KAIST 전산학과 최광무

(정의) 아래의 세 가지 initial function과 세 가지 함수의 결합으로만 이루어진 함수를 **primitive recursive function**이라고 정의한다.

단 k-tuple $\vec{n} = (n_1, \cdots, n_k) \in N^k$이라하자.

1. Initial functions
   (1) **zero**: $\forall k \geq 0 : \zeta : N^k \to \{0\}$
   $\forall \vec{n} \in N^k : \zeta(\vec{n}) \stackrel{\text{def}}{=} 0.$
   (2) **successor**: $\sigma : N \to N$
   $\forall n \in N : \sigma(n) \stackrel{\text{def}}{=} n+1.$
   (3) **projection**: $\forall k \geq 1 : 1 \leq \forall i \leq k : \pi_i^k : N^k \to N,$
   $\forall \vec{n} \in N^k : \pi_i^k(\vec{n}) \stackrel{\text{def}}{=} n_i$ 단 $\vec{n} = (n_1, \cdots, n_i, \cdots, n_k) \in N^k.$
2. 함수의 결합
   (1) **combination** of functions: $f : N^k \to N^n$, $g : N^k \to N^m$일 때,
   $f \times g : N^k \to N^{n+m} : \forall \vec{n} \in N^k : f \times g(\vec{n}) \stackrel{\text{def}}{=} (f(\vec{n}), g(\vec{n})).$
   (2) **composition** of functions: $f : N^k \to N^n$, $g : N^n \to N^m$일 때,
   $f \circ g : N^k \to N^m : \forall \vec{n} \in N^k : f \circ g(\vec{n}) \stackrel{\text{def}}{=} g(f(\vec{n}))$[1].
   (3) **primitive recursion** of functions: $g : N^k \to N^m$, $h : N^{k+m+1} \to N^m$일 때,
   $f : N^{k+1} \to N^m : \forall \vec{n} \in N^k : \forall r \in N :$
   $f(\vec{n}, \zeta(r)) \stackrel{\text{def}}{=}_{\text{B}} g(\vec{n}),$
   $f(\vec{n}, \sigma(r)) \stackrel{\text{def}}{=}_{\text{R}} h(\vec{n}, r, f(\vec{n}, r)).$
   또는
   $f(\vec{n}, 0) \stackrel{\text{def}}{=}_{\text{B}} g(\vec{n}),$
   $f(\vec{n}, r+1) \stackrel{\text{def}}{=}_{\text{R}} h(\vec{n}, r, f(\vec{n}, r)).$

함수 zero, successor, projection, combination, composition, primitive recursion은 모두 TM **computable**이므로 primitive recursive function은 TM **computable**다.

임의의 k-tuple은 자연수와 1:1 대응하므로, $N^k \leftrightarrow^{1:1} N$, 위 정의의 k-tuple, m-tuple, n-tuple 또는 심지어 (k+1)-tuple 까지도 자연수로 바꾸어도 되고 역으로 자연수도 k-tuple로 변환하면 되므로, combination이 필요 없고, **zero**, **successor**, composition과 primitive recursion만 정의하면 된다. 심지어는 zero와 successor를 없애고 **constant**를 정의하여도 되나, zero와 successor는 자연수를 정리한 Peano의 Lemma의 기본이므로 남겨두고 정의하면 아래와 같다.

(정의) k-tuple과 자연수간의 변환 $N^k \leftrightarrow^{1:1} N$을 아래와 같이 정의한다.

$(n_1, n_2, \cdots, n_k) \leftrightarrow (n_1 \# n_2 \# \cdots \# n_k).$

임의의 함수 $f : N^k \to N^m$를 위의 변환을 이용하여 $f' : N \to N$으로 바꿀 수 있다. 따라서 아

1) 고등학교에서는 $f(g(\vec{n}))$으로 정의하였음에 주의하라.

래의 정의와 같이 변환을 이용하여 자연수로 정의역을 표시할 수 있다. 하지만 함수에 정의역에 여러 개의 자연수를 가지는 것에 익숙하므로 그대로 정의역을 N^k라고 보기도 한다.

(정의) 두 가지 initial function **zero**와 **successor**, **primitive recursion**으로만 이루어진 함수를 **primitive recursive function**이라고 정의한다.

1. **zero**: $\forall k \geq 0$: $\zeta : N \to N$, $\forall n \in N$: $\zeta(\vec{n}) \stackrel{\text{def}}{=} 0$.
2. **successor**: $\sigma : N \to N$, $\forall n \in N$: $\sigma(n) \stackrel{\text{def}}{=} n+1$.
3. **primitive recursion** $g : N \to N$, $h : N \to N$일 때,

$f : N \to N$: $\forall n \in N$: $\forall r \in N$:

$$f(n\#\zeta(\mathrm{r})) \stackrel{\text{def}}{=}_{\mathrm{B}} g(n),$$
$$f(n\#\sigma(\mathrm{r})) \stackrel{\text{def}}{=}_{\mathrm{R}} h(n\#\mathrm{r}\#\mathrm{f}(\mathrm{n}\#\mathrm{r})).$$

또는

$$f(n\#0) \stackrel{\text{def}}{=}_{\mathrm{B}} g(n),$$
$$f(n\#(\mathrm{r}+1)) \stackrel{\text{def}}{=}_{\mathrm{R}} h(n\#\mathrm{r}\#\mathrm{f}(\mathrm{n}\#\mathrm{r})).$$

primitive recursive 함수의 예

1. **Constant**: $\forall a \in N$: $K_a : N \to N$, $\forall n \in N$: $K_a(n) \stackrel{\text{def}}{=} \zeta \circ \sigma^a(n) = a$.

Primitive recursive function $K_a(n)$나 $\zeta \circ \sigma^a(n)$ 대신에 상수 a를 그대로 쓴다.

2. Arithmetic function

$plus : N \times N \to N$

$$plus(n,0) \stackrel{\text{def}}{=}_{\mathrm{B}} n,$$
$$plus(n,m+1) \stackrel{\text{def}}{=}_{\mathrm{R}} \sigma(+(n,m)).$$

$+ : N \times N \to N$

$$n+0 \stackrel{\text{def}}{=}_{\mathrm{B}} n,$$
$$n+(m+1) \stackrel{\text{def}}{=}_{\mathrm{R}} \sigma(n+m) = (n+m)+1.$$

오른 쪽 정의와 같이 연산자 +가 infix operator임을 그대로 이용해도 된다.

$\mathrm{mult} : \mathrm{N} \times \mathrm{N} \to \mathrm{N}$

$$\mathrm{mult}(\mathrm{n},\zeta(\mathrm{m})) \stackrel{\text{def}}{=}_{\mathrm{B}} \zeta(n),$$
$$\mathrm{mult}(\mathrm{n},\sigma(\mathrm{m})) \stackrel{\text{def}}{=}_{\mathrm{R}} plus(n,\mathrm{mult}(\mathrm{n},\mathrm{m})).$$

$\cdot : N \times N \to N$

$$n \cdot 0 \stackrel{\text{def}}{=}_{\mathrm{B}} 0,$$
$$n(m+1) \stackrel{\text{def}}{=}_{\mathrm{R}} nm+n.$$

$\mathrm{expo} : N \times N \to N$

$$\mathrm{expo}(n,0) \stackrel{\text{def}}{=}_{\mathrm{B}} 1,$$
$$\mathrm{expo}(n,\sigma(m)) \stackrel{\text{def}}{=}_{\mathrm{R}} \mathrm{mult}(\mathrm{n},\mathrm{expo}(n,m)).$$

$\dot{\cdot} : N \times N \to N$

$$n^0 \stackrel{\text{def}}{=}_{\mathrm{B}} 1,$$
$$n^{m+1} \stackrel{\text{def}}{=}_{\mathrm{R}} n^m \cdot n.$$

Successor 함수(σ)의 역을 생각해 보자.

$\sigma^{-1} : N \to N$

$$\sigma^{-1}(0) \stackrel{\text{def}}{=}_{\mathrm{B}} 0,$$
$$\sigma^{-1}(n+1) \stackrel{\text{def}}{=}_{\mathrm{R}} n.$$

$\dot{-} : N \times N \to N$ $\qquad n \dot{-} m \stackrel{\text{def}}{=}$ **if** $n \geq m \to n - m \mid n \leq m \to 0$ **fi**

$n \dot{-} 0 \stackrel{\text{def}}{=}_{B} n,$

$n \dot{-} (m+1) \stackrel{\text{def}}{=}_{R} \sigma^{-1}(n \dot{-} m).$

3. Boolean functions

$= : N \times N \to \{0,1\}$ $\qquad n = m \stackrel{\text{def}}{=}$ **if** $n = m \to 1 \mid n \neq m \to 0$ **fi**

$n = m \stackrel{\text{def}}{=} 1 \dot{-} ((n \dot{-} m) + (m \dot{-} n))$

$\neq, >, \geq, <, \leq : N \times N \to \{0,1\}$도 정의하여 보아라.

4. Boolean operators

$\vee : \{0,1\} \times \{0,1\} \to \{0,1\}$

$\wedge : \{0,1\} \times \{0,1\} \to \{0,1\}$

$\neg : \{0,1\} \to \{0,1\}$

도 정의하여 보아라.

Ackerman Function(1928)

$A : N^2 \to N$

$A(0,y) = y+1$

$A(x+1,0) = A(x,1)$

$A(x+1,y+1) = A(x, A(x+1,y))$

$$\begin{aligned}
A(2,1) &= A(1, A(2,0)) \\
&= A(1, A(1,1)) \\
&= A(1, A(0, A(0,1))) \\
&= A(1, A(0,2)) \\
&= A(1,3) \\
&= A(0, A(1,2)) \\
&= A(0, A(0, A(1,1))) \\
&= A(0, A(0, A(0, A(1,0)))) \\
&= A(0, A(0, A(0, A(0,1)))) \\
&= A(0, A(0, A(0,2))) \\
&= A(0, A(0,3)) \\
&= A(0,4) \\
&= 5
\end{aligned}$$

$A(3,2)$를 한 번 해 보시오!

Ackerman function은 **computable**이지만, **primitive recursion**은 아니다[2].

2) 증명은 recursion의 큰 수(big number)로 하는데 생략한다.