Simulation of TMs by recursive functions

Church

Simulation of 1-tape Turing machines by recursive functions

Lemma 1. *Let M be a 1-tape TM and $f_M : \mathbb{N}_0^k \to \mathbb{N}_0$ be the function computed by M. Then M is μ-recursive.*

Simulation of 1-tape Turing machines by recursive functions

Lemma 1. *Let M be a 1-tape TM and $f_M : \mathbb{N}_0^k \to \mathbb{N}_0$ be the function computed by M. Then M is μ-recursive.*

1 Coding strings by numbers (in the obvious way)

Let

$$M = (Z, A, \delta, z_0, E)$$

w.l.o.g assume

$$Z \cap A = \emptyset \quad , \quad Z \cup A = \{a_1, \dots, a_p\}$$

For configurations k of M:

$$k \in (A \cup Z)^+$$

1.1 coding $(A \cup Z)^+$ in $\mathbb{N}$:

- single symbols:

$$\psi(a_i) = i$$

- strings $b = b[1 : L]$ interpreted as numbers with base $p+1$

$$\psi(b[1 : L]) = \sum_{i=1}^{L} \psi(b_i)(p+1)^{L-i}$$

injective as $\Psi(a_i) \neq 0$, thus no 'leading zeros'.

1.1 coding $(A \cup Z)^+$ in $\mathbb{N}$:

- single symbols:

$$\psi(a_i) = i$$

- strings $b = b[1:L]$ interpreted as numbers with base $p+1$

$$\psi(b[1:L]) = \sum_{i=1}^{L} \psi(b_i)(p+1)^{L-i}$$

injective as $\Psi(a_i) \neq 0$, thus no 'leading zeros'.

example: p+1 = 10, b[1:6] = 296314, $\Psi(i) = i$

using natural numbers = decimal numbers:

$$x = \Psi(b) = b$$

1.2 operations on coded strings

Let $x = \psi(b)$ with $b = b[1:L]$.

- length of b

$$L(x) = \min\{j \ : \ j \leq x \ , \ (p+1)^j > x\}$$

pr (bounded μ-operator)

$$10^5 = 10000 < 296314 < 10^6$$

1.1 coding $(A \cup Z)^+$ in $\mathbb{N}$:

- single symbols:

$$\psi(a_i) = i$$

- strings $b = b[1:L]$ interpreted as numbers with base $p+1$

$$\psi(b[1:L]) = \sum_{i=1}^{L} \psi(b_i)(p+1)^{L-i}$$

injective as $\Psi(a_i) \neq 0$, thus no 'leading zeros'.

example: p+1 = 10, b[1:6] = 296314, $\Psi(i) = i$

using natural numbers = decimal numbers:

$$x = \Psi(b) = b$$

1.2 operations on coded strings

Let $x = \psi(b)$ with $b = b[1:L]$.

- length of b

$$L(x) = \min\{j \ : \ j \leq x \ , \ (p+1)^j > x\}$$

pr (bounded μ-operator)

$$10^5 = 10000 < 296314 < 10^6$$

- prefix

$$\begin{aligned} prefix(x,i) &= \psi(b[1:i]) \\ &= \lfloor x/(p+1)^{L(x)-i} \rfloor \end{aligned}$$

$$i = 2$$

$$\lfloor 296314/10^{6-2} \rfloor = \lfloor 29{,}6314 \rfloor$$

1.1 coding $(A \cup Z)^+$ in $\mathbb{N}$:

- single symbols:

$$\psi(a_i) = i$$

- strings $b = b[1:L]$ interpreted as numbers with base $p+1$

$$\psi(b[1:L]) = \sum_{i=1}^{L} \psi(b_i)(p+1)^{L-i}$$

injective as $\Psi(a_i) \neq 0$, thus no 'leading zeros'.

example: p+1 = 10, b[1:6] = 296314, $\Psi(i) = i$

using natural numbers = decimal numbers:

$$x = \Psi(b) = b$$

1.2 operations on coded strings

Let $x = \psi(b)$ with $b = b[1:L]$.

- length of b

$$L(x) = \min\{j \ : \ j \leq x \ , \ (p+1)^j > x\}$$

pr (bounded μ-operator)

$$10^5 = 10000 < 296314 < 10^6$$

- prefix

$$\begin{aligned} prefix(x,i) &= \psi(b[1:i]) \\ &= \lfloor x/(p+1)^{L(x)-i} \rfloor \end{aligned}$$

$$i = 2$$

$$\lfloor 296314/10^{6-2} \rfloor = \lfloor 29{,}6314 \rfloor = 29$$

- suffix

$$\begin{aligned} suffix(x,i) &= \psi(b[i:L]) \\ &= rem(x,(p+1)^{L(x)-i+1}) \end{aligned}$$

$$rem(296314{,}10^{6-1}) = rem(296314{,}100000) = 96314$$

1.1 coding $(A \cup Z)^+$ in $\mathbb{N}$:

- single symbols:
$$\psi(a_i) = i$$
- strings $b = b[1:L]$ interpreted as numbers with base $p+1$
$$\psi(b[1:L]) = \sum_{i=1}^{L} \psi(b_i)(p+1)^{L-i}$$
injective as $\Psi(a_i) \neq 0$, thus no 'leading zeros'.

1.2 operations on coded strings

Let $x = \psi(b)$ with $b = b[1:L]$.

- length of b
$$L(x) = \min\{j \ : \ j \leq x \ , \ (p+1)^j > x\}$$
pr (bounded μ-operator)
- prefix
$$\begin{aligned} prefix(x,i) &= \psi(b[1:i]) \\ &= \lfloor x/(p+1)^{L(x)-i} \rfloor \end{aligned}$$
- suffix
$$\begin{aligned} suffix(x,i) &= \psi(b[i:L]) \\ &= rem(x,(p+1)^{L(x)-i+1}) \end{aligned}$$
- head
$$\begin{aligned} hd(x) &= \psi(b_1) \\ &= prefix(x,1) \end{aligned}$$

1.1 coding $(A \cup Z)^+$ in $\mathbb{N}$:

- single symbols:

$$\psi(a_i) = i$$

- strings $b = b[1:L]$ interpreted as numbers with base $p+1$

$$\psi(b[1:L]) = \sum_{i=1}^{L} \psi(b_i)(p+1)^{L-i}$$

  injective as $\Psi(a_i) \neq 0$, thus no 'leading zeros'.

1.2 operations on coded strings

Let $x = \psi(b)$ with $b = b[1:L]$.

- length of b

$$L(x) = \min\{j \;:\; j \leq x \;,\; (p+1)^j > x\}$$

  pr (bounded μ-operator)

- prefix

$$\begin{aligned} prefix(x,i) &= \psi(b[1:i]) \\ &= \lfloor x/(p+1)^{L(x)-i} \rfloor \end{aligned}$$

- suffix

$$\begin{aligned} suffix(x,i) &= \psi(b[i:L]) \\ &= rem(x,(p+1)^{L(x)-i+1}) \end{aligned}$$

- head

$$\begin{aligned} hd(x) &= \psi(b_1) \\ &= prefix(x,1) \end{aligned}$$

- select

$$\begin{aligned} select(x,i) &= x_{(i)} \\ &= \psi(b_i) \\ &= hd(suffix(x,i)) \end{aligned}$$

1.1 coding $(A \cup Z)^+$ in $\mathbb{N}$:

- single symbols:

$$\psi(a_i) = i$$

- strings $b = b[1:L]$ interpreted as numbers with base $p+1$

$$\psi(b[1:L]) = \sum_{i=1}^{L} \psi(b_i)(p+1)^{L-i}$$

  injective as $\Psi(a_i) \neq 0$, thus no 'leading zeros'.

1.2 operations on coded strings

Let $x = \psi(b)$ with $b = b[1:L]$.

- length of b

$$L(x) = \min\{j \ : \ j \leq x \ , \ (p+1)^j > x\}$$

  pr (bounded μ-operator)

- prefix

$$\begin{aligned} prefix(x,i) &= \psi(b[1:i]) \\ &= \lfloor x/(p+1)^{L(x)-i} \rfloor \end{aligned}$$

- suffix

$$\begin{aligned} suffix(x,i) &= \psi(b[i:L]) \\ &= rem(x,(p+1)^{L(x)-i+1}) \end{aligned}$$

- head

$$\begin{aligned} hd(x) &= \psi(b_1) \\ &= prefix(x,1) \end{aligned}$$

- select

$$\begin{aligned} select(x,i) &= x_{(i)} \\ &= \psi(b_i) \\ &= hd(suffix(x,i)) \end{aligned}$$

- concatenation. Let $x = \psi(b)$ and $y = \psi(c)$

$$\begin{aligned} [x,y] &= \psi(bc) \\ &= x(p+1)^{L(y)} + y \end{aligned}$$

  decomposition theorem of base $p+1$ numbers

1.1 coding $(A \cup Z)^+$ in $\mathbb{N}$:

- single symbols:
$$\psi(a_i) = i$$
- strings $b = b[1:L]$ interpreted as numbers with base $p+1$
$$\psi(b[1:L]) = \sum_{i=1}^{L} \psi(b_i)(p+1)^{L-i}$$
injective as $\Psi(a_i) \neq 0$, thus no 'leading zeros'.

1.2 operations on coded strings

Let $x = \psi(b)$ with $b = b[1:L]$.

- length of b
$$L(x) = \min\{j \;:\; j \leq x \;,\; (p+1)^j > x\}$$
pr (bounded μ-operator)

- prefix
$$\begin{aligned} prefix(x,i) &= \psi(b[1:i]) \\ &= \lfloor x/(p+1)^{L(x)-i} \rfloor \end{aligned}$$

- suffix
$$\begin{aligned} suffix(x,i) &= \psi(b[i:L]) \\ &= rem(x,(p+1)^{L(x)-i+1}) \end{aligned}$$

- head
$$\begin{aligned} hd(x) &= \psi(b_1) \\ &= prefix(x,1) \end{aligned}$$

- select
$$\begin{aligned} select(x,i) &= x_{(i)} \\ &= \psi(b_i) \\ &= hd(suffix(x,i)) \end{aligned}$$

- concatenation. Let $x = \psi(b)$ and $y = \psi(c)$
$$\begin{aligned} [x,y] &= \psi(bc) \\ &= x(p+1)^{L(y)} + y \end{aligned}$$
decomposition theorem of base $p+1$ numbers

$$[u,v,w] = [[u,v],w] \quad , \quad [u,v,w,x,y] = [[u,v,w],x,y]$$

etc.

2 Operations on configurations

- code of a state Let

$$\psi(Z) = \{\psi(z) \ : \ z \in Z\}$$

  Then predicate

$$y \in \psi(Z)$$

  pr (true for finitely many y)

2 Operations on configurations

- code of a state Let
$$\psi(Z) = \{\psi(z) \ : \ z \in Z\}$$
Then predicate
$$y \in \psi(Z)$$
pr (true for finitely many y)

- position i of state in a configuration
$$k = uazbv$$
with non empty u, a, b, v Let $x = \Psi(k)$:
$$\begin{aligned} \tilde{z}(x) &= i \quad \text{with} \quad k_i \in Z \\ &= \min\{j \ : \ j \leq x \ , \ x_{(j)} \in \psi(Z)\} \end{aligned}$$

2 Operations on configurations

- code of a state Let
$$\psi(Z) = \{\psi(z) \ : \ z \in Z\}$$
Then predicate
$$y \in \psi(Z)$$
pr (true for finitely many y)

- position i of state in a configuration
$$k = uazbv$$
with non empty u, a, b, v Let $x = \Psi(k)$:
$$\begin{aligned} \tilde{z}(x) &= i \quad \text{with} \quad k_i \in Z \\ &= \min\{j \ : \ j \leq x \ , \ x_{(j)} \in \psi(Z)\} \end{aligned}$$

- extracting u, a, z, b, v:
$$\begin{aligned} a(x) &= x_{(\tilde{z}(x)-1)} \\ z(x) &= x_{(\tilde{z}(x))} \\ b(x) &= x_{(\tilde{z}(x)+1)} \\ u(x) &= prefix(x, \tilde{z}(x) - 2) \\ v(x) &= suffix(x, \tilde{z}(x) + 2) \end{aligned}$$

3 successor configuration and computations

- code of a state Let

$$\psi(Z) = \{\psi(z) \;:\; z \in Z\}$$

Then predicate

$$y \in \psi(Z)$$

pr (true for finitely many y)

- position i of state in a configuration

$$k = uazbv$$

with non empty u, a, b, v Let $x = \Psi(k)$:

$$\begin{aligned} \tilde{z}(x) &= i \quad \text{with} \quad k_i \in Z \\ &= \min\{j \;:\; j \leq x\,,\, x_{(j)} \in \psi(Z)\} \end{aligned}$$

- extracting u, a, z, b, v:

$$\begin{aligned} a(x) &= x_{(\tilde{z}(x)-1)} \\ z(x) &= x_{(\tilde{z}(x))} \\ b(x) &= x_{(\tilde{z}(x)+1)} \\ u(x) &= prefix(x, \tilde{z}(x)-2) \\ v(x) &= suffix(x, \tilde{z}(x)+2) \end{aligned}$$

- coding the transition function by

$$\tilde{\delta} : \mathbb{N} \to \mathbb{N}$$

the 3 lines
which contain the world
of computation

$$\boxed{\tilde{\delta}(\Psi(azb)) = \begin{cases} \psi(az'c) & \delta(z,a) = (z',c,N) \\ \psi(acz') & \delta(z,a) = (z',c,R) \\ \psi(za'c) & \delta(z,a) = (z',c,L) \end{cases}}$$

$$\tilde{\delta}(y) = 0 \quad \text{if } y \neq \psi(azb) \text{ for some } a, z, b$$

where are other 9 cases of δ?

3 successor configuration and computations

- code of a state Let
$$\psi(Z) = \{\psi(z) \ : \ z \in Z\}$$
Then predicate
$$y \in \psi(Z)$$
pr (true for finitely many y)

- position i of state in a configuration
$$k = uazbv$$
with non empty u,a,b,v Let $x = \Psi(k)$:
$$\begin{aligned} \tilde{z}(x) &= i \quad \text{with} \quad k_i \in Z \\ &= \min\{j \ : \ j \leq x \ , \ x_{(j)} \in \psi(Z)\} \end{aligned}$$

- extracting u,a,z,b,v:
$$\begin{aligned} a(x) &= x_{(\tilde{z}(x)-1)} \\ z(x) &= x_{(\tilde{z}(x))} \\ b(x) &= x_{(\tilde{z}(x)+1)} \\ u(x) &= prefix(x, \tilde{z}(x)-2) \\ v(x) &= suffix(x, \tilde{z}(x)+2) \end{aligned}$$

- coding the transition function by
$$\tilde{\delta} : \mathbb{N} \rightarrow \mathbb{N}$$

the 3 lines
which contain the world
of computation

$$\boxed{\tilde{\delta}(\Psi(azb)) = \begin{cases} \psi(az'c) & \delta(z,a) = (z',c,N) \\ \psi(acz') & \delta(z,a) = (z',c,R) \\ \psi(za'c) & \delta(z,a) = (z',c,L) \end{cases}}$$
$$\tilde{\delta}(y) = 0 \quad \text{if } y \neq \psi(azb) \text{ for some } \ a,z,b$$

where are other 9 cases of δ?

- will code initial configuration of the form
$$k_0 = BBz_0wBB$$

- coded next configuration: surround by B's.
$$\Delta(x) = [\psi(B), u(x), \tilde{\delta}([a(x), z(x), b(x)], v(x), \psi(B)]$$

3 successor configuration and computations

- code of a state Let
$$\psi(Z) = \{\psi(z) \ : \ z \in Z\}$$
Then predicate
$$y \in \psi(Z)$$
pr (true for finitely many y)

- position i of state in a configuration
$$k = uazbv$$
with non empty u, a, b, v Let $x = \Psi(k)$:
$$\begin{aligned} \tilde{z}(x) &= i \quad \text{with} \quad k_i \in Z \\ &= \min\{j \ : \ j \leq x \ , \ x_{(j)} \in \psi(Z)\} \end{aligned}$$

- extracting u, a, z, b, v:
$$\begin{aligned} a(x) &= x_{(\tilde{z}(x)-1)} \\ z(x) &= x_{(\tilde{z}(x))} \\ b(x) &= x_{(\tilde{z}(x)+1)} \\ u(x) &= prefix(x, \tilde{z}(x) - 2) \\ v(x) &= suffix(x, \tilde{z}(x) + 2) \end{aligned}$$

- coding the transition function by
$$\tilde{\delta} : \mathbb{N} \to \mathbb{N}$$

the 3 lines
which contain the world
of computation

$$\boxed{\tilde{\delta}(\Psi(azb)) = \begin{cases} \psi(az'c) & \delta(z,a) = (z', c, N) \\ \psi(acz') & \delta(z,a) = (z', c, R) \\ \psi(za'c) & \delta(z,a) = (z', c, L) \end{cases}}$$
$$\tilde{\delta}(y) = 0 \quad \text{if } y \neq \psi(azb) \text{ for some } \ a, z, b$$

where are other 9 cases of δ?

- will code initial configuration of the form
$$k_0 = BBz_0wBB$$

- coded next configuration: surround by B's.
$$\Delta(x) = [\psi(B), u(x), \tilde{\delta}([a(x), z(x), b(x)], v(x), \psi(B)]$$

- n'th successor configuration
$$\begin{aligned} I(0, x) &= x \\ I(n+1, x) &= \Delta(I(n, x)) \end{aligned}$$

3 successor configuration and computations

- coding the transition function by

$$\tilde{\delta} : \mathbb{N} \to \mathbb{N}$$

$$\boxed{\tilde{\delta}(\Psi(azb)) = \begin{cases} \psi(az'c) & \delta(z,a) = (z',c,N) \\ \psi(acz') & \delta(z,a) = (z',c,R) \\ \psi(za'c) & \delta(z,a) = (z',c,L) \end{cases}}$$

$$\tilde{\delta}(y) = 0 \quad \text{if } y \neq \psi(azb) \text{ for some } a,z,b$$

where are other 9 cases of δ?

- will code initial configuration of the form

$$k_0 = BBz_0wBB$$

- coded next configuration: surround by B's.

$$\Delta(x) = [\psi(B), u(x), \tilde{\delta}([a(x), z(x), b(x)], v(x), \psi(B)]$$

- n'th successor configuration

$$\begin{aligned} I(0,x) &= x \\ I(n+1,x) &= \Delta(I(n,x)) \end{aligned}$$

- remaining run time

$$T(x) = \min\{j \; : \; z(I(j,x)) \in \psi(E)\}$$

unbounded μ-operator; may not be defined.

3 successor configuration and computations

- coding the transition function by

$$\tilde{\delta} : \mathbb{N} \to \mathbb{N}$$

$$\boxed{\tilde{\delta}(\Psi(azb)) = \begin{cases} \psi(az'c) & \delta(z,a) = (z',c,N) \\ \psi(acz') & \delta(z,a) = (z',c,R) \\ \psi(za'c) & \delta(z,a) = (z',c,L) \end{cases}}$$

$$\tilde{\delta}(y) = 0 \quad \text{if } y \neq \psi(azb) \text{ for some } a,z,b$$

where are other 9 cases of δ?

- will code initial configuration of the form

$$k_0 = BBz_0wBB$$

- coded next configuration: surround by B's.

$$\Delta(x) = [\psi(B), u(x), \tilde{\delta}([a(x), z(x), b(x)], v(x), \psi(B)]$$

- n'th successor configuration

$$\begin{aligned} I(0,x) &= x \\ I(n+1,x) &= \Delta(I(n,x)) \end{aligned}$$

- remaining run time

$$T(x) = \min\{j \ : \ z(I(j,x)) \in \psi(E)\}$$

unbounded μ-operator; may not be defined.

- resulting end configuration (if existing):

$$R(x) = I(T(x),x)$$

4 Preprocessor

$$f : \mathbb{N}_0^r \to \mathbb{N}_0$$

result of preprocessor

$$P(x_1, \ldots, x_r) = \psi(BBz_0 bin(x_1)\# \ldots \# bin(x_r)BB)$$

4 Preprocessor

$$f : \mathbb{N}_0^r \to \mathbb{N}_0$$

result of preprocessor

$$P(x_1, \dots, x_r) = \psi(BBz_0bin(x_1)\#\dots\#bin(x_r)BB)$$

Lemma 2. *If*

$$g : \mathbb{N}_0^r + 1 \to \mathbb{N}_0$$

is pr, then also

$$h : \mathbb{N}_0^r + 1 \to \mathbb{N}_0$$

with

$$h(n,x) = \sum_{i=0}^{n} g(i,x)$$

was an exercise

4 Preprocessor

$$f : \mathbb{N}_0^r \to \mathbb{N}_0$$

result of preprocessor

$$P(x_1,\ldots,x_r) = \psi(BBz_0bin(x_1)\#\ldots\#bin(x_r)BB)$$

Lemma 2. *If*

$$g : \mathbb{N}_0^r + 1 \to \mathbb{N}_0$$

is pr, then also

$$h : \mathbb{N}_0^r + 1 \to \mathbb{N}_0$$

with

$$h(n,x) = \sum_{i=0}^{n} g(i,x)$$

$$\begin{array}{rcl} h(0,x) & = & g(0,x) \\ h(n+1,x) & = & h(n,x) + g(n+1,x) \end{array}$$

4 Preprocessor

$$f : \mathbb{N}_0^r \to \mathbb{N}_0$$

result of preprocessor

$$P(x_1,\ldots,x_r) = \psi(BBz_0bin(x_1)\#\ldots\#bin(x_r)BB)$$

Lemma 2. *If*

$$g : \mathbb{N}_0^r + 1 \to \mathbb{N}_0$$

is pr, then also

$$h : \mathbb{N}_0^r + 1 \to \mathbb{N}_0$$

with

$$h(n,x) = \sum_{i=0}^{n} g(i,x)$$

$$\begin{aligned} h(0,x) &= g(0,x) \\ h(n+1,x) &= h(n,x) + g(n+1,x) \end{aligned}$$

converting $x \in \mathbb{N}$ into $\Psi(bin(x))$:

- length of $bin(x)$:

$$\begin{aligned} s(x) &= |bin(x)| \\ &= \min\{j \,:\, j \leq x+1 \,,\, 2^j > x\} \end{aligned}$$

4 Preprocessor

$$f : \mathbb{N}_0^r \to \mathbb{N}_0$$

result of preprocessor

$$P(x_1,\ldots,x_r) = \psi(BBz_0bin(x_1)\#\ldots\#bin(x_r)BB)$$

Lemma 2. *If*

$$g : \mathbb{N}_0^r + 1 \to \mathbb{N}_0$$

is pr, then also

$$h : \mathbb{N}_0^r + 1 \to \mathbb{N}_0$$

with

$$h(n,x) = \sum_{i=0}^{n} g(i,x)$$

$$\begin{aligned} h(0,x) &= g(0,x) \\ h(n+1,x) &= h(n,x) + g(n+1,x) \end{aligned}$$

converting $x \in \mathbb{N}$ into $\Psi(bin(x))$:

- length of $bin(x)$:

$$\begin{aligned} s(x) &= |bin(x)| \\ &= \min\{j \;:\; j \leq x+1 \;,\; 2^j > x\} \end{aligned}$$

- let $bin(x) = y[s(x)-1:0]$

$$\begin{aligned} y_i &= y(x,i) \\ &= \lfloor rem(x,2^{i+1})/2^i \rfloor \end{aligned}$$

4 Preprocessor

$$f : \mathbb{N}_0^r \to \mathbb{N}_0$$

result of preprocessor

$$P(x_1,\dots,x_r) = \psi(BBz_0bin(x_1)\#\dots\#bin(x_r)BB)$$

Lemma 2. *If*

$$g : \mathbb{N}_0^r + 1 \to \mathbb{N}_0$$

is pr, then also

$$h : \mathbb{N}_0^r + 1 \to \mathbb{N}_0$$

with

$$h(n,x) = \sum_{i=0}^{n} g(i,x)$$

$$\begin{aligned} h(0,x) &= g(0,x) \\ h(n+1,x) &= h(n,x) + g(n+1,x) \end{aligned}$$

converting $x \in \mathbb{N}$ into $\Psi(bin(x))$:

- length of $bin(x)$:

$$\begin{aligned} s(x) &= |bin(x)| \\ &= \min\{j \,:\, j \leq x+1\,,\ 2^j > x\} \end{aligned}$$

- let $bin(x) = y[s(x)-1:0]$

$$\begin{aligned} y_i &= y(x,i) \\ &= \lfloor rem(x,2^{i+1})/2^i \rfloor \end{aligned}$$

then

$$\psi(bin(x)) = \sum_{i=0}^{s(x)-1} \psi(y(x,i)) \cdot (p+1)^i$$

- preprocessor

$$P(x_1,\dots,x_r) = [\psi(BB), \psi(z_0), \psi(bin(x_1)), \psi(\#), \dots, \psi(\#), \psi(bin(x_r)), \psi(BB)]$$

5 post processor

result $y \in \mathbb{N}_0$ leads to coded end configuration

$$x = \psi(B \dots B z_e bin(y) B \dots B)$$

5 post processor

result $y \in \mathbb{N}_0$ leads to coded end configuration

$$x = \psi(B \dots B z_e bin(y) B \dots B)$$

- cut off leading B's and z_e:

$$F_1(x) = suffix(x, \tilde{z}(x) + 1)$$

5 post processor

result $y \in \mathbb{N}_0$ leads to coded end configuration

$$x = \psi(B \dots B z_e bin(y) B \dots B)$$

- cut off leading B's and z_e:

$$F_1(x) = suffix(x, \tilde{z}(x) + 1)$$

- position of first B

$$F_2(x) = \min\{i \leq x \; : \; x_{(i)} = \psi(B)\}$$

5 post processor

result $y \in \mathbb{N}_0$ leads to coded end configuration

$$x = \psi(B \ldots B z_e bin(y) B \ldots B)$$

- cut off leading B's and z_e:

$$F_1(x) = suffix(x, \tilde{z}(x) + 1)$$

- position of first B

$$F_2(x) = \min\{i \leq x \ : \ x_{(i)} = \psi(B)\}$$

- cutting off trailing B's

$$\begin{aligned} F_3(x) &= prefix(F_1(x), F_2(F_1(x)) - 1) \\ &= \Psi(y) \\ &= z \end{aligned}$$

5 post processor

result $y \in \mathbb{N}_0$ leads to coded end configuration

$$x = \psi(B \dots B z_e bin(y) B \dots B)$$

- cut off leading B's and z_e:

$$F_1(x) = suffix(x, \tilde{z}(x) + 1)$$

- position of first B

$$F_2(x) = \min\{i \leq x \ : \ x_{(i)} = \psi(B)\}$$

- cutting off trailing B's

$$\begin{aligned} F_3(x) &= prefix(F_1(x), F_2(F_1(x)) - 1) \\ &= \Psi(y) \\ &= z \end{aligned}$$

- fix coding and convert z to number

$$\underline{\psi(0) = 1 \quad , \psi(1) = 2}$$

$$F_4(z) = \sum_{i=0}^{L(z)-1} (\lfloor rem(z, (p+1)^{i+1})/(p+1)^i \rfloor \underline{-1}) \cdot 2^i$$

5 post processor

result $y \in \mathbb{N}_0$ leads to coded end configuration

$$x = \psi(B \dots B z_e bin(y) B \dots B)$$

- cut off leading B's and z_e:

$$F_1(x) = suffix(x, \tilde{z}(x) + 1)$$

- position of first B

$$F_2(x) = \min\{i \leq x \ : \ x_{(i)} = \psi(B)\}$$

- cutting off trailing B's

$$\begin{aligned} F_3(x) &= prefix(F_1(x), F_2(F_1(x)) - 1) \\ &= \Psi(y) \\ &= z \end{aligned}$$

- fix coding and convert z to number

$$\psi(0) = 1 \quad , \psi(1) = 2$$

$$F_4(z) = \sum_{i=0}^{L(z)-1} (\lfloor rem(z, (p+1)^{i+1})/(p+1)^i \rfloor - 1) \cdot 2^i$$

- post processor

$$Q(x) = F_4(F_3(x)) = y$$

6 computing f

$$f(x) = Q(R(P(x_1, \ldots, x_r)))$$

6 computing f

$$f(x) = Q(R(P(x_1, \ldots, x_r)))$$

Kleene normal form

Lemma 3. *If*

$$f : \mathbb{N}_0^r \to \mathbb{N}_0$$

be μ-recursive. Then there are primitive recursive functions

$$g, h : \mathbb{N}_0^{r+1} \to \mathbb{N}_0$$

such that

$$f(x) = h(\mu g(x), x)$$

6 computing f

$$f(x) = Q(R(P(x_1,\ldots,x_r)))$$

Kleene normal form

Lemma 3. *If*

$$f : \mathbb{N}_0^r \to \mathbb{N}_0$$

be μ-recursive. Then there are primitive recursive functions

$$g,h : \mathbb{N}_0^{r+1} \to \mathbb{N}_0$$

such that

$$f(x) = h(\mu g(x),x)$$

i.e. the unlimited μ-operator is only applied once.

- compute f by 1-tape TM as shown before.
- simulate TM by μ-recursive functions as just shown