Computability Theory **Fall 2020**

Lecture 5: Turing Machines for Functions

Lecturer: Renjie Yang

5.1 Turing Computablity

Definition 5.1 *An* ***expression*** *is a finite sequence (possibly empty) of symbols chosen from the list:* $q_1, q_2, q_3, \ldots \in Q$*;* $S_0, S_1, S_2, S_3, \ldots \in \Gamma$*;* R, L.

Definition 5.2 *A* ***quadruple*** *is an expression having one of the following forms:*

1. $q_i S_j S_k q_l$
2. $q_i S_j R q_l$
3. $q_i S_j L q_l$

Definition 5.3 *A* ***Turing machine*** *is a finite (nonempty) set of quadruples that contains no two quadruples whose first two symbols are the same.*

The q*'s and* S*'s that occur in the quadruples of a Turing machine are called its* ***internal configurations*** *and its* ***alphabet****, respectively.*

Definition 5.4 *An* ***instantaneous description*** *is an expression that contains exactly one* q_i*, neither* R *nor* L*, and is such that* q_i *is not the rightmost symbol.*

If Z *is a Turing machine and* α *is an instantaneous description, then we say that* α *is* ***an instantaneous description of*** Z *if the* q_i *that occurs in* α *is an internal configuration of* Z *and if the* S_i*'s that occur in* α *are part of the alphabet of* Z.

Definition 5.5 *An expression that consists entirely of the letters S; is called a* ***tape expression****.*

Definition 5.6 *Let* Z *be a Turing machine, and let* α *be an instantaneous description of* Z*, where* q_i *is the internal configuration that occurs in* α *and where* S_j *is the symbol immediately to the right of* q_i*. Then we call* q_i *the* ***internal configuration of*** Z ***at*** α*, and we call* S_j ***the symbol scanned by*** Z ***at*** α*. The tape expression obtained on removing* q_i *from* α *is called the* ***expression on the tape of*** Z ***at*** α*.*

Definition 5.7 *Let* Z *be a Turing machine, and let* α, β *be instantaneous descriptions. Then we write* $\alpha \to \beta$ (Z) *to mean that one of the following alternatives holds:*

1. *There exist expressions* P *and* Q *(possibly empty) such that*
$$\alpha \text{ is } Pq_iS_jQ,$$
$$\beta \text{ is } Pq_lS_kQ,$$
where
$$Z \text{ contains } q_iS_jS_kq_l.$$

2. *There exist expressions* P *and* Q *(possibly empty) such that*
$$\alpha \text{ is } Pq_iS_jS_kQ,$$
$$\beta \text{ is } PS_jq_lS_kQ,$$
where
$$Z \text{ contains } q_iS_jRq_l.$$

3. *There exist expressions P and Q (possibly empty) such that*
$$\alpha \text{ is } Pq_iS_j,$$
$$\beta \text{ is } PS_jq_lS_0,$$
where
$$Z \text{ contains } q_iS_jRq_l.$$
4. *There exist expressions P and Q (possibly empty) such that*
$$\alpha \text{ is } PS_kq_iS_jQ,$$
$$\beta \text{ is } Pq_lS_kS_jQ,$$
where
$$Z \text{ contains } q_iS_jLq_l.$$
5. *There exist expressions P and Q (possibly empty) such that*
$$\alpha \text{ is } q_iS_jQ,$$
$$\beta \text{ is } q_lS_0S_jQ,$$
where
$$Z \text{ contains } q_iS_jLq_l.$$

Theorem 5.8 *If $\alpha \to \beta$ (Z) and $\alpha \to \gamma$ (Z), then $\beta = \gamma$.*

Theorem 5.9 *If $\alpha \to \beta$ (Z) and $Z \subset Z'$, then $\alpha \to \beta$ (Z').*

Definition 5.10 *An instantaneous description α is called terminal with respect to Z if for no β do we have $\alpha \to \beta$ (Z).*

Definition 5.11 *By a* ***computation of a Turing machine*** *Z is meant a finite sequence $\alpha_1, \alpha_2, \ldots, \alpha_p$ of instantaneous descriptions such that $\alpha_i \to \alpha_{i+1}$ Z for $1 \leqslant i < p$ and such that α_p is terminal with respect to Z. In such a case, we write $\alpha_p = Res_Z(\alpha_1)$ and we call α_p the* ***resultant*** *of α_1 with respect to Z.*

Definition 5.12 *With each number n we associate the tape expression $\overline{n}$ where $\overline{n} = 1^{n+1}$.*

Definition 5.13 *Withe each k-tuple $(n_1, n_2, \ldots, n_k)$ of integers we associate the tape expression $\overline{(n_1, n_2, \ldots, n_k)}$, where*
$$\overline{(n_1, n_2, \ldots, n_k)} = \overline{n_1}B\overline{n_2}B \cdots B\overline{n_k}$$

Definition 5.14 *Let M be any expression. Then $\langle M \rangle$ is the number of occurrences of 1 in M.*

Definition 5.15 *Let Z be a Turing machine. Then, for each n, we associate with Z an n-ary function*
$$\Psi_Z^{(n)}(x_1, x_2, \cdots, x_n)$$
as follows. For each n-tuple $(m_1, m_2, \ldots, m_n)$, we set $a_1 = q_1\overline{(m_1, m_2, \ldots, m_n)}$ and we distinguish between two cases:

1. *There exists a computation of Z, $a_1, \ldots, a_p$. In this case we set*
$$\Psi_Z^{(n)}(m_1, m_2, \ldots, m_n) = \langle a_p \rangle = \langle Res_Z(\alpha_1) \rangle.$$
2. *There exists no computation $a_1, \ldots, a_p$. In this case we leave $\Psi_Z^{(n)}(x_1, x_2, \cdots, x_n)$ undefined.*

For $\Psi_Z^{(1)}(x)$ we write $\Psi_Z(x)$.

Definition 5.16 *An n-ary function $f(x_1, \ldots, x_n)$ is* ***partially computable*** *if there exists a Turing machine Z such that*

$$f(x_1, \ldots, x_n) = \Psi_Z^{(n)}(x_1, \ldots, x_n).$$

In this case we say that Z computes f. If, in addition, $f(x_1, \ldots, x_n)$ is a total function, then it is called ***computable****.*

5.2 Computable Functions

Examples

- $f(x, y) = x + y$, p12
- $S(x) = x + 1$, p12
- $f(x, y) = x - y$, p12
- $f(x, y) = x \dot{-} y$, p15
- $I(x) = x$, p16
- $U_i(x_1, x_2, \ldots, x_n) = x_i$, $1 \leqslant i \leqslant n$, p16
- $f(x, y) = (x + 1)(y + 1)$, p17

Definition 5.17 *If Z is a Turing machine we let $\theta(Z)$ be the largest number i such that q_i is an internal configuration of Z.*

Definition 5.18 *A Turing machine Z is called* ***n-regular*** *($n > 0$) if*

1. *There is an $s > 0$ such that, whenever $Res_Z[q_1\overline{(m_1, \ldots, m_n)}]$ is defined, it has the form $q_{\theta(Z)}\overline{(r_1, \ldots, r_s)}$ for suitable $r_1, \ldots, r_s$, and*
2. *No quadruple of Z begins with $q_{\theta(Z)}$.*

Definition 5.19 *Let Z be a Turing machine. Then $Z^{(n)}$ is the Turing machine obtained from Z by replacing each internal configuration q_i, at all of its occurrences in quadruples of Z, by q_{n+i}.*

Lemma 5.20 *(LEMMA 1 in the book) For every Turing machine Z, we can find a Turing machine Z' such that, for each n, Z' is n-regular, and*

$$Res_{Z'}[q_1\overline{(m_1, \ldots, m_n)}] = q_{\theta(Z')}\overline{\Psi_Z^{(n)}(m_1, \ldots, m_n)}$$

Proof: Page 26. See the demo in class. ■

Lemma 5.21 *(LEMMA 2) For each n-regular Turing machine Z and each $p > 0$, there is a $(p + n)$-regular Turing machine Z_p such that, whenever*

$$Res_Z[q_1\overline{(m_1, \ldots, m_n)}] = q_{\theta(Z)}\overline{(r_1, \ldots, r_s)},$$

it is also the case that

$$Res_Z[q_1\overline{(k_1, \ldots, k_p, m_1, \ldots, m_n)}] = q_{\theta(Z)}\overline{(k_1, \ldots, k_p, r_1, \ldots, r_s)},$$

whereas, whenever $Res_Z[q_1\overline{(m_1, \ldots, m_n)}]$ is undefined, so is $Res_Z[q_1\overline{(k_1, \ldots, k_p, , m_1, \ldots, m_n)}]$.

Page 29. See the demo in class. Same for the following results.

Example The copying machine C_p
For each $n > 0$ and $p \geqslant 0$, we shall define a $(p+n)$-regular Turing machine such that

$$Res_{C_p}[q_1\overline{(k_1,\dots,k_p,m_1,\dots,m_n)}] = q_{p+16}\overline{(m_1,\dots,m_n,k_1,\dots,k_p,m_1,\dots,m_n)}.$$

Example The transfer machine R_p
For each $n > 0$ and $p \geqslant 0$, we shall define a $(p+n)$-regular Turing machine such that

$$Res_{R_p}[q_1\overline{(k_1,\dots,k_p,m_1,\dots,m_n)}] = q_{p+16}\overline{(m_1,\dots,m_n,k_1,\dots,k_p)}.$$

Lemma 5.22 *(LEMMA 3) For each n-regular Turing machine Z, there is an n-regular Turing machine Z' such that, whenever*

$$Res_Z[q_1\overline{(m_1,\dots,m_n)}] = q_{\theta(Z)}\overline{(r_1,\dots,r_s)},$$

it is also the case that

$$Res_{Z'}[q_1\overline{(m_1,\dots,m_n)}] = q_{\theta(Z')}\overline{(r_1,\dots,r_s,m_1,\dots,m_n)},$$

whereas, whenever $Res_Z[q_1\overline{(m_1,\dots,m_n)}]$ is undefined, so is $Res_{Z'}[q_1\overline{(m_1,\dots,m_n)}]$.

Proof: By Lemma 2, there is a $2n$-regular Turing machine U such that

$$Res_Z[q_1\overline{(m_1,\dots,m_n,m_1,\dots,m_n)}] = q_{\theta(Z)}\overline{(m_1,\dots,m_n,r_1,\dots,r_s)}.$$

Then we can take

$$Z' = C_0 \bigcup U^{(15)} \bigcup R_n^{(14+\theta(U))}.$$

Z' does the following:

$$\begin{aligned} q_1\overline{(m_1,\dots,m_n)} &\to q_{16}\overline{(m_1,\dots,m_n,m_1,\dots,m_n)} \quad \text{(by } C_0\text{)} \\ &\to q_{\theta(U^{(15)})}\overline{(m_1,\dots,m_n,r_1,\dots,r_s)} \quad \text{(by } U^{(15)}\text{)} \\ &\to q_{\theta(Z')}\overline{(m_1,\dots,m_n,r_1,\dots,r_s)} \quad \text{(by } R_n^{(14+\theta(U))}\text{)} \end{aligned}$$

■

Lemma 5.23 *(LEMMA 4) Let $Z_1,\dots,Z_p$ be Turing machines. Let $n > 0$. Then, there exists an n-regular Turing machine Z' such that*

$$Res_{Z'}[q_1\overline{(m_1,\dots,m_n)}] = q_{\theta(Z')}\overline{(\Psi_{Z_1}^{(n)}(m_1,\dots,m_n),\dots,\Psi_{Z_p}^{(n)}(m_1,\dots,m_n))}.$$

Proof: Prove by induction on p. For $p = 1$, the result is Lemma 1. Assume the result holds for $p = k$; show that it also holds for $p = k+1$.
Let $Z_1, Z_2, \dots, Z_{k+1}$ be given Turing machines. Let

$$r_i = \Psi_{Z_i}^{(n)}(m_1,\dots,m_n)$$

for $1 \leqslant i \leqslant k+1$. By the inductive hypothesis, there is an n-regular Turing machine Y_1 such that

$$Res_{Y_1}[q_1\overline{(m_1,\dots,m_n)}] = q_{\theta(Y_1)}\overline{(r_1,\dots,r_k)}.$$

By Lemma 3, there is an n-regular Turing machine Y_2 such that

$$Res_{Y_1}[q_1\overline{(m_1,\ldots,m_n)}] = q_{\theta(Y_2)}\overline{(r_1,\ldots,r_k,m_1,\ldots,m_n)}.$$

By Lemma 1, there is an n-regular Turing machine Y_3 such that

$$Res_{Y_3}[q_1\overline{(m_1,\ldots,m_n)}] = q_{\theta(Y_3)}\overline{r_{k+1}}.$$

By Lemma 2, there is an n-regular Turing machine Y_4 such that

$$Res_{Y_1}[q_1\overline{(r_1,\ldots,r_k,m_1,\ldots,m_n)}] = q_{\theta(Y_4)}\overline{(r_1,\ldots,r_k,r_{k+1})}.$$

We can take $Z' = Y_2 \bigcup Y_4^{(\theta(Y_2)-1)}$ to obtain the result for $p = k+1$. ■

5.3 Composition and Minimization

Definition 5.24 *The operation of composition associates with the functions $f(y^{(m)})$, $g_1(x^{(n)})$, $g_2(x^{(n)})$, …, $g_m(x^{(n)})$, the function*

$$h(x^{(n)}) = f(g_1(x^{(n)}), g_2(x^{(n)}), \ldots, g_m(x^{(n)})).$$

Theorem 5.25 *Let $f(y^{(m)})$, $g_1(x^{(n)})$, $g_2(x^{(n)})$, …, $g_m(x^{(n)})$ be (partially) computable. Let*

$$h(x^{(n)}) = f(g_1(x^{(n)}), g_2(x^{(n)}), \ldots, g_m(x^{(n)})).$$

Then $h(x^{(n)})$ is (partially) computable.

Proof: By Lemma 4, there is an n-regular Turing machine Z such that

$$Res_Z[q_1\overline{(x^{(n)}}] = q_{\theta(Z)}\overline{(g_1(x^{(n)}), g_2(x^{(n)}), \ldots, g_m(x^{(n)}))}.$$

Let Z_1 be chosen so that

$$\Psi_{Z_1}^{(m)}(x^{(m)}) = f(x^{(m)}),$$

let $Z' = Z \bigcup Z_1^{(\theta(Z)-1)}$. Then

$$\Psi_{Z'}^{(n)}(\overline{x^{(n)}}) = f(g_1(x^{(n)}), g_2(x^{(n)}), \ldots, g_m(x^{(n)})) = h(x^{(n)})$$

■

Corollary 5.26 *The class of (partially) computable functions is closed under the operation of composition.*

Corollary 5.27 *xy is computable.*

Proof:

$$xy = (x+1)(y+1) \dot{-} (y+1) \dot{-} x$$

■

Example $x^k = x^{(k-1)}x$
Example $|x-y| = (x \dot{-} y) + (y \dot{-} x)$

Definition 5.28 *The operation of minimization associates with each total function $f(y, x^{(n)})$ the function $h(x^{(n)})$ whose value for given $x^{(n)}$ is the least value of y, if one such exists, for which $f(y, x^{(n)}) = 0$, and which is undefined if no such y exists:*

$$h(x^{(n)}) = min_y[f(y, x^{(n)}) = 0].$$

Definition 5.29 *The total function $f(y, x^{(n)})$ is called regular if*

$$min_y[f(y, x^{(n)}) = 0]$$

is total.

Theorem 5.30 *If $f(y, x^{(n)})$ is computable, then*

$$h(x^{(n)}) = min_y[f(y, x^{(n)}) = 0]$$

is partially computable. Moreover, if $f(y, x^{(n)})$ is regular, then $h(x^{(n)})$ is computable.

Page 38.