, n+3) \\
I_{n+7} &: T(m+n+3, 1)
\end{aligned}
$$

Definability of Minimization

Fact. If $f(\widetilde{x}, y)$ is URM-definable, then the minimization function $\mu y(f(\widetilde{x}, y) = 0)$ is URM-definable.

Definability of Minimization

Suppose F computes $f(\widetilde{x}, y)$. Let m be $\max\{n+1, \rho(F)\}$.

Definability of Minimization

Suppose F computes $f(\widetilde{x}, y)$. Let m be $\max\{n+1, \rho(F)\}$.

Registers: $[\ldots]_1^m [\widetilde{x}]_{m+1}^{m+n} [k]_{m+n+1}^{m+n+1} [0]_{m+n+2}^{m+n+2}$.

Definability of Minimization

Suppose F computes $f(\widetilde{x}, y)$. Let m be $\max\{n+1, \rho(F)\}$.

Registers: $[\ldots]_1^m [\widetilde{x}]_{m+1}^{m+n} [k]_{m+n+1}^{m+n+1} [0]_{m+n+2}^{m+n+2}$.

Program:

$$
\begin{aligned}
I_1 &: T(1, m+1) \\
&\vdots \\
I_n &: T(n, m+n) \\
I_{n+1} &: F[m+1, m+2, \ldots, m+n+1 \ \rightarrow\ 1] \\
I_{n+2} &: J(1, m+n+2, n+5) \\
I_{n+3} &: S(m+n+1) \\
I_{n+4} &: J(1, 1, n+1) \\
I_{n+5} &: T(m+n+1, 1)
\end{aligned}
$$

Main Result

Theorem. All recursive functions are URM-definable.

Homework

- Read the proof that Ackermann function is not primitive.
- Try to solve the exercises in Chapter 1 & 2 as many as possible.