Si definisca la seguente grammatica per le liste di naturali:

$L ::= [\,] \mid n{:}L$ con $n \in \mathbb{N}$

L'insieme di regole SOS è stato definito con in mente il seguente pseudocodice per l'algoritmo Bubble Sort:

```
proc bubblesort(A : array of int) begin
    repeat
        swapped := false;
        for i := 1 to n-1 do
            if A[i-1] > A[i] then
                swap(A[i-1], A[i]);
                swapped := true;
            end if
        endfor
    until not swapped
end proc
```

Le regole di derivazione per bubblesort sono definite sulla base della procedura bubble che avrà il compito di simulare il ciclo for dello pseudocodice sopra.
Le regole per bubble restituiscono una coppia contenente una lista e un booleano e indicano se è stato effettuato uno swap per quella chiamata di bubble.

$$\frac{}{bubble([\,]) \Rightarrow ([\,], false)} \qquad \frac{}{bubble(n_1{:}[\,]) \Rightarrow (n_1{:}[\,], false)}$$

$$\frac{n_1 \le n_2 \quad bubble(n_2{:}\ell) \Rightarrow (\ell_1, b)}{bubble(n_1{:}n_2{:}\ell) \Rightarrow (n_1{:}\ell_1, b)} \quad b \in \{true, false\}$$

$$\frac{n_1 > n_2 \quad bubble(n_1{:}\ell) \Rightarrow (\ell_1, b)}{bubble(n_1{:}n_2{:}\ell) \Rightarrow (n_2{:}\ell_1, true)} \quad b \in \{true, false\}$$

$$\frac{bubble(\ell) \Rightarrow (\ell', tt) \quad bubblesort(\ell') \Rightarrow \ell''}{bubblesort(\ell) \Rightarrow \ell''}$$

$$\frac{bubble(\ell) \Rightarrow (\ell', ff)}{bubblesort(\ell) \Rightarrow \ell'}$$

TEORIA DEI DOMINI

Costruire il grafo del dominio: $(T_\bot \times O) \to O$

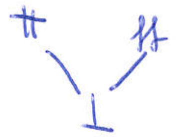
Definiamo gli information system relativi ai domini $T_\bot$ e O per poi andare a costruire i successivi domini.

- $T_\bot$ è il CPO di tre elementi: $\bot$, tt, ff con le seguenti relazioni d'ordine $\bot \sqsubseteq tt$ e $\bot \sqsubseteq ff$.

  L'information system corrispondente è la tripla

  $T = (\{tt, ff\}, \{\{tt\}, \{ff\}, \emptyset\}, \{(\{tt\}, tt), (\{ff\}, ff)\})$

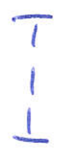
  Da cui risulta: $|T| = T_\bot$ il cui grafo è:



- O è il CPO di due elementi: $\bot \sqsubseteq \top$ e l'information system che lo genera è solitamente indicato con 1 o $0_\bot$ (zero liftato):

  $1 = (\{*\}, \{\{*\}, \emptyset\}, \{(\{*\}, *)\})$ in cui $\bot \equiv \emptyset$ e $\top \equiv \{*\}$

  Il grafo è:



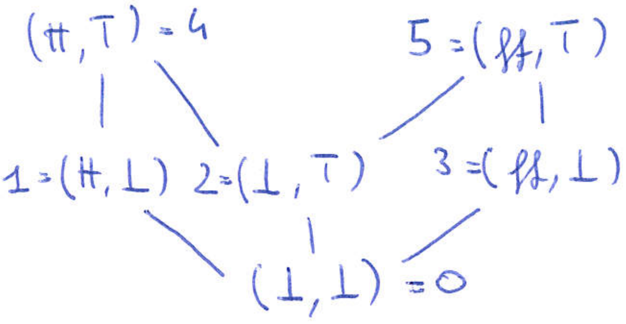
- $T_\bot \times O$: Si calcola l'IS per $T_\bot \times O$ a partire dagli IS di $T_\bot$ e O.

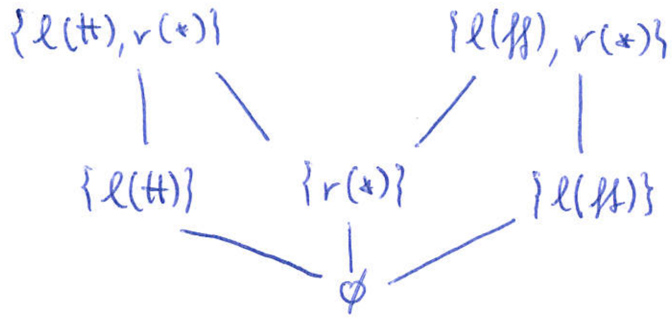
$$IS_{T_\bot \times O} = (\{l(tt), l(ff), r(*)\},$$
$$\{\{l(tt)\}, \{l(ff)\}, \{r(*)\}, \{l(tt), r(*)\}, \{l(ff), r(*)\}, \emptyset\},$$
$$\{(\{l(tt)\}, l(tt)), (\{l(ff)\}, l(ff)), (\{r(*)\}, r(*)), (\{l(tt), r(*)\}, l(tt)),$$
$$(\{l(tt), r(*)\}, r(*)), (\{l(ff), r(*)\}, l(ff)), (\{l(ff), r(*)\}, r(*))\})$$

Gli insiemi di token di $IS_{T_\bot \times O}$ chiusi per le relazioni di entailment sono:

$\emptyset, \{l(ff)\}, \{l(tt)\}, \{r(*)\}, \{l(tt), r(*)\}$ e $\{l(ff), r(*)\}$

Di seguito si presentano i grafi per $T_\bot \times O$ e $IS_{T_\bot \times O}$.





Infine andiamo a definire il CPO delle funzioni continue
$T_\bot \times O \to O$

Nella definizione delle funzioni verrà omesso l'output per alcuni input, se tale output si può inferire dalle proprietà di monotonia delle funzioni del nostro CPO. In particolare:

- Se una data funzione assume il valore $\top$ in un punto x, allora assumerà lo stesso valore per tutti gli y t.c. $x \sqsubseteq y$:
  Se $f(x) = \top$ allora $\forall y.\ x \sqsubseteq y \Rightarrow f(y) = \top$.
- Se invece una data funzione assume valore $\bot$ in un punto y, allora assumerà lo stesso valore per tutti gli $x \sqsubseteq y$:
  Se $f(y) = \bot$ allora $\forall x.\ x \sqsubseteq y \Rightarrow f(x) = \bot$.

Le coppie di input del dominio $T_\bot \times O$ sono state numerate come sopra, da 0 a 5. Ogni funzione è indicata come un insieme di coppie (input, output).

Il grafo del dominio è riportato nella pagina che segue.

L'IS per $T_\bot \times O \to O$ è così definito: $IS_{T_\bot \times O \to O} = (A, Con, \vdash)$

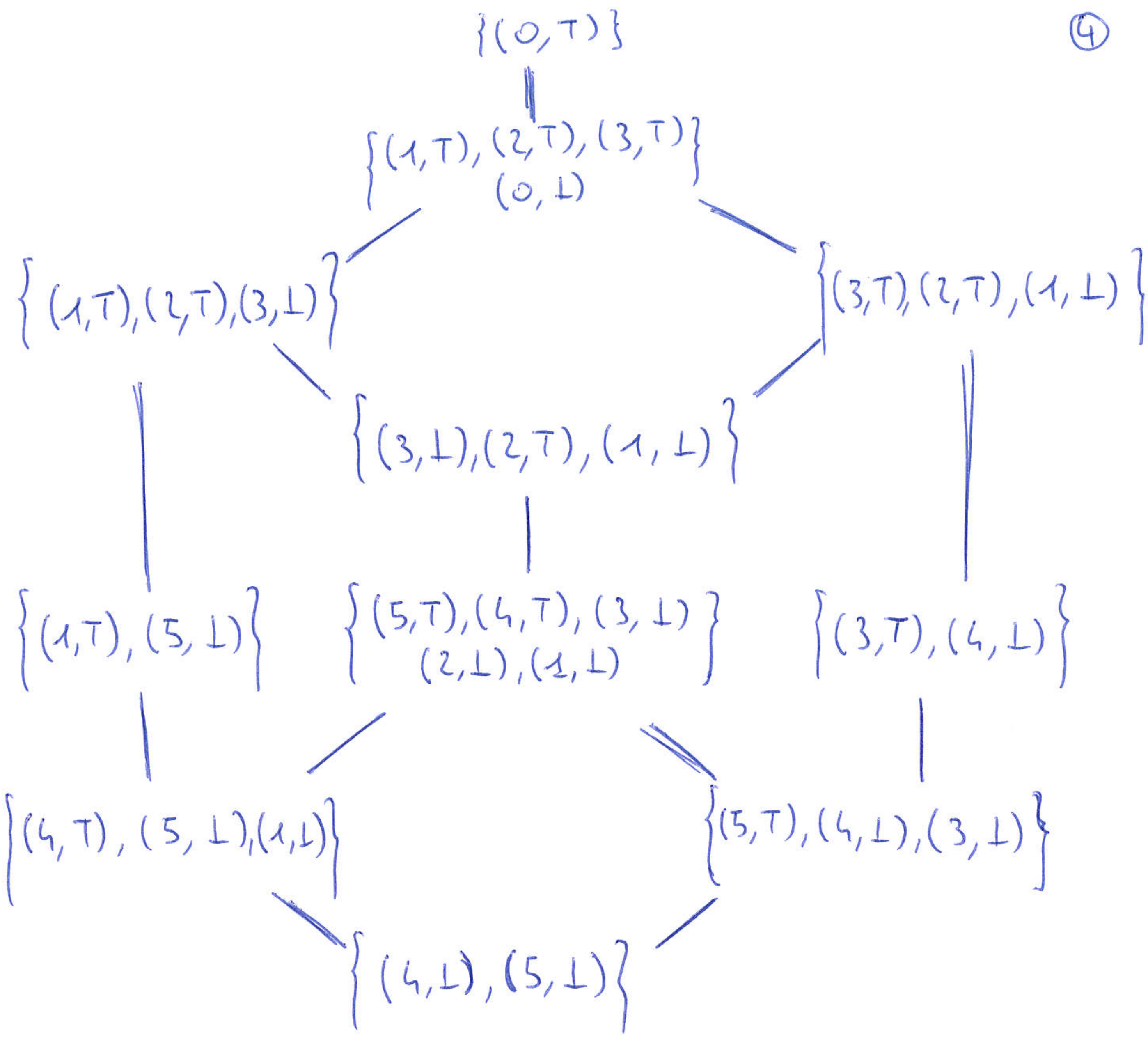
$A = \{(\emptyset, *), (\{l(tt)\}, *), (\{l(ff)\}, *), (\{r(*)\}, *), (\{l(tt), r(*)\}, *), (\{l(ff), r(*)\}, *)\}$

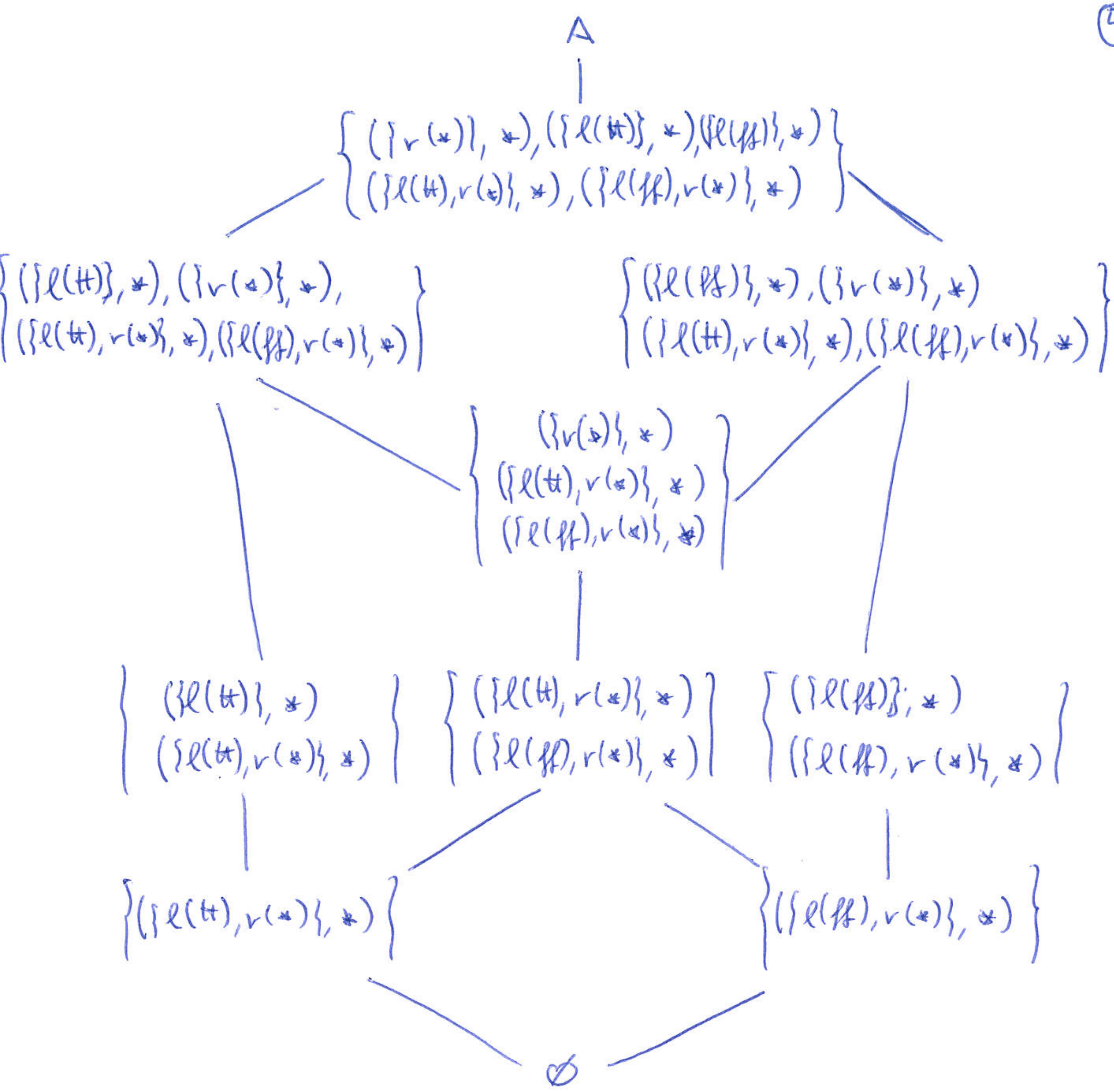
$Con = \mathcal{P}(A)$

Date le dimensioni di $\vdash$, mostriamo solo le relazioni di entailment per gli insiemi singoletto di $\mathcal{P}(A)$:

$\vdash = \{(\{(\emptyset, *)\}, (X, *)) \mid (X, *) \in A\} \cup \{(\{X\}, X) \mid X \in A\} \cup$
$\{(\{(\{l(b)\}, *)\}, (\{l(b), r(*)\}, *)) \mid b \in \{tt, ff\}\} \cup$
$\{(\{(\{r(*)\}, *)\}, (\{l(b), r(*)\}, *)) \mid b \in \{tt, ff\}\} \cup \ldots$

Unito tutte le restanti coppie non descritte dagli insiemi sopra.


{(0,⊤)}
{(1,⊤),(2,⊤),(3,⊤) (0,⊥)}
{(1,⊤),(2,⊤),(3,⊥)}
{(3,⊤),(2,⊤),(1,⊥)}
{(3,⊥),(2,⊤),(1,⊥)}
{(1,⊤),(5,⊥)}
{(5,⊤),(4,⊤),(3,⊥) (2,⊥),(1,⊥)}
{(3,⊤),(4,⊥)}
{(4,⊤),(5,⊥),(1,⊥)}
{(5,⊤),(4,⊥),(3,⊥)}
{(4,⊥),(5,⊥)}


A
{({r(*)}, *), ({l(tt)}, *), ({l(ff)}, *)
({l(tt), r(*)}, *), ({l(ff), r(*)}, *)}
{({l(tt)}, *), ({r(*)}, *),
({l(tt), r(*)}, *), ({l(ff), r(*)}, *)}
{({l(ff)}, *), ({r(*)}, *)
({l(tt), r(*)}, *), ({l(ff), r(*)}, *)}
{({r(*)}, *)
({l(tt), r(*)}, *)
({l(ff), r(*)}, *)}
{({l(tt)}, *)
({l(tt), r(*)}, *)}
{({l(tt), r(*)}, *)
({l(ff), r(*)}, *)}
{({l(ff)}, *)
({l(ff), r(*)}, *)}
{({l(tt), r(*)}, *)}
{({l(ff), r(*)}, *)}
∅

LINGUAGGI FUNZIONALI

Si definisca il seguente assegnamento di tipo per le variabili:

- z ha tipo Nat
- y ha tipo τ_y
- g ha tipo $\tau_g \equiv \tau_y \to \tau_y$
- x ha tipo τ_x
- f ha tipo $\tau_f \equiv \tau_x \to \tau_x$

Si scopre inoltre che il secondo termine viene applicato a x, quindi $\tau_x \equiv \tau_y \to \tau_y$ e che quindi $\tau_f \equiv (\tau_y \to \tau_y) \to (\tau_y \to \tau_y)$

Il terzo termine $(\lambda z. z+1)$ ha tipo $Nat \to Nat$.

Si calcola la semantica denotazionale dei singoli termini del nostro programma:

$$\llbracket \overbrace{\lambda f. \lambda x. (f(f\,x))}^{t_1} \rrbracket = \lambda \rho. \lfloor \lambda v_f : V_{\tau_f}. \llbracket \lambda x. (f(f\,x)) \rrbracket \rho[v_f/f] \rfloor$$

$$= \lambda \rho. \lfloor \lambda v_f : V_{\tau_f}. \lfloor \lambda v_x : V_{\tau_x}. \llbracket (f(f\,x)) \rrbracket \underbrace{\rho[v_f/f][v_x/x]}_{\rho'} \rfloor \rfloor$$

$$= \lambda \rho. \lfloor \lambda v_f : V_{\tau_f}. \lfloor \lambda v_x : V_{\tau_x}. \text{let } v_1 \Leftarrow \llbracket f \rrbracket \rho'. \text{let } v_2 \Leftarrow \llbracket (f\,x) \rrbracket \rho'. v_1(v_2) \rfloor \rfloor$$

$$= \lambda \rho. \lfloor \lambda v_f : V_{\tau_f}. \lfloor \lambda v_x : V_{\tau_x}. \text{let } v_1 \Leftarrow \lfloor v_f \rfloor. \text{let } v_2 \Leftarrow \llbracket (f\,x) \rrbracket \rho'. v_1(v_2) \rfloor \rfloor$$

$$= \lambda \rho. \lfloor \lambda v_f : V_{\tau_f}. \lfloor \lambda v_x : V_{\tau_x}. \text{let } v_1 \Leftarrow \lfloor v_f \rfloor. \text{let } v_2 \Leftarrow (\text{let } v_3 \Leftarrow \llbracket f \rrbracket \rho'. \text{let } v_4 \Leftarrow \llbracket x \rrbracket \rho'. v_3(v_4)). v_1(v_2) \rfloor \rfloor$$

$$= \lambda \rho. \lfloor \lambda v_f : V_{\tau_f}. \lfloor \lambda v_x : V_{\tau_x}. \text{let } v_1 \Leftarrow \lfloor v_f \rfloor. \text{let } v_2 \Leftarrow (\text{let } v_3 \Leftarrow \lfloor v_f \rfloor. \text{let } v_4 \Leftarrow \lfloor v_x \rfloor. v_3(v_4)). v_1(v_2) \rfloor \rfloor$$

$$= \lambda \rho. \lfloor \lambda v_f : V_{\tau_f}. \lfloor \lambda v_x : V_{\tau_x}. \text{let } v_1 \Leftarrow \lfloor v_f \rfloor. \text{let } v_2 \Leftarrow v_f(v_x). v_1(v_2) \rfloor \rfloor$$

$$= \lambda \rho. \lfloor \lambda v_f : V_{\tau_f}. \lfloor \lambda v_x : V_{\tau_x}. \text{let } v_2 \Leftarrow v_f(v_x). v_f(v_2) \rfloor \rfloor$$

Siccome $v_f(v_x)$ non sappiamo se sia un valore liftato o bottom usiamo l'operatore $*$ così come visto in classe (Winskel p. 131)

$$= \lambda \rho. \lfloor \lambda v_f : V_{\tau_f}. \lfloor \lambda v_x : V_{\tau_x}. v_f^*(v_f(v_x)) \rfloor \rfloor$$

$$\llbracket \overbrace{\lambda g.\lambda y.(g(g(g\ y)))}^{t_2} \rrbracket = \lambda \rho.\lfloor \lambda v_g : V_{\tau_g}.\llbracket \lambda y.(g(g(g\ y))) \rrbracket \rho[v_g/g] \rfloor$$

$$= \lambda \rho.\lfloor \lambda v_g : V_{\tau_g}.\lfloor \lambda v_y : V_{\tau_y}.\llbracket (g(g(g\ y))) \rrbracket \underbrace{\rho[v_g/g][v_y/y]}_{\rho'} \rfloor\rfloor$$

$$= \lambda \rho.\lfloor \lambda v_g : V_{\tau_g}.\lfloor \lambda v_y : V_{\tau_y}.\text{let } v_1 \Leftarrow \llbracket g \rrbracket \rho'.\ \text{let } v_2 \Leftarrow \llbracket (g(g\ y)) \rrbracket \rho'.\ v_1(v_2) \rfloor\rfloor$$

$$\stackrel{!}{=} \lambda \rho.\lfloor \lambda v_g : V_{\tau_g}.\lfloor \lambda v_y : V_{\tau_y}.\text{let } v_1 \Leftarrow \lfloor v_g \rfloor.$$
$$\text{let } v_2 \Leftarrow (\text{let } v_3 \Leftarrow \llbracket g \rrbracket \rho'.\ \text{let } v_4 \Leftarrow \llbracket (g\ y) \rrbracket \rho'.\ v_3(v_4)).\ v_1(v_2) \rfloor\rfloor$$

$$= \lambda \rho.\lfloor \lambda v_g : V_{\tau_g}.\lfloor \lambda v_y : V_{\tau_y}.\text{let } v_1 \Leftarrow \lfloor v_g \rfloor.$$
$$\text{let } v_2 \Leftarrow (\text{let } v_3 \Leftarrow \lfloor v_g \rfloor.$$
$$\text{let } v_4 \Leftarrow (\text{let } v_5 \Leftarrow \llbracket g \rrbracket \rho'.$$
$$\text{let } v_6 \Leftarrow \llbracket y \rrbracket \rho'.\ v_5(v_6)).\ v_3(v_4)).\ v_1(v_2) \rfloor\rfloor$$

$$= \lambda \rho.\lfloor \lambda v_g : V_{\tau_g}.\lfloor \lambda v_y : V_{\tau_y}.\text{let } v_1 \Leftarrow \lfloor v_g \rfloor.$$
$$\text{let } v_2 \Leftarrow (\text{let } v_3 \Leftarrow \lfloor v_g \rfloor.$$
$$\text{let } v_4 \Leftarrow (\text{let } v_5 \Leftarrow \lfloor v_g \rfloor.$$
$$\text{let } v_6 \Leftarrow \lfloor v_y \rfloor.\ v_5(v_6)).\ v_3(v_4)).\ v_1(v_2) \rfloor\rfloor$$

$$= \lambda \rho.\lfloor \lambda v_g : V_{\tau_g}.\lfloor \lambda v_y : V_{\tau_y}.\text{let } v_1 \Leftarrow \lfloor v_g \rfloor.$$
$$\text{let } v_2 \Leftarrow (\text{let } v_3 \Leftarrow \lfloor v_g \rfloor.\ \text{let } v_4 \Leftarrow v_g(v_y).\ v_3(v_4)).\ v_1(v_2) \rfloor\rfloor$$

$$= \lambda \rho.\lfloor \lambda v_g : V_{\tau_g}.\lfloor \lambda v_y : V_{\tau_y}.\text{let } v_1 \Leftarrow \lfloor v_g \rfloor.\ \text{let } v_2 \Leftarrow v_g^*(v_g(v_y)).\ v_1(v_2) \rfloor\rfloor$$

$$\stackrel{!}{=} \lambda \rho.\lfloor \lambda v_g : V_{\tau_g}.\lfloor \lambda v_y : V_{\tau_y}.\ v_g^*(v_g^*(v_g(y))) \rfloor\rfloor$$

$$\llbracket \overbrace{\lambda z.z+1}^{t_3} \rrbracket = \lambda \rho.\lfloor \lambda v_z : V_{\tau_z}.\llbracket z+1 \rrbracket \rho[v_z/z] \rfloor$$

$$= \lambda \rho.\lfloor \lambda v_z : V_{\tau_z}.\llbracket z \rrbracket \rho[v_z/z] +_\perp \llbracket 1 \rrbracket \rho[v_z/z] \rfloor$$

$$\stackrel{!}{=} \lambda \rho.\lfloor \lambda v_z : V_{\tau_z}.\lfloor v_z \rfloor +_\perp \lfloor 1 \rfloor \rfloor$$

$$\stackrel{!}{=} \lambda \rho.\lfloor \lambda v_z : V_{\tau_z}.\lfloor v_z + 1 \rfloor \rfloor$$

$$\llbracket \underbrace{0}_{t_4} \rrbracket = \lambda \rho.\lfloor 0 \rfloor$$

Si procede ora al calcolo della semantica per il programma:

$$(((t_1\ t_2)\ t_3)\ t_4)$$

$$[\![(t_1\ t_2)]\!] = \lambda\rho.\ \text{let } v_1 \Leftarrow [\![t_1]\!]\rho.\ \text{let } v_2 \Leftarrow [\![t_2]\!]\rho.\ v_1(v_2)$$

$$= \lambda\rho.\ \text{let } v_1 \Leftarrow \lfloor \lambda v_f : V_{\tau_f}.\ \lfloor \lambda v_x : V_{\tau_x}.\ v_f^*(v_f(v_x)) \rfloor \rfloor.$$
$$\quad \text{let } v_2 \Leftarrow \lfloor \lambda v_g : V_{\tau_g}.\ \lfloor \lambda v_y : V_{\tau_y}.\ v_g^*(v_g^*(v_g(v_y))) \rfloor \rfloor.\ v_1(v_2)$$

$$= \lambda\rho.\ \text{let } v_2 \Leftarrow \lfloor \lambda v_g : V_{\tau_g}.\ \lfloor \lambda v_y : V_{\tau_y}.\ v_g^*(v_g^*(v_g(v_y))) \rfloor \rfloor.$$
$$\quad ((\lambda v_f : V_{\tau_f}.\ \lfloor \lambda v_x : V_{\tau_x}.\ v_f^*(v_f(v_y)) \rfloor)\ v_2)$$

$$= \lambda\rho.\ \text{let } v_2 \Leftarrow \lfloor \lambda v_g : V_{\tau_g}.\ \lfloor \lambda v_y : V_{\tau_y}.\ v_g^*(v_g^*(v_g(v_y))) \rfloor \rfloor.$$
$$\quad \lfloor \lambda v_x : V_{\tau_x}.\ v_2^*(v_2(v_x)) \rfloor$$

$$= \lambda\rho.\ \lfloor \lambda v_x : V_{\tau_x}.\ ((\lambda v_g : V_{\tau_g}.\ \lfloor \lambda v_y : V_{\tau_y}.\ v_g^*(v_g^*(v_g(v_y))) \rfloor)^*$$
$$\quad ((\lambda v_g : V_{\tau_g}.\ \lfloor \lambda v_y : V_{\tau_y}.\ v_g^*(v_g^*(v_g(v_y))) \rfloor)\ v_x)) \rfloor$$

$$= \lambda\rho.\ \lfloor \lambda v_x : V_{\tau_x}.\ ((\lambda v_g : V_{\tau_g}.\ \lfloor \lambda v_y : V_{\tau_y}.\ v_g^*(v_g^*(v_g(v_y))) \rfloor)^*$$
$$\quad (\lfloor \lambda v_y : V_{\tau_y}.\ v_x^*(v_x^*(v_x(v_y))) \rfloor)) \rfloor$$

$$= \lambda\rho.\ \lfloor \lambda v_x : V_{\tau_x}.\ \lfloor \lambda v_y : V_{\tau_y}.\ ((\lambda v_y' : V_{\tau_y}.\ v_x^*(v_x^*(v_x(v_y'))))^*$$
$$\quad ((\lambda v_y' : V_{\tau_y}.\ v_x^*(v_x^*(v_x(v_y'))))^*$$
$$\quad ((\lambda v_y' : V_{\tau_y}.\ v_x(v_x(v_x(v_y'))))\ v_y))) \rfloor \rfloor$$

$$= \lambda\rho.\ \lfloor \lambda v_x : V_{\tau_x}.\ \lfloor \lambda v_y : V_{\tau_y}.\ ((\lambda v_y' : V_{\tau_y}.\ v_x^*(v_x^*(v_x(v_y'))))^*$$
$$\quad ((\lambda v_y' : V_{\tau_y}.\ v_x^*(v_x^*(v_x(v_y'))))^*$$
$$\quad (v_x^*(v_x^*(v_x(v_y)))))) \rfloor \rfloor$$

$$= \lambda\rho.\ \lfloor \lambda v_x : V_{\tau_x}.\ \lfloor \lambda v_y : V_{\tau_y}.\ v_x^*(v_x^*(v_x^*(v_x^*(v_x^*(v_x^*(v_x^*(v_x^*(v_x(v_y))))))))) \rfloor \rfloor$$

Abbiamo ottenuto una funzione che prende in input una funzione di tipo $\tau_y \to \tau_y$ e un valore di tipo τ_y e applica la funzione 9 volte.

$$[\![((t_1\ t_2)\ t_3)]\!] = \lambda\rho.\ \text{let } v_1 \Leftarrow [\![(t_1\ t_2)]\!]\rho.\ \text{let } v_2 \Leftarrow [\![t_3]\!].\ v_1(v_2)$$

$$= \lambda\rho.\ \lfloor \lambda v_y : V_{Nat}.\ \lfloor 1+1+1+1+1+1+1+1+1+v_y \rfloor \rfloor \equiv \lambda\rho.\ \lfloor \lambda v_y : V_{\tau_y}.\ \lfloor 9+v_y \rfloor \rfloor$$

$$[\![(((t_1\ t_2)\ t_3)\ t_4)]\!] = \lambda\rho.\ \text{let } v_1 \Leftarrow [\![((t_1\ t_2)\ t_3)]\!].\ \text{let } v_2 \Leftarrow [\![t_4]\!].\ v_1(v_2)$$

$$= \lambda\rho.\ ((\lambda v_y : V_{Nat}.\ \lfloor 9+v_y \rfloor)\ 0) = \lambda\rho.\ \lfloor 9+0 \rfloor = \lambda\rho.\ \lfloor 9 \rfloor$$

SEMANTICA DENOTAZIONALE

$$\frac{t_1 \Rightarrow \lambda f.\lambda x.(f(f\,x)) \quad t_2 \Rightarrow \lambda g.\lambda y.(g(g(g\,y))) \quad \overbrace{\lambda x.((\lambda g.\lambda y.(g(g(g\,y))))\,((\lambda g.\lambda y.(g(g(g\,y))))\,x))}^{t_1'} \Rightarrow t_1'}{(t_1\;t_2) \Rightarrow t_1'}$$

$$\frac{t_2 \Rightarrow t_2 \quad \dfrac{t_2 \Rightarrow t_2 \quad t_3 \Rightarrow t_3 \quad \lambda y.(t_3(t_3(t_3\,y))) \Rightarrow \overbrace{\lambda y.(t_3(t_3(t_3\,y)))}^{t_5}}{(t_2\;t_3) \Rightarrow \lambda y.(t_3(t_3(t_3\,x)))} \quad \lambda y'.(t_5(t_5(t_5\,y'))) \Rightarrow \overbrace{\lambda y'.(t_5(t_5(t_5\,y')))}^{t_6}}{}$$

$$\frac{(t_1\;t_2) \Rightarrow t_1' \quad t_3 \Rightarrow \lambda z.z+1 \quad ((\lambda g.\lambda y'.(g(g(g\,y))))\;((\lambda g.\lambda y.(g(g(g\,y))))\;(\lambda z.z+1))) \Rightarrow t_6}{((t_1\;t_2)\;t_3) \Rightarrow t_6}$$

$$\dfrac{\dfrac{\dfrac{t_3 \Rightarrow t_3 \quad 0 \Rightarrow 0 \quad 0+1 \Rightarrow 1}{t_3 \Rightarrow t_3 \quad (t_3\,0) \Rightarrow 1 \quad 1+1 \Rightarrow 2}}{t_3 \Rightarrow \lambda z.z+1 \quad (t_3(t_3\,0)) \Rightarrow 2 \quad 2+1 \Rightarrow 3}}{t_5 \Rightarrow t_5 \quad 0 \Rightarrow 0 \quad (t_3(t_3(t_3\,0))) \Rightarrow 3}$$

$$\dfrac{\dfrac{\dfrac{t_3 \Rightarrow t_3 \quad 3 \Rightarrow 3 \quad 3+1 \Rightarrow 4}{t_3 \Rightarrow t_3 \quad (t_3\,3) \Rightarrow 4 \quad 4+1 \Rightarrow 5}}{t_3 \Rightarrow \lambda z.z+1 \quad (t_3(t_3\,3)) \Rightarrow 5 \quad 5+1 \Rightarrow 6}}{(t_3(t_3(t_3\,3))) \Rightarrow 6}$$

$$\dfrac{\dfrac{\dfrac{t_3 \Rightarrow t_3 \quad 6 \Rightarrow 6 \quad 6+1 \Rightarrow 7}{t_3 \Rightarrow t_3 \quad (t_3\,6) \Rightarrow 7 \quad 7+1 \Rightarrow 8}}{t_3 \Rightarrow \lambda z.z+1 \quad (t_3(t_3\,6)) \Rightarrow 8 \quad 8+1 \Rightarrow 9}}{(t_3(t_3(t_3\,6))) \Rightarrow 9}$$

$$\dfrac{\dfrac{t_5 \Rightarrow t_5 \quad (t_5\,0) \Rightarrow 3 \quad (t_3(t_3(t_3\,3))) \Rightarrow 6}{t_5 \Rightarrow \lambda y.(t_3(t_3(t_3\,y))) \quad (t_5(t_5\,0)) \Rightarrow 6 \quad (t_3(t_3(t_3\,6))) \Rightarrow 9}}{((t_1\;t_2)\;t_3) \Rightarrow t_6 \quad t_4 \Rightarrow 0 \quad (t_5(t_5(t_5\,0))) \Rightarrow 9}$$

$$(((t_1\;t_2)\;t_3)\;t_4) \Rightarrow 9$$

DOMINI RICORSIVI

La soluzione dell'equazione di dominio ricorsive può essere calcolata come minimo punto fisso della funzione:

$$F(X) = (O_\bot + X)_\bot$$

Siccome le operazioni di lifting e somma sugli IS sono funzioni continue nel CPO degli information system con relazione d'ordine $\trianglelefteq$, F è continua e per il teorema del punto fisso otteniamo che

$$fix(F) = \bigsqcup_{n \in \omega} F^n(\bot)$$

è il minimo punto fisso di F. Di seguito vengono calcolate le prime approssimazioni della funzione F.

$F^0 = \bot$

$F^1 = (O_\bot + \bot)_\bot = (O_\bot)_\bot = ((\{*_2\}, \{\{*_2\}, \emptyset\}, \{(\{*_2\}, *_1)\}))_\bot \equiv 1_\bot$

$= (\{*_1, *_2\},$

$\{\emptyset, \{*_1\}, \{*_2\}, \{*_1, *_2\}\},$

$\{(\{*_1\}, *_2), (\{*_1\}, *_2), (\{*_2, *_2\}, *_2), (\{*_1, *_2\}, *_2)\})$

In cui: $*_1 = l(l(r(*)))$ e $*_2 = r(*)$

Il grafo del dominio è:

$\{*_1, *_2\}$
|
$\{*_2\}$
|
$\emptyset$

$F^2 = (O_\bot + F^1(\bot))_\bot = ((\{*_1\} \uplus \{*_1, *_2\}, Con_{1+1_\bot}, \vdash_{1+1_\bot}))_\bot$

$= ((\{*_1, *_2, *_3\},$

$\{\emptyset, \{*_1\}, \{*_2\}, \{*_3\}, \{*_2, *_3\}\},$

$\{(\{*_2\}, *_1), (\{*_2\}, *_2), (\{*_3\}, *_3), (\{*_2, *_3\}, *_2), (\{*_2\}, *_3), (\{*_2, *_3\}, *_3)\}))_\bot$

$= \big(\{*_1, *_2, *_3, *_4\},$

$\{\emptyset, \{*_1\}, \{*_2\}, \{*_3\}, \{*_4\}, \{*_2, *_3\}, \{*_1, *_4\}, \{*_2, *_4\}, \{*_3, *_4\}, \{*_2, *_3, *_4\}\},$

$\{(\{*_1\}, *_1), (\{*_1\}, *_4), (\{*_2\}, *_2), (\{*_2\}, *_3), (\{*_2\}, *_4), (\{*_3\}, *_3), (\{*_3\}, *_4),$

$(\{*_4\}, *_4), (\{*_2, *_3\}, *_2), (\{*_2, *_3\}, *_3), (\{*_2, *_3\}, *_4), (\{*_1, *_4\}, *_1), (\{*_1, *_4\}, *_4),$

$(\{*_2, *_4\}, *_2), (\{*_2, *_4\}, *_4), (\{*_3, *_4\}, *_3), (\{*_3, *_4\}, *_4), (\{*_2, *_4\}, *_3),$

$(\{*_2, *_3, *_4\}, *_2), (\{*_2, *_3, *_4\}, *_3), (\{*_2, *_3, *_4\}, *_4)\}\big)$

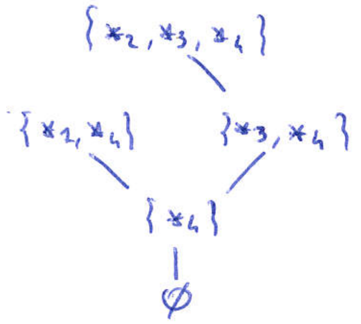
con $*_1 = \ell(\ell(r(*)))$

$*_2 = \ell(r(\ell(\ell(r(*)))))$

$*_3 = \ell(r(r(*)))$

$*_4 = r(*)$

Il grafo associato al dominio $\lceil F^1 \rceil$ è:



$$F_3 = (O_\perp + F^2(\perp)_\perp) = \Big(\big(\{\ast_1\} \uplus \{\ast_1, \ast_2, \ast_3, \ast_4\}, \mathrm{Con}_{1+F^2(\perp)}, \vdash_{1+F^2(\perp)} \big) \Big)_\perp$$

$$= \Big(\big(\{\ast_1, \ast_2, \ast_3, \ast_4, \ast_5\},$$

$$\{\emptyset, \{\ast_1\}, \{\ast_2\}, \{\ast_3\}, \{\ast_4\}, \{\ast_5\}, \{\ast_3, \ast_4\}, \{\ast_2, \ast_5\}, \{\ast_3, \ast_5\}, \{\ast_4, \ast_5\},$$
$$\{\ast_3, \ast_4, \ast_5\}\},$$

$$\{(\{\ast_1\}, \ast_1), (\{\ast_2\}, \ast_2), (\{\ast_2\}, \ast_5), (\{\ast_3\}, \ast_3), (\{\ast_3\}, \ast_4), (\{\ast_3\}, \ast_5), (\{\ast_4\}, \ast_4),$$
$$(\{\ast_4\}, \ast_5), (\{\ast_5\}, \ast_5), (\{\ast_3, \ast_4\}, \ast_3), (\{\ast_3, \ast_4\}, \ast_4), (\{\ast_3, \ast_4\}, \ast_5), (\{\ast_2, \ast_5\}, \ast_2),$$
$$(\{\ast_2, \ast_5\}, \ast_5), (\{\ast_3, \ast_5\}, \ast_3), (\{\ast_3, \ast_5\}, \ast_4), (\{\ast_3, \ast_5\}, \ast_5), (\{\ast_4, \ast_5\}, \ast_4),$$
$$(\{\ast_4, \ast_5\}, \ast_5), (\{\ast_3, \ast_4, \ast_5\}, \ast_3), (\{\ast_3, \ast_4, \ast_5\}, \ast_4), (\{\ast_3, \ast_4, \ast_5\}, \ast_5)\} \big) \Big)_\perp$$

$$= \big(\{\ast_1, \ast_2, \ast_3, \ast_4, \ast_5, \ast_6\},$$

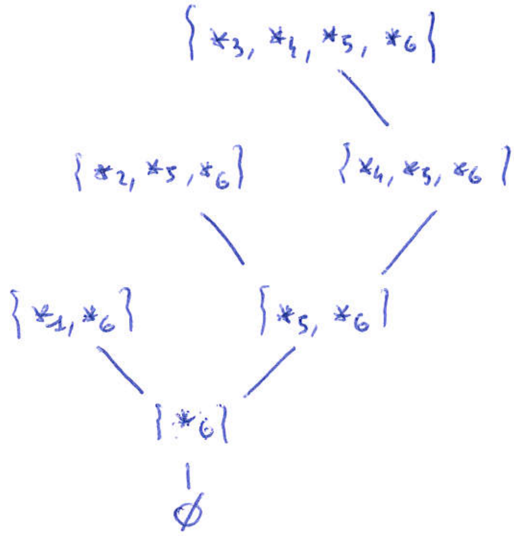
$$\{\emptyset, \{\ast_1\}, \{\ast_2\}, \{\ast_3\}, \{\ast_4\}, \{\ast_5\}, \{\ast_6\}, \{\ast_1, \ast_6\}, \{\ast_2, \ast_6\}, \{\ast_3, \ast_6\}, \{\ast_4, \ast_6\},$$
$$\{\ast_5, \ast_6\}, \{\ast_3, \ast_4\}, \{\ast_2, \ast_5\}, \{\ast_3, \ast_5\}, \{\ast_4, \ast_5\}, \{\ast_3, \ast_4, \ast_6\}, \{\ast_2, \ast_5, \ast_6\}, \{\ast_3, \ast_5, \ast_6\},$$
$$\{\ast_4, \ast_5, \ast_6\}, \{\ast_3, \ast_4, \ast_5\}, \{\ast_3, \ast_4, \ast_5, \ast_6\}\},$$

$$\{(\{\ast_1\}, \ast_1), (\{\ast_1\}, \ast_6), (\{\ast_2\}, \ast_2), (\{\ast_2\}, \ast_5), (\{\ast_2\}, \ast_6), (\{\ast_3\}, \ast_3), (\{\ast_3\}, \ast_4),$$
$$(\{\ast_3\}, \ast_5), (\{\ast_3\}, \ast_6), (\{\ast_4\}, \ast_4), (\{\ast_4\}, \ast_5), (\{\ast_4\}, \ast_6), (\{\ast_5\}, \ast_5), (\{\ast_5\}, \ast_6),$$
$$(\{\ast_1, \ast_6\}, \ast_1), (\{\ast_1, \ast_6\}, \ast_6), (\{\ast_2, \ast_6\}, \ast_2), (\{\ast_2, \ast_6\}, \ast_5), (\{\ast_2, \ast_6\}, \ast_6),$$
$$(\{\ast_3, \ast_6\}, \ast_3), (\{\ast_3, \ast_6\}, \ast_4), (\{\ast_3, \ast_6\}, \ast_5), (\{\ast_3, \ast_6\}, \ast_6), (\{\ast_4, \ast_6\}, \ast_4),$$
$$(\{\ast_4, \ast_6\}, \ast_5), (\{\ast_4, \ast_6\}, \ast_6), (\{\ast_5, \ast_6\}, \ast_5), (\{\ast_5, \ast_6\}, \ast_6), (\{\ast_3, \ast_4\}, \ast_3),$$
$$(\{\ast_3, \ast_4\}, \ast_4), (\{\ast_3, \ast_4\}, \ast_5), (\{\ast_3, \ast_4\}, \ast_6), (\{\ast_2, \ast_5\}, \ast_2), (\{\ast_2, \ast_5\}, \ast_5),$$
$$(\{\ast_2, \ast_5\}, \ast_6), (\{\ast_3, \ast_5\}, \ast_3), (\{\ast_3, \ast_5\}, \ast_4), (\{\ast_3, \ast_5\}, \ast_5), (\{\ast_3, \ast_5\}, \ast_6),$$
$$(\{\ast_4, \ast_5\}, \ast_4), (\{\ast_4, \ast_5\}, \ast_5), (\{\ast_4, \ast_5\}, \ast_6), (\{\ast_3, \ast_4, \ast_6\}, \ast_3), (\{\ast_3, \ast_4, \ast_6\}, \ast_4),$$
$$(\{\ast_3, \ast_4, \ast_6\}, \ast_5), (\{\ast_3, \ast_4, \ast_6\}, \ast_6), (\{\ast_2, \ast_5, \ast_6\}, \ast_2), (\{\ast_2, \ast_5, \ast_6\}, \ast_5),$$
$$(\{\ast_2, \ast_5, \ast_6\}, \ast_6), (\{\ast_3, \ast_5, \ast_6\}, \ast_3), (\{\ast_3, \ast_5, \ast_6\}, \ast_4), (\{\ast_3, \ast_5, \ast_6\}, \ast_5),$$
$$(\{\ast_3, \ast_5, \ast_6\}, \ast_6), (\{\ast_4, \ast_5, \ast_6\}, \ast_4), (\{\ast_4, \ast_5, \ast_6\}, \ast_5), (\{\ast_4, \ast_5, \ast_6\}, \ast_6),$$
$$(\{\ast_3, \ast_4, \ast_5\}, \ast_3), (\{\ast_3, \ast_4, \ast_5\}, \ast_4), (\{\ast_3, \ast_4, \ast_5\}, \ast_5), (\{\ast_3, \ast_4, \ast_5\}, \ast_6),$$
$$(\{\ast_3, \ast_4, \ast_5, \ast_6\}, \ast_3), (\{\ast_3, \ast_4, \ast_5, \ast_6\}, \ast_4), (\{\ast_3, \ast_4, \ast_5, \ast_6\}, \ast_5),$$
$$(\{\ast_3, \ast_4, \ast_5, \ast_6\}, \ast_6)\} \big)$$

Gli insiemi chiusi per entailment di F^3 sono:

$\emptyset, \{*_6\}, \{*_1, *_6\}, \{*_5, *_6\}, \{*_2, *_5, *_6\}, \{*_4, *_5, *_6\}, \{*_3, *_4, *_5, *_6\}$

Il grafo del dominio associato a $\lceil F^3 \rceil$ è:



Con:

- $*_1 = l(l(r(*)))$
- $*_2 = l(r(l(l(r(*)))))$
- $*_3 = l(r(l(r(l(l(r(*)))))))$
- $*_4 = l(r(l(r(r(*)))))$
- $*_5 = l(r(r(*)))$
- $*_6 = r(*)$

In generale avremo che l'approssimazione F^n sarà un albero binario con n foglie e n nodi interni con radice un ulteriore nodo a rappresentare $\bot$. Il dominio soluzione è quello associato al least upper bound:

$$\bigsqcup_n F^n(\bot) = \left(\bigcup_n A_n, \bigcup_n Con_n, \bigcup_n \vdash_n\right)$$

Un tipo di dato Haskell che si avvicina alle nostre equazioni ricorsive potrebbe essere così definito:

```
data Nat a = Zero | Succ a
data X = Lift (Nat X) | Bottom
```

e avremo ad esempio che: Zero $\equiv l(r(*))$ e Succ(Bottom) $\equiv r(r(*))$

Quindi:

- $*_1$ = Lift Zero
- $*_2$ = Lift (Succ (Lift Zero))
- $*_3$ = Lift (Succ (Lift (Succ (Lift Zero))))
- $*_4$ = Lift (Succ (Lift (Succ Bottom)))
- $*_5$ = Lift (Succ Bottom)
- $*_6$ = Bottom