

Primitive Recursive Algorithms

Examples of primitive recursion

$a_0(x, y) = x + 1$	$S(x) = x + 1$ so $a_0(x, y) = S(x)$
$a_1(x, y) = add(x, y)$	$\begin{cases} add(0, y) = y \\ add(S(x), y) = S(add(x, y)) \end{cases}$
	Note: $S(add(x, y)) = a_0(a_1(x, y), y)$
$a_2(x, y) = mult(x, y)$	$\begin{cases} mult(0, y) = 0 \\ mult(S(x), y) = add(mult(x, y), y) \end{cases}$ i.e. $(x + 1).y = xy + y$
	Note: $add(mult(x, y), y) = a_1(a_2(x, y), y)$
$a_3(x, y) = exp(x, y)$	$\begin{cases} exp(0, y) = 1 \quad \text{i.e. } y^0 = 1 \\ exp(S(x), y) = mult(exp(x, y), y) \end{cases}$ $y^{x+1} = y^x \cdot y$
	Note: $mult(exp(x, y), y) = a_2(a_3(x, y), y)$
$a_4(x, y) = hypexp(x, y)$	$\begin{cases} hypexp(0, y) = y \\ hypexp(S(x), y) = exp(hypexp(x, y), y) \end{cases}$
	Note: $exp(hypexp(x, y), y) = a_3(a_4(x, y), y)$
$a_{n+1}(x, y)$	$\begin{cases} a_{n+1}(0, y) = y \\ a_{n+1}(S(x), y) = a_n(a_{n+1}(x, y), y) \end{cases}$

We can think of $a_n(x, y)$ as a function of n, x, y . This is one form of *Ackermann's function*. It is not primitive recursive.