Primitive recursive functions

Functions $f : \cancel{\mathbb{N} \mapsto \mathbb{N}}$ $\underline{\underline{\mathbb{N}^k}} \longmapsto \underline{\underline{\mathbb{N}^\ell}}$ $k \geq 0$

Basic primitive recursive functions:

- **Zero function:** $Z() = 0$, Constant function: $C^k_n(x_1, \dots x_k) = n$
- **Successor function:** $Succ(n) = n + 1$
- **Projection function:** $P_i(x_1, \dots, x_n) = x_i$

$h \circ (g_1, g_2 \dots g_m)$

Operations:

$g_1(x_1 \dots x_k)$ $g_2(x_1 \dots x_k)$ $\dots g_m(x_1 \dots x_k)$

- **Composition** $h\,(\,y_1 \quad y_2 \quad y_m\,)$
- **Primitive recursion:** if f and g are p.r. of arity k and $k+2$, there is a p.r. h of arity $k+1$:

$$\begin{aligned} h(0, x_1, \dots, x_k) &= f(x_1, \dots, x_k) \\ h(n+1, x_1, \dots, x_k) &= g(h(n, x_1, \dots, x_k), n, x_1, \dots, x_k) \end{aligned}$$

$h(n, x_1, \dots x_k), n, (x_1, \dots x_k)$

Composition:

p.r.

$$\begin{cases} g_1 : \mathbb{N}^k \longrightarrow \mathbb{N} & g_1(x_1, \dots x_k) \longrightarrow y_1 \\ \vdots & \\ g_m : \mathbb{N}^k \longrightarrow \mathbb{N} & g_m(x_1 \dots x_k) \longrightarrow y_m \\ h : \mathbb{N}^m \longrightarrow \mathbb{N} & \end{cases}$$

$$h \circ (g_1, \dots g_m) [x_1 \dots x_k)] \rightarrow h\left[g_1(x_1 \dots x_k), \; g_2(x_1 \dots x_k) \; \vdots \; g_m(x_1 \dots x_k)\right]$$

$\downarrow$

will be p.r. obtaind by composition.

Addition:

$$\begin{aligned} \overset{h}{Add}(0, y) &= y \qquad f(y) = y \\ h\, Add(n+1, y) &= Succ(Add(n, y)) \end{aligned}$$

$Succ(P_1(Add(n, y), n, y))$

h: $Succ \circ P_1$

Addition:

$$Add(0, y) = y$$
$$Add(n+1, y) = Succ(Add(n, y))$$

$Succ(P_1(Add(n,y), n, y))$

Multiplication:

$$Mult(0, y) = Z()$$
$$Mult(n+1, y) = Add(Mult(n, y), y)$$

$P_1(Mult(n,y), n, y) = Mult(n,y)$

$P_3(Mult(n,y), n, y) = y$

$Add \circ (P_1, P_3) : (Mult(n,y), n, y)$

$= Add(P_1(\;), P_3(\;))$

$= Add(Mult(n,y), y)$

Addition:

$$
\begin{aligned}
Add(0, y) &= y \\
Add(n+1, y) &= Succ(Add(n, y))
\end{aligned}
$$

Multiplication:

$$
\begin{aligned}
Mult(0, y) &= Z() \\
Mult(n+1, y) &= Add(Mult(n, y), y)
\end{aligned}
$$

Exponentiation 2^n:

$$
\begin{aligned}
Exp(0) &= Succ(Z()) \\
Exp(n+1) &= \underline{Mult(Exp}(n), 2)
\end{aligned}
$$

$P_1[Exp(n), n]$

C_2^2

$Mult \circ (P_1, C_2^2)$

Addition:

$$\begin{aligned} Add(0, y) &= y \\ Add(n+1, y) &= Succ(Add(n, y)) \end{aligned}$$

Multiplication:

$$\begin{aligned} Mult(0, y) &= Z() \\ Mult(n+1, y) &= Add(Mult(n, y), y) \end{aligned}$$

Exponentiation 2^n:

$$\begin{aligned} Exp(0) &= Succ(Z()) \\ Exp(n+1) &= Mult(Exp(n), 2) \end{aligned}$$

Hyper-exponentiation (tower of n two-s):

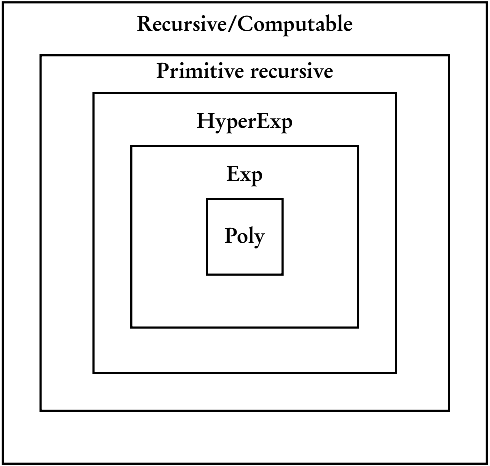
$$\begin{aligned} HyperExp(0) &= Succ(Z()) \\ HyperExp(\underline{n+1}) &= Exp(HyperExp(n)) \end{aligned}$$

$HyperExp(1) = 2$

$HyperExp(2) = 2^2$

$(3) = 2^{2^2}$

$(4) = 2^{2^{2^2}}$



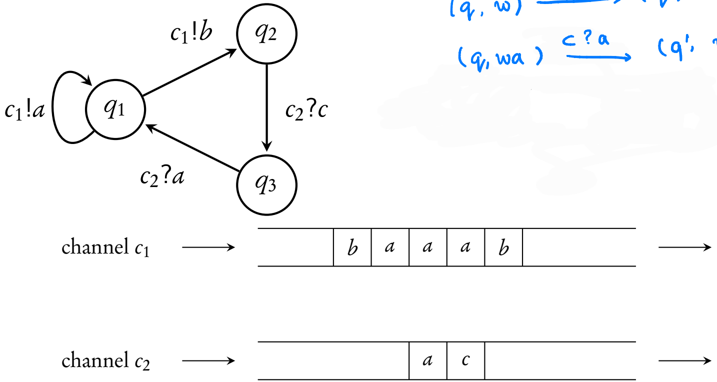
Recursive but not primitive rec.: Ackermann function, Sudan function

Coming next: a problem that has complexity non-primitive recursive

Channel systems

$(q, w) \xrightarrow{c!a} (q', aw)$

$(q, wa) \xrightarrow{c?a} (q', w)$



Finite state description of communication protocols

G. von Bochmann. 1978

On communicating finite-state machines

D. Brand and P. Zafiropulo. 1983

Theorem [BZ'83]

Reachability in channel systems is **undecidable**

Coming next: modifying the model for decidability

Lossy channel systems

Finkel'94, Abdulla and Jonsson'96

Messages stored in channel can be **lost** during transition

$(q, w) \xrightarrow{c!a} (q', w')$ where w' is a subword of aw

$(q, wa) \xrightarrow{c?a} (q', w'')$ where w'' is a subword of w

Lossy channel systems

Finkel'94, Abdulla and Jonsson'96

Messages stored in channel can be **lost** during transition

Theorem [Schnoebelen'2002]

Reachability for **lossy one-channel** systems is **non-primitive recursive**

Reachability problem for **lossy one-channel** systems can be reduced to emptiness problem for **purely universal 1-clock ATA**