About algorithmic completeness of primitive recursive languages

P. Cegielski[1], J. Cervelle[1], S. Grigorieff[2], P. Valarcher[1]

1 University Paris 12, LACL

2 University Paris 7, IRIF

[WORK IN PROGRESS]

We consider programming languages that compute the set of primitive recursive functions.

$0, S, \pi, \circ, P$	0, var, var+1, var−1
	$x :=$ expression | assignement
$f(0,\vec{y}) = g(\vec{y})$	$I_1; \dots ; I_n$ | sequence
$f(S(x),\vec{y}) = h(x,\vec{y},f(x,\vec{y}))$	LOOP var P END LOOP | bounded loop
PRC schema	
PR	**LOOP**

PRV

$$\begin{cases} if_0(0, t, s) = t \\ if_0(S(n), t, s) = s \end{cases}$$

conditional by name

$$\begin{cases} f(0, \vec{y}) = g(\vec{y}) \\ f(S(x), \vec{y}) = h(x, f(x, j(x, \vec{y})), \vec{y}) \end{cases}$$

recursion with variable parameters scheme

$LOOP_{exit}$

if $x = 0$
then P_1
else P_2

conditional

loop x exceptif $y = 0$
P
end loop

breakable bounded loop

PR — LOOP

PRV — $LOOP_{exit}$

All the 4 languages compute the same set of functions but definitely not in the same manner

The time of computation distinguishes the way they compute

We speak about the algorithmic expressiveness of programming language

A language for PR : combinator PRC

- 0 is a PRC of arity 0
- $Succ$ is a PRC of arity 1
- Π_i^n is a PRC of arity n (with $1 \leq i \leq n$)
- $S_m^n(c; \underline{c_1, \dots, c_n})$ is a PRC of arity m with c (PRC, n), c_i (PRC, m)
- $Rec(b, s)$ is PRC of arity $n+1$ with b (PRC, n) s (PRC, $n+2$)

Examples:

- Π_1^1 is the identity
- $S_0^2(0;)$ binary null function
- $\boxed{Rec(\Pi_1^1, S_1^3(Succ; \Pi_2^3))}$

$$\equiv \quad add(0, y) = g(y) \quad [g = \Pi_1^1]$$
$$add(Succ(x), y) = Succ(add(x,y)) \quad [h = Succ \circ \Pi_2^3]$$
$$= h(x, add(x,y), y)$$

Operational semantics : one step $\to$

$Succ^i(0)$ is irreducible [a value]

$0[t_1, \dots, t_n] \longrightarrow 0$

$Succ\,[v] \longrightarrow Succ^{i+1}(0)$ with v the value $Succ^i(0)$

$\Pi_n^i[t_1, \dots t_n] \longrightarrow t_i$

$S_m^n(c; c_1, \dots, c_n)[v_1, \dots, v_m] \longrightarrow$ $s \circ (c_1, \dots, c_n)$

transitive closure

$c\,[w_1, \dots, w_m]$ with $c_i[v_1, \dots, v_n] \to^* w_i$

$Rec(b, s)\,[0, t_2, \dots, t_n] \longrightarrow b[t_2, \dots, t_n]$

$$\text{Rec}(b,s)\,[\text{Succ}(t_1), t_2, \dots, t_n] \to S_n^{n+1}(s;\, \text{Rec}(b,s), \Pi_1, \dots, \Pi_n)[t_1, \dots, t_n]$$

if $t_i \to t_i'$ then

$$t[t_1, \dots, t_{i-1}, t_i, t_{i+1}, \dots, t_n] \to t[t_1, \dots, t_{i-1}, t_i', t_{i+1}, \dots, t_n]$$

$$\text{Rec}(\overbrace{\Pi_1^1}^{b}, \overbrace{S_1^3(\text{Succ}; \Pi_2^3)}^{s})\,[\text{Succ}^2(0), \text{Succ}^3(0)]$$

$$\to S_2^3\left(S_1^3(\text{Succ}; \Pi_2^3);\ \text{Rec}(\Pi_1^1, S_1^3(\text{Succ}; \Pi_2^3)), \Pi_1, \Pi_2\right)[\text{Succ}(0), \text{Succ}^3(0)]$$

$$\to^* S_1^3(\text{Succ}; \Pi_2^3)[w_1, w_2, w_3]$$

with

$$w_1 \leftarrow \Pi_1[\text{Succ}(0), \text{Succ}^3(0)]$$

$$\text{Succ}^4(0) = w_2 \xleftarrow{*} \text{Rec}(\Pi_1^1, S_1^3(\text{Succ}; \Pi_2^3))[\text{Succ}(0), \text{Succ}^3(0)]$$

$$w_3 \leftarrow \Pi_2[\text{Succ}(0), \text{Succ}^3(0)]$$

$$\to \text{Succ}[\Pi_2^3[\text{Succ}(0), \text{Succ}^4(0), \text{Succ}^3(0)]]$$

$$\to \text{Succ}[\text{Succ}^4(0)]$$

$$\to \text{Succ}^5(0)$$

We denote by $\text{cost}(c\,[v_1, \dots, v_n])$ the minimum number of one step reductions that leads to a value

Main theorem: for a n-ary PRC c

- there exist $n+2$ constants $k_c, a, r_1, \dots, r_n$

such that for every $x_1, \dots x_n$ we have

$$x_1 \geq r_1, \dots \; x_n \geq r_n \implies 1 \leq \mathrm{cost}(c[x_1,\dots,x_n]) \leq k_c$$

- or there exist an index $i \leq n$ and $n+1$ const. $a, r_1, \dots, r_n$ such that

$$x_1 \geq r_1, \dots, x_n \geq r_n \implies \mathrm{cost}(c[x_1,\dots,x_n]) \geq x_i - a$$

Applications

[Colson] The function $x, y \longmapsto \min(x,y)$ is PR but there is no PRC to evaluate it with a cost in $\min(x,y)$.

[log] The function $x \longmapsto \lfloor \log(x) \rfloor$ is PR but there is no PRC to evaluate it with a cost in $\log(x)$.

$O(\log(x))$

FFT

	$O(\log(n))$	$O(\inf(n,m))$	$O(n)$	$\dots$	$O(n.\log n)$	$\dots$	$O(n^2)$
PRC+div2	No No	No	Yes				Yes
PRV+div2	No Yes	Yes	Yes		Yes		Yes
LOOP+div2	No No	No	Yes		Yes		Yes
LOOP_{exit}+div2	No Yes	Yes	Yes		Yes		Yes

Question

Let f be PR (complexity function)

Let P a programming language for PR

$\exists ?\, prg \in P$ s.t. $cost(prg) \in O(f)$

Thank You!