Understanding and Using Spector's Bar Recursion

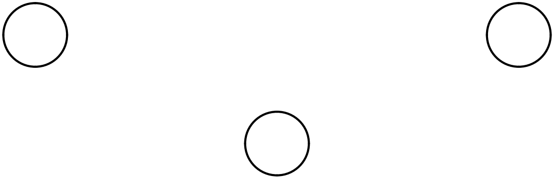
Paulo Oliva

Queen Mary, University of London, UK
(pbo@dcs.qmul.ac.uk)

Swansea, 4 July 2006

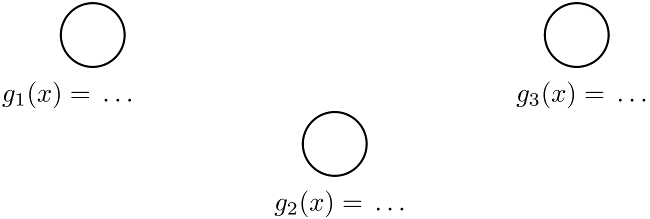
A 3-player game

1. Person $i \in \{1, 2, 3\}$ builds a (non-zero) function $g_i(x)$
2. Person i is assigned the number $x_i := g_i(i)$
3. $g_1(x_1) = x_2 + x_3$ and $g_2(x_2) = x_1 + x_3$ and $g_3(x_3) = x_1 + x_2$



A 3-player game

1. Person $i \in \{1, 2, 3\}$ builds a (non-zero) function $g_i(x)$
2. Person i is assigned the number $x_i := g_i(i)$
3. $g_1(x_1) = x_2 + x_3$ and $g_2(x_2) = x_1 + x_3$ and $g_3(x_3) = x_1 + x_2$



A 3-player game

1. Person $i \in \{1, 2, 3\}$ builds a (non-zero) function $g_i(x)$
2. Person i is assigned the number $x_i := g_i(i)$
3. $g_1(x_1) = x_2 + x_3$ and $g_2(x_2) = x_1 + x_3$ and $g_3(x_3) = x_1 + x_2$

$g_1(x) = \ldots$

$x_1 := g_1(1)$

$g_3(x) = \ldots$

$x_3 := g_3(3)$

$g_2(x) = \ldots$

$x_2 := g_2(2)$

A 3-player game

1. Person $i \in \{1, 2, 3\}$ builds a (non-zero) function $g_i(x)$
2. Person i is assigned the number $x_i := g_i(i)$
3. $g_1(x_1) = x_2 + x_3$ and $g_2(x_2) = x_1 + x_3$ and $g_3(x_3) = x_1 + x_2$

$g_1(x) = c_1$

$x_1 := g_1(1)$

$g_3(x) = c_3$

$x_3 := g_3(3)$

$g_2(x) = c_2$

$x_2 := g_2(2)$

A 3-player game

1. Person $i \in \{1, 2, 3\}$ builds a (non-zero) function $g_i(x)$
2. Person i is assigned the number $x_i := g_i(i)$
3. $g_1(x_1) = x_2 + x_3$ and $g_2(x_2) = x_1 + x_3$ and $g_3(x_3) = x_1 + x_2$

$g_1(x) = \lambda x.(5x + 4)$

$x_1 := g_1(1)$

$g_3(x) = \lambda x.29$

$x_3 := g_3(3)$

$g_2(x) = \lambda x.(x + 18)$

$x_2 := g_2(2)$

A 3-player game

1. Person $i \in \{1,2,3\}$ builds a (non-zero) function $g_i(x)$
2. Person i is assigned the number $x_i := g_i(i)$
3. $g_1(x_1) = x_2 + x_3$ and $g_2(x_2) = x_1 + x_3$ and $g_3(x_3) = x_1 + x_2$

9

$g_1(x) = \lambda x.(5x + 4)$

$x_1 := g_1(1)$

29

$g_3(x) = \lambda x.29$

$x_3 := g_3(3)$

20

$g_2(x) = \lambda x.(x + 18)$

$x_2 := g_2(2)$

A 3-player game

1. Person $i \in \{1, 2, 3\}$ builds a (non-zero) function $g_i(x)$
2. Person i is assigned the number $x_i := g_i(i)$
3. $g_1(x_1) = x_2 + x_3$ and $g_2(x_2) = x_1 + x_3$ and $g_3(x_3) = x_1 + x_2$

A 3-player game

1. Person $i \in \{1, 2, 3\}$ builds a (non-zero) function $g_i(x)$
2. Person i is assigned the number $x_i := g_i(i)$
3. $g_1(x_1) = x_2 + x_3$ and $g_2(x_2) = x_1 + x_3$ and $g_3(x_3) = x_1 + x_2$

$$G_3[x_1, x_2] := \lambda x_3 . x_1 + x_2$$

$$X_3[x_1, x_2] := x_1 + x_2$$

A 3-player game

1. Person $i \in \{1, 2, 3\}$ builds a (non-zero) function $g_i(x)$
2. Person i is assigned the number $x_i := g_i(i)$
3. $g_1(x_1) = x_2 + x_3$ and $g_2(x_2) = x_1 + x_3$ and $g_3(x_3) = x_1 + x_2$

$$G_3[x_1, x_2] := \lambda x_3 . x_1 + x_2$$

$$X_3[x_1, x_2] := x_1 + x_2$$

$$G_2[x_1] := \lambda x_2 . x_1 + X_3[x_1, x_2]$$

$$X_2[x_1] := x_1 + X_3[x_1, 2]$$

A 3-player game

1. Person $i \in \{1, 2, 3\}$ builds a (non-zero) function $g_i(x)$
2. Person i is assigned the number $x_i := g_i(i)$
3. $g_1(x_1) = x_2 + x_3$ and $g_2(x_2) = x_1 + x_3$ and $g_3(x_3) = x_1 + x_2$

$$G_3[x_1, x_2] := \lambda x_3 . x_1 + x_2$$

$$X_3[x_1, x_2] := x_1 + x_2$$

$$G_2[x_1] := \lambda x_2 . x_1 + X_3[x_1, x_2]$$

$$X_2[x_1] := x_1 + X_3[x_1, 2]$$

$$g_1 := \lambda x_1 . X_2[x_1] + X_3[x_1, X_2[x_1]]$$

$$x_1 := X_2[1] + X_3[1, X_2[1]]$$

A 3-player game

1. Person $i \in \{1, 2, 3\}$ builds a (non-zero) function $g_i(x)$
2. Person i is assigned the number $x_i := g_i(i)$
3. $g_1(x_1) = x_2 + x_3$ and $g_2(x_2) = x_1 + x_3$ and $g_3(x_3) = x_1 + x_2$

$$G_3[x_1, x_2] := \lambda x_3 . x_1 + x_2$$

$$X_3[x_1, x_2] := x_1 + x_2$$

$$G_2[x_1] := \lambda x_2 . x_1 + X_3[x_1, x_2]$$

$$X_2[x_1] := x_1 + X_3[x_1, 2]$$

$$g_1 := \lambda x_1 . X_2[x_1] + X_3[x_1, X_2[x_1]] \quad = \lambda x_1 . (5x_1 + 4)$$

$$x_1 := X_2[1] + X_3[1, X_2[1]] \quad = 9$$

A 3-player game

1. Person $i \in \{1, 2, 3\}$ builds a (non-zero) function $g_i(x)$
2. Person i is assigned the number $x_i := g_i(i)$
3. $g_1(x_1) = x_2 + x_3$ and $g_2(x_2) = x_1 + x_3$ and $g_3(x_3) = x_1 + x_2$

$$G_3[x_1, x_2] := \lambda x_3.x_1 + x_2$$

$$X_3[x_1, x_2] := x_1 + x_2$$

$$G_2[x_1] := \lambda x_2.x_1 + X_3[x_1, x_2] \qquad = \lambda x_2.(x_2 + 18)$$

$$X_2[x_1] := x_1 + X_3[x_1, 2] \qquad = 20$$

$$g_1 := \lambda x_1.X_2[x_1] + X_3[x_1, X_2[x_1]] \qquad = \lambda x_1.(5x_1 + 4)$$

$$x_1 := X_2[1] + X_3[1, X_2[1]] \qquad = 9$$

A 3-player game

1. Person $i \in \{1,2,3\}$ builds a (non-zero) function $g_i(x)$
2. Person i is assigned the number $x_i := g_i(i)$
3. $g_1(x_1) = x_2 + x_3$ and $g_2(x_2) = x_1 + x_3$ and $g_3(x_3) = x_1 + x_2$

$$
\begin{aligned}
G_3[x_1, x_2] &:= \lambda x_3 . x_1 + x_2 && = \lambda x_3 . 29 \\
X_3[x_1, x_2] &:= x_1 + x_2 && = 29 \\
G_2[x_1] &:= \lambda x_2 . x_1 + X_3[x_1, x_2] && = \lambda x_2 . (x_2 + 18) \\
X_2[x_1] &:= x_1 + X_3[x_1, 2] && = 20 \\
g_1 &:= \lambda x_1 . X_2[x_1] + X_3[x_1, X_2[x_1]] && = \lambda x_1 . (5x_1 + 4) \\
x_1 &:= X_2[1] + X_3[1, X_2[1]] && = 9
\end{aligned}
$$

A 3-player game

1. Person $i \in \{1, 2, 3\}$ builds a (non-zero) function $g_i(x)$
2. Person i is assigned the number $x_i := g_i(i)$
3. $g_1(x_1) = x_2 + x_3$ and $g_2(x_2) = x_1 + x_3$ and $g_3(x_3) = x_1 + x_2$

A ω-player game

1. Person $i \in \{1, 2, 3, \ldots\}$ builds a (non-zero) function $g_i(x)$
2. Person i is assigned the number $x_i := g_i(i)$
3. $g_1(x_1) = x_2 + x_3$ and $g_2(x_2) = x_1 + x_3$ and $g_3(x_3) = x_1 + x_2$

A ω-player game

1. Person $i \in \{1, 2, 3, \ldots\}$ builds a (non-zero) function $g_i(x)$
2. Person i is assigned the number $x_i := \Phi_i(g_i)$
3. $g_1(x_1) = x_2 + x_3$ and $g_2(x_2) = x_1 + x_3$ and $g_3(x_3) = x_1 + x_2$

A ω-player game

1. Person $i \in \{1, 2, 3, \ldots\}$ builds a (non-zero) function $g_i(x)$
2. Person i is assigned the number $x_i := \Phi_i(g_i)$
3. $g_i(x_i) = \Delta(x_1, x_2, \ldots)$

Outline

1. Bar recursion
   - Finite bar recursion
   - Spector's bar recursion
2. An application

Outline

1. Bar recursion
   - Finite bar recursion
   - Spector's bar recursion
2. An application

On Bar Recursion

- Facts:
  - Classical computational interpretation of countable choice (due to Spector'62)
  - In particular, provides interpretation of full comprehension
  - Difficult to understand

On Bar Recursion

- Facts:
  - Classical computational interpretation of countable choice (due to Spector'62)
  - In particular, provides interpretation of full comprehension
  - Difficult to understand
- Goal:
  - Explain bar recursion
  - Use it in simple (practical) examples
  - Understand how it solves the problem

Interpreting countable choice

- Give classical computation interpretation of cAC

$$\forall n^{\mathbb{N}} \exists y^{\tau} A(n,y) \to \exists f \forall n A(n, fn)$$

Interpreting countable choice

- Give classical computation interpretation of cAC

$$\forall n^{\mathbb{N}} \exists y^{\tau} A(n, y) \to \exists f \forall n A(n, fn)$$

- Need to interpret the negative translation of cAC

$$\forall n^{\mathbb{N}} \neg\neg\exists y^{\tau} A^{\dagger}(n, y) \to \neg\neg\exists f \forall n A^{\dagger}(n, fn)$$

Interpreting countable choice

- Give classical computation interpretation of cAC

$$\forall n^{\mathbb{N}} \exists y^{\tau} A(n, y) \rightarrow \exists f \forall n A(n, fn)$$

- Need to interpret the negative translation of cAC

$$\forall n^{\mathbb{N}} \neg\neg \exists y^{\tau} A^{\dagger}(n, y) \rightarrow \neg\neg \exists f \forall n A^{\dagger}(n, fn)$$

- Consider cAC for universal formulas

$$\forall n^{\mathbb{N}} \neg\neg \exists y^{\tau} \forall x^{\sigma} A_{\mathsf{qf}}(n, y, x) \rightarrow \neg\neg \exists f \forall n, x^{\sigma} A_{\mathsf{qf}}(n, fn, x)$$

Interpreting countable choice

- Give classical computation interpretation of cAC

$$\forall n^{\mathbb{N}} \exists y^{\tau} A(n, y) \rightarrow \exists f \forall n A(n, fn)$$

- Need to interpret the negative translation of cAC

$$\forall n^{\mathbb{N}} \neg\neg \exists y^{\tau} A^{\dagger}(n, y) \rightarrow \neg\neg \exists f \forall n A^{\dagger}(n, fn)$$

- Consider cAC for universal formulas

$$\forall n^{\mathbb{N}} \neg\neg \exists y^{\tau} \forall x^{\sigma} A_{\mathsf{qf}}(n, y, x) \rightarrow \neg\neg \exists f \forall n, x^{\sigma} A_{\mathsf{qf}}(n, fn, x)$$

Interpretation asks for functionals n, g, f depending on Φ, Ψ, Δ s.t.

$$\neg\neg A_{\mathsf{qf}}(n, \Phi_n g, g(\Phi_n g)) \rightarrow \neg\neg A_{\mathsf{qf}}(\Psi f, f(\Psi f), \Delta f)$$

Interpreting countable choice

How to produce n, g, f (parametrised by Φ, Ψ, Δ) such that

$$\neg\neg A_{\mathsf{qf}}(n, \Phi_n g, g(\Phi_n g)) \to \neg\neg A_{\mathsf{qf}}(\Psi f, f(\Psi f), \Delta f)$$

Interpreting countable choice

How to produce n, g, f (parametrised by Φ, Ψ, Δ) such that

$$\neg\neg A_{\mathsf{qf}}(n, \Phi_n g, g(\Phi_n g)) \to \neg\neg A_{\mathsf{qf}}(\Psi f, f(\Psi f), \Delta f)$$

Enough to satisfy equations:

$$\left\{ \begin{array}{lcl} n & \stackrel{\mathbb{N}}{=} & \Psi f \\ fn & \stackrel{\tau}{=} & \Phi_n g \\ g(fn) & \stackrel{\sigma}{=} & \Delta f \end{array} \right\}$$

Interpreting countable choice

How to produce n, g, f (parametrised by Φ, Ψ, Δ) such that

$$\neg\neg A_{\mathsf{qf}}(n, \Phi_n g, g(\Phi_n g)) \to \neg\neg A_{\mathsf{qf}}(\Psi f, f(\Psi f), \Delta f)$$

Enough to satisfy equations:

$$\left\{\begin{array}{lcl} n & \stackrel{\mathbb{N}}{=} & \Psi f \\ fn & \stackrel{\tau}{=} & \Phi_n g \\ g(fn) & \stackrel{\sigma}{=} & \Delta f \end{array}\right\} \quad \Rightarrow \quad \left\{\begin{array}{lcl} i & \leq & |\boldsymbol{x}| \\ x_i & \stackrel{\tau}{=} & \Phi_i g_i \\ g_i(x_i) & \stackrel{\sigma}{=} & \Delta \boldsymbol{x} \end{array}\right\}$$

Interpreting countable choice

How to produce n, g, f (parametrised by Φ, Ψ, Δ) such that

$$\neg\neg A_{\mathsf{qf}}(n, \Phi_n g, g(\Phi_n g)) \to \neg\neg A_{\mathsf{qf}}(\Psi f, f(\Psi f), \Delta f)$$

Enough to satisfy equations:

$$\left\{ \begin{array}{lcl} n & \stackrel{\mathbb{N}}{=} & \Psi f \\ fn & \stackrel{\tau}{=} & \Phi_n g \\ g(fn) & \stackrel{\sigma}{=} & \Delta f \end{array} \right\} \Rightarrow \left\{ \begin{array}{lcl} i & \leq & |\boldsymbol{x}| \\ x_i & \stackrel{\tau}{=} & \Phi_i g_i \\ g_i(x_i) & \stackrel{\sigma}{=} & \Delta \boldsymbol{x} \end{array} \right\}$$

Given $\Psi\hat{\boldsymbol{x}} < |\boldsymbol{x}|$ then $f := \hat{\boldsymbol{x}}$ and $n := \Psi\hat{\boldsymbol{x}}$ and $g := g_n$.

A particular case

Let's consider the particular case in which $\Psi \leq 3$

$$
\begin{aligned}
i &\leq 3 \\
x_i &\stackrel{\tau}{=} \Phi_i g_i \\
g_i(x_i) &\stackrel{\sigma}{=} \Delta \boldsymbol{x}
\end{aligned}
$$

A particular case

Let's consider the particular case in which $\Psi \leq 3$

$$\begin{array}{lcl} i & \leq & 3 \\ x_i & \stackrel{\tau}{=} & \Phi_i g_i \\ g_i(x_i) & \stackrel{\sigma}{=} & \Delta \boldsymbol{x} \end{array}$$

$$G_3[x_1, x_2] := \lambda x_3 . \Delta(x_1, x_2, x_3)$$
$$X_3[x_1, x_2] := \Phi_3(G_3[x_1, x_2])$$

A particular case

Let's consider the particular case in which $\Psi \leq 3$

$$\begin{array}{lcl} i & \leq & 3 \\ x_i & \stackrel{\tau}{=} & \Phi_i g_i \\ g_i(x_i) & \stackrel{\sigma}{=} & \Delta \boldsymbol{x} \end{array}$$

$$G_3[x_1, x_2] := \lambda x_3.\Delta(x_1, x_2, x_3)$$
$$X_3[x_1, x_2] := \Phi_3(G_3[x_1, x_2])$$

$$G_2[x_1] := \lambda x_2.\Delta(x_1, x_2, X_3[x_1, x_2])$$
$$X_2[x_1] := \Phi_2(G_2[x_1])$$

A particular case

Let's consider the particular case in which $\Psi \leq 3$

$$\begin{aligned} i &\leq 3 \\ x_i &\stackrel{\tau}{=} \Phi_i g_i \\ g_i(x_i) &\stackrel{\sigma}{=} \Delta \boldsymbol{x} \end{aligned}$$

$$G_3[x_1, x_2] := \lambda x_3.\Delta(x_1, x_2, x_3)$$
$$X_3[x_1, x_2] := \Phi_3(G_3[x_1, x_2])$$

$$G_2[x_1] := \lambda x_2.\Delta(x_1, x_2, X_3[x_1, x_2])$$
$$X_2[x_1] := \Phi_2(G_2[x_1])$$

$$g_1 := \lambda x_1.\Delta(x_1, X_2[x_1], X_3[x_1, X_2[x_1]])$$
$$x_1 := \Phi_1(g_1)$$

A particular case

Let's consider the particular case in which $\Psi \leq 3$

$$\begin{aligned} i &\leq 3 \\ x_i &\stackrel{\tau}{=} \Phi_i g_i \\ g_i(x_i) &\stackrel{\sigma}{=} \Delta \boldsymbol{x} \end{aligned}$$

$$G_3[x_1, x_2] := \lambda x_3.\Delta(x_1, x_2, x_3)$$

$$X_3[x_1, x_2] := \Phi_3(G_3[x_1, x_2])$$

$$g_2 := \quad G_2[x_1] := \lambda x_2.\Delta(x_1, x_2, X_3[x_1, x_2])$$

$$x_2 := \quad X_2[x_1] := \Phi_2(G_2[x_1])$$

$$g_1 := \lambda x_1.\Delta(x_1, X_2[x_1], X_3[x_1, X_2[x_1]])$$

$$x_1 := \Phi_1(g_1)$$

A particular case

Let's consider the particular case in which $\Psi \leq 3$

$$\begin{array}{lcl} i & \leq & 3 \\ x_i & \overset{\tau}{=} & \Phi_i g_i \\ g_i(x_i) & \overset{\sigma}{=} & \Delta \boldsymbol{x} \end{array}$$

$$g_3 := \quad G_3[x_1, x_2] := \lambda x_3.\Delta(x_1, x_2, x_3)$$

$$x_3 := \quad X_3[x_1, x_2] := \Phi_3(G_3[x_1, x_2])$$

$$g_2 := \quad G_2[x_1] := \lambda x_2.\Delta(x_1, x_2, X_3[x_1, x_2])$$

$$x_2 := \quad X_2[x_1] := \Phi_2(G_2[x_1])$$

$$g_1 := \lambda x_1.\Delta(x_1, X_2[x_1], X_3[x_1, X_2[x_1]])$$

$$x_1 := \Phi_1(g_1)$$

Finite bar recursion

General case (with a fixed bound k)

$$\begin{array}{lcl} i & \leq & k \\ x_i & \stackrel{\tau}{=} & \Phi_i g_i \\ g_i(x_i) & \stackrel{\sigma}{=} & \Delta(x_1, \ldots, x_k) \end{array}$$

Finite bar recursion

General case (with a fixed bound k)

$$\begin{array}{lcl} i & \leq & k \\ x_i & \stackrel{\tau}{=} & \Phi_i g_i \\ g_i(x_i) & \stackrel{\sigma}{=} & \Delta(x_1, \ldots, x_k) \end{array}$$

General solution can be constructed as follows ($\boldsymbol{x}_{i-1} \equiv x_1, \ldots, x_{i-1}$)

$$\mathsf{fB}(\boldsymbol{x}_{i-1}) = \begin{cases} x_1, \ldots, x_k & k = i - 1 \\ \mathsf{fB}(\boldsymbol{x}_{i-1}, X_i[\boldsymbol{x}_{i-1}]) & \text{otherwise} \end{cases}$$

where $X_i[\boldsymbol{x}_{i-1}] := \Phi_i G_i[\boldsymbol{x}_{i-1}]$ and $G_i[\boldsymbol{x}_{i-1}] := \lambda x_i.\Delta(\mathsf{fB}(\boldsymbol{x}_{i-1}, x_i))$.

Finite bar recursion

General case (with a fixed bound k)

$$\begin{array}{lcl} i & \leq & k \\ x_i & \stackrel{\tau}{=} & \Phi_i g_i \\ g_i(x_i) & \stackrel{\sigma}{=} & \Delta(x_1, \ldots, x_k) \end{array}$$

General solution can be constructed as follows ($\boldsymbol{x}_{i-1} \equiv x_1, \ldots, x_{i-1}$)

$$\mathsf{fB}(\boldsymbol{x}_{i-1}) = \begin{cases} x_1, \ldots, x_k & k = i - 1 \\ \mathsf{fB}(\boldsymbol{x}_{i-1}, X_i[\boldsymbol{x}_{i-1}]) & \text{otherwise} \end{cases}$$

where $X_i[\boldsymbol{x}_{i-1}] := \Phi_i G_i[\boldsymbol{x}_{i-1}]$ and $G_i[\boldsymbol{x}_{i-1}] := \lambda x_i.\Delta(\mathsf{fB}(\boldsymbol{x}_{i-1}, x_i))$.

Then take $\langle x_1, \ldots, x_k \rangle := \mathsf{fB}(\langle\ \rangle)$ and $g_i := G_i[\boldsymbol{x}_{i-1}]$.

Spector's bar recursion

Back to the original problem

$$
\begin{aligned}
i &\leq |\boldsymbol{x}| \\
x_i &\stackrel{\tau}{=} \Phi_i g_i \\
g_i(x_i) &\stackrel{\sigma}{=} \Delta \boldsymbol{x}
\end{aligned}
$$

Spector's bar recursion

Back to the original problem

$$\begin{array}{lcl} i & \leq & |\boldsymbol{x}| \\ x_i & \stackrel{\tau}{=} & \Phi_i g_i \\ g_i(x_i) & \stackrel{\sigma}{=} & \Delta \boldsymbol{x} \end{array}$$

can be solved with $(\boldsymbol{x}_{i-1} \equiv x_1, \ldots, x_{i-1})$

$$\mathsf{BR}(\boldsymbol{x}_{i-1}) = \begin{cases} \boldsymbol{x}_{i-1} & \Psi\hat{\boldsymbol{x}} < i - 1 \\ \mathsf{BR}(\boldsymbol{x}_{i-1}, X_i[\boldsymbol{x}_{i-1}]) & \text{otherwise} \end{cases}$$

where $X_i[\boldsymbol{x}_{i-1}] := \Phi_i G_i[\boldsymbol{x}_{i-1}]$ and $G_i[\boldsymbol{x}_{i-1}] := \lambda x_i.\Delta(\mathsf{BR}(\boldsymbol{x}_{i-1}, x_i))$.

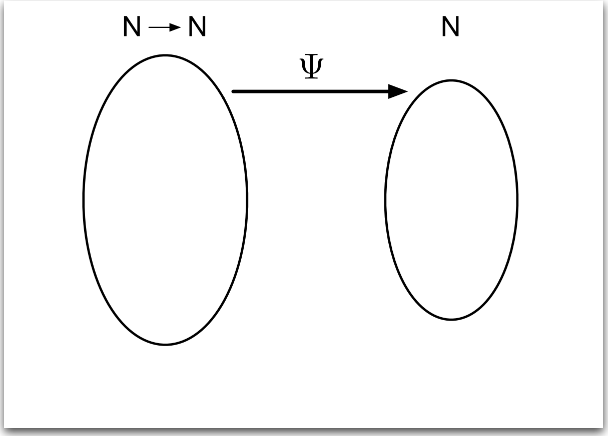
Finally, take $\boldsymbol{x} := \mathsf{fB}(\langle\ \rangle)$ and $g_i := G_i[\boldsymbol{x}_{i-1}]$.

Outline

No injection from $\mathbb{N} \to \mathbb{N}$ to $\mathbb{N}$

Theorem

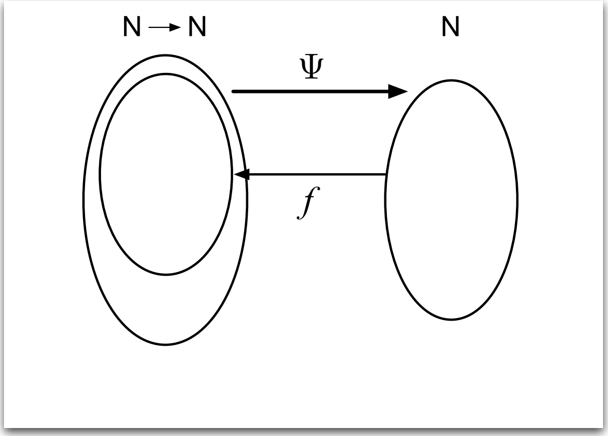
$\forall \Psi^{1\to 0} \exists \alpha^1, \beta^1 (\alpha \neq \beta \wedge \Psi\alpha = \Psi\beta)$.



No injection from $\mathbb{N} \to \mathbb{N}$ to $\mathbb{N}$

Theorem

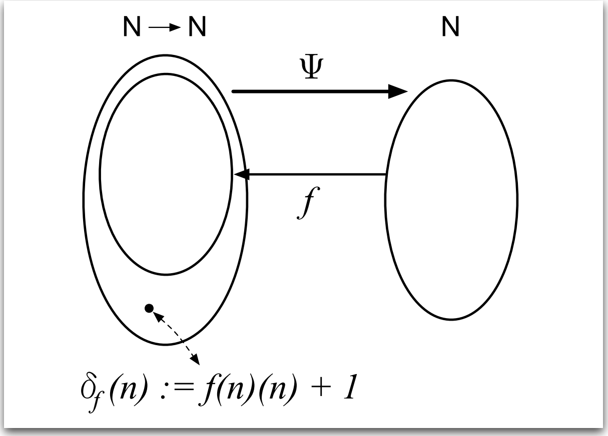
$\forall \Psi^{1\to 0} \exists \alpha^1, \beta^1 (\alpha \neq \beta \wedge \Psi\alpha = \Psi\beta).$



No injection from $\mathbb{N} \to \mathbb{N}$ to $\mathbb{N}$

Theorem

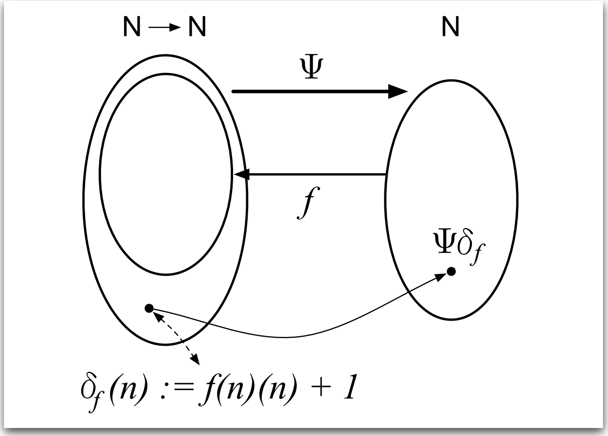
$\forall \Psi^{1\to 0} \exists \alpha^1, \beta^1 (\alpha \neq \beta \wedge \Psi\alpha = \Psi\beta).$



No injection from $\mathbb{N} \to \mathbb{N}$ to $\mathbb{N}$

Theorem

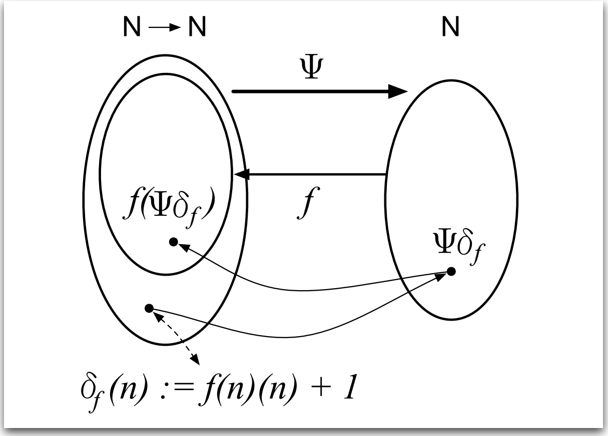
$\forall \Psi^{1\to 0} \exists \alpha^1, \beta^1 (\alpha \neq \beta \wedge \Psi\alpha = \Psi\beta)$.



No injection from $\mathbb{N} \to \mathbb{N}$ to $\mathbb{N}$

Theorem

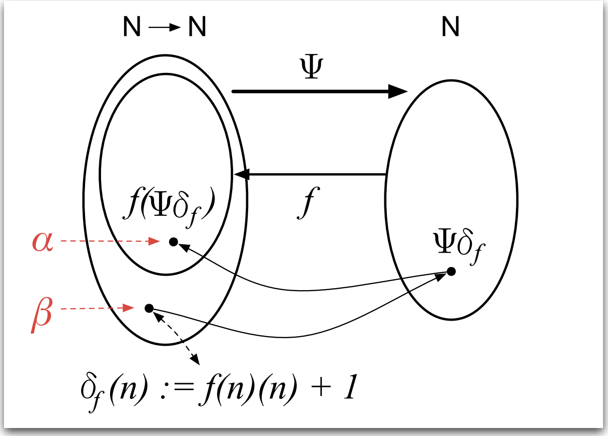
$\forall \Psi^{1\to 0} \exists \alpha^1, \beta^1 (\alpha \neq \beta \wedge \Psi\alpha = \Psi\beta)$.



No injection from $\mathbb{N} \to \mathbb{N}$ to $\mathbb{N}$

Theorem

$\forall \Psi^{1\to 0} \exists \alpha^1, \beta^1 (\alpha \neq \beta \wedge \Psi\alpha = \Psi\beta)$.



Bar recursive solution

Key in solution is the construction of the enumeration $f : \mathbb{N} \to (\mathbb{N} \to \mathbb{N})$

$$\forall k(\exists \beta(\Psi\beta = k) \to \Psi(fk) = k)$$

Bar recursive solution

Key in solution is the construction of the enumeration $f : \mathbb{N} \to (\mathbb{N} \to \mathbb{N})$

$$\forall k(\exists \beta(\Psi\beta = k) \to \Psi(fk) = k)$$

Could try to build a *finite approximation* t for f

$$\forall k < |t|(\exists \beta(\Psi\beta = k) \to \Psi(t_k) = k)$$

Bar recursive solution

Key in solution is the construction of the enumeration $f : \mathbb{N} \to (\mathbb{N} \to \mathbb{N})$

$$\forall k(\exists \beta(\Psi \beta = k) \to \Psi(fk) = k)$$

Could try to build a *finite approximation* t for f

$$\forall k < |t|(\exists \beta(\Psi \beta = k) \to \Psi(t_k) = k)$$

This is still too strong. We build finite approximation t satisfying

$$\forall k < |t|(\Psi(\delta_{\hat{t}}) = k \to \Psi(t_k) = k) \text{ and } \Psi(\delta_{\hat{t}}) \leq |t|$$

Bar recursive solution

Key in solution is the construction of the enumeration $f : \mathbb{N} \to (\mathbb{N} \to \mathbb{N})$

$$\forall k(\exists \beta(\Psi\beta = k) \to \Psi(fk) = k)$$

Could try to build a *finite approximation* t for f

$$\forall k < |t|(\exists \beta(\Psi\beta = k) \to \Psi(t_k) = k)$$

This is still too strong. We build finite approximation t satisfying

$$\forall k < |t|(\Psi(\delta_{\hat{t}}) = k \to \Psi(t_k) = k) \text{ and } \Psi(\delta_{\hat{t}}) \leq |t|$$

Let

$$\mathsf{B}(s, k) := \begin{cases} s & \Psi\delta_{\hat{s}} < k \\ r & \Psi\delta_{\hat{r}} \neq k \quad (\text{and } \Psi\delta_{\hat{s}} \geq k) \\ \mathsf{B}(s * \delta_{\hat{r}}, k+1) & \Psi\delta_{\hat{r}} = k \quad (\text{and } \Psi\delta_{\hat{s}} \geq k) \end{cases}$$

where $r := \mathsf{B}(s * 0^1, k+1)$. Then take $t := \mathsf{B}(\langle\, \rangle, 0)$.

Final remarks

- Other application in the paper
  - compute fixed-point for update procedures (Avigad'02)

Final remarks

- Other application in the paper
  - compute fixed-point for update procedures (Avigad'02)
- Models of bar recursion
  - Total continuous functions (Scarpellini'71)
  - Strongly majorizable functions (Bezem'85)

Final remarks

- Other application in the paper
  - compute fixed-point for update procedures (Avigad'02)
- Models of bar recursion
  - Total continuous functions (Scarpellini'71)
  - Strongly majorizable functions (Bezem'85)
- Interpretation used
  - Dialectica interpretation (Gödel'58)
  - Using realizability interpretations:
    Modified bar recursion (Berardi et al.'98, Berger/O.'05)