Ribbon Proofs for Separation Logic

John Wickerson, University of Cambridge

(joint work with Matthew Parkinson and Mike Dodds)

LICS, Dubrovnik, 27th June 2012

A problem

Our solution

A worked example

Where now?

```
mchunkptr b, p;
idx += ~smallbits & 1; /* Uses next bin if idx empty */
```

$$\left\{\begin{array}{l}\exists\{U_i \mid i\in[0,63)\}, n.\ arena(A_{\mathsf{a}} \uplus (\biguplus_{i=0}^{64}. U_i)_{\mathsf{u}}) \ast \texttt{least\_addr} = 5\mathsf{w}\\ \ast\ n\mathsf{w} = \lceil\texttt{bytes}\rceil_{\mathsf{w}} \ast 8\texttt{idx} \geq (n+1)\mathsf{w} \ast 2 \leq \texttt{idx} < 32 \ast \texttt{smallmap}_{[\texttt{idx}]} = 1\\ \ast \bigstar_{i=0}^{32}. smallbin_i(U_i) \ast \bigstar_{i=0}^{32}. treebin_i(U_{i+32})\end{array}\right\}$$

```
b = smallbin_at(gm, idx);
```

$$\left\{\begin{array}{l}\exists\{U_i \mid i\in[0,63)\}, n.\ arena(A_{\mathsf{a}} \uplus (\biguplus_{i=0}^{64}. U_i)_{\mathsf{u}}) \ast \texttt{least\_addr} = 5\mathsf{w}\\ \ast\ n\mathsf{w} = \lceil\texttt{bytes}\rceil_{\mathsf{w}} \ast 8\texttt{idx} \geq (n+1)\mathsf{w} \ast 2 \leq \texttt{idx} < 32 \ast \texttt{smallmap}_{[\texttt{idx}]} = 1\\ \ast\ \texttt{b} = \texttt{smallbins} + 8\texttt{idx} \ast bin(|\texttt{idx}|, \texttt{b}, U_{\texttt{idx}}) \ast U_{\texttt{idx}} \neq \{\}\\ \ast \bigstar_{i\in[0..32)-\texttt{idx}}. smallbin_i(U_i) \ast \bigstar_{i=0}^{32}. treebin_i(U_{i+32})\end{array}\right\}$$

```
// rename U_idx to U_idx++[p+2w->8idx-1w]
```

$$\left\{\begin{array}{l}\exists\{U_i \mid i\in[0,63)\}, p, n.\ arena(A_{\mathsf{a}} \uplus (\biguplus_{i=0}^{64}. U_i)_{\mathsf{u}} \uplus \{p+2\mathsf{w} \mapsto_{\mathsf{u}} 8\texttt{idx} - 1\mathsf{w}\})\\ \ast\ \texttt{least\_addr} = 5\mathsf{w} \ast n\mathsf{w} = \lceil\texttt{bytes}\rceil_{\mathsf{w}} \ast 8\texttt{idx} \geq (n+1)\mathsf{w} \ast 2 \leq \texttt{idx} < 32\\ \ast\ \texttt{smallmap}_{[\texttt{idx}]} = 1 \ast \texttt{b} = \texttt{smallbins} + 8\texttt{idx}\\ \ast\ \texttt{b} \xmapsto{\mathsf{fd}} p \ast p \xmapsto{\mathsf{bk}} \texttt{b} \ast (bnode\,|\texttt{idx}|)^{*}(p, \texttt{b}, U_{\texttt{idx}} \uplus \{p+2\mathsf{w} \mapsto 8\texttt{idx} - 1\mathsf{w}\})\\ \ast \bigstar_{i\in[0..32)-\texttt{idx}}. smallbin_i(U_i) \ast \bigstar_{i=0}^{32}. treebin_i(U_{i+32})\end{array}\right\}$$

```
p = b->fd;
```

$$\left\{\begin{array}{l}\exists\{U_i \mid i\in[0,63)\}, n, F.\ arena(A_{\mathsf{a}} \uplus (\biguplus_{i=0}^{64}. U_i)_{\mathsf{u}} \uplus \{\texttt{p}+2\mathsf{w} \mapsto_{\mathsf{u}} 8\texttt{idx} - 1\mathsf{w}\})\\ \ast\ \texttt{least\_addr} = 5\mathsf{w} \ast n\mathsf{w} = \lceil\texttt{bytes}\rceil_{\mathsf{w}} \ast 8\texttt{idx} \geq (n+1)\mathsf{w} \ast 2 \leq \texttt{idx} < 32\\ \ast\ \texttt{smallmap}_{[\texttt{idx}]} = 1 \ast \texttt{b} = \texttt{smallbins} + 8\texttt{idx}\\ \ast\ \texttt{b} \xmapsto{\mathsf{fd}} \texttt{p} \ast \texttt{p} \xmapsto{\mathsf{bk}} \texttt{b} \ast \frac{1}{2}(\texttt{p} \xmapsto{\mathsf{size}} 8\texttt{idx}) \ast \texttt{p} \xmapsto{\mathsf{fd}} F \ast F \xmapsto{\mathsf{bk}} \texttt{p} \ast (bnode\,|\texttt{idx}|)^{*}(F, \texttt{b}, U_{\texttt{idx}})\\ \ast \bigstar_{i\in[0..32)-\texttt{idx}}. smallbin_i(U_i) \ast \bigstar_{i=0}^{32}. treebin_i(U_{i+32})\end{array}\right\}$$

```
//assert(chunksize(p) == small_index2size(idx));
unlink_first_small_chunk(gm, b, p, idx);
```

$$\left\{\begin{array}{l}\exists\{U_i \mid i\in[0,63)\}, n.\ arena(A_{\mathsf{a}} \uplus (\biguplus_{i=0}^{64}. U_i)_{\mathsf{u}} \uplus \{\texttt{p}+2\mathsf{w} \mapsto_{\mathsf{u}} 8\texttt{idx} - 1\mathsf{w}\})\\ \ast\ \texttt{least\_addr} = 5\mathsf{w} \ast n\mathsf{w} = \lceil\texttt{bytes}\rceil_{\mathsf{w}} \ast 8\texttt{idx} \geq (n+1)\mathsf{w} \ast 2 \leq \texttt{idx} < 32\\ \ast\ \frac{1}{2}(\texttt{p} \xmapsto{\mathsf{size}} 8\texttt{idx}) \ast \texttt{p} \xmapsto{\mathsf{fd}} \_ \ast \texttt{p} \xmapsto{\mathsf{bk}} \_ \ast \bigstar_{i=0}^{32}. smallbin_i(U_i) \ast \bigstar_{i=0}^{32}. treebin_i(U_{i+32})\end{array}\right\}$$

$$\left\{\begin{array}{l}\exists\{U_i \mid i\in[0,63)\}, B_1, B_2, n.\ coallesced(A_{\mathsf{a}} \uplus (\biguplus_{i=0}^{64}. U_i)_{\mathsf{u}} \uplus \{\texttt{p}+2\mathsf{w} \mapsto_{\mathsf{u}} 8\texttt{idx} - 1\mathsf{w}\})\\ \ast\ \texttt{start} \xmapsto{\mathsf{prevfoot}} \_ \ast \texttt{start} \xmapsto{\mathsf{pinuse}} 1 \ast ublock(\texttt{top}, \texttt{top} + \texttt{topsize}, \_)\\ \ast\ block^{*}(\texttt{start}, \texttt{p}, B_1) \ast ublock(\texttt{p}, \texttt{p} + 8\texttt{idx}, \{\texttt{p}+2\mathsf{w} \mapsto_{\mathsf{u}} 8\texttt{idx} - 1\mathsf{w}\})\\ \ast\ block^{*}(\texttt{p} + 8\texttt{idx}, \texttt{top}, B_2) \ast B_1 \uplus B_2 = A_{\mathsf{a}} \uplus (\biguplus_{i=0}^{64}. U_i)_{\mathsf{u}}\\ \ast\ \texttt{least\_addr} = 5\mathsf{w} \ast n\mathsf{w} = \lceil\texttt{bytes}\rceil_{\mathsf{w}} \ast 8\texttt{idx} \geq (n+1)\mathsf{w} \ast 2 \leq \texttt{idx} < 32\\ \ast\ \frac{1}{2}(\texttt{p} \xmapsto{\mathsf{size}} 8\texttt{idx}) \ast \texttt{p} \xmapsto{\mathsf{fd}} \_ \ast \texttt{p} \xmapsto{\mathsf{bk}} \_ \ast \bigstar_{i=0}^{32}. smallbin_i(U_i) \ast \bigstar_{i=0}^{32}. treebin_i(U_{i+32})\end{array}\right\}$$

Proof outlines are bad

$\{x \mapsto 0 * y \mapsto 0 * z \mapsto 0\}$

$[x] := 1$

$\{x \mapsto 1 * y \mapsto 0 * z \mapsto 0\}$

$[y] := 1$

$\{x \mapsto 1 * y \mapsto 1 * z \mapsto 0\}$

$[z] := 1$

$\{x \mapsto 1 * y \mapsto 1 * z \mapsto 1\}$

× Repetitive

Proof outlines are bad

$\{x \mapsto 0 * y \mapsto 0 * z \mapsto 0\}$

$[x] := 1$

$\{x \mapsto 1 * y \mapsto 0 * z \mapsto 0\}$

$[y] := 1$

$\{x \mapsto 1 * y \mapsto 1 * z \mapsto 0\}$

$[z] := 1$

$\{x \mapsto 1 * y \mapsto 1 * z \mapsto 1\}$

- × Repetitive
- × Hard to interpret effect of each instruction

Proof outlines are bad

$\{x \mapsto 0 * y \mapsto 0 * z \mapsto 0\}$

$[x]{:=}1$

$\{x \mapsto 1 * y \mapsto 0 * z \mapsto 0\}$

$[y]{:=}1$

$\{x \mapsto 1 * y \mapsto 1 * z \mapsto 0\}$

$[z]{:=}1$

$\{x \mapsto 1 * y \mapsto 1 * z \mapsto 1\}$

- × Repetitive
- × Hard to interpret effect of each instruction

Proof outlines are bad

$$\frac{}{\{x \mapsto 0\}\ [x] := 1\ \{x \mapsto 1\}}\ \text{Upd}$$

$$\frac{\{x \mapsto 0\}\ [x] := 1\ \{x \mapsto 1\}}{\left\{\begin{array}{r} x \mapsto 0 \\ * \ y \mapsto 0 \\ * \ z \mapsto 0 \end{array}\right\} [x] := 1 \left\{\begin{array}{r} x \mapsto 1 \\ * \ y \mapsto 0 \\ * \ z \mapsto 0 \end{array}\right\}}\ \text{Frm}$$

$$\frac{}{\{y \mapsto 0\}\ [y] := 1\ \{y \mapsto 1\}}\ \text{Upd}$$

$$\frac{\{y \mapsto 0\}\ [y] := 1\ \{y \mapsto 1\}}{\left\{\begin{array}{r} x \mapsto 1 \\ * \ y \mapsto 0 \\ * \ z \mapsto 0 \end{array}\right\} [y] := 1 \left\{\begin{array}{r} x \mapsto 1 \\ * \ y \mapsto 1 \\ * \ z \mapsto 0 \end{array}\right\}}\ \text{Frm}$$

$$\frac{}{\{z \mapsto 0\}\ [z] := 1\ \{z \mapsto 1\}}\ \text{Upd}$$

$$\frac{\{z \mapsto 0\}\ [z] := 1\ \{z \mapsto 1\}}{\left\{\begin{array}{r} x \mapsto 1 \\ * \ y \mapsto 1 \\ * \ z \mapsto 0 \end{array}\right\} [z] := 1 \left\{\begin{array}{r} x \mapsto 1 \\ * \ y \mapsto 1 \\ * \ z \mapsto 1 \end{array}\right\}}\ \text{Frm}$$

$$\frac{\left\{\begin{array}{r} x \mapsto 1 \\ * \ y \mapsto 0 \\ * \ z \mapsto 0 \end{array}\right\} [y] := 1 \left\{\begin{array}{r} x \mapsto 1 \\ * \ y \mapsto 1 \\ * \ z \mapsto 0 \end{array}\right\} \qquad \left\{\begin{array}{r} x \mapsto 1 \\ * \ y \mapsto 1 \\ * \ z \mapsto 0 \end{array}\right\} [z] := 1 \left\{\begin{array}{r} x \mapsto 1 \\ * \ y \mapsto 1 \\ * \ z \mapsto 1 \end{array}\right\}}{\left\{\begin{array}{r} x \mapsto 1 \\ * \ y \mapsto 0 \\ * \ z \mapsto 0 \end{array}\right\} [y] := 1 \ ; \ [z] := 1 \left\{\begin{array}{r} x \mapsto 1 \\ * \ y \mapsto 1 \\ * \ z \mapsto 1 \end{array}\right\}}\ \text{Seq}$$

$$\frac{\left\{\begin{array}{r} x \mapsto 0 \\ * \ y \mapsto 0 \\ * \ z \mapsto 0 \end{array}\right\} [x] := 1 \left\{\begin{array}{r} x \mapsto 1 \\ * \ y \mapsto 0 \\ * \ z \mapsto 0 \end{array}\right\} \qquad \left\{\begin{array}{r} x \mapsto 1 \\ * \ y \mapsto 0 \\ * \ z \mapsto 0 \end{array}\right\} [y] := 1 \ ; \ [z] := 1 \left\{\begin{array}{r} x \mapsto 1 \\ * \ y \mapsto 1 \\ * \ z \mapsto 1 \end{array}\right\}}{\left\{\begin{array}{r} x \mapsto 0 \\ * \ y \mapsto 0 \\ * \ z \mapsto 0 \end{array}\right\} [x] := 1 \ ; \ [y] := 1 \ ; \ [z] := 1 \left\{\begin{array}{r} x \mapsto 1 \\ * \ y \mapsto 1 \\ * \ z \mapsto 1 \end{array}\right\}}\ \text{Seq}$$

Proof outlines are bad

$$
\dfrac{
\dfrac{\dfrac{}{\{x\mapsto 0\}\ [x] := 1\ \{x\mapsto 1\}}\,\text{Upd}}{\left\{\begin{array}{r} x\mapsto 0 \\ *\ y\mapsto 0 \\ *\ z\mapsto 0 \end{array}\right\} [x] := 1 \left\{\begin{array}{r} x\mapsto 1 \\ *\ y\mapsto 0 \\ *\ z\mapsto 0 \end{array}\right\}}\,\text{Frm}
\qquad
\dfrac{
\dfrac{
\dfrac{\dfrac{}{\{y\mapsto 0\}\ [y] := 1\ \{y\mapsto 1\}}\,\text{Upd}}{\left\{\begin{array}{r} y\mapsto 0 \\ *\ z\mapsto 0 \end{array}\right\} [y] := 1 \left\{\begin{array}{r} y\mapsto 0 \\ *\ z\mapsto 0 \end{array}\right\}}\,\text{Frm}
\quad
\dfrac{\dfrac{}{\{z\mapsto 0\}\ [z] := 1\ \{z\mapsto 1\}}\,\text{Upd}}{\left\{\begin{array}{r} y\mapsto 0 \\ *\ z\mapsto 0 \end{array}\right\} [z] := 1 \left\{\begin{array}{r} y\mapsto 0 \\ *\ z\mapsto 0 \end{array}\right\}}\,\text{Frm}
}{\left\{\begin{array}{r} y\mapsto 0 \\ *\ z\mapsto 0 \end{array}\right\} [y] := 1\ ;\ [z] := 1 \left\{\begin{array}{r} y\mapsto 1 \\ *\ z\mapsto 1 \end{array}\right\}}\,\text{Seq}
}{\left\{\begin{array}{r} x\mapsto 1 \\ *\ y\mapsto 0 \\ *\ z\mapsto 0 \end{array}\right\} [y] := 1\ ;\ [z] := 1 \left\{\begin{array}{r} x\mapsto 1 \\ *\ y\mapsto 1 \\ *\ z\mapsto 1 \end{array}\right\}}\,\text{Frm}
}{\left\{\begin{array}{r} x\mapsto 0 \\ *\ y\mapsto 0 \\ *\ z\mapsto 0 \end{array}\right\} [x] := 1\ ;\ [y] := 1\ ;\ [z] := 1 \left\{\begin{array}{r} x\mapsto 1 \\ *\ y\mapsto 1 \\ *\ z\mapsto 1 \end{array}\right\}}\,\text{Seq}
$$

Proof outlines are bad

$\{x \mapsto 0 * y \mapsto 0 * z \mapsto 0\}$

[x]:=1

$\{x \mapsto 1 * y \mapsto 0 * z \mapsto 0\}$

[y]:=1

$\{x \mapsto 1 * y \mapsto 1 * z \mapsto 0\}$

[z]:=1

$\{x \mapsto 1 * y \mapsto 1 * z \mapsto 1\}$

- × Repetitive
- × Hard to interpret effect of each instruction
- × Many 'equivalent' ways to apply Frame rule
- × Inflexible

A problem

Our solution

A worked example

Where now?

Proof outline vs. Ribbon proof

$\{x \mapsto 0 * y \mapsto 0 * z \mapsto 0\}$

$[x] := 1$

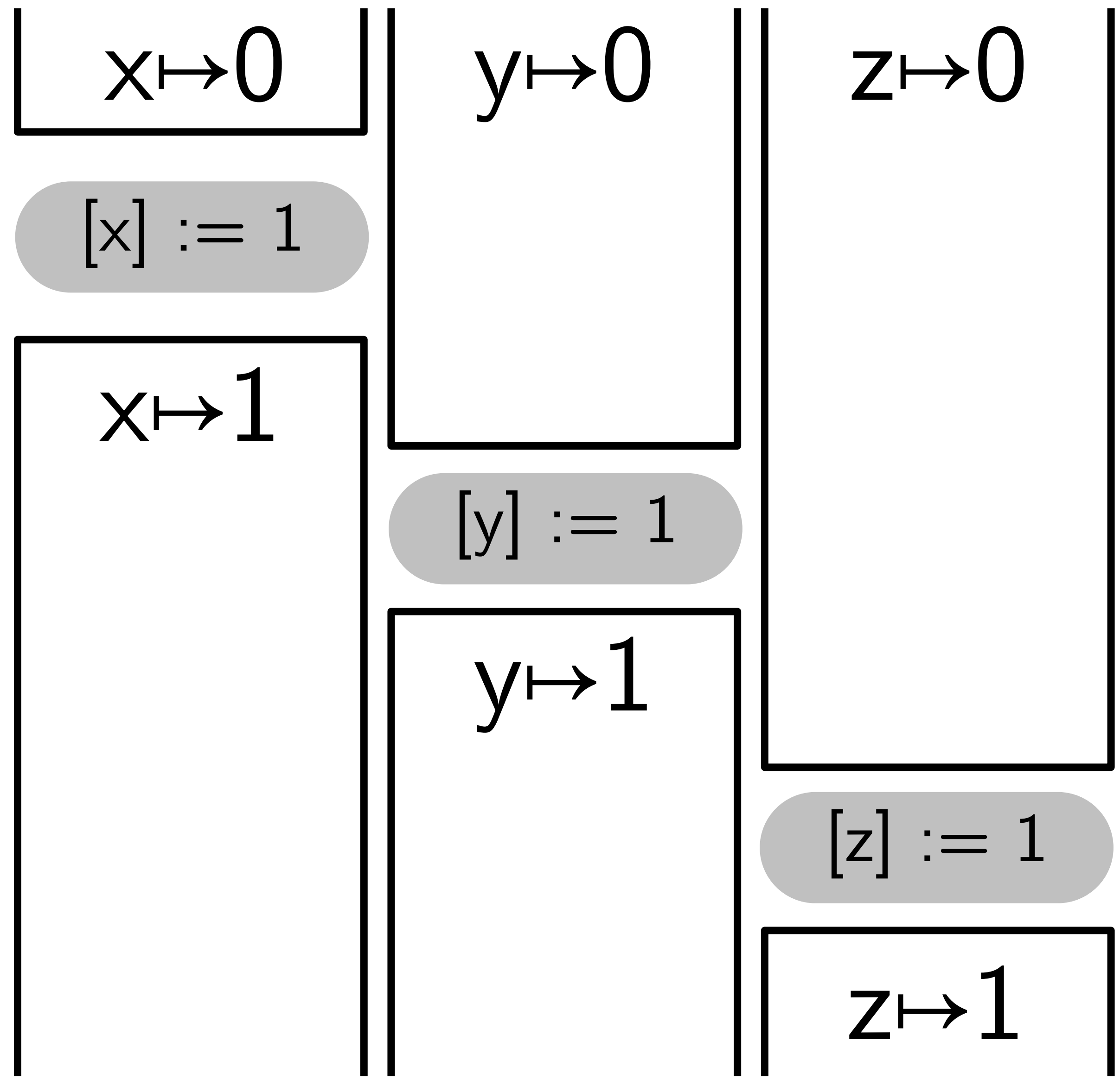
$\{x \mapsto 1 * y \mapsto 0 * z \mapsto 0\}$

$[y] := 1$

$\{x \mapsto 1 * y \mapsto 1 * z \mapsto 0\}$

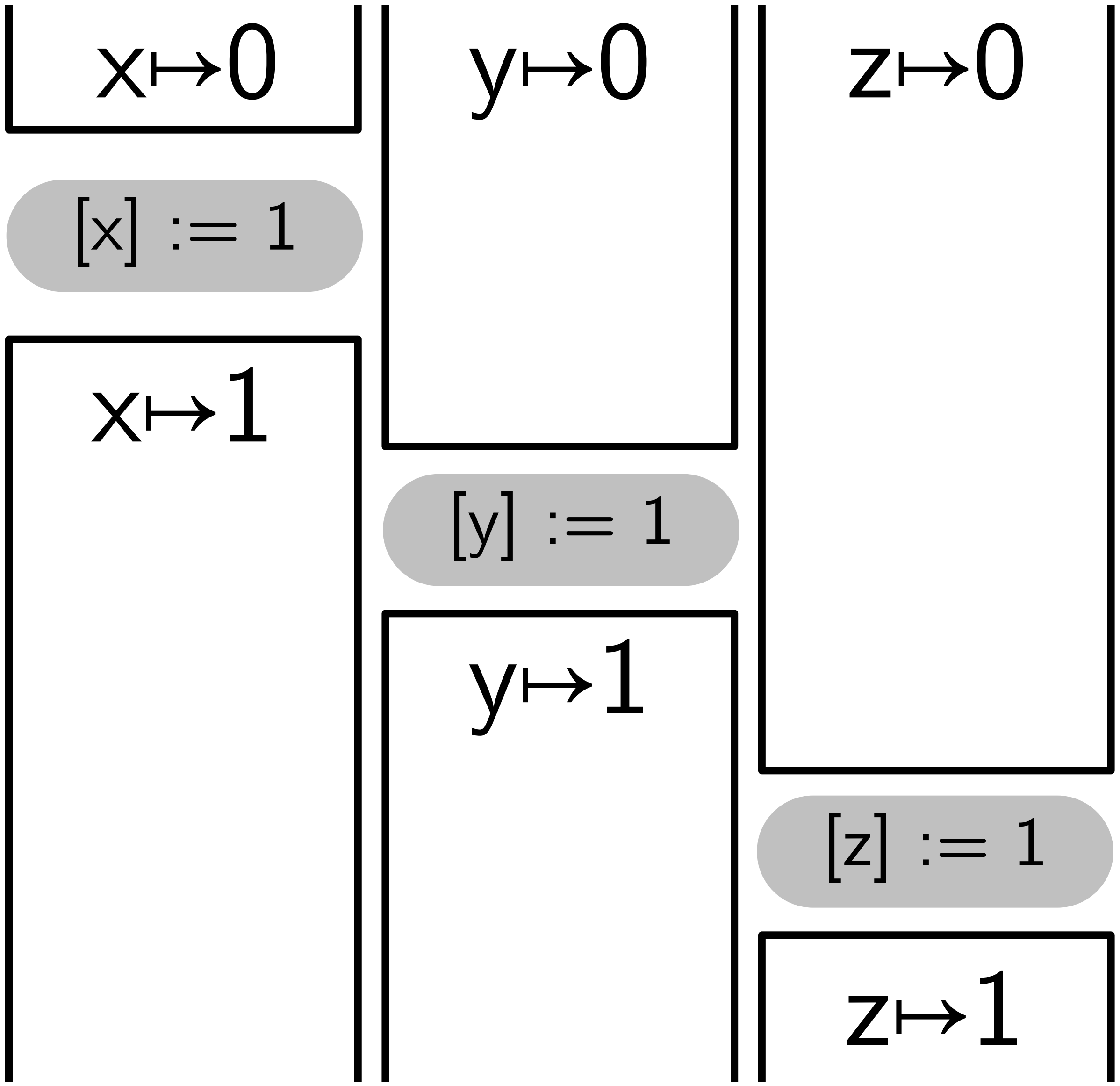
$[z] := 1$

$\{x \mapsto 1 * y \mapsto 1 * z \mapsto 1\}$



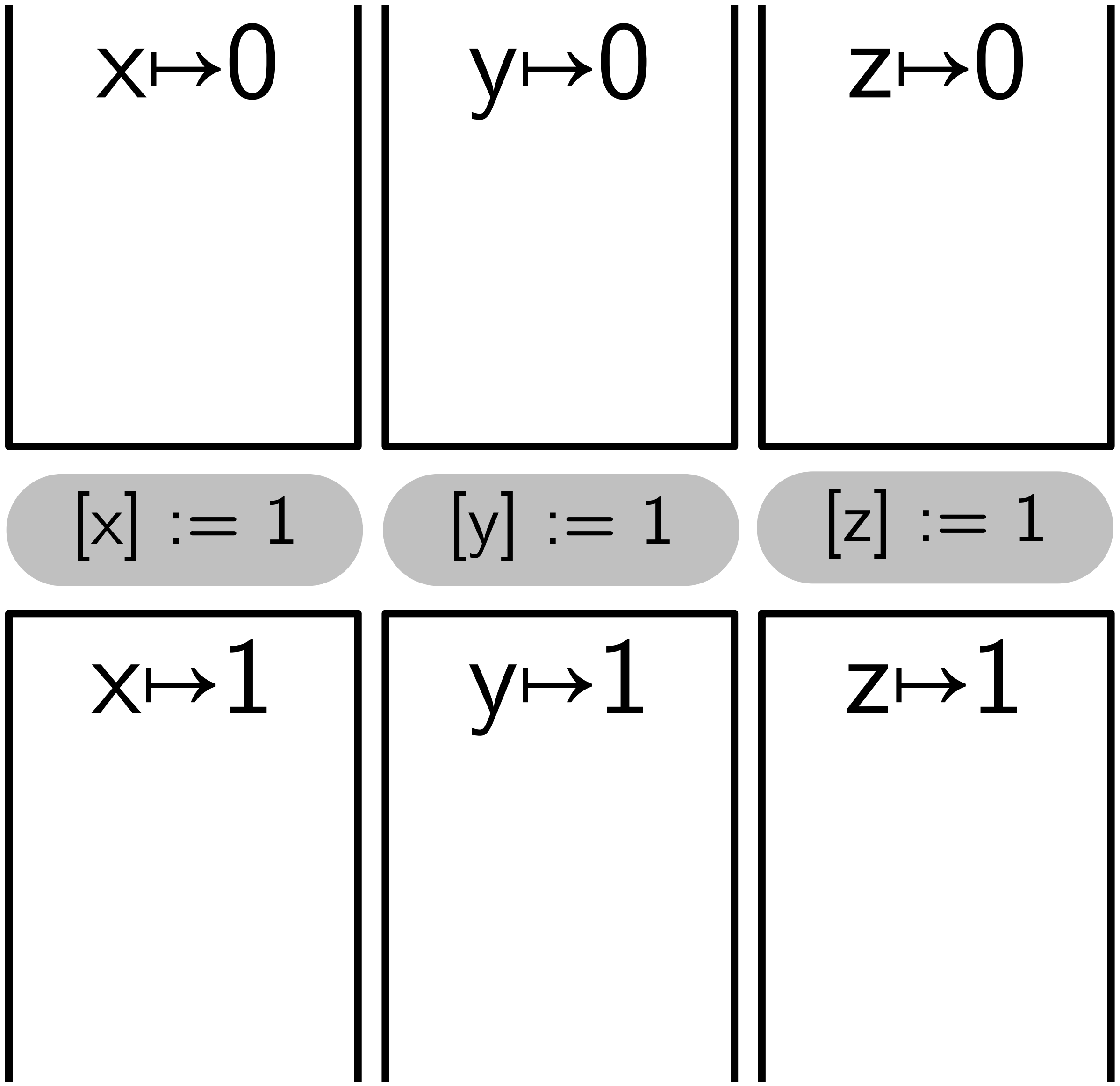
Proof outline vs. Ribbon proof

- ✓ Repetition is gone
- ✓ Effect of each instruction is clear
- ✓ Frame rule is applied implicitly
- ✓ Flexible



Proof outline vs. Ribbon proof

- ✓ Repetition is gone
- ✓ Effect of each instruction is clear
- ✓ Frame rule is applied implicitly
- ✓ Flexible



A problem

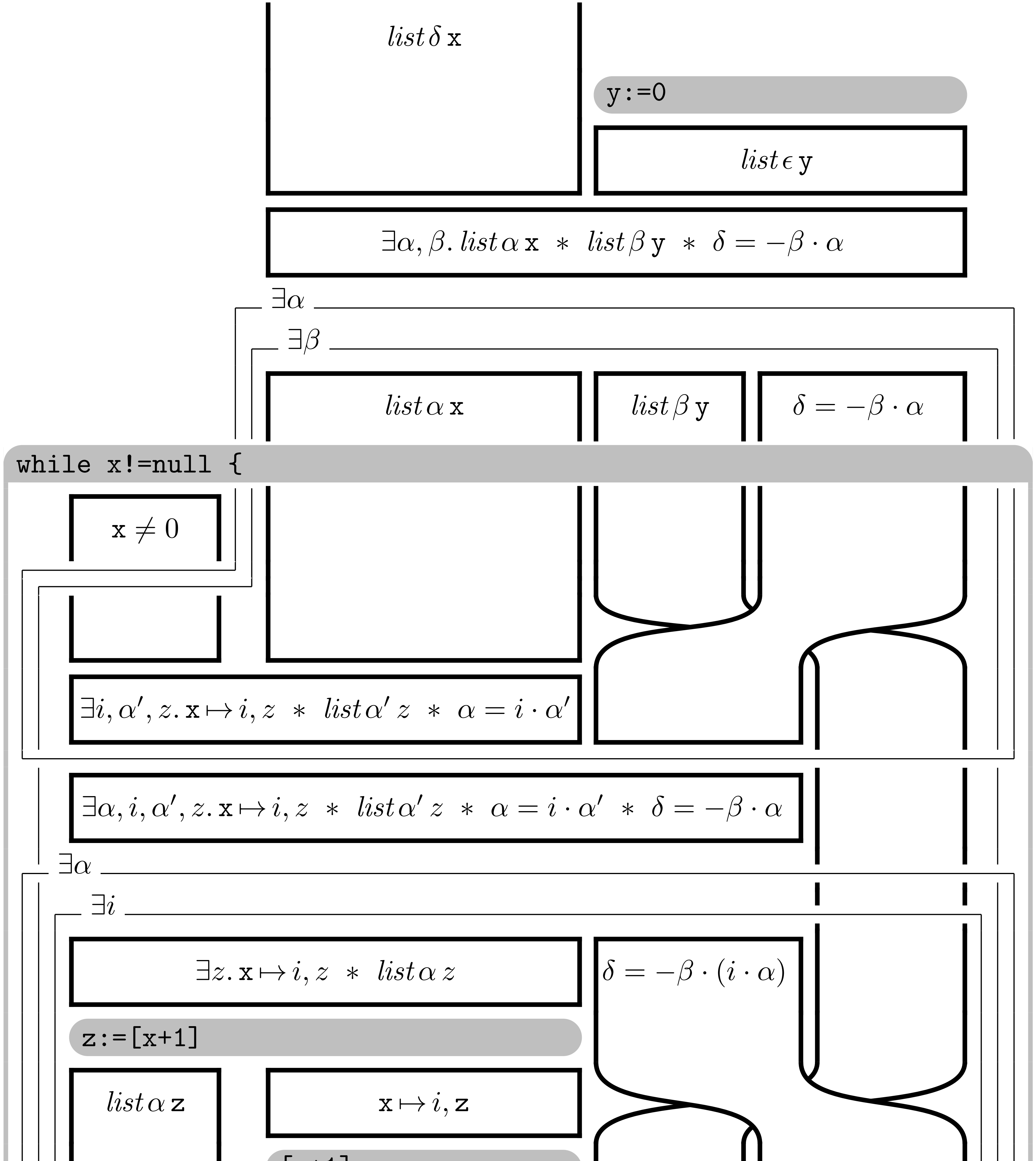
Our solution

A worked example

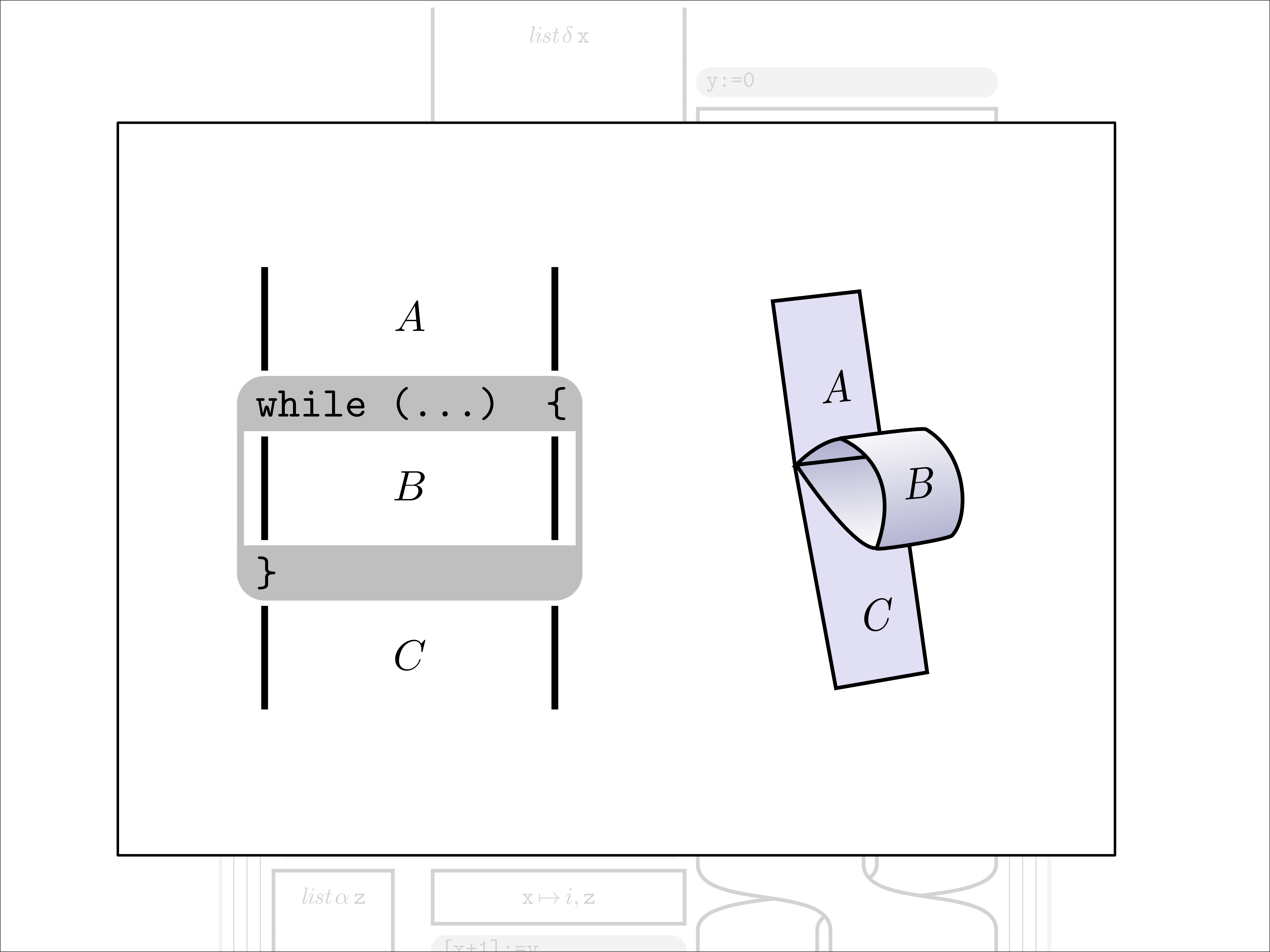
Where now?

List reversal

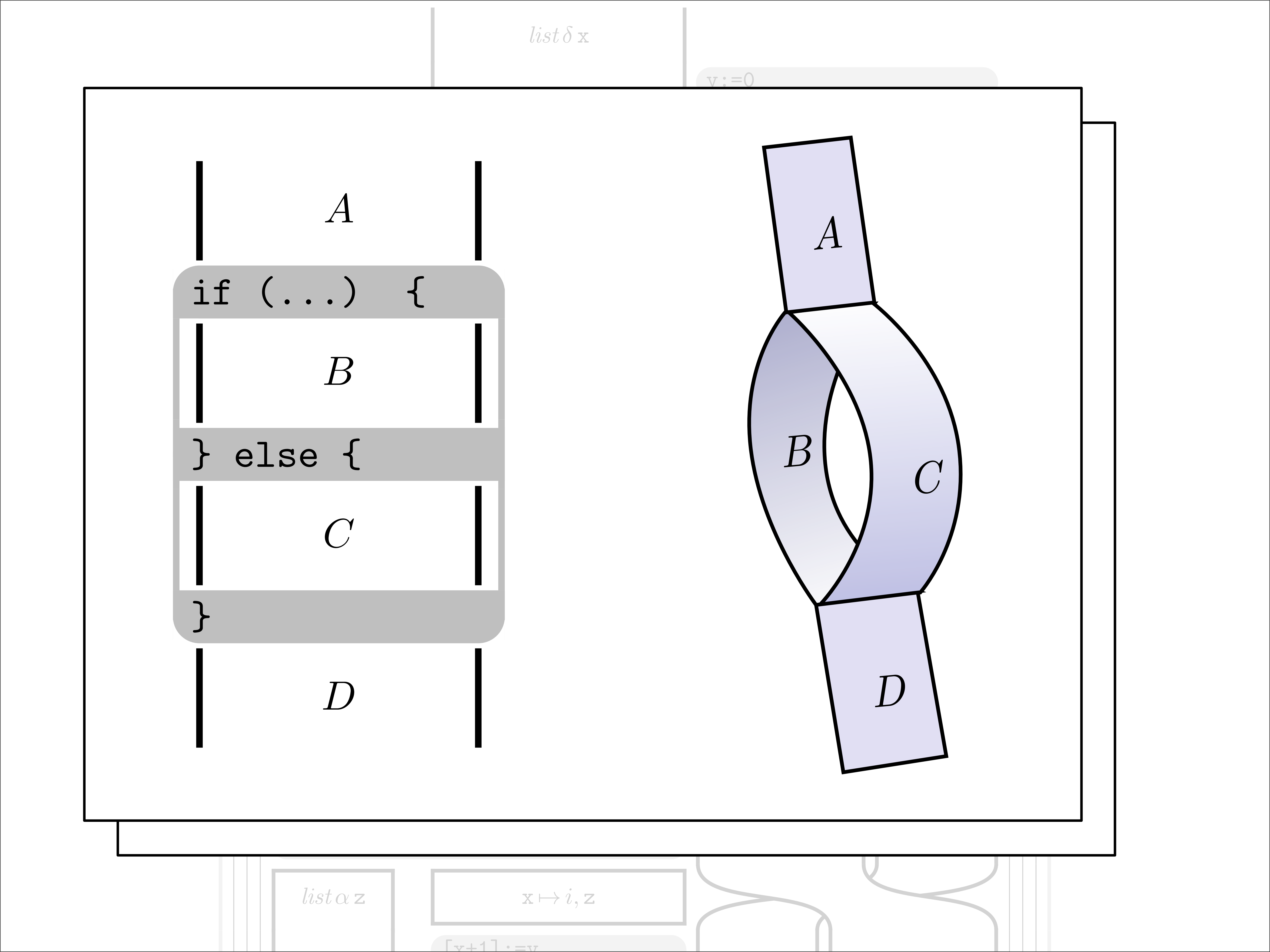
```
{list δ x}
y := 0;
while {∃α,β. list α x ∗ list β y ∗ δ ≐ -β·α} (x≠0) do {
    {∃i,α,β,Z. x ↦ i,Z ∗ list α Z ∗ list β y ∗ δ ≐ -β·(i·α)}
    z := [x+1];
    {∃i,α,β. x ↦ i,z ∗ list α z ∗ list β y ∗ δ ≐ -(i·β)·α}
    [x+1] := y;
    {∃i,α,β. x ↦ i,y ∗ list α z ∗ list β y ∗ δ ≐ -(i·β)·α}
    {∃α,β. list α z ∗ list β x ∗ δ ≐ -β·α}
    y := x; x := z;
    {∃α,β. list α x ∗ list β y ∗ δ ≐ -β·α}
}
{list -δ y}
```



$list\,\delta\,\mathtt{x}$
y:=0
$list\,\epsilon\,\mathtt{y}$
$\exists\alpha,\beta.\,list\,\alpha\,\mathtt{x} \ast list\,\beta\,\mathtt{y} \ast \delta = -\beta\cdot\alpha$
$\exists\alpha$
$\exists\beta$
$list\,\alpha\,\mathtt{x}$
$list\,\beta\,\mathtt{y}$
$\delta = -\beta\cdot\alpha$
while x!=null {
$\mathtt{x} \neq 0$
$\exists i,\alpha',z.\,\mathtt{x}\mapsto i,z \ast list\,\alpha'\,z \ast \alpha = i\cdot\alpha'$
$\exists\alpha,i,\alpha',z.\,\mathtt{x}\mapsto i,z \ast list\,\alpha'\,z \ast \alpha = i\cdot\alpha' \ast \delta = -\beta\cdot\alpha$
$\exists\alpha$
$\exists i$
$\exists z.\,\mathtt{x}\mapsto i,z \ast list\,\alpha\,z$
$\delta = -\beta\cdot(i\cdot\alpha)$
z:=[x+1]
$list\,\alpha\,\mathtt{z}$
$\mathtt{x}\mapsto i,\mathtt{z}$



A
while (...) {
B
}
C
A
B
C
list δ x
y:=0
list α z
x ↦ i, z


A
if (...) {
B
} else {
C
}
D
A
B
C
D

List reversal

$\{\text{list } \delta\ x\}$

y := 0;

while $\{\exists\alpha,\beta.\ \text{list } \alpha\ x * \text{list } \beta\ y * \delta \doteq {-\beta}\cdot\alpha\}$ (x≠0) do {

$\{\exists i,\alpha,\beta,Z.\ x \mapsto i,Z * \text{list } \alpha\ Z * \text{list } \beta\ y * \delta \doteq {-\beta}\cdot(i\cdot\alpha)\}$

z := [x+1];

$\{\exists i,\alpha,\beta.\ x \mapsto i,z * \text{list } \alpha\ z * \text{list } \beta\ y * \delta \doteq -(i\cdot\beta)\cdot\alpha\}$

[x+1] := y;

$\{\exists i,\alpha,\beta.\ x \mapsto i,y * \text{list } \alpha\ z * \text{list } \beta\ y * \delta \doteq -(i\cdot\beta)\cdot\alpha\}$

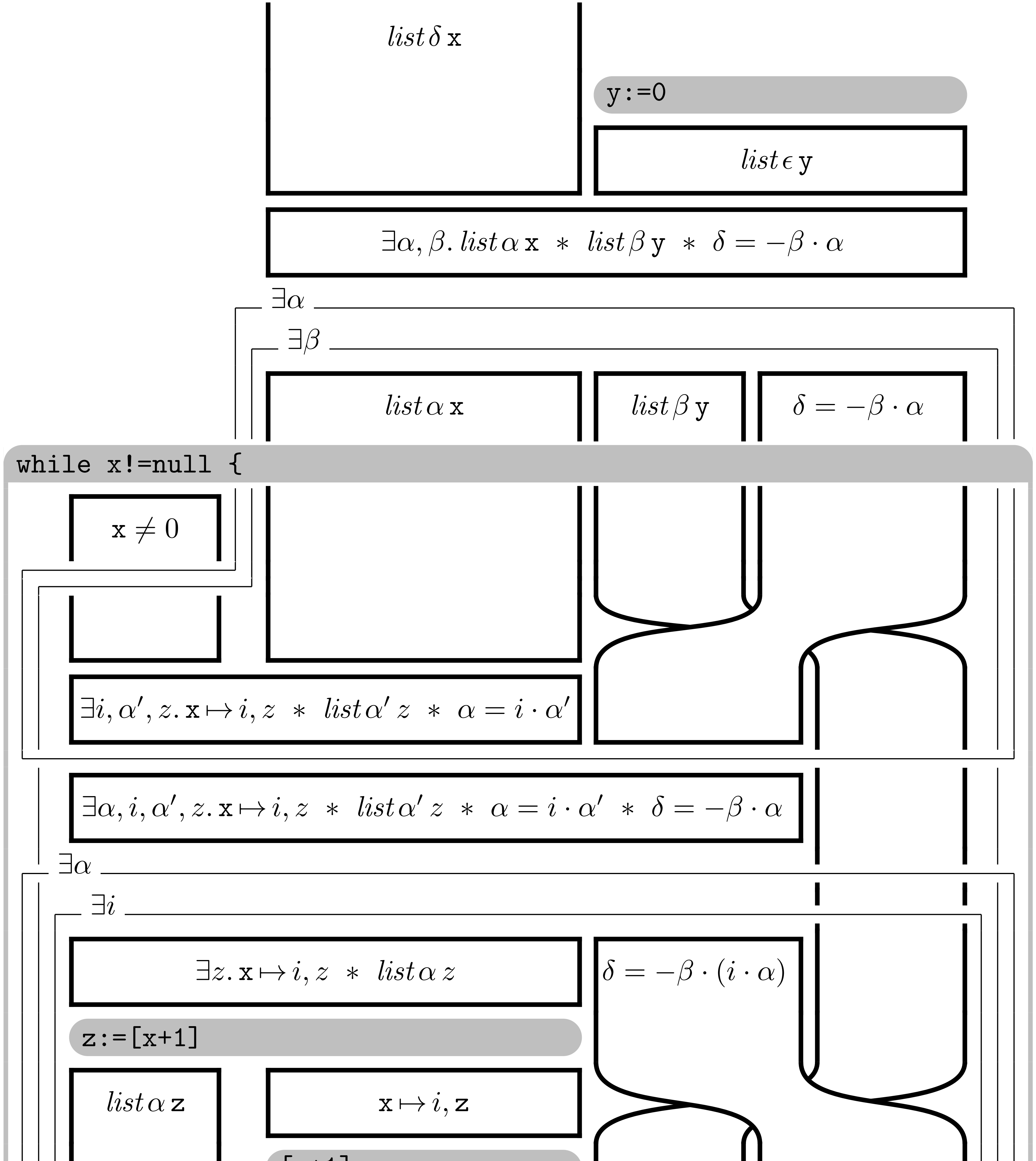
$\{\exists\alpha,\beta.\ \text{list } \alpha\ z * \text{list } \beta\ x * \delta \doteq {-\beta}\cdot\alpha\}$

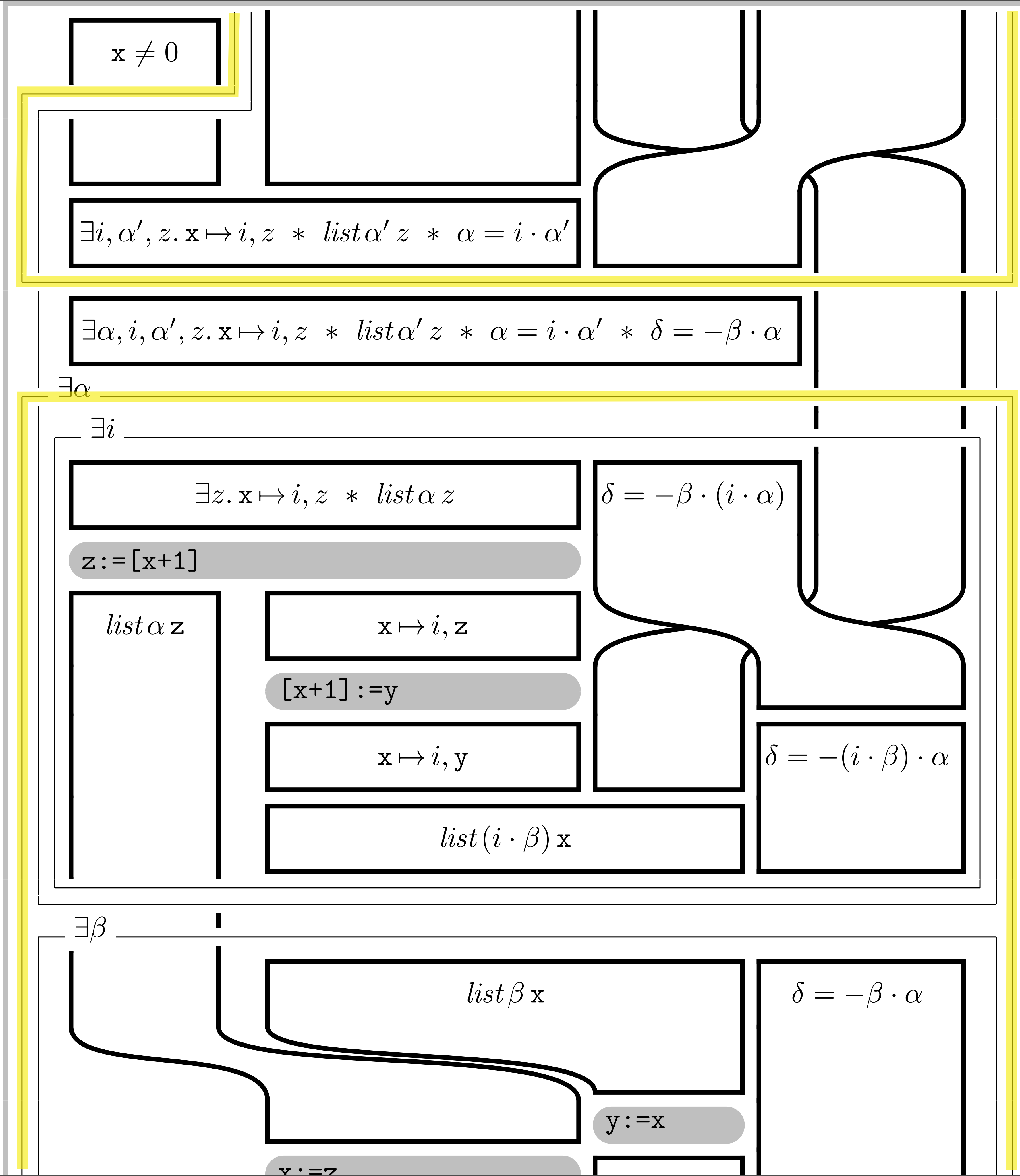
y := x; x := z;

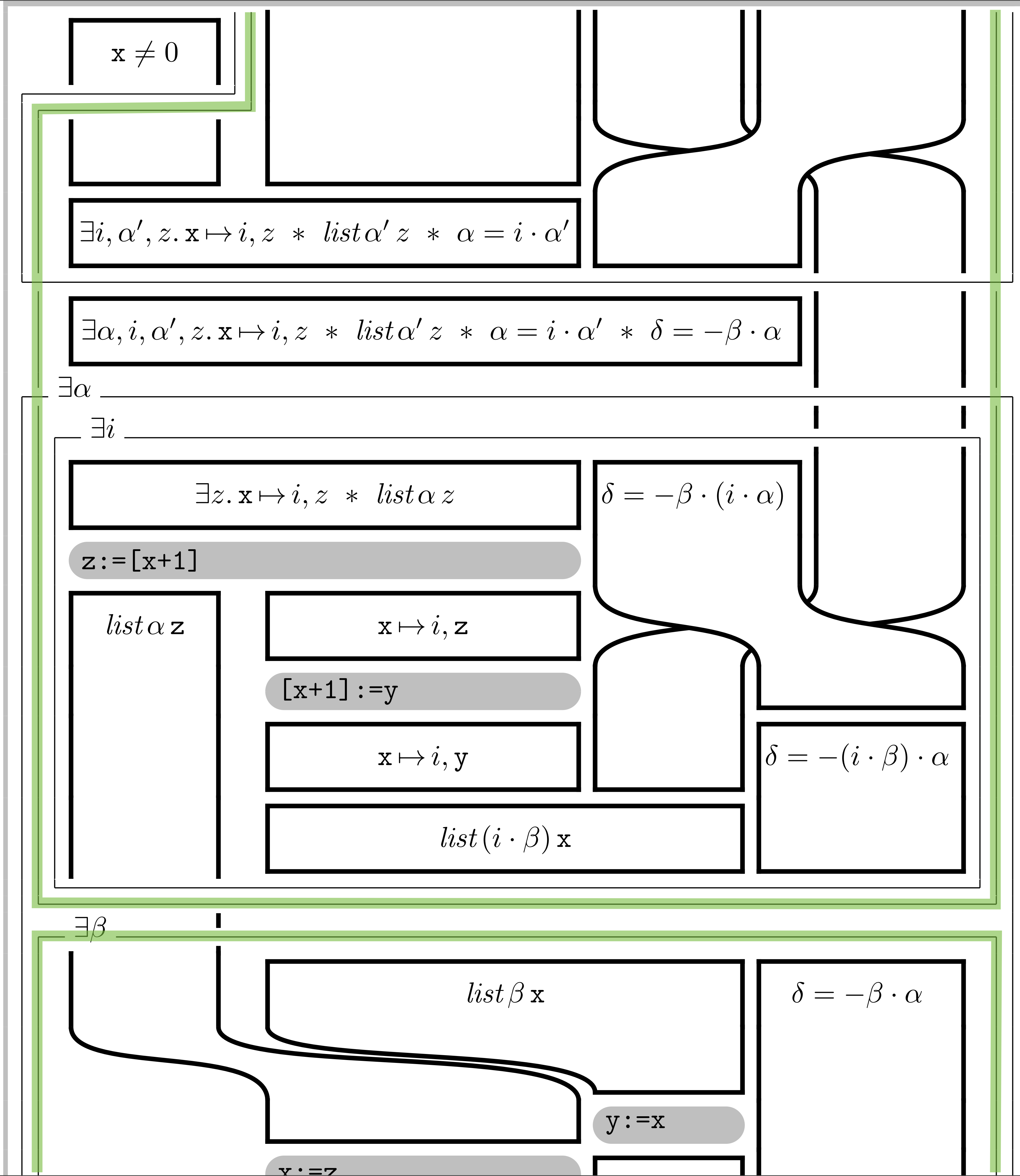
$\{\exists\alpha,\beta.\ \text{list } \alpha\ x * \text{list } \beta\ y * \delta \doteq {-\beta}\cdot\alpha\}$

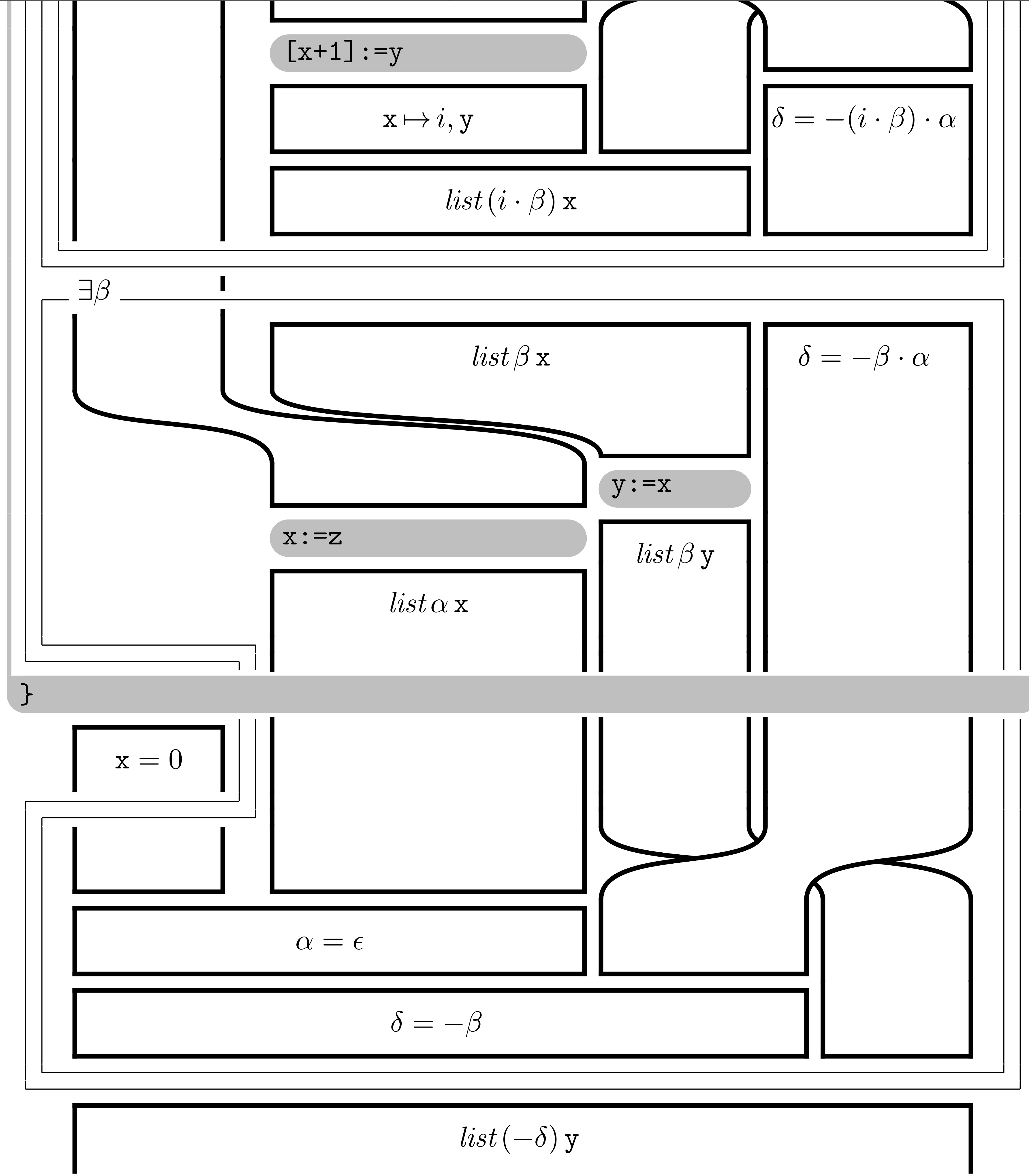
}

$\{\text{list } {-\delta}\ y\}$


list δ x
y:=0
list ε y
∃α, β. list α x ∗ list β y ∗ δ = −β · α
∃α
∃β
list α x
list β y
δ = −β · α
while x!=null {
x ≠ 0
∃i, α′, z. x ↦ i, z ∗ list α′ z ∗ α = i · α′
∃α, i, α′, z. x ↦ i, z ∗ list α′ z ∗ α = i · α′ ∗ δ = −β · α
∃α
∃i
∃z. x ↦ i, z ∗ list α z
δ = −β · (i · α)
z:=[x+1]
list α z
x ↦ i, z


x ≠ 0
∃i, α′, z. x ↦ i, z ∗ list α′ z ∗ α = i · α′
∃α, i, α′, z. x ↦ i, z ∗ list α′ z ∗ α = i · α′ ∗ δ = −β · α
∃α
∃i
∃z. x ↦ i, z ∗ list α z
δ = −β · (i · α)
z:=[x+1]
list α z
x ↦ i, z
[x+1]:=y
x ↦ i, y
δ = −(i · β) · α
list (i · β) x
∃β
list β x
δ = −β · α
y:=x


x ≠ 0
∃i, α′, z. x ↦ i, z ∗ list α′ z ∗ α = i · α′
∃α, i, α′, z. x ↦ i, z ∗ list α′ z ∗ α = i · α′ ∗ δ = −β · α
∃α
∃i
∃z. x ↦ i, z ∗ list α z
δ = −β · (i · α)
z:=[x+1]
list α z
x ↦ i, z
[x+1]:=y
x ↦ i, y
δ = −(i · β) · α
list (i · β) x
∃β
list β x
δ = −β · α
y:=x



[x+1]:=y
$\mathtt{x} \mapsto i, \mathtt{y}$
$\delta = -(i \cdot \beta) \cdot \alpha$
$list\,(i \cdot \beta)\,\mathtt{x}$
$\exists\beta$
$list\,\beta\,\mathtt{x}$
$\delta = -\beta \cdot \alpha$
y:=x
x:=z
$list\,\beta\,\mathtt{y}$
$list\,\alpha\,\mathtt{x}$
}
$\mathtt{x} = 0$
$\alpha = \epsilon$
$\delta = -\beta$
$list\,(-\delta)\,\mathtt{y}$

A problem

Our solution

A worked example

Where now?

Proof of `unlink_first_small_chunk`

$smallbin_{\lfloor \mathtt{S}/8 \rfloor}(U \uplus \{\mathtt{P}+2\mathtt{w} \mapsto \mathtt{S}-1\mathtt{w}\})$

Defn of *smallbin*

$\exists x$

$\exists i$

$\mathtt{S} = 8i \;*\; 0 \leq i < 32$
$*\; x = \mathtt{smallbin} + 2i\mathtt{w}$

$bin(|i|, x, U \uplus \{\mathtt{P}+2\mathtt{w} \mapsto \mathtt{S}-1\mathtt{w}\})$

$\mathtt{smallmap}_{[i]} = 1$

Defn of *bin*

$\exists y$

$x \overset{\mathsf{fd}}{\longmapsto} y$

$y \overset{\mathsf{bk}}{\longmapsto} x$

$(bnode\,|i|)^*(y, x, U \uplus \{\mathtt{P}+2\mathtt{w} \mapsto \mathtt{S}-1\mathtt{w}\})$

Split *bnode* list into three.

$\exists U_1, U_2$

$U = U_1 \uplus U_2$

$(bnode\,|i|)^*(y, \mathtt{P}, U_1)$

$\exists F.\, \mathtt{P} \overset{\mathsf{fd}}{\longmapsto} F$
$*\; F \overset{\mathsf{bk}}{\longmapsto} \mathtt{P}$
$*\; \frac{1}{2}(\mathtt{P} \overset{\mathsf{size}}{\longmapsto} \mathtt{S})$
$*\; (bnode\,|i|)^*(F, x, U_2)$

Unroll RTC one step.

$(y = \mathtt{P} \;*\; U_1 = \{\}) \vee (\exists B.$
$(bnode\,|i|)^*(y, B, U_1 \uplus\!\!- \{B+2\mathtt{w} \mapsto \_\})$
$*\; B \overset{\mathsf{fd}}{\longmapsto} \mathtt{P} \;*\; \mathtt{P} \overset{\mathsf{bk}}{\longmapsto} B)$

`mchunkptr F = P->fd;`

$\mathtt{F} \overset{\mathsf{bk}}{\longmapsto} \mathtt{P}$

$(bnode\,|i|)^*(\mathtt{F}, x, U_2)$

$\mathtt{P} \overset{\mathsf{fd}}{\longmapsto} \mathtt{F}$

$\frac{1}{2}(\mathtt{P} \overset{\mathsf{size}}{\longmapsto} \mathtt{S})$

Extend scope of $\exists B$. Choose $B = x$ in first disjunct.

$\exists B$

$(y = \mathtt{P} \;*\; U_1 = \{\} * B = x) \vee$
$((bnode\,|i|)^*(y, B, U_1 \uplus\!\!- \{B+2\mathtt{w} \mapsto \_\})$
$*\; B \overset{\mathsf{fd}}{\longmapsto} \mathtt{P} \;*\; \mathtt{P} \overset{\mathsf{bk}}{\longmapsto} B)$

Distribute $x \overset{\mathsf{fd}}{\longmapsto} y \;*\; y \overset{\mathsf{bk}}{\longmapsto} x$ into disjunction.

$(B \overset{\mathsf{fd}}{\longmapsto} \mathtt{P} \;*\; \mathtt{P} \overset{\mathsf{bk}}{\longmapsto} B \;*\; y = \mathtt{P} \;*\; U_1 = \{\} \;*\; B = x) \vee$
$((bnode\,|i|)^*(y, B, U_1 \uplus\!\!- \{B+2\mathtt{w} \mapsto \_\})$
$*\; B \overset{\mathsf{fd}}{\longmapsto} \mathtt{P} \;*\; \mathtt{P} \overset{\mathsf{bk}}{\longmapsto} B \;*\; x \overset{\mathsf{fd}}{\longmapsto} y \;*\; y \overset{\mathsf{bk}}{\longmapsto} x)$

Distribute $B \overset{\mathsf{fd}}{\longmapsto} \mathtt{P} \;*\; \mathtt{P} \overset{\mathsf{bk}}{\longmapsto} B$ out of disjunction. Forget $y = \mathtt{P}$ from first disjunct.

$(U_1 = \{\} \;*\; B = x) \vee$
$((bnode\,|i|)^*(y, B, U_1 \uplus\!\!- \{B+2\mathtt{w} \mapsto \_\})$
$*\; x \overset{\mathsf{fd}}{\longmapsto} y \;*\; y \overset{\mathsf{bk}}{\longmapsto} x)$

$B \overset{\mathsf{fd}}{\longmapsto} \mathtt{P}$

$\mathtt{P} \overset{\mathsf{bk}}{\longmapsto} B$

`mchunkptr B = P->bk;`

$(U_1 = \{\} \;*\; \mathtt{B} = x) \vee$
$((bnode\,|i|)^*(y, \mathtt{B}, U_1 \uplus\!\!- \{\mathtt{B}+2\mathtt{w} \mapsto \_\})$
$*\; x \overset{\mathsf{fd}}{\longmapsto} y \;*\; y \overset{\mathsf{bk}}{\longmapsto} x)$

$\mathtt{B} \overset{\mathsf{fd}}{\longmapsto} \mathtt{P}$

$\mathtt{P} \overset{\mathsf{bk}}{\longmapsto} \mathtt{B}$

$(U_1 = \{\} \;*\; \mathtt{B} = x) \vee \exists y.$
$((bnode\,|i|)^*(y, \mathtt{B}, U_1 \uplus\!\!- \{\mathtt{B}+2\mathtt{w} \mapsto \_\})$
$*\; x \overset{\mathsf{fd}}{\longmapsto} y \;*\; y \overset{\mathsf{bk}}{\longmapsto} x)$

$y = \mathtt{P}$ from first disjunct.

$(U_1 = \{\} \ * \ B = x) \vee$
$((bnode\,|i|)^*(y, B, U_1 \uplus\!\!- \{B + 2\mathsf{w} \mapsto \_\})$
$* \ x \overset{\mathsf{fd}}{\longmapsto} y \ * \ y \overset{\mathsf{bk}}{\longmapsto} x)$

$B \overset{\mathsf{fd}}{\longmapsto} \mathtt{P}$

$\mathtt{P} \overset{\mathsf{bk}}{\longmapsto} B$

```
mchunkptr B = P->bk;
```

$(U_1 = \{\} \ * \ \mathtt{B} = x) \vee$
$((bnode\,|i|)^*(y, \mathtt{B}, U_1 \uplus\!\!- \{\mathtt{B} + 2\mathsf{w} \mapsto \_\})$
$* \ x \overset{\mathsf{fd}}{\longmapsto} y \ * \ y \overset{\mathsf{bk}}{\longmapsto} x)$

$\mathtt{B} \overset{\mathsf{fd}}{\longmapsto} \mathtt{P}$

$\mathtt{P} \overset{\mathsf{bk}}{\longmapsto} \mathtt{B}$

$(U_1 = \{\} \ * \ \mathtt{B} = x) \vee \exists y.$
$((bnode\,|i|)^*(y, \mathtt{B}, U_1 \uplus\!\!- \{\mathtt{B} + 2\mathsf{w} \mapsto \_\})$
$* \ x \overset{\mathsf{fd}}{\longmapsto} y \ * \ y \overset{\mathsf{bk}}{\longmapsto} x)$

```
bindex_t I = small_index(S);
```

$\mathtt{S} = 8\mathtt{I} \ * \ 0 \le \mathtt{I} < 32$
$* \ x = \mathtt{smallbin} + 2\mathtt{I}\mathtt{w}$

$(U_1 = \{\} \ * \ \mathtt{B} = x) \vee \exists y.$
$((bnode\,|\mathtt{I}|)^*(y, \mathtt{B}, U_1 \uplus\!\!- \{\mathtt{B} + 2\mathsf{w} \mapsto \_\})$
$* \ x \overset{\mathsf{fd}}{\longmapsto} y \ * \ y \overset{\mathsf{bk}}{\longmapsto} x)$

$(bnode\,|\mathtt{I}|)^*(\mathtt{F}, x, U_2)$

$\mathtt{smallmap}_{[\mathtt{I}]} = 1$

$\mathtt{P} \overset{\mathsf{bk}}{\longmapsto} \_$

$\mathtt{P} \overset{\mathsf{fd}}{\longmapsto} \_$

```
if (F==B) {
```

$\mathtt{B} = \mathtt{F}$

Ribbons 1 and 4 contradict second disjunct of ribbon 2; hence $\mathtt{B} = \mathtt{F} = x$ and $U_1 = \{\}$. Since $\mathtt{F} = x$, ribbon 5 implies $U_2 = \{\}$. (See Lemma 1, above.)

$x \overset{\mathsf{fd}}{\longmapsto} \_$

$x \overset{\mathsf{bk}}{\longmapsto} \_$

$U_1 = \{\} \ * \ U_2 = \{\}$

$U = \{\}$

```
clear_smallmap(M,I);
```

$U = \{\}$

$\mathtt{smallmap}_{[\mathtt{I}]} = (U \neq \{\})$

Defn of *bin*.

$bin(|\mathtt{I}|, x, U)$

```
} else {
```

$\mathtt{B} \neq \mathtt{F}$

$(U_1 = \{\} \ * \ \mathtt{B} = x) \vee \exists y.$
$((bnode\,|\mathtt{I}|)^*(y, \mathtt{B}, U_1 \uplus\!\!- \{\mathtt{B} + 2\mathsf{w} \mapsto \_\})$
$* \ x \overset{\mathsf{fd}}{\longmapsto} y \ * \ y \overset{\mathsf{bk}}{\longmapsto} x)$

$\mathtt{B} \overset{\mathsf{fd}}{\longmapsto} \mathtt{P}$

$\mathtt{F} \overset{\mathsf{bk}}{\longmapsto} \mathtt{P}$

$(bnode\,|\mathtt{I}|)^*(\mathtt{F}, x, U_2)$

$U = U_1 \uplus U_2$

$\mathtt{smallmap}_{[\mathtt{I}]} = 1$

From ribbons 1, 2 and 3, deduce U_1 and U_2 can't both be empty. Extend scope of $\exists y$ in ribbon 2, choosing $y = \mathtt{F}$ in first disjunct.

$\exists y. \, (y = \mathtt{F} \ * \ U_1 = \{\} \ * \ \mathtt{B} = x) \vee$
$((bnode\,|\mathtt{I}|)^*(y, \mathtt{B}, U_1 \uplus\!\!- \{\mathtt{B} + 2\mathsf{w} \mapsto \_\})$
$* \ x \overset{\mathsf{fd}}{\longmapsto} y \ * \ y \overset{\mathsf{bk}}{\longmapsto} x)$

```
B->fd = F
```

```
F->bk = B
```

$\mathtt{B} \overset{\mathsf{fd}}{\longmapsto} \mathtt{F}$

$\mathtt{F} \overset{\mathsf{bk}}{\longmapsto} \mathtt{B}$

$(bnode\,|\mathtt{I}|)^*(\mathtt{F}, x, U_2)$

$U = U_1 \uplus U_2$

$\mathtt{smallmap}_{[\mathtt{I}]} = (U \neq \{\})$

Distribute $\mathtt{B} \overset{\mathsf{fd}}{\longmapsto} \mathtt{F} \ * \ \mathtt{F} \overset{\mathsf{bk}}{\longmapsto} \mathtt{B}$ into disjunction.

$\exists y$

$x \overset{fd}{\longmapsto} \_$ | $x \overset{bk}{\longmapsto} \_$ | $U_1 = \{\} \ * \ U_2 = \{\}$

$U = \{\}$

`clear_smallmap(M,I);`

$U = \{\}$ | $\texttt{smallmap}_{[\texttt{I}]} = (U \neq \{\})$

Defn of *bin*.

$bin(|\texttt{I}|, x, U)$

`} else {`

$\texttt{B} \neq \texttt{F}$ | $(U_1 = \{\} \ * \ \texttt{B} = x) \vee \exists y. ((bnode\,|\texttt{I}|)^*(y, \texttt{B}, U_1 \uplus\!\!- \{\texttt{B} + 2\texttt{w} \mapsto \_\}) \ * \ x \overset{fd}{\longmapsto} y \ * \ y \overset{bk}{\longmapsto} x)$ | $\texttt{B} \overset{fd}{\longmapsto} \texttt{P}$ | $\texttt{F} \overset{bk}{\longmapsto} \texttt{P}$ | $(bnode\,|\texttt{I}|)^*(\texttt{F}, x, U_2)$ | $U = U_1 \uplus U_2$ | $\texttt{smallmap}_{[\texttt{I}]} = 1$

From ribbons 1, 2 and 3, deduce U_1 and U_2 can't both be empty. Extend scope of $\exists y$ in ribbon 2, choosing $y = \texttt{F}$ in first disjunct.

$\exists y. (y = \texttt{F} \ * \ U_1 = \{\} \ * \ \texttt{B} = x) \vee ((bnode\,|\texttt{I}|)^*(y, \texttt{B}, U_1 \uplus\!\!- \{\texttt{B} + 2\texttt{w} \mapsto \_\}) \ * \ x \overset{fd}{\longmapsto} y \ * \ y \overset{bk}{\longmapsto} x)$ | $(bnode\,|\texttt{I}|)^*(\texttt{F}, x, U_2)$ | $U = U_1 \uplus U_2$ | $\texttt{smallmap}_{[\texttt{I}]} = (U \neq \{\})$

`B->fd = F` | `F->bk = B`

$\texttt{B} \overset{fd}{\longmapsto} \texttt{F}$ | $\texttt{F} \overset{bk}{\longmapsto} \texttt{B}$

Distribute $\texttt{B} \overset{fd}{\longmapsto} \texttt{F} \ * \ \texttt{F} \overset{bk}{\longmapsto} \texttt{B}$ into disjunction.

$\exists y$

$(y = \texttt{F} \ * \ U_1 = \{\} \ * \ \texttt{B} = x \ * \ \texttt{B} \overset{fd}{\longmapsto} \texttt{F} \ * \ \texttt{F} \overset{bk}{\longmapsto} \texttt{B}) \vee ((bnode\,|\texttt{I}|)^*(y, \texttt{B}, U_1 \uplus\!\!- \{\texttt{B} + 2\texttt{w} \mapsto \_\}) \ * \ x \overset{fd}{\longmapsto} y \ * \ y \overset{bk}{\longmapsto} x \ * \ \texttt{B} \overset{fd}{\longmapsto} \texttt{F} \ * \ \texttt{F} \overset{bk}{\longmapsto} \texttt{B})$

Distribute $x \overset{fd}{\longmapsto} y \ * \ y \overset{bk}{\longmapsto} x$ out of disjunction.

$x \overset{fd}{\longmapsto} y$ | $y \overset{bk}{\longmapsto} x$ | $(y = \texttt{F} \ * \ U_1 = \{\}) \vee ((bnode\,|\texttt{I}|)^*(y, \texttt{B}, U_1 \uplus\!\!- \{\texttt{B} + 2\texttt{w} \mapsto \_\}) \ * \ \texttt{B} \overset{fd}{\longmapsto} \texttt{F} \ * \ \texttt{F} \overset{bk}{\longmapsto} \texttt{B})$

Roll RTC.

$(bnode\,|\texttt{I}|)^*(y, \texttt{F}, U_1)$

Merge two *bnode* lists.

$(bnode\,|\texttt{I}|)^*(y, x, U)$

Defn of *bin*.

$bin(|\texttt{I}|, x, U)$

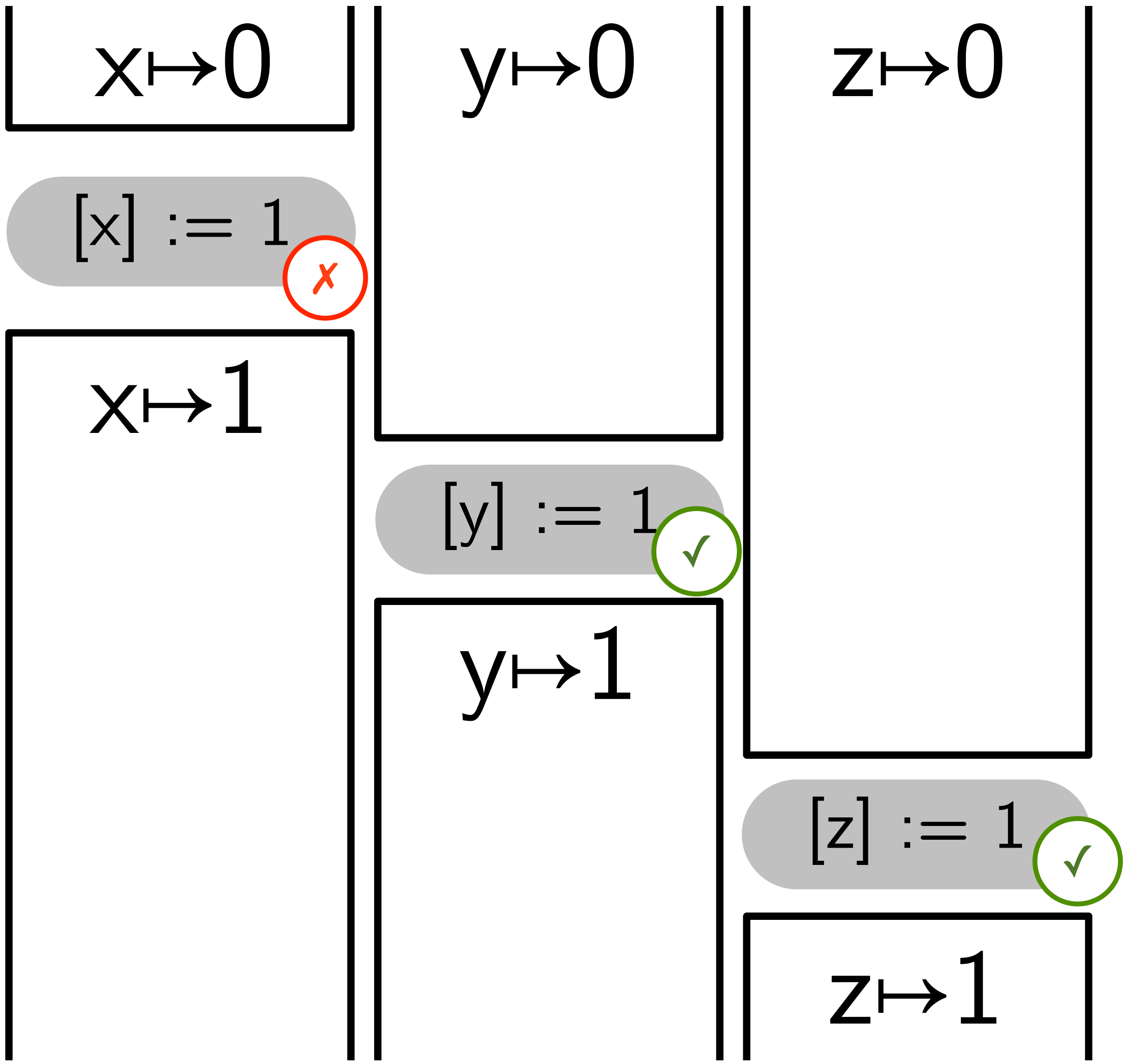
`}`

Defn of *smallbin*.

$smallbin_{\lfloor \texttt{S}/8 \rfloor}(U)$

$smallbin_{\lfloor \texttt{S}/8 \rfloor}(U)$ | $\texttt{P} \overset{bk}{\longmapsto} \_$ | $\texttt{P} \overset{fd}{\longmapsto} \_$ | $\frac{1}{2}(\texttt{P} \overset{size}{\longmapsto} \texttt{S})$

Tool support



Conclusion

- **Proof outlines** are
  - ✗ big
  - ✗ clumsy
  - ✗ hard-to-read
- **Ribbon proofs** are
  - ✓ scalable
  - ✓ flexible
  - ✓ readable

- In our full paper:
  - formalisation in Isabelle
  - soundness and completeness results
  - description of prototype tool
  - ribbon proof of Unix V7 malloc

References

- Jules Bean. *Ribbon proofs.* In MFPS 2003.
- John C. Reynolds. *Separation logic: a logic for shared mutable data structures.* In LICS 2002.
- Peter O'Hearn. *Resources, concurrency and local reasoning.* In CONCUR 2004.