Outline

1. Primitive Recursive Predicates
2. Iterated Operations and Bounded Quantifiers
Predicates are total functions that range in the two-element set \( \{0, 1\} \). Therefore, the notion of “primitive recursive” makes sense for predicates.

**Example**

The predicate \( x = y \) corresponds to the function

\[
f(x, y) = \begin{cases} 
1 & \text{if } x = y, \\
0 & \text{otherwise.}
\end{cases}
\]

\( f \) is primitive recursive because

\[
f(x, y) = \alpha(|x - y|).
\]
The predicate $x \leq y$ is primitive recursive because it is just $\alpha(x \div y)$. 
Theorem

Let $C$ be a PRC class. If $P, Q \in C$, then $\neg P$, $P \lor Q$ and $P \land Q$ all belong to $C$.

Proof.

Note that $\neg P = \alpha(P)$, so $\neg P \in C$.

We have $P \land Q = P \cdot Q$, so $P \land Q \in C$.

Finally, $P \lor Q \in C$ because

$$P \lor Q = \sim (\sim P \land \sim Q).$$
A similar result holds for computable predicates:

**Corollary**

*If $P$ and $Q$ are computable predicates, then so are $\neg P$, $P \& Q$, and $P \lor Q$.***
Example

\( x < y \) is primitive recursive because

\[ x < y \iff \neg (y \leq x). \]
Theorem

Definition by Cases:
Let \( C \) be a PRC class. If the functions \( g, h \) and the predicate \( P \) belongs to \( C \), then the function \( f \) defined as

\[
f(x_1, \ldots, x_n) = \begin{cases} 
g(x_1, \ldots, x_n) & \text{if } P(x_1, \ldots, x_n) \\
h(x_1, \ldots, x_n) & \text{otherwise} \end{cases}
\]

belongs to \( C \).

Proof.

The result follows from the equality

\[
f(x_1, \ldots, x_n) = g(x_1, \ldots, x_n) \cdot P(x_1, \ldots, x_n) + h(x_1, \ldots, x_n) \cdot \alpha(P(x_1, \ldots, x_n)).
\]
Corollary

Let $\mathcal{C}$ be a PRC class. If the functions $g_1, \ldots, g_m, h$ and the predicates $P_1, \ldots, P_m$ belong to $\mathcal{C}$ such that

$$P_i(x_1, \ldots, n) \& P_j(x_1, \ldots, n) = 0$$

for all $1 \leq i < j \leq m$ and $x_1, \ldots, x_n$. If $f$ defined as

$$f(x_1, \ldots, x_n) = \begin{cases} 
    g_1(x_1, \ldots, x_n) & \text{if } P_1(x_1, \ldots, x_n) \\
    g_2(x_1, \ldots, x_n) & \text{if } P_2(x_1, \ldots, x_n) \\
    \vdots & \\
    g_m(x_1, \ldots, x_n) & \text{if } P_m(x_1, \ldots, x_n) \\
    h(x_1, \ldots, x_n) & \text{otherwise, }
\end{cases}$$

then $f$ belongs to $\mathcal{C}$. 
Proof.

The proof is by induction on $m$. For $m = 1$ the statement holds by the previous theorem. Suppose that the statement is true and let $h'$ be

$$h'(x_1, \ldots, x_n) = \begin{cases} g_{m+1}(x_1, \ldots, x_n) & \text{if } P_{m+1}(x_1, \ldots, x_n) \\ h(x_1, \ldots, x_n) & \text{otherwise} \end{cases}$$

Since $h' \in \mathcal{C}$ (by the theorem on Slide 8) and

$$f(x_1, \ldots, x_n) = \begin{cases} g_1(x_1, \ldots, x_n) & \text{if } P_1(x_1, \ldots, x_n) \\ g_2(x_1, \ldots, x_n) & \text{if } P_2(x_1, \ldots, x_n) \\ \vdots \\ g_m(x_1, \ldots, x_n) & \text{if } P_m(x_1, \ldots, x_n) \\ h'(x_1, \ldots, x_n) & \text{otherwise}, \end{cases}$$

it follows that $f \in \mathcal{C}$. 

Theorem

Let $\mathcal{C}$ be a PRC class. If $f(t, x_1, \ldots, x_n)$ belongs to $\mathcal{C}$, then so do the functions:

$$g(y, x_1, \ldots, x_n) = \sum_{t=0}^{y} f(t, x_1, \ldots, x_n),$$

and

$$h(y, x_1, \ldots, x_n) = \prod_{t=0}^{y} f(t, x_1, \ldots, x_n).$$
Proof.

The recursion equations can be written as

\[
\begin{align*}
g(0, x_1, \ldots, x_n) &= f(0, x_1, \ldots, x_n), \\
g(t + 1, x_1, \ldots, x_n) &= g(t, x_1, \ldots, x_n) + f(t + 1, x_1, \ldots, x_n).
\end{align*}
\]

Since addition belongs to \( C \), \( g \in C \).

Similarly, since

\[
\begin{align*}
h(0, x_1, \ldots, x_n) &= f(0, x_1, \ldots, x_n), \\
h(t + 1, x_1, \ldots, x_n) &= g(t, x_1, \ldots, x_n) \cdot f(t + 1, x_1, \ldots, x_n),
\end{align*}
\]

it follows that \( h \in C \).
**Question:**

Can we prove by induction on $y$ that $g(y, x_1, \ldots, x_n) \in C$?

**NO!** because such a proof would show only that the functions

$$g(0, x_1, \ldots, x_n), g(1, x_1, \ldots, x_n), \ldots$$

belong to $C$ and not that $g(y, x_1, \ldots, x_n) \in C$!
A variant of Theorem from Slide 11

**Theorem**

Let $C$ be a PRC class. If $f(t, x_1, \ldots, x_n)$ belongs to $C$, then so do the functions:

$$g(y, x_1, \ldots, x_n) = \sum_{t=1}^{y} f(t, x_1, \ldots, x_n),$$

and

$$h(y, x_1, \ldots, x_n) = \prod_{t=1}^{y} f(t, x_1, \ldots, x_n).$$
Proof.

For this variant take the initial recursion equations

\[ g(0, t_1, \ldots, t_n) = 0, \]
\[ h(0, t_1, \ldots, t_n) = 1. \]

with the remaining equations as in the previous proof. This defines a vacuous sum as 0 and a vacuous product to be 1.
Theorem

If the predicate $P(t, x_1, \ldots, x_n)$ belongs to some PRC $C$ then so do the predicates

$$(\forall t)_{\leq y} P(t, x_1, \ldots, x_n) \text{ and } (\exists t)_{\leq y} P(t, x_1, \ldots, x_n).$$

The defined predicates are obtained through bounded quantification.
Proof.

Note that

\[(\forall t \leq y) P(t, x_1, \ldots, x_n) = \left( \prod_{t=0}^{y} P(t, x_1, \ldots, t_n) \right) = 1\]

\[(\exists t \leq y) P(t, x_1, \ldots, x_n) = \left( \sum_{t=0}^{y} P(t, x_1, \ldots, t_n) \right) \neq 0.\]
Alternatively, we could have written

$$(\forall t)_{\leq y} P(t, x_1, \ldots, x_n) = \prod_{t=0}^{y} P(t, x_1, \ldots, t_n).$$
Another mode for using quantifiers is \((\forall t)_{t<y}\) and \((\exists t)_{t<y}\).

The result follows from the recursion equations

\[
(\exists t)_{t<y} P(t, x_1, \ldots, c_n) = (\exists t)_{\leq y} [t \neq y \& P(t, x_1, \ldots, x_n)]
\]

\[
(\forall t)_{t<y} P(t, x_1, \ldots, c_n) = (\forall t)_{\leq y} [t = y \lor P(t, x_1, \ldots, x_n)].
\]
Example

$y \mid x$ (which stands for “$y$ divides $x$”); for example, $3 \mid 12$ is TRUE while $5 \mid 12$ is FALSE.

The predicate is primitive recursive because

$$y \mid x \iff (\exists t)_{\leq x} (y \cdot t = x).$$
Example

Prime(x) which is TRUE when x is a prime number is primitive recursive because

$$\text{Prime}(x) \iff x > 1 \& (\forall t)_{\leq x} \{ t = 1 \lor t = x \lor \sim (t | x) \},$$

which expresses that a number is prime if it is greater than 1 and has no divisors other than 1 and itself.