Mathematics for Informatics

Carlos Areces and Patrick Blackburn

areces@loria.fr blackbur@loria.fr

http://www.loria.fr/~areces http://www.loria.fr/~blacbur

INRIA Lorraine
Nancy, France

2007/2008

Summatory and Productory (from 0)

Theorem

Let $\mathcal{C}$ be a PRC class. If $f : \mathbb{N}^{n+1} \rightarrow \mathbb{N}$ is in $\mathcal{C}$ then the functions

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

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

are also in $\mathcal{C}$.

Summatory and Productory (from 0)

Theorem

Let $\mathcal{C}$ be a PRC class. If $f : \mathbb{N}^{n+1} \to \mathbb{N}$ is in $\mathcal{C}$ then the functions

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

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

are also in $\mathcal{C}$.

Proof.

$$\begin{aligned} g(0, x_1, \ldots, x_n) &= f(0, x_1, \ldots, x_n) \\ g(t+1, x_1, \ldots, x_n) &= g(t, x_1, \ldots, x_n) + f(t+1, x_1, \ldots, x_n) \end{aligned}$$

Idem for h with $\cdot$ instead of $+$. $\square$

Observe that it is not important in which variable we do the recursion (we can define $g'(x, t)$ as in the last class and then $g(t, x) = g'(x, t)$)

Sumatorias y productorias (desde 1)

Theorem

Let $\mathcal{C}$ be a PRC class. If $f : \mathbb{N}^{n+1} \to \mathbb{N}$ is in $\mathcal{C}$ then also the following furncions are in $\mathcal{C}$

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

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

(as usual, empty summatory = 0, empty productory = 1)

Sumatorias y productorias (desde 1)

Theorem

Let $\mathcal{C}$ be a PRC class. If $f : \mathbb{N}^{n+1} \to \mathbb{N}$ is in $\mathcal{C}$ then also the following furncions are in $\mathcal{C}$

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

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

(as usual, empty summatory = 0, empty productory = 1)

Proof.

$$\begin{aligned} g(0, x_1, \ldots, x_n) &= 0 \\ g(t+1, x_1, \ldots, x_n) &= g(t, x_1, \ldots, x_n) + f(t+1, x_1, \ldots, x_n) \end{aligned}$$

Idem for h with $\cdot$ instead of $+$ and 1 instead of 0 in the base case. $\square$

Bounded quantification

Let $p : \mathbb{N}^{n+1} \to \{0, 1\}$ be a predicate

$(\forall t)_{\leq y}\ p(t, x_1, \ldots, x_n)$ is true iff

- $p(0, x_1, \ldots, x_n)$ is true and

  $\vdots$
- $p(y, x_1, \ldots, x_n)$ is true

$(\exists t)_{\leq y}\ p(t, x_1, \ldots, x_n)$ is true iff

- $p(0, x_1, \ldots, x_n)$ is true or

  $\vdots$
- $p(y, x_1, \ldots, x_n)$ is true

We can also define versions with $< y$ instead of $\leq y$.

$$(\exists t)_{<y}\ p(t, x_1, \ldots, x_n) \quad \text{and} \quad (\forall t)_{<y}\ p(t, x_1, \ldots, x_n)$$

Bounded quantification (with $\leq$)

Theorem

Let $p : \mathbb{N}^{n+1} \to \{0,1\}$ be a predicate in a PRC class $\mathcal{C}$. The following predicates are also in $\mathcal{C}$:

$$(\forall t)_{\leq y}\ p(t, x_1, \ldots, x_n)$$

$$(\exists t)_{\leq y}\ p(t, x_1, \ldots, x_n)$$

Bounded quantification (with $\leq$)

Theorem

Let $p : \mathbb{N}^{n+1} \to \{0, 1\}$ *be a predicate in a PRC class* $\mathcal{C}$*. The following predicates are also in* $\mathcal{C}$*:*

$$(\forall t)_{\leq y}\ p(t, x_1, \dots, x_n)$$

$$(\exists t)_{\leq y}\ p(t, x_1, \dots, x_n)$$

Proof.

$(\forall t)_{\leq y}\ p(t, x_1, \dots, x_n)$ iff $\prod_{t=0}^{y}\ p(t, x_1, \dots, x_n) = 1$

Bounded quantification (with $\leq$)

Theorem

Let $p : \mathbb{N}^{n+1} \to \{0, 1\}$ be a predicate in a PRC class $\mathcal{C}$. The following predicates are also in $\mathcal{C}$:

$$(\forall t)_{\leq y}\ p(t, x_1, \ldots, x_n)$$

$$(\exists t)_{\leq y}\ p(t, x_1, \ldots, x_n)$$

Proof.

$(\forall t)_{\leq y}\ p(t, x_1, \ldots, x_n)$ iff $\prod_{t=0}^{y}\ p(t, x_1, \ldots, x_n) = 1$
$(\exists t)_{\leq y}\ p(t, x_1, \ldots, x_n)$ iff $\sum_{t=0}^{y}\ p(t, x_1, \ldots, x_n) \neq 0$

Bounded quantification (with $\leq$)

Theorem

Let $p : \mathbb{N}^{n+1} \to \{0,1\}$ be a predicate in a PRC class $\mathcal{C}$. The following predicates are also in $\mathcal{C}$:

$$(\forall t)_{\leq y}\ p(t, x_1, \ldots, x_n)$$

$$(\exists t)_{\leq y}\ p(t, x_1, \ldots, x_n)$$

Proof.

$(\forall t)_{\leq y}\ p(t, x_1, \ldots, x_n)$ iff $\prod_{t=0}^{y}\ p(t, x_1, \ldots, x_n) = 1$
$(\exists t)_{\leq y}\ p(t, x_1, \ldots, x_n)$ iff $\sum_{t=0}^{y}\ p(t, x_1, \ldots, x_n) \neq 0$

- summatory and productory are in $\mathcal{C}$
- comparison by $=$ is in $\mathcal{C}$

$\square$

Bounded quantification (with $<$)

Theorem

Let $p : \mathbb{N}^{n+1} \to \{0,1\}$ be a predicate in a PRC class $\mathcal{C}$. Then the following predicates are also in $\mathcal{C}$:

$$(\forall t)_{<y}\ p(t, x_1, \ldots, x_n)$$

$$(\exists t)_{<y}\ p(t, x_1, \ldots, x_n)$$

Proof.

$(\forall t)_{<y}\ p(t, x_1, \ldots, x_n)$ iff $(\forall t)_{\leq y}\ (t = y \vee p(t, x_1, \ldots, x_n))$
$(\exists t)_{<y}\ p(t, x_1, \ldots, x_n)$ iff $(\exists t)_{\leq y}\ (t \neq y \wedge p(t, x_1, \ldots, x_n))$ $\square$

More examples of primitive recursive functions

- $y|x$ iff y divides x. It can be defined as

$$(\exists t)_{\leq x}\ y \cdot t = x$$

  Note that with this definition $0|0$.
- $prime(x)$ iff x is prime.

Minimization

Let $p : \mathbb{N}^{n+1} \to \{0,1\}$ be a predicate in a PRC class $\mathcal{C}$.

$$g(y, x_1, \ldots, x_n) = \sum_{u=0}^{y} \prod_{t=0}^{u} \alpha(p(t, x_1, \ldots, x_n))$$

What does g do?

Minimization

Let $p : \mathbb{N}^{n+1} \to \{0, 1\}$ be a predicate in a PRC class $\mathcal{C}$.

$$g(y, x_1, \dots, x_n) = \sum_{u=0}^{y} \prod_{t=0}^{u} \alpha(p(t, x_1, \dots, x_n))$$

What does g do?

- suppose that there is $t \leq y$ suth that $p(t, x_1, \dots, x_n)$ is true

Minimization

Let $p : \mathbb{N}^{n+1} \to \{0, 1\}$ be a predicate in a PRC class $\mathcal{C}$.

$$g(y, x_1, \ldots, x_n) = \sum_{u=0}^{y} \prod_{t=0}^{u} \alpha(p(t, x_1, \ldots, x_n))$$

What does g do?

- suppose that there is $t \leq y$ suth that $p(t, x_1, \ldots, x_n)$ is true
  - let t_0 be the maximal such t

Minimization

Let $p : \mathbb{N}^{n+1} \to \{0, 1\}$ be a predicate in a PRC class $\mathcal{C}$.

$$g(y, x_1, \dots, x_n) = \sum_{u=0}^{y} \prod_{t=0}^{u} \alpha(p(t, x_1, \dots, x_n))$$

What does g do?

- suppose that there is $t \leq y$ suth that $p(t, x_1, \dots, x_n)$ is true
  - let t_0 be the maximal such t
  - $p(t, x_1, \dots, x_n) = 0$ for each $t < t_0$

Minimization

Let $p : \mathbb{N}^{n+1} \to \{0,1\}$ be a predicate in a PRC class $\mathcal{C}$.

$$g(y, x_1, \ldots, x_n) = \sum_{u=0}^{y} \prod_{t=0}^{u} \alpha(p(t, x_1, \ldots, x_n))$$

What does g do?

- suppose that there is $t \leq y$ suth that $p(t, x_1, \ldots, x_n)$ is true
  - let t_0 be the maximal such t
  - $p(t, x_1, \ldots, x_n) = 0$ for each $t < t_0$
  - $p(t_0, x_1, \ldots, x_n) = 1$

Minimization

Let $p : \mathbb{N}^{n+1} \to \{0,1\}$ be a predicate in a PRC class $\mathcal{C}$.

$$g(y, x_1, \ldots, x_n) = \sum_{u=0}^{y} \prod_{t=0}^{u} \alpha(p(t, x_1, \ldots, x_n))$$

What does g do?

- suppose that there is $t \leq y$ suth that $p(t, x_1, \ldots, x_n)$ is true
  - let t_0 be the maximal such t
  - $p(t, x_1, \ldots, x_n) = 0$ for each $t < t_0$
  - $p(t_0, x_1, \ldots, x_n) = 1$
  - $\prod_{t=0}^{u} \alpha(p(t, x_1, \ldots, x_n)) = \begin{cases} 1 & \text{if } u < t_0 \\ 0 & \text{otherwise} \end{cases}$

Minimization

Let $p : \mathbb{N}^{n+1} \to \{0,1\}$ be a predicate in a PRC class $\mathcal{C}$.

$$g(y, x_1, \ldots, x_n) = \sum_{u=0}^{y} \prod_{t=0}^{u} \alpha(p(t, x_1, \ldots, x_n))$$

What does g do?

- suppose that there is $t \leq y$ suth that $p(t, x_1, \ldots, x_n)$ is true
  - let t_0 be the maximal such t
  - $p(t, x_1, \ldots, x_n) = 0$ for each $t < t_0$
  - $p(t_0, x_1, \ldots, x_n) = 1$
  - $\prod_{t=0}^{u} \alpha(p(t, x_1, \ldots, x_n)) = \begin{cases} 1 & \text{if } u < t_0 \\ 0 & \text{otherwise} \end{cases}$
  - $g(y, x_1, \ldots, x_n) = \underbrace{1 + 1 + \cdots + 1}_{t_0 \text{ veces}} = t_0$
  - then $g(x_1, \ldots, x_n)$ is the maximal t such that $p(t, x_1, \ldots, x_n)$ is true

Minimization

Let $p : \mathbb{N}^{n+1} \to \{0, 1\}$ be a predicate in a PRC class $\mathcal{C}$.

$$g(y, x_1, \ldots, x_n) = \sum_{u=0}^{y} \prod_{t=0}^{u} \alpha(p(t, x_1, \ldots, x_n))$$

What does g do?

- suppose that there is $t \leq y$ suth that $p(t, x_1, \ldots, x_n)$ is true
  - let t_0 be the maximal such t
  - $p(t, x_1, \ldots, x_n) = 0$ for each $t < t_0$
  - $p(t_0, x_1, \ldots, x_n) = 1$
  - $\prod_{t=0}^{u} \alpha(p(t, x_1, \ldots, x_n)) = \begin{cases} 1 & \text{if } u < t_0 \\ 0 & \text{otherwise} \end{cases}$
  - $g(y, x_1, \ldots, x_n) = \underbrace{1 + 1 + \cdots + 1}_{t_0 \text{ veces}} = t_0$
  - then $g(x_1, \ldots, x_n)$ is the maximal t such that $p(t, x_1, \ldots, x_n)$ is true
- if there is no such t, then $g(y, x_1, \ldots, x_n) = y + 1$

Minimization

We note

$$\min_{t \le y} p(t, x_1, \ldots, x_n) = \begin{cases} \text{smallest } t \le y \text{ such that} \\ p(t, x_1, \ldots, x_n) \text{ is true} & \text{if thre is such a } t \\ & \\ 0 & \text{otherwise} \end{cases}$$

Theorem

Let $p : \mathbb{N}^{n+1} \to \{0, 1\}$ *be a predicate in a PRC class* $\mathcal{C}$*. Then the function*

$$\min_{t \le y} p(t, x_1, \ldots, x_n)$$

is also in $\mathcal{C}$*.*

More examples of primitive recursive functions

- x div y is the integer division of x by y

$$\min_{t \le x}((t+1) \cdot y > x)$$

Note that with this definition 0 div 0 is false.

More examples of primitive recursive functions

- x div y is the integer division of x by y

$$\min_{t \leq x}((t+1) \cdot y > x)$$

Note that with this definition 0 div 0 is false.

- $x \mod y$ is the reminder of dividing x by y

More examples of primitive recursive functions

- x div y is the integer division of x by y

$$\min_{t \leq x}((t+1) \cdot y > x)$$

  Note that with this definition 0 div 0 is false.

- $x \mod y$ is the reminder of dividing x by y
- p_n is the n-th prime $(n > 0)$. It is defined as $p_0 = 0, p_1 = 2, p_2 = 3, p_3 = 5, \ldots$

More examples of primitive recursive functions

- x div y is the integer division of x by y

$$\min_{t \leq x}((t+1) \cdot y > x)$$

  Note that with this definition 0 div 0 is false.

- $x \mod y$ is the reminder of dividing x by y
- p_n is the n-th prime $(n > 0)$. It is defined as $p_0 = 0, p_1 = 2, p_2 = 3, p_3 = 5, \ldots$

$$\begin{array}{rcl} p_0 & = & 0 \\ p_{n+1} & = & \min_{t}(primo(t) \wedge t > p_n) \end{array}$$

More examples of primitive recursive functions

- x div y is the integer division of x by y

$$\min_{t \leq x}((t+1) \cdot y > x)$$

  Note that with this definition 0 div 0 is false.

- $x \mod y$ is the reminder of dividing x by y
- p_n is the n-th prime $(n > 0)$. It is defined as $p_0 = 0, p_1 = 2, p_2 = 3, p_3 = 5, \ldots$

$$\begin{aligned} p_0 &= 0 \\ p_{n+1} &= \min_{t \leq K(n)} (primo(t) \wedge t > p_n) \end{aligned}$$

  We need a good bound $K(n)$, i.e.
  - sufficiently big and
  - primitive recursive

More examples of primitive recursive functions

- x div y is the integer division of x by y

$$\min_{t \leq x}((t+1) \cdot y > x)$$

  Note that with this definition 0 div 0 is false.

- $x \mod y$ is the reminder of dividing x by y
- p_n is the n-th prime $(n > 0)$. It is defined as $p_0 = 0, p_1 = 2, p_2 = 3, p_3 = 5, \dots$

$$\begin{array}{rcl} p_0 & = & 0 \\ p_{n+1} & = & \min_{t \leq K(n)} (primo(t) \wedge t > p_n) \end{array}$$

  We need a good bound $K(n)$, i.e.

  - sufficiently big and
  - primitive recursive

  $K(n) = p_n! + 1$ is enough (note that $p_{n+1} \leq p_n! + 1$).

Bounded minimization

Recall the definition of bounded minimization:

$$\min_{t \le y} p(t, x_1, \dots, x_n) = \begin{cases} \text{smallest } t \le y \text{ such that } p(t, x_1, \dots, x_n) \text{ is true} & \text{if ther is such a } t \\ 0 & \text{otherwise} \end{cases}$$

Bounded minimization

Recall the definition of bounded minimization:

$$\min_{t \leq y} p(t, x_1, \ldots, x_n) = \begin{cases} \text{smallest } t \leq y \text{ such that } p(t, x_1, \ldots, x_n) \text{ is true} & \text{if ther is such a } t \\ 0 & \text{otherwise} \end{cases}$$

What happens if there is no bound? We define unbounded minimization as

$$\min_{t} p(t, x_1, \ldots, x_n) = \begin{cases} \text{smallest } t \text{ such that } p(t, x_1, \ldots, x_n) \text{ is true} & \text{if there is such a } t \\ \uparrow & \text{otherwise} \end{cases}$$

Unbounded minimisation

Theorem

If $p : \mathbb{N}^{n+1} \longrightarrow \{0, 1\}$ *is a computable predicate then*

$$\min_t p(t, x_1, \ldots, x_n)$$

is a partially computable predicate.

Unbounded minimisation

Theorem

If $p : \mathbb{N}^{n+1} \to \{0,1\}$ is a computable predicate then

$$\min_t p(t, x_1, \ldots, x_n)$$

is a partially computable predicate.

Proof.

The following program computes $\min_t p(t, x_1, \ldots, x_n)$:

$$\begin{array}{ll} [A] & \text{IF } p(X_1, \ldots, X_n, Y) = 1 \text{ GOTO } E \\ & Y \leftarrow Y + 1 \\ & \text{GOTO } A \end{array}$$

□

Bounded and unbounded minimization

What we know:

- a `while` can simulate an unbounded minimization
- a `for` can simulate a bounded minimization

Bounded and unbounded minimization

What we know:

- a `while` can simulate an unbounded minimization
- a `for` can simulate a bounded minimization

Questions:

- Can we write every program of $\mathscr{S}$ as a for-programa?
- can we find, for any unbounded minimization, a proper bound to turn it into a bounded minimization?
- computable $=$ primitive recursive?