Recursive Function Theory

Abhishek Kr Singh

TIFR Mumbai.

21 August 2014
Primitive Recursive Functions

- Initial Functions.
  - \( s(x) = x + 1 \)
  - \( n(x) = 0 \)
  - \( u^n_i (x_1, \ldots, x_n) = x_i \), where \( 1 \leq i \leq n \).

- Composition:
  - Let \( h(x_1, \ldots, x_n) = f(g_1(x_1, \ldots, x_n), \ldots, g_k(x_1, \ldots, x_n)) \).
  - Then \( h \) is said to be obtained from \( f \) and \( g_1, \ldots, g_k \) by composition.

- Primitive Recursion:
  - Let \( h(x_1, \ldots, x_n, 0) = f(x_1, \ldots, x_n) \), and
    \( h(x_1, \ldots, x_n, t + 1) = g(t, h(x_1, \ldots, x_n, t), x_1, \ldots, x_n) \).
  - Then \( h \) is said to be obtained from \( f \) and \( g \) by primitive recursion, or simply recursion.

Definition: A function is called **primitive recursive** if it can be obtained from the initial functions by a finite number of applications of composition and recursion.
Some Primitive Recursive Functions

1. \( x + y \)
2. \( x \cdot y \)
3. \( x! \)
4. \( x^y \)
5. \( p(x) \) the predecessor function
6. \( x \div y \)
7. \( |x - y| \)
8. \( \alpha(x) \) the IsZero predicate
9. \( x = y \)
10. \( x \leq y \)
11. \( x < y \)
12. \( y \mid x \) y divides x
13. \( \text{Prime}(x) \)
14. \( \lfloor x/y \rfloor \)
15. \( R(x, y) \)
16. \( p_n \) the nth prime number
17. \( < x, y > \) the pairing function
18. \( [a_1, ..., a_n] \) the Godel number
19. \( Lt(x) \) where \( x = [a_1, ..., a_n] \)
20. \( ([a_1, ..., a_n])_i \)
Bounded Quantifiers:

- If the predicate $P(t, x_1, \ldots, x_n)$ is primitive recursive then so are the predicates $(\forall t)_{\leq y} P(t, x_1, \ldots, x_n)$ and $(\exists t)_{\leq y} P(t, x_1, \ldots, x_n)$.

Bounded Minimalization:

- If the predicate $P(t, x_1, \ldots, x_n)$ is primitive recursive then so is the predicate $\min_{t \leq y} P(t, x_1, \ldots, x_n)$.

Programs and Computable Functions

- Programming language $S$.

  - Our concept of computable function will be based on programming language $S$ which has following instruction types.

    1. $V \leftarrow V$
    2. $V \leftarrow V + 1$
    3. $V \leftarrow V - 1$
    4. $\text{IF } V \neq 0 \text{ GOTO } L$

- A program in $S$ is a sequence of labeled or unlabeled instructions of above type.
Syntax of the language $S$

- Conventions:
  - Input variables $X_1, X_2, X_3, \ldots$
  - Output variable $Y$ and
  - Local Variables $Z_1, Z_2, Z_3, \ldots$

- State $\sigma$ and snapshot $s = (i, \sigma)$ of program $P$.

- A Computation of a program $P$ is defined to be a sequence $s_1, s_2, s_3, \ldots s_k$ of snapshots of $P$ such that $s_{i+1}$ is the successor of $s_i$ for each $i$ and $s_k$ is the terminal snapshot.

Computable Functions

- For any program $P$ and any positive integer $m$, $\psi_P^m(x_1, \ldots, x_m)$ represents the value of function computed by program $P$ on input $x_1, \ldots, x_m$.

- A given partial function $g$ is said to be partially computable if it is computed by some program.

- A function $g$ is called computable if it is both total and partially computable.
Primitive recursive Vs computable functions.

- Every primitive recursive function is computable.
- Coding program by numbers
  - \( \#(I) = < a, < b, c > \)
  - \( \#(P) = [\#(I_1), \#(I_2), \ldots, \#(I_k)] - 1 \).
- Universality Theorem:
  - Let \( \phi^n(x_1, \ldots, x_n, y) = \psi_P^n(x_1, \ldots, x_n) \), where \( \#(P) = y \).
  - Then for each \( n > 0 \), the function \( \phi^n(x_1, \ldots, x_n, y) \) is partially computable.
- Step-Counter Theorem:
  - Let \( STP^n(x_1, \ldots, x_n, y, t) \leftrightarrow \) Program number \( y \) halts after \( t \) or fewer steps on inputs \( x_1, \ldots, x_n \)
  - Then for each \( n > 0 \), the predicate \( STP^n(x_1, \ldots, x_n, y, t) \) is primitive recursive.
- Normal Form Theorem:
  - Let \( f(x_1, \ldots, x_n) \) be a partially computable function. Then there is a primitive recursive predicate \( R(x_1, \ldots, x_n, y) \) such that \( f(x_1, \ldots, x_n) = l(min_z R(x_1, \ldots, x_n, z)) \).