Some Primitive Recursive Functions

Example 3: $h(x) = x!$
Here are the recursion equations:
\[
\begin{align*}
h(0) &= 1 \\
h(t + 1) &= h(t) \cdot s(t)
\end{align*}
\]
This can be rewritten as:
\[
\begin{align*}
h(0) &= s(n(0)) \\
h(t + 1) &= g(t, h(t))
\end{align*}
\]
where
\[
g(x_1, x_2) = s(x_1) \cdot s(x_2),
\]
which can be written as:
\[
g(x_1, x_2) = s(u_1(x_1, x_2)) \cdot s(u_2(x_1, x_2)).
\]
Multiplication is already known to be primitive recursive. Therefore, $h(x) = x!$ is primitive recursive.

Some Primitive Recursive Functions

Example 4: $x^y$
The recursion equations are:
\[
\begin{align*}
x^0 &= 1 \\
x^{y+1} &= x \cdot x^y
\end{align*}
\]

Some Primitive Recursive Functions

Example 5: The predecessor function $p(x)$
It is defined as follows:
\[
\begin{align*}
p(x) &= x - 1 & \text{if } x \neq 0 \\
      &= 0 & \text{if } x = 0
\end{align*}
\]
The recursion equations are:
\[
\begin{align*}
p(0) &= 0 \\
p(t + 1) &= p(t)
\end{align*}
\]

Some Primitive Recursive Functions

Example 6: $x - y$ (monus)
It is defined as follows:
\[
\begin{align*}
x - y &= x - 0 & \text{if } x \geq y \\
      &= 0 & \text{if } x < y
\end{align*}
\]
The recursion equations are:
\[
\begin{align*}
x^0 &= x \\
x^{t+1} &= p(x^t)
\end{align*}
\]

Some Primitive Recursive Functions

Example 7: $|x - y|$ The function $|x - y|$ is defined as the absolute value of the difference between $x$ and $y$.
It can be written as follows:
\[
|x - y| = (x - y) + (y - x)
\]
Therefore, $|x - y|$ is primitive recursive.

Some Primitive Recursive Functions

Example 8: $\alpha(x)$ ("negation")
The function $\alpha(x)$ is defined as follows:
\[
\begin{align*}
\alpha(x) &= 1 & \text{if } x = 0 \\
        &= 0 & \text{if } x \neq 0
\end{align*}
\]
$\alpha(x)$ is primitive recursive, because:
\[
\alpha(x) = 1 - x
\]
If we prefer, we can just write down the recursion equations:
\[
\begin{align*}
\alpha(0) &= 1 \\
\alpha(t + 1) &= 0
\end{align*}
\]
Primitve Recursive Predicates

**Example 9:** \( x = y \)

We can describe this predicate by the function \( d(x, y) \):

\[
d(x, y) = \begin{cases} 
1 & \text{if } x = y \\
0 & \text{if } x \neq y 
\end{cases}
\]

\( d(x, y) \) is primitive recursive because of the following equation:

\[
d(x, y) = \alpha(x - y)
\]

---

**Primitive Recursive Predicates**

**Theorem 5.1:** Let \( C \) be a PRC class.

If \( P, Q \) are predicates that belong to \( C \), then so are \( \sim P, P \lor Q, \) and \( P \land Q \).

**Proof:**

Since \( \sim P = \alpha(P) \), it follows that \( \sim P \) belongs to \( C \).

Since \( P \land Q = P \lor Q \), it follows that \( P \land Q \) belongs to \( C \).

---

**Example 10:** \( x \leq y \)

Again, we can describe this predicate by the function \( d(x, y) \):

\[
d(x, y) = \begin{cases} 
1 & \text{if } x \leq y \\
0 & \text{if } x > y 
\end{cases}
\]

\( d(x, y) \) is primitive recursive because of the following equation:

\[
d(x, y) = \alpha(x - y)
\]

---

**Primitive Recursive Predicates**

**Theorem 5.2:** According to Corollary 5.2, \( x < y \) is also a primitive recursive predicate.

---

**Corollary 5.3:**

If \( P, Q \) are computable predicates, then so are \( \sim P, P \lor Q, \) and \( P \land Q \).

Correspondingly, we have the following corollaries:

**Corollary 5.2:** If \( P, Q \) are primitive recursive predicates, then so are \( \sim P, P \lor Q, \) and \( P \land Q \).

**Corollary 5.3:** If \( P, Q \) are computable predicates, then so are \( \sim P, P \lor Q, \) and \( P \land Q \).

---

**Example 11:** \( x < y \)

So far, we only know that \( x \leq y \) and \( x = y \) are primitive recursive predicates (Examples 9 and 10).

Since we have the tautology

\[
x < y \iff x \leq y \land \sim (x = y)
\]

or even simpler:

\[
x < y \iff \sim (y \leq x)
\]

according to Corollary 5.2, \( x < y \) is also a primitive recursive predicate.

---

**Theorem 5.4:** Let \( C \) be a PRC class.

Let the functions \( g, h \) and the predicate \( P \) belong to \( C \).

Let

\[
f(x_1, \ldots, x_n) = g(x_1, \ldots, x_n) \text{ if } P(x_1, \ldots, x_n) \\
= h(x_1, \ldots, x_n) \text{ otherwise.}
\]

Then \( f \) belongs to \( C \).

**Proof:** \( f \) obviously belongs to \( C \), because it can be written as:

\[
f(x_1, \ldots, x_n) = g(x_1, \ldots, x_n) \land P(x_1, \ldots, x_n) + \\
h(x_1, \ldots, x_n) \land \sim P(x_1, \ldots, x_n)
\]
Theorem 5.4 provides the case for $m = 1$, so we just have to do the inductive step. Let
\[
 f(x_1, \ldots, x_n) = g_1(x_1, \ldots, x_n) \quad \text{if} \quad P_1(x_1, \ldots, x_n) \\
 = g_2(x_1, \ldots, x_n) \quad \text{if} \quad P_2(x_1, \ldots, x_n) \\
 \quad \vdots \\
 = g_m(x_1, \ldots, x_n) \quad \text{if} \quad P_m(x_1, \ldots, x_n) \\
 = h(x_1, \ldots, x_n) \quad \text{otherwise.}
\]

Then $f$ belongs to $C$.

The previous steps showed that

1. $f$ belongs to $C$ for $m = 1$
2. whenever $f$ belongs to $C$ for a particular $m$, then $f$ also belongs to $C$ for $m + 1$.

Conclusion: $f$ belongs to $C$ for any natural number $m$. 

To prove this theorem, we will show that $g$ and $h$ can be obtained from $f$ by primitive recursion.