Esercizio 1

$\Gamma' = \Gamma, f : T_1$

$Id := fn\ y : bool \Rightarrow bool$

$T_1 \equiv \cancel{(bool \to bool)}^{T_2 \to T_3} \to \cancel{T}^{(T_A \to T_B)}$

$T_2 \equiv bool \to bool$

posso dire $T_C = T_{fix}$ ovvero T_C è il tipo del fix

$T_3 \equiv T_2 \to (T_A \to T_B) \to (T_A \to T_B)$

$$\dfrac{\dfrac{\dfrac{\dfrac{\Gamma'''(h) = T_3 = T_2 \to T_E}{\Gamma''' \vdash h : \cancel{T_A}^{T_2} \to T_B}\,(var) \qquad \dfrac{\Gamma'''(g) = T_2}{\Gamma''' \vdash g : \cancel{T}\ T_2}\,(var)}{\Gamma''' \vdash hg : (T_A \to T_B) \to (T_A \to T_B) \equiv T_E}\,(App)}{\Gamma''' = (\Gamma'', h : T_3) \vdash fix(hg) : T_C \equiv T_A \to T_B}\,(fix)}{\dfrac{\Gamma'' = (\Gamma_{,g}, T_2) \vdash fn\ h : T_3 \Rightarrow fix.(hg) : \cancel{T_D}\ T \equiv T_3 \to T_C}{\Gamma \vdash (fn\ g.\ T_2 \Rightarrow fn\ h.\ T_3 \Rightarrow fix(hg)) : \cancel{T_A}\ T_4 = T_2 \to \cancel{T_D}\ T}\,(fn)}$$

$$\dfrac{\dfrac{\dfrac{\Gamma'(f) = T_1 = (bool \to bool) \to T}{\Gamma' \vdash f : T_A \to T}\,(var) \qquad \dfrac{}{\Gamma' \vdash Id : T_A = bool \to bool}\,(fn)}{\Gamma' = \Gamma, f : T_1 \vdash f\ Id : T}\,(App)}{\Gamma \vdash (fn.\ T_1 \Rightarrow f\ Id) : \cancel{T_A}^{T_1} \to T}\,(fn)$$

$$\Gamma \vdash (fn\ f : T_1 \Rightarrow f\ Id)(fn\ g.\ T_2 \Rightarrow (fn\ h : T_3 \Rightarrow fix.(hg))) : T$$

Esercizio 2

$T_1 \equiv unit \to T$

$T_2 = unit$

$T = T_3 \to bool$

$T_3 \equiv bool \to (T_2 \to bool) \to (T_2 \to bool)$

$$\dfrac{\dfrac{\dfrac{\dfrac{\Gamma'(f) = T_1 \equiv unit \to T}{\Gamma' \vdash f : \cancel{T_B}^{unit} \to T}\,(var) \qquad \dfrac{}{\Gamma \vdash skip : \cancel{T_B}^{unit}}\,(skip)}{\Gamma'' = (\Gamma, f.\ T_1) \vdash f\ skip : T}\,(App)}{\Gamma \vdash (fn\ f.\ T_1 \Rightarrow f\ skip) : \cancel{T}^{T_1} \to T}\,(fn)}$$

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\Gamma'''(h) = T_3 = bool \to T_E}{\Gamma''' \vdash h : \cancel{T_D}^{bool} \to T_E}\,(var) \qquad \dfrac{}{\Gamma''' \vdash true : \cancel{T_D}\ bool}\,(bool)}{\Gamma''' \vdash (h\ true) : (T_2 \to bool) \to (T_2 \to bool)}\,(App)\ \cancel{T_B}}{\Gamma''' \vdash (fix.(h\ true)) : \cancel{T_C}^{unit} \to bool} \qquad \dfrac{\Gamma'''(g) = \cancel{T_A}^{unit} = T_C}{\Gamma''' \vdash g : \cancel{T_C}\ T_A}}{\Gamma''' \vdash (fix.(h\ true))\ g : bool}\,(App)}{\Gamma''' \equiv (\Gamma'', h : T_3) \vdash \neg(fix.(h\ true)\ g) : \cancel{T_S}\ bool}\,(not)}{\Gamma'' \equiv (\Gamma_{,g} : T_2) \vdash (fn\ h.\ T_3 \to \neg(fix(h\ true)\ g)) : \cancel{T_R}\ T \equiv T_3 \to T_S}\,(fix)}{\Gamma \vdash fn\ g.\ T_2 \Rightarrow (fn\ h.\ T_3 \Rightarrow \neg(fix.(h\ true)\ g)) : \cancel{T_A}\ T_1 \equiv \cancel{T}^{unit} \to \cancel{T}^{T}\ *}\,(fn)$$

$$\Gamma \vdash (fn\ f.\ T_1 \Rightarrow f\ skip)(fun(g : T_2) \Rightarrow (fun(h : T_3) \Rightarrow \neg(fix.(h\ true)\ g))) : T$$

(App)

let $g : T_3$
let $x : T_L = \text{fun } 5 \text{ in } \underset{T_{int}}{g}\ (\text{fun } z : \text{ref}\{a : \text{int}\} \Rightarrow \ldots)(\text{ref } x)$

$\rightarrow$ let $x : T_1 = \{a = 5, b = 60\}$ in .....

$\rightarrow$ $g\,(\text{fn } z : \text{ref}\,\underbrace{\{a : \text{int}\}}_{\text{non tipabile}} \Rightarrow \ldots)\,(\text{ref}\,\{a = 5, b = 60\})$

Considerando che nell'evaluazione la tipabilità si mantiene, non essendo tipabile alla fine non lo si nominerà all'inizio

$$\Gamma' = \dfrac{\dfrac{\Gamma'(g) = \text{int} = T_A}{\Gamma' \vdash g : \cancel{T_A}\ \text{int}}\,(\text{var}) \qquad \dfrac{\dfrac{\Gamma'(g) = \text{int}}{\Gamma' \vdash g : \text{int}}\,(\text{var})}{\Gamma' \vdash g + g : \cancel{T_B}\ \text{int}}\,(\text{sum})}{(\Gamma, g, \text{int}) \vdash \{a = g, b = g + g\} : T_1 = \{a : T_A, b : T_B\} \equiv \{a.\text{int}, b.\text{int}\}}\,(\text{rec})$$

$$\dfrac{\Gamma \vdash (\text{fn } y : \text{int} \Rightarrow \ldots) : T_A \to T_1 \qquad \Gamma \vdash 5 : \cancel{T_A}\ \text{int}}{\Gamma \vdash (\text{fn } y : \text{int} \to \{a = g, b = g + g\})\,5 : T_1}\,(\text{App})$$

$$\dfrac{\dfrac{\dfrac{\dfrac{\Gamma''' \vdash \text{fn } x.h : T_A \to T_R \qquad \dfrac{\dfrac{\Gamma'''(x) = T_A \to T_1}{\Gamma''' \vdash x : \cancel{T_S}\ T_1}}{\Gamma''' \vdash \text{ref } x : T_C = \text{ref } \cancel{T_S}\ T_1}}{\Gamma''' \vdash (\text{fn } z : \text{ref}\{a : \text{int}\} \Rightarrow \ldots) : T_C \to T_A}}{\Gamma''' \vdash (\text{fn } z : \text{ref}\{a : \text{int}\} \Rightarrow \ldots)(\text{ref } x) : T_A}}{\Gamma''' = (\Gamma'', h : T_2) \vdash (\text{fix } h.h)\,(\text{fn } z : \text{ref}\{a.\text{int}\} \Rightarrow \ldots)(\text{ref } x) \equiv T_R}}{(\Gamma, x.T_1) \overset{\Gamma'''}{\vdash} \text{fn } h : T_2 \Rightarrow (\text{fix } h)(\ldots) : T}\,(\text{let})$$

$$\Gamma \vdash \text{let } x.T_1 = \{\text{fun } y : \text{int} \Rightarrow \{a = g, b = g + g\}\}\,5 \sim \text{fn } h : T_2 \Rightarrow (\text{fix } h)\,(\text{fn } z : \text{ref}\{a.\text{int}\} \Rightarrow \ldots)(\text{ref } x) : T$$

Esercizi 7/11/2023

$T_1 \equiv T_2 \to \{a:int, b:bool, c:int\}$

$T_2 = \{a:bool, b:int\}$

$T_1 = T_2 \to T_x$

$T_2 = \{a:T_B, b:T_A\}$

$T = bool$

$T_x = \{a:T_A, b:T_B, c:\cancel{T_C}\ T_A\}$

$$\dfrac{\dfrac{\dfrac{\Gamma'(x) = T_2 = \{\ldots b:T_A \ldots\}}{\Gamma' \vdash x:\{\ldots b:T_A \ldots\}}\,(var)}{\Gamma' \vdash \#bx : T_A}\,(proj) \qquad \dfrac{\dfrac{\Gamma'(x) = T_2 = \{\ldots a:T_B \ldots\}}{\Gamma' \vdash x:\{\ldots a:T_B \ldots\}}\,(var)}{\Gamma' \vdash \#ax . T_B}\,(proj) \qquad \dfrac{\dfrac{\Gamma' \vdash x: <\ldots b:T_C \ldots\}}{\Gamma' \vdash x.\{\ldots b:T_C \ldots\}}\,(var)}{\Gamma' \vdash \#bx . T_C}\,(proj)}{(\Gamma, x.T_2) \vdash \{a = \#bx, b = \#ax, c = \#bx\} : T_x = \ldots}$$

$$\dfrac{\dfrac{\Gamma \vdash (fn\ g:\{a:int, c:int\} \Rightarrow \ldots) : \{a:int, c:int\} \Rightarrow bool \qquad \Gamma'' \vdash f\{a=false, b=6, c=true\}) : \{a:int, c:int\}}{\Gamma \vdash (fn\ g:\{a:int, c:int\} \Rightarrow \ldots)((f\{a=false, b=6, c=true\})) : bool}}{\Gamma, f:T_1 \vdash (\ldots \wedge (fn\ g.\{a:int, c:int\} \Rightarrow \ldots)(f\{\ldots\})) : \cancel{T}\ bool}$$

2 subtyping

$$\dfrac{\dfrac{\Gamma \vdash fn\ x:T_2 \Rightarrow \{a = \#bx, b = \#ax, c = \#bx\}) : T_1 = T_2 \to T_x}{\Gamma \vdash let\ f:T_1 = fn\ x:T_2 \to \{a = \#bx, b = \#ax, c = \#bx\}\ in\ (\sim \wedge (fn\ g:\{a:int, c:int\} \to \ldots)(f\{a=false, b=6, c=true\})) : T}}{}$$

Esercitazione 9/11/2023

$T_1 \equiv T_2 \to T_x$

$T_2 \equiv \{a: \cancel{T_B}^{bool}, b: \cancel{T_A}^{int}\}$

$T \equiv bool$

$T_x \equiv \{a: \cancel{T_A}^{int}, b: \cancel{T_B}^{bool}, c: \cancel{T_A}^{int}\}$

$$
\dfrac{
\dfrac{\Gamma'' (f) = T_1 \equiv T_2 \to T_x}{\Gamma'' \vdash f: \cancel{T_A}^{T_2} \to \{a: int, b: T_B, c: int\}}
\quad
\dfrac{\dfrac{\Gamma \vdash ff: bool \quad \Gamma \vdash 6: int \quad \Gamma \vdash tt: bool}{\Gamma'' \vdash \{a = ff, b = 6, c = tt\}: \{a: bool, b: int, c: bool\}}\,(vec)}{\Gamma \vdash \{a: false, b = 6, c = true\}: \cancel{T_A}^{T_2 = \{a: \cancel{T_B}^{bool}, b: int\}}}\,(App)
}{\Gamma'' \vdash f\{a = false, b = 6, c = true\}: \{a: T_A, b: T_B, c: T_A\}}
$$

per aggiustare il tipo argomento di f.

$$
\dfrac{\{a: \cancel{T_A}^{int}, b: T_B, c: \cancel{T_A}^{int}\} <: \{a: int, c: int\}}{}\,(sub)
$$

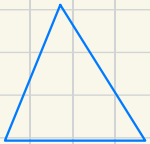
$$
\Gamma'' \vdash f\{a = false, b = 6, c = true\}: \{a: int, c: int\}
$$

$\Gamma'' = \Gamma', f: T_1$

$$\text{(var)}\ \frac{\Gamma''(f) = T_1 \equiv T_2 \to T_x}{\Gamma'' \vdash f : T_2 \to T_x}$$

Tipo Argomento

$$\text{(sub)}\ \frac{\{a:T_B, b:T_A\} \quad // \quad T_2 := \{a:bool, b:int, c:bool\}}{\{a:T_B, b:T_A\} <: \{a:int, c:int\}}$$



$$\text{(sub)}\ \frac{\{a:T_A, b:T_B, c:T_A\} = T_x <: \{a:int, c:int\}}{\Gamma'' \vdash f : T_A \to \{a:int, c:int\}}$$

$$\text{(rec)}\ \frac{\Gamma'' \vdash ff : bool \qquad \Gamma'' \vdash 6 : int \qquad \Gamma'' \vdash tt : bool}{\Gamma'' \vdash \{a=ff, b=6, c=tt\} : \{a:bool, b:int, c:bool\}}$$

$$\text{(App)}\ \frac{\Gamma'' \vdash f : T_A \to \{a:int, c:int\} \qquad \Gamma'' \vdash \{a=ff, b=6, c=tt\} : \{a:bool, b:int, c:bool\}}{\Gamma \vdash \{a=false, b=6, c=true\} : \{a:int, c:int\}}$$

$\Gamma'' = \Gamma, f : T_1$

$$\Gamma'' = \Gamma, f : T_1$$

$$\dfrac{\dfrac{\dfrac{\Gamma''(f) = T_1 = T_2 \to T_x = \{\ldots\} \to \{\ldots\}}{\Gamma'' \vdash f : \{a.T_B, b.T_A\} \to \{a:T_A, b:T_B, c:T_A\}}\,(\text{var}) \qquad \dfrac{\dfrac{\triangle}{\{a.T_B, b:T_A\} :> \{a:\text{bool}, b:\text{int}, c:\text{bool}\}} \qquad \dfrac{\triangle}{\{a:T_A, b:T_B, c:T_A\} <: \{a:\text{int}, c:\text{int}\}}\,(\text{sub rec})}{(\{a:T_B, b:T_A\} \to \{a:T_A, b:T_B, c:T_A\}) <: (\{a:\text{bool}, b:\text{int}, c:\text{bool}\} \to \{a:\text{int}, c:\text{int}\})}\,(\text{sub fun})}{\Gamma'' \vdash f : \{a.\text{bool}, b.\text{int}, c.\text{bool}\} \to \{a.\text{int}, c.\text{int}\}}}{\Gamma'' \vdash f\{a = \text{false}, b = 6, c = \text{true}\} : \{a:\text{int}, c:\text{int}\}}$$

$\text{ref } T < \text{ref } T'$

no regola di Tipaggio

$$\dfrac{\dfrac{\Gamma \vdash e : T \qquad T < T'}{\Gamma \vdash e : T'}\ (sub)}{\Gamma \vdash \text{ref } e : \text{ref } T'}\ (ref)$$

$T_1 = \{a:bool, c:int\} \rightarrow \text{ref}\{a:int, b:bool, c:bool\}$

$\{b:bool, r:int\} \rightarrow \text{ref}\{c:bool\}$

$T_2 = \{b:bool, c:int\}$

$$\dfrac{\dfrac{\dfrac{\dfrac{\Gamma \vdash \{\ldots\} : \{a:T_A, b:T_B, c:T_C\} \qquad \{a:T_A, b:T_B, c:T_C\} <: \{c:bool\}}{\Gamma' \vdash \{a=\#cx, b=\#cx \geq 0, c=\#bx\} : \{c:bool\}}}{(\Gamma, x:T_2) \overset{=\Gamma'}{\vdash} \text{ref}\{a=\#cx, b=\#cx \geq 0, c=\#bx\} : \text{ref}\{c:bool\}}}{\Gamma \vdash \text{fn } x:T_2 \rightarrow \text{ref}\{a=\#cx, b=\#cx \geq 0, c=\#bx\} : T_1 = T_2 \rightarrow T_x} \qquad \Gamma', f:T_1 \vdash (\text{fn } g:\text{ref}\{c:bool\} \Rightarrow \#c!g)(f\{\}) : T_A}{\Gamma \vdash \text{let } f:T_1 = \text{fn } x:T_2 \Rightarrow \text{ref}\{\} \text{ in } (\text{fn } g:\text{ref}\{c:bool\} \Rightarrow \#c!g)(f\{a=false, b=true, c=2\}) : T}$$

Esercitazione 14/11/2023

$fix\ f.\ T: \{a:int, b:int\} \to \{a:bool, b:bool, c:bool\}$

$f: T_1 \equiv \overset{T_A \to T_B}{T_2} \to \overbrace{T_3 \to \{a:bool, b:bool, c:bool\}}$

$T_2 = \cancel{?}\ T_3 \to \{a:bool, b:bool, c:bool, d:bool\}$

$T_3 = \{a:int\}$

$$\frac{\Gamma \vdash f:(T_A \to T_B) \to (T_A \to T_B)}{\Gamma \vdash fix\ f.\ T_1 \to T_B}$$

$Fix\ f.\ T_3 \to \{....\}$ $\qquad T_1 \equiv (\{a:int\} \to \{a:bool, ..., d:bool\}) \to \{a:int\} \to \{a:bool, d.bool\}$

$$\frac{\Gamma'(f) = T_1 = \quad \Gamma' \vdash f \vdash (\{a:int\} \to \overset{T_2}{\{a:bool, d:bool\}}) \to \overset{T_3}{\{a:int\}} \to \{a:bool, ..., d:bool\}}{\Gamma' \vdash fix\ f: \{a:int\} \to \{a:bool, d.bool\}}$$

$$\frac{\{a:int\} \underset{:>}{} \{a:int\} \qquad \{a:bool, ..., d:bool\} \underset{<:}{} \{a:bool, ..., c:bool\}}{\{a:int\} \to \{a:bool, d:bool\} \underset{<}{} \{a:int\} \to \{a:bool, c:bool\}}$$

$$\frac{\Gamma' \vdash 5,6:int \quad \Gamma' \vdash \{a=5, b=6\}: \{a:int, b:int\}}{\{a:int, b:int\} < \{a.int\}}\ (rec)$$

$$\frac{\Gamma' \vdash fix\ f: \overset{\{a:int\}}{\cancel{T_W}} \to \{a:bool, b:bool, c:bool\} \quad \Gamma' \vdash \{a=5, b=6\}: \{a:int\}}{\Gamma' \vdash (fix\ f)\ \{a=5, b=6\}: \{a:bool, b:bool, c:bool\}}\ (SUB)$$

$$\frac{\Gamma' \vdash (fix\ f)\ \{a=5, b=6\}: \{a:bool, b:bool, c:bool\} \quad \{a:bool, b:bool, c:bool\} < \{a:bool, b:bool\}}{\Gamma' \vdash (fix\ f)\ \{a=5, b=6\}: \{a:bool, b:bool\}}$$

$$\frac{\Gamma' \vdash (fix\ f)\ \{a=5, b=6\}: \{a:bool, b:bool\} \quad \Gamma, g: \{a:bool, b:bool\} \vdash \#bg.bool}{\Gamma' \vdash (let\ g: \{a:bool, b:bool\} = (fix\ f)\{a=5, b=6\}\ in\ \#bg):bool}$$

$$\frac{\Gamma' \vdash (let\ g: \{a:bool, b:bool\} = (fix\ f)\{a=5, b=6\}\ in\ \#bg):bool \quad \Gamma' \vdash ....: bool}{\Gamma' \vdash (\ \ )\wedge(...): bool}\ (AND)$$

$$\frac{\Gamma'' \vdash tt:bool \qquad \Gamma'' \vdash ff:bool}{\Gamma'' = (\Gamma'', x.T_3) \vdash \{a=tt, b=ff, c=tt, d=tt\}: \{a:bool, ..., d:bool\}}$$

$$\frac{\Gamma'' = (\Gamma, y:T_2): fn\ x:T_3 \to \{a=tt, b=ff, c=tt, d=tt\}: T_x}{\Gamma \vdash fn\ y:T_2 \Rightarrow fn\ x.T_3 \Rightarrow \{a=tt, b=ff, c=tt, d=tt\}: T_1 = T_2 \to T_x}\ (fn)$$

$$\frac{\Gamma' = (\Gamma, f:T_1) \vdash (\ \ )\wedge(...): bool}{\Gamma \vdash let\ f:T_1 = fn\ y.T_2 \to fn\ x.T_3 \to \{a=tt, b=ff, c=tt, d=tt\}\ in\ ((let\ g:\{a:bool, b:bool\} = (fix\ f)\ \{a=5, b=6\}\ in\ \#bg)\wedge...):T}\ (let)$$

1 Sottotipaggio referenze covariante:

$$\frac{T_1 <: T_2}{\text{ref } T_1 <: \text{ref } T_2}\ (\text{sub-cov})$$

Ragionamento x assurdo

Esempio: ref $\{a:int, b:int\} <:$ ref $\{a:int\}$

Term = let x: ref $\{a:int, b:int\} = \{a=6, b=7\}$ in $x := \{a=5\}$; $(\#a!x)+1$

$$\frac{\dfrac{\dfrac{}{\Gamma \vdash \{a=5\}:\{a:int\}}}{\Gamma \vdash x := \{a=5\}: unit} \quad \dfrac{\Gamma \vdash x: \text{ref}\{a:int, b:int\} \quad \dfrac{}{\text{ref}\{a:int\} <: \text{ref}\{a:int, b:int\}}}{\Gamma \vdash x: \text{ref}\{a:int\}}\ (\text{sub})}$$

Il sottotipaggio covariante NON funziona

$$\cancel{\frac{T_1 <: T_2}{\text{ref } T_1 <: \text{ref } T_2}}\ (\text{sub-cov})$$

2 Sottotipaggio referenze covariante:

$$\frac{T_1 :> T_2}{\text{ref } T_1 <: \text{ref } T_2}\ (\text{sub-contr})$$

Esempio: ref $\{a:int\} <:$ ref $\{a:int, b:int\}$

Term = let x: ref $\{a:int\} = \{a=6\}$ in $x = \{a=7, b=8\}$; $(\#b!x)+1$

$$\cancel{\frac{T_1 :> T_2}{\text{ref } T_1 <: \text{ref } T_2}}\ (\text{sub-contr})$$

Il sottotipaggio controvariante non esiste

$$\frac{\Gamma \vdash x: \text{ref}\{a:int\} \quad \dfrac{}{\text{ref}\{a:int\} <: \text{ref}\{a:int, b:int\}}\ (\text{sub-contrav})}{\Gamma \vdash x. \text{ref}\{a:int, b:int\}}\ (\text{sub})$$

(sub-rec) taglio c

$\text{let } x:T_1 = \text{fn } z:T_2 \Rightarrow \text{fn } y:T_3 = \text{ref}\{b=\#b!y, c=1\} \text{ in}$

$\text{let } w:T_4 = ((\text{fix } x)(\text{ref}\{a=tt, b=5\})) \text{ in } (\text{fix } x)w$

(sub-rec taglio a)

$\overset{x}{\vdots}\ T_1 = \overset{z}{\vdots}\ T_2 \to \overset{y}{\vdots}\ T_3 \to \text{ref}\{b.T_B, c.T_C\}$ e se fosse $\overset{x}{\vdots}\ T_1 = \overset{z}{\vdots}\ T_2 \to \overset{y}{\vdots}\ T_3 \to \text{ref}\{b.T_B\}$

$T_2 = T_3 \to \text{ref}\{b.T_B, c.T_C\}$ (dal fatto che faccio fix x)

$T_3 = \text{ref}\{b:T_B, a:T_A\}$ e se fosse $T_3 = \text{ref}\{b:T_B\}$

$T_4 = T_3 = \text{ref}\{b:T_B, a:T_A\}$ e se fosse $T_3 = \text{ref}\{b:T_B\}$

Non posso fare subtyping di tipo ref ma posso fare questo → Si intende, non posso sottotipare gli oggetti di tipo ref, ma posso sottotipare le sue sottocomponenti

$$\boxed{\dfrac{\dfrac{\Gamma \vdash e:T_1 \quad T_1 <: T_2}{\Gamma \vdash e:T_2}\,(sub)}{\Gamma \vdash \text{ref } e : \text{ref } T_2}\,ref}$$

$T_1 = T_2 \to T_3 \to \text{ref}\{b:T_B\}$

$T_3 = \text{ref}\{b:T_B\}$

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\Gamma'(y) = \text{ref}\{b:T_B, \dots\} \neq T_3}{\Gamma' y = \text{ref}\{b:T_B \dots\}}}{\Gamma' \vdash !y : \{b:T_B \dots\}}}{\Gamma' \vdash \#b!y : T_B} \quad \dfrac{-}{\Gamma' \vdash 1 : int}}{\Gamma' \vdash \{b=\#b!y, c=1\} : \{b:T_B, c:int\}} \quad \{b:T_B, c:int\} <: \{\}\,(sub)}{\Gamma' \vdash \{b=\#b!y, c=1\} = TREF = \{\}'}$$

ricordiamoci T_3

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\Gamma''(y) = T_3 = \text{ref}\{b:T_B \dots,\}}{\Gamma'' \vdash y : \text{ref}\{b:T_B, \dots\}}\,(var)}{\Gamma'' \vdash !y\{b:T_B \dots\}}\,(def)}{\Gamma'' \vdash \#b!y : T_B}\,(prog)}{\vdots}}{\Gamma'' \vdash \#b!y : T_B} \quad \triangle_{\Gamma'' \vdash 1 : int}}{\Gamma'' \vdash \{b=\#b!y, c=1\} : \{b:T_B, c:int\}} \quad \dfrac{\triangle}{\{b:T_B, c:int\} <: \{b:T_B\}}\,(sub\text{-}rec)}{\Gamma'' \vdash \{b=\#b!y, c=1\} : \{b:T_B\}}\,(sub)}{(\Gamma', y:T_3)^{\leq \Gamma''} \vdash \text{ref}\{b=\#b!y, c=1\} : T_{REF} = \text{ref}\{b:T_B\}}\,(ref)}{(\Gamma, z:T_2)^{\leq \Gamma'} \vdash \text{fn } y:T_3 \to \text{ref}\{b=\#b!y, c=1\} = T_y}\,(fun)\ T_3 \to T_{REF}}{\Gamma \vdash \text{fn } z:T_2 \to \text{fn } y:T_3 \to \text{ref}\{b=\#b!y, c=1\} : T_1}\,(fun)\ T_2 \to T_y$$

$T_2 \to T_3 \to \text{ref}\{b:T_B\}$

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\Gamma'(x) = T_1 = (T_A \to T_4) \to (T_A \to T_4)}{\Gamma' \vdash x : (T_A \to T_4) \to (T_A \to T_4)}}{\Gamma' \vdash (\text{fix } x) : T_A \to T_4}}{\Gamma' \vdash (\text{fix } x)\text{ref}\{a=tt, b=5\} : T_4} \quad \dfrac{\dfrac{\dfrac{\triangle_{\{a:bool, b:int\} <: \{b:int\}}}{\Gamma' \vdash \{a=tt, b=5\} : \{b:T_B^{int}\}}}{\Gamma' \vdash \text{ref}\{a=tt, b=5\} = T_A = \text{ref}\{b:T_B^{int}\}}}{\Gamma', w:T_4 \vdash (\text{fix } w) : T}}{\Gamma'' = (\Gamma, x, T_1) \vdash \text{let } w:T_4 = (\text{fix } x)(\text{ref}\{a=tt, b=5\}) \text{ in } (\text{fix } x\ w) : T}\,(let)$$

$\Gamma \vdash \text{let } x.T_1 = \text{funzione in let } w:T_4 = (\text{fix } x)(\dots) \text{ in } (\text{fix})w : T$

Esercitazione 30/11

ⓐ $\partial_;(\mathrm{skip};E_1 \| \mathrm{skip};E_2) \overset{?}{\approx} \mathrm{skip};(a,E_1 \| \alpha;E_2)$

$$\begin{array}{ccc}
\downarrow * & & \downarrow \mathrm{skip} \ ① \\
(\mathrm{skip},E_1 \| \mathrm{skip},E_2) & \overset{?}{\not\approx} & a,E_1 \| a,E_2 \\
\downarrow \alpha & & \downarrow \alpha \\
\mathrm{skip},E_1 \| \mathrm{skip},E_2 & \overset{?}{\not\approx} & E_1 \| \alpha,E_2 \\
\downarrow \mathrm{skip} & & \downarrow \ell=1 \\
\ell=1 & & \\
\mathrm{skip} \| \mathrm{skip},E_2 & \overset{?}{\not\approx} & \mathrm{skip} \| \alpha,E_2 \\
\not\downarrow \alpha & & \downarrow \\
 & & \alpha \equiv \ell:=5
\end{array}$$

Controesempio

$\alpha = \ell:=5$

$E_1 \equiv \ell:=1;\mathrm{skip}$

$E_2 \equiv \mathrm{skip}$

$$P \approx Q \overset{\text{non vale in questo verso}}{\not\Leftarrow} \Rightarrow \begin{cases} P \sqsubseteq Q \\ \wedge \\ P \sqsupseteq Q \end{cases}$$

$$\underbrace{\partial_;(\mathrm{skip};E_1 \| \mathrm{skip};E_2)}_{P} \overset{?}{\sqsubseteq} \underbrace{\mathrm{skip};(a,E_1 \| \alpha;E_2)}_{Q} \qquad \forall \alpha\ E_1,E_2$$

$\mathcal{R} = \{(P,Q),(P',Q')\} \cup Id$

① Analizziamo la 1^a coppia

1.1 $P \xrightarrow{\alpha} (\mathrm{skip},E_1 \| \mathrm{skip},E_2) \equiv P'$ allora $\exists Q'$ $Q \xrightarrow{\alpha} E_1 \| E_2 = Q'$, purché $(P',Q') \in \mathcal{R}$

② Analizziamo la 2^a coppia

2.1 Se $P' \xrightarrow{\mathrm{skip}} E_1 \| \mathrm{skip},E_2 \equiv P''$, $\exists Q'$, t.c $Q' \xrightarrow{\alpha} Q''$ e $(P'',Q') \in \mathcal{R}$

2.2 Se $P' \xrightarrow{\mathrm{skip}} \mathrm{skip},E_1 \| E_2 \equiv P'''$ $\exists Q'$, t.c $Q' \xrightarrow{\alpha} Q''$ e $(P''',Q) \in \mathcal{R}$

③ Analizziamo la 3^a coppia

3.1 Sia $P' \equiv E_1 \| \mathrm{skip},E_2 \xrightarrow{\mathrm{skip}} E_1 \| E_2 \equiv Q'$, $\exists Q' \xrightarrow{\alpha} Q'$, $(Q',Q') \in Id \subseteq \mathcal{R}$

3.2 Sia $P'' \equiv E_1 \| \mathrm{skip},E_2 \to E_1' \| \mathrm{skip} = P''$ allora $\exists Q' \equiv E_1 \| E_2 \to E_1' \| E_2 = Q'$.
con $(P'',Q') \in \mathcal{R}$

ⓑ $\overbrace{\text{await true } (\ ): \ell:=1 \ \underline{\text{end}}}^{P} \simeq \overbrace{(\text{await } \ell=0 \quad) \| (\text{await } \ell=0 \quad)}^{Q}$

$$\begin{array}{ccccc}
\forall & \sqsubseteq & \exists & & P \sqsubseteq Q \ \exists \\
\downarrow & & \downarrow * & \searrow \forall & \downarrow \quad \downarrow * \\
 & & & & P' \sqsubseteq Q'
\end{array}$$

$\mathcal{R} = \{(P,Q) \ldots\ldots\} \cup Id$

1.1 Analizziamo la 1^a coppia da sx us dx

1.1.1) La $P \xrightarrow{\alpha E_1 \ell=1} P' \equiv \mathrm{skip}$ allora $\exists Q'$ t.c $Q \xrightarrow{\alpha E_1 \ell} \mathrm{skip} \| (\text{await } \ell=0 \ldots) \equiv Q'$, purché

$(P',Q') \in \mathcal{R}$

1.1.2) La $P \xrightarrow{\alpha E_1 \ell=1} P'' \equiv \mathrm{skip}$, allora $\exists Q''$ t.c $Q \xrightarrow{\beta E_2 \ell=1} Q'' \equiv \text{await } \ell=\emptyset \ldots\ldots)$ purché
$(\mathrm{skip},Q'') \in \mathcal{R}$

1.2

1.2.1) La $Q \xrightarrow{\alpha E_2 \ell=1} \mathrm{skip} \| \text{await} \equiv Q'$ allora $\exists P \xrightarrow{\alpha E_1 \ell=1} \mathrm{skip}$ con $(\mathrm{skip},Q') \in \mathcal{R}$
1.2.2)

8 febbraio 2021

$$\mathcal{R} = \{ \underset{①}{(P,Q)}, \underset{\alpha E_1 \ell=1}{\overset{②}{(skip, Q')}}, \underset{\beta E_1 \ell=1}{\overset{③}{(skip, Q'')}}, \underset{④}{(skip, Q''')} \alpha E_1 \ell=1 \}$$

2.1) Analizziamo la 2ª coppia da sx verso dx

Il processo da sx non ha nessuna azione da simulare poiché $\equiv$ skip

2.2) Analizziamo la 2ª coppia da dx verso sx

La $Q' \equiv skip \parallel (await\ \ell=0)$

2.2.1) La $Q' \xrightarrow{skip} Q'' \equiv$ await allora $skip \xrightarrow{*} skip$, con $(skip, Q'') \in \mathcal{R}$

3.1) Analizziamo la 3ª coppia da sx verso dx

Non vi sono azioni da analizzare

3.2) Analizzando la 3ª coppia da dx vs sx

4.1) Analizziamo la 4ª coppia da sx verso dx

No azioni

4.2) Analizziamo la 4ª coppia da dx verso sx

4.2.1) La $Q''' = (await\ \ell=0 \ldots)$

Essendo $\ell=1$, l'await ha guardia falsa e non ci sono errori da simulare

$$\begin{tikzcd}
P \arrow[d, "\alpha", "④"'] \arrow[r, dashed, "?", "\not\approx"'] & Q \arrow[d, "\alpha", "skip"'] \arrow[d, shift left=6, "②"] \\
\ell=1 \quad E_1 \arrow[d, "*"] \arrow[r, dashed, "?", "\not\approx"'] & E_1 + skip; E_2 \arrow[d, "skip", "③"'] \\
E_1 \arrow[d, "\not\downarrow"'] \arrow[r, dashed, "\not\approx"] & E_2 \arrow[d, "m=1", "⑤"'] \\
 & \text{se non lo sono qui non lo erano neanche prima}
\end{tikzcd}$$

$$(\text{let con})\ \frac{E_1 \Downarrow v_1 \qquad E_2\{^{v}/_{x}\} \Downarrow v}{\text{let } x = E_1 \text{ in } E_2 \Downarrow v}$$

$$(\text{let}_1)\ \frac{E_1 \to E_1'}{\text{let } x = E_1 \text{ in } E_2 \to \text{let } x = E_1' \text{ in } E_2}$$

$$(\text{let}_2)\ \frac{-}{\text{let } x = v \text{ in } E_2 \to E_2\{^{x}/_{v}\}}$$

Teorema

Per $E \in LANG$ se $E \Downarrow v$ then $E \to^* v$

Posso fare induzione strutturale
Posso fare rule induction

Proof

By structural induction con E

Caso Base 1

$E \equiv n$. Ho allora $E \equiv n \Downarrow n$ e $E \equiv n \to^* n$

per la B-num

essendo che non abbiamo regole

Caso Base 2

$E \equiv x$ non si applica perché $\nexists\, v$ t.c $E \equiv x \Downarrow v$

Il teorema è vero perché l'ipotesi è falsa. $\nexists\, v$ t.c $x \Downarrow v$

↳ La big step è definita solo su termini chiusi, x è solo un place holder

Caso induttivo 1

$E \equiv E_1 + E_2$. Ne consegue che $E \Downarrow v$ è stato ottenuto applicando la (B-Add)

$$\frac{E_1 \Downarrow v_1 \qquad E_2 \Downarrow v_2}{E \equiv E_1 + E_2 \Downarrow v}$$

con $v = add(v_1, v_2)$ Ho allora per (IH) ho che $E_1 \to^* v_1$ e $E_2 \to^* v_2$. Quindi applicando la S-Right

applicando la S-right

$E \equiv E_1 + E_2 \to^* v_1 + E_2 \to^* v_1 + v_2 \to v$ Cioè $E \to^* v$ come volevasi dimostrare

applicando la S-left

$v = add(v_1, v_2)$

Caso induttivo 2

$E \equiv \text{let } x = E_1 \text{ in } E_2$, per qualche E_1 ed E_2. Ho allora ho applicato la regola (B-let)

$$\frac{E_1 \Downarrow v_1 \qquad E_2\{^{v_1}/_{x}\} \Downarrow v}{\text{let } x = E_1 \text{ in } E_2 \Downarrow v}$$

Per IH da $E_1 \Downarrow v_1$ deriva che $E_1 \to^* v_1$ Per IH da $E_2\{^{v_1}/_{x}\}$ deriva cosa?

↳ In generale NON è una sottostruttura della struttura iniziale

non posso provarla con la structure induction

↳ la structure induction è sintattica

$E_1 = E_1 + E_2 \to^* v_1 + E_2 \to^* ?$

left

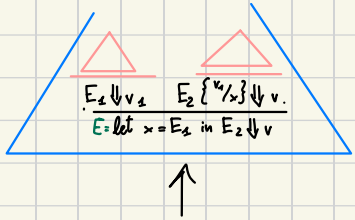
Teorema Per $E \in Lang$, if $E \Downarrow v$ then $E \to^* v$

Proof (rule induction)

Per rule induction su come ho derivato $E \Downarrow v$

CASO BASE Sia $E \Downarrow v$ dell'applicazione della (B-NUM). Ne deriva che $E \equiv n$, per qualche n
Quindi $E \equiv n \to^* n$

CASO INDUTTIVO 1 Sia $E \Downarrow v$ derivato applicando le B-let: $\dfrac{E_1 \Downarrow v_1 \quad E_2\{^{v_1}/_x\} \Downarrow v}{let\ x = E_1\ in\ E_2 \Downarrow v}$. Ma allora guardando la profondità degli alberi di derivazione, per (IH) abbiamo $\begin{cases} E_1 \to^* v_1 \\ E_2\{^{v_1}/_x\} \to^* v_2 \end{cases}$



Schema
$\triangle \to$ è un albero di derivazione più piccolo

Ma allora

$$E = let\ x = E_1\ in\ E_2 \xrightarrow{let\ 1}{}^* let\ x = v_1\ in\ E_2 \xrightarrow{let\ 2} E_2\{^{v_1}/_x\} \xrightarrow{per\ (IH)}{}^* v.$$

Come volevasi dimostrare

$$\alpha;(E_1 + skip;E_2) + \alpha;E_1 \approx \alpha;(skip;(E_1+skip;E_2)) + \alpha;E_2$$

$$(1)\ \alpha \downarrow \qquad\qquad (2)$$

$$E_1 \qquad\qquad E_1 + skip;E_2$$

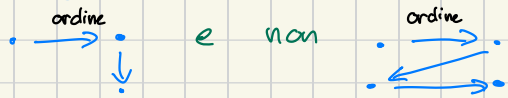
$$\downarrow skip\ (3)$$

$$E_2$$

$E_1 \equiv l := 5$

$E_2 \equiv m := 7$

C'è stato un cambio di turno → si è rotto



non cambiano il programma

$$\underbrace{\alpha;(skip;E_1 + \alpha;E_2) + a;E_2}_{P} \approx \underbrace{a;(a;E_1 + \alpha;E_2) + \alpha;a;E_1}_{Q}$$

skip

Proviamo che la relazione $\mathcal{R}$ è una bisimulazione $\quad \mathcal{R} = \{(P,Q),(P',Q'),(E_2,\alpha;E_2)_{\alpha}\dots\} \cup Id$

1.a) Analizziamo la 1ª coppia da sx verso dx

1.a.1) Sia $P \xrightarrow{\alpha} E_2$ ma allora $\exists\, Q \xrightarrow{a}\xrightarrow{skip} E_2$ con $(E_2, E_2) \in Id \subseteq \mathcal{R}$

1.a.2) Sia $P \xrightarrow{\alpha} (skip;E_1 + \alpha;E_2) \equiv P'$, ma allora $\exists Q'$ t.c $Q \xrightarrow{\alpha} (\alpha;E_1 + skip;E_2) \equiv Q'$, con $(P',Q') \in \mathcal{R}$

1.b) Analizziamo la 1ª coppia da dx verso sx

1.b.1) Sia $Q \xrightarrow{\alpha} Q'$, ma allora $\exists P'$ t.c $P \xrightarrow{\alpha} P'$, con $(P',Q') \in \mathcal{R}$

1.b.2) Sia $Q \xrightarrow{a} a;E_2$, ma allora $\exists P \xrightarrow{a}\xrightarrow{skip} E_1$, con $(E_1, \alpha;E_1)_{\alpha} \in \mathcal{R}$

2.a) Analizziamo la 2ª coppia da sx verso dx

2.a.1) Sia $P' \rightarrow skip_3 E_1$, ma allora, poiché ho già fatto α, $\exists Q' \overset{\alpha}{\Rightarrow} E_1$ con $(E_1, E_1) \in Id \subseteq R$

2.a.2) Sia $P' \rightarrow E_2$, ma allora poiché ho già fatto α, $\exists Q' \overset{skip}{\rightarrow} E_2$, con $(E_2, E_2) \in Id \subseteq R$

2.b) Analizziamo la 2ª coppia da dx verso sx

2.b.1) Sia $Q' \underset{skip}{\rightarrow} E_1$, " " " " , $\exists P' \overset{skip}{\rightarrow} E_1$, con $(E_1, E_1) \in Id \subseteq R$

2.b.2) Sia $Q' \underset{skip}{\rightarrow} E_2$, " " " , $\exists P' \overset{\alpha}{\rightarrow} E_2$, con $(E_2, E_2) \in Id \subseteq R$

$s = \{ \ell \mapsto 3, m \mapsto 2$

$e \equiv m = !m - 2 \; ; \; \ell = !m + 2$

1) $B \equiv \text{ReqSeq}(e) \equiv e ; e ; e ; e ; \ldots\ldots$

Sia $B \xrightarrow{*} B'$ allora m e ℓ sono in questa relazione

1) ℓ e m sono monotone non crescenti
2) $\ell - m$ o vale 2 oppure 4. (oppure nello stato iniziale $\ell = 3, m = 2$)

2) $C \equiv \text{RepAwa}(e) = \text{await}(e) \,||\, \text{await}(e) \,||\, \text{await}(e) \ldots\ldots\ldots\ldots$

Sia $C \xrightarrow{*} C'$ allora m e ℓ sono così caratterizzati

1) ℓ e m sono monotone non-crescenti
2) $\ell - m = 2$, sempre a parte lo stato iniziale

1) $A \equiv \text{RepPar}(e) \equiv e \,||\, e \,||\, e \ldots\ldots$

1) ℓ e m non hanno comportamento monotono
2) $\ell > m$? NO, $\ell < m$? NO
3) ℓ e m sono sempre pari a regime

Quale vale?

$B \sqsubseteq C$	✗
$C \sqsubseteq B$	✓

Quale vale?

1	$A \sqsubseteq B$	→ Falso
2	$B \sqsubseteq A$	← Vero
3	$A \sqsubseteq C$	→ Falso
4	$C \sqsubseteq A$	← Vero

Vere quelle in cui A è il simulante "perché ha più gioco"

Esercizio 3b compito 4-feb-2018

$(\text{await true protect } (E_1 \| E_2) \text{ end}) \approx_r (\text{await true protect } E_2 \text{ end}) \| (\text{await true protect } E_1 \text{ end})$

$\text{await } (P \| Q) \not\equiv \text{await } P \| \text{ await } Q$

devo eseguire tutto $P \| Q$ o stare fermo perciò non ho una risposta adeguata

ho sia $l=5$ che $m=7$

$\xrightarrow{l:=5}$ $\text{skip} \| \text{await } Q$

$P \equiv l:=5$

$Q \equiv m:=7$

$\text{await } (P \| Q) \equiv \text{await } P \| \text{ await } Q$

$s=5$
$l=6$ skip

devo eseguire per forza tutto P o tutto Q e non posso procedere per accedermi all'altro stato

faccio $l=5;$ $\to$ $s:=!l+1$ $\to$ $l=6$

$P \equiv l=5; l=6$

$Q \equiv s:=!l$

Esercizio 3A

$a_j(\beta_j E_1 \| \beta_j E_2) \approx_r skip_j\, a_j\, \beta_j (E_1 \| E_2)$

non è più un incognita su questa strada

$$
\begin{tikzcd}
P \arrow[d, "2"'] & \cancel{?}\ \exists & Q \arrow[d, "skip\ 1"] \\
P \arrow[d, "4"', "\alpha"] & \cancel{?}\ \exists & Q' \arrow[d, "\alpha\ (3)"] \\
P' \arrow[d, "\beta"] & \cancel{?}\ \exists & Q'' \arrow[d, "\beta\ (5)"] \\
E_1 \| E_2 & Id & Q'' \arrow[d] \\
 & & E_1 \| E_2
\end{tikzcd}
$$

$\mathcal{R} = \{(Q,P),(Q',P),(Q'',P')\} \cup Id$

no fatto skip

no fatto α

a: $m=5$

b: $m=7$

α $l=5$

b $m=7$

E_1 $m=6$

E_2 $m=4$

$P \not\approx Q$

$$
\begin{tikzcd}
 & P \arrow[d, "\alpha"'] & \overset{?}{\exists} & Q \arrow[d, "\alpha"] & \\
l=5 & P' \arrow[d, "\beta"'] & \overset{?}{\exists} & Q' \arrow[d, "\beta"] & l=5 \\
l=5, m=7 & E_1 \| \beta E_2 \arrow[d, "E_1"'] & & E_1 \| E_2 \arrow[d, "E_1"'] & l=5, m=7 \\
l=5, m=6 & skip \| \beta E_2 \arrow[d, "\beta"'] & & skip \| E_2 \ \cancel{\downarrow} & l=5, m=6 \\
l=5, m=7 & & & & l=5, m=\ \bigcirc
\end{tikzcd}
$$

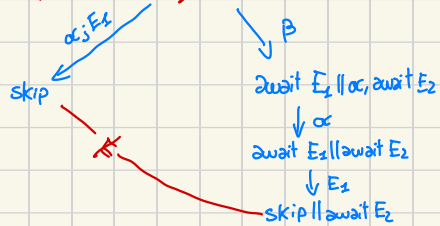
non può diventare 7

Esame 30 gennaio 2020

3b

$$(\text{await true protect}(\alpha ; E_1 \oplus \beta ; E_2)\ \text{end}) \approx_r (\beta ; \text{await true protect } E_1 \text{ end}) \parallel (\alpha ; \text{await true protect } E_2 \text{ end})$$

Q simula P? $\quad P \overset{?}{\not\approx} Q$



se α e β scrivono su locazioni diverse non vale perché ho una locazione in più su Q

se α e β usano sulla stessa locazione

La simulazione richiede che si vada nello stesso stato avendo una locazione in più NON sono nello stesso stato

nel caso in cui α e β lavorino sulla stessa locazione

$$\mathcal{R} = \{(P, Q), (\text{skip}, \text{skip} \parallel \text{await}(E_2))_{\alpha E_1}, (\text{skip}, \text{await}(E_1) \parallel \text{skip})_{\alpha E_2}\}$$

Inizio es da solo

T_1

fun prende in input T_2

che va in funzione che prende in input T_3 che va in ref

T_1 : $T_2 \to (T_3 \to ref)$

qui ho fix quindi per forza T_2 perche ho fix point dato in output ciò che prende in input

T_2 : $T_3 \to ref$

a: int b: bool c: int

T_3: ref {b: int}

T_1 =

T_2 =

T_3 =

ref {b: int} → ref {a: int, b: bool, c: int} → ref {b: int} → ref {a: int, b: bool, c: int}

$$
\dfrac{
\dfrac{
\dfrac{
\dfrac{
\dfrac{\Gamma'(h) = T_3 = \text{ref}\{b:int\}}{\Gamma' \vdash h : \text{ref}\{T_2\}\ b:int}
}{\Gamma' \vdash !h : T_2\{b:int\}}
\quad
\dfrac{}{\Gamma' \vdash b : bool}
\quad
\dfrac{
\dfrac{\Gamma'(h) = T_3 = \text{ref}\{b:int\}}{\Gamma' \vdash h : \text{ref}\{T_2\}\ b:int}\ (var)
}{\Gamma' \vdash !h : T_2\{b:int\}}\ (rec\ proj)
}{\Gamma' \vdash (\#b!h) : T_x\ int \quad \Gamma' \vdash b : bool \quad \Gamma' \vdash (\#b!h) : T_x\ int}
}{\Gamma'', h:T_3 \vdash \text{ref}\{a = (\#b!h), b = true, c = \#b!h\} : \text{ref}\{a:int,\ b:bool,\ c:int\}}
}{\Gamma, g:T_2, \text{fun}(h:T_2) \Rightarrow \text{fun}(h:T_3) \Rightarrow \text{ref}\{a = (\#b!h), b = true, c = \#b!h\} : T_c - (T_3 \to T_{ref})}\ (fun)
}{\Gamma \vdash \text{fun}(g:T_2) \Rightarrow \text{fun}(h:T_3) \Rightarrow \text{ref}\{a = (\#b!h), b = true, c = (\#b!h)\} : T_1}
$$

ref {b: int}

{a: int, b: bool, c: int}

passo questo

$$
\dfrac{
\dfrac{
\dfrac{
\dfrac{\Gamma'' \vdash !y : T_L\ \{b:bool\}}{\Gamma'' \vdash (\#b!y) : bool}\ (rec\ proj)
}{\Gamma', y:\text{ref}\{b:bool\} \vdash \neg(\#b!y) : bool}\ not
}{\Gamma' \vdash (\text{fun}\ y:\text{ref}\{b:bool\} \Rightarrow \neg(\#b!y))}\ fn
\quad
\dfrac{
\dfrac{
\dfrac{\Gamma \vdash \text{fix}\ f :}{\Gamma \vdash ((\text{fix}\ f)(\text{ref}\{a=7, b=2\})) : \text{ref}\ T_B}\ app
}{\Gamma \vdash !((\text{fix}\ f)(\text{ref}\{a=7, b=2\})) : T_B}\ deref
\quad T_B <: \{b:bool\}
}{\Gamma \vdash !((\text{fix}\ f)(\text{ref}\{a=7, b=2\})) : \{b:bool\}}\ sub
}{\Gamma \vdash !((\text{fix}\ f)(\text{ref}\{a=7, b=2\})) : \text{ref}\{b:bool\}}\ ref
}{\Gamma \vdash (\text{fun}\ y:\text{ref}\{b:bool\} \Rightarrow \neg(\#b!y))(\text{ref}\ !((\text{fix}\ f)(\text{ref}\{a=7, b=2\})) : bool}\ app
$$

$$
\dfrac{
\dfrac{
\dfrac{
\dfrac{
\dfrac{\Gamma \vdash (\text{fun}\ y:\text{ref}\{a:int\} \Rightarrow \text{ref}\{a:int\} \Rightarrow (\#a!y) > 0)(\text{ref}\ !((\text{fix}\ f))(\text{ref}\{a=false, b=6, c=7\})) : bool}{\Gamma \vdash (\text{fun}\ y:\text{ref}\{a:int\} \Rightarrow \text{ref}\{a:int\} \Rightarrow (\#a!y) > 0)(\text{ref}\ !((\text{fix}\ f))(\text{ref}\{a=false, b=6, c=7\})) : bool}\ (\wedge)
}{\Gamma \vdash (\text{fun}\ y:\text{ref}\{b:bool\} \Rightarrow \neg(\#b!y))(\text{ref}\ !((\text{fix}\ f)(\text{ref}\{a=7, b=2\})) \wedge (\text{fun}\ y:\text{ref}\{a:int\} \Rightarrow \text{ref}\{a:int\} \Rightarrow (\#a!y) > 0)(\text{ref}\ !((\text{fix}\ f))(\text{ref}\{a=false, b=6, c=7\})) : bool}\ (let)
}{\Gamma \vdash \text{let}\ f:T_1 = \text{fun}(g:T_2) \Rightarrow \text{fun}(h:T_3) \Rightarrow \text{ref}\{a = (\#b!h), b = true, c = (\#b!h)\}\ \text{in}\ (\text{fun}\ y:\text{ref}\{b:bool\} \Rightarrow \neg(\#b!y))(\text{ref}\ !((\text{fix}\ f)(\text{ref}\{a=7, b=2\})) \wedge (\text{fun}\ y:\text{ref}\{a:int\} \Rightarrow \text{ref}\{a:int\} \Rightarrow (\#a!y) > 0)(\text{ref}\ !((\text{fix}\ f))(\text{ref}\{a=false, b=6, c=7\}))
}
$$

c: int perche lo è a in input y

mi si fix viene deref e poi ref

sicuramente T_1 adesso ricorsivo perche f è la funzione di tipo T_1

entrambi applicati a fix di f l'unico in accordo è b: int, gli altri vengono utilizzati

$$\mathsf{fun}(f : T_1) \Rightarrow$$
$$(\mathsf{fn}\ y : \mathsf{ref}\{b : \mathsf{int}\} \Rightarrow \#b\,y > 0)(\mathsf{ref}\ ((\mathsf{fix}.f)(\mathsf{ref}\ \{a = \mathsf{true}, b = \mathsf{false}\})))$$
$$\wedge$$
$$(\mathsf{fn}\ z : \mathsf{ref}\{a : \mathsf{bool}\} \Rightarrow \#a\,z)(\mathsf{ref}\ ((\mathsf{fix}.f)(\mathsf{ref}\ \{a = \mathsf{false}, b = 6, c = 7\})))$$

$$T_1 \equiv \mathsf{ref}\{a{:}\mathsf{bool}\} \to \{a{:}\mathsf{bool}, b{:}\mathsf{int}\} \to \mathsf{ref}\{a{:}\mathsf{bool}\} \to \{a{:}\mathsf{bool}, b{:}\mathsf{int}\}$$

$$\wedge$$

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\Gamma'(y) = \{b{:}\mathrm{int}\}}{\Gamma', y{:}\mathsf{ref}\{b{:}\mathrm{int}\} \vdash y : \{b{:}\mathrm{int}\}}\ \mathsf{recproj} \quad \dfrac{-}{\Gamma', y{:}\mathsf{ref}\{b{:}\mathrm{int}\} \vdash 0 : \mathrm{int}}}{\Gamma', y{:}\mathsf{ref}\{b{:}\mathrm{int}\} \vdash \#b\,y : \mathrm{int}}\ \mathsf{op>}}{\Gamma', y{:}\mathsf{ref}\{b{:}\mathrm{int}\} \vdash \#b\,y > 0 : T_R\ \mathrm{bool}}}{\Gamma', f, T_1 \vdash \mathsf{fn}\ y{:}\mathsf{ref}\{b{:}\mathrm{int}\} \Rightarrow \#b\,y > 0 : T_R \to T_R\ \mathrm{bool}}\ \mathsf{fn} \quad \dfrac{\dfrac{\Gamma, f{:}T_1 \vdash f : (T_1 \to \mathrm{bool}) \to (T_1 \to \mathrm{bool})}{\Gamma', f, T_1 \vdash \mathsf{fix}\ f : (T_1 \to \mathrm{bool})} \quad \dfrac{\dfrac{\Gamma \vdash \{a = \mathrm{true}\ b = \mathrm{false}\} : \{a{:}\mathrm{bool}, b{:}\mathrm{bool}\} \quad \{a{:}\mathrm{bool}, b{:}\mathrm{bool}\} < \{a{:}\mathrm{bool}\}}{\Gamma' \vdash \{a = \mathrm{true}, b = \mathrm{false}\} : \{a{:}\mathrm{bool}\}}\ \mathsf{sub}}{\Gamma' \vdash (\mathsf{ref}\{a = \mathrm{true}, b = \mathrm{false}\}) : T_1}}{\Gamma', f, T_1 \vdash (\mathsf{fix}\ f)(\mathsf{ref}\{a = \mathrm{true}, b = \mathrm{false}\}) : T_L}\ \mathsf{app}}{\Gamma', f, T_1 \vdash \mathsf{ref}\ ((\mathsf{fix}\ f)(\mathsf{ref}\{a = \mathrm{true}, b = \mathrm{false}\})) : T_R\ \mathsf{ref}\ T_L}\ \mathsf{ref}}{\Gamma', f, T_1 \vdash (\mathsf{fn}\ y{:}\mathsf{ref}\{b{:}\mathrm{int}\} \Rightarrow \#b\,y > 0)(\mathsf{ref}\ ((\mathsf{fix}\ f)(\mathsf{ref}\{a = \mathrm{true}, b = \mathrm{false}\}))) : T_R \to T_R\ \mathrm{bool}}\ \mathsf{app} \quad \Gamma, f, T_1 \vdash (\mathsf{fn}\ z{:}\mathsf{ref}\{a{:}\mathrm{bool}\} \Rightarrow \#a\,z)(\mathsf{ref}\ ((\mathsf{fix}\ f)(\mathsf{ref}\{a = \mathrm{false}, b = 6, c = 7\}))) : T_S \to T_K\ \mathrm{bool?}}{\Gamma, f{:}T_1 \vdash (\mathsf{fn}\ y{:}\mathsf{ref}\{b{:}\mathrm{int}\} \Rightarrow \#b\,y > 0)(\mathsf{ref}\ ((\mathsf{fix}\ f)(\mathsf{ref}\{a = \mathrm{true}, b = \mathrm{false}\}))) \wedge (\mathsf{fn}\ z{:}\mathsf{ref}\{a{:}\mathrm{bool}\} \Rightarrow \#a\,z)(\mathsf{ref}\ ((\mathsf{fix}\ f)(\mathsf{ref}\{a = \mathrm{false}, b = 6, c = 7\}))) : T'\ \mathrm{bool}}\ (\mathsf{and})$$

$$\Gamma \vdash \mathsf{fun}(f : T_1) \Rightarrow (\mathsf{fn}\ y{:}\mathsf{ref}\{b{:}\mathrm{int}\} \Rightarrow \#b\,y > 0)(\mathsf{ref}\ ((\mathsf{fix}\ f)(\mathsf{ref}\{a = \mathrm{true}, b = \mathrm{false}\}))) \wedge (\mathsf{fn}\ z{:}\mathsf{ref}\{a{:}\mathrm{bool}\} \Rightarrow \#a\,z)(\mathsf{ref}\ ((\mathsf{fix}\ f)(\mathsf{ref}\{a = \mathrm{false}, b = 6, c = 7\}))) : T_1 \to T'\ \mathrm{bool}$$

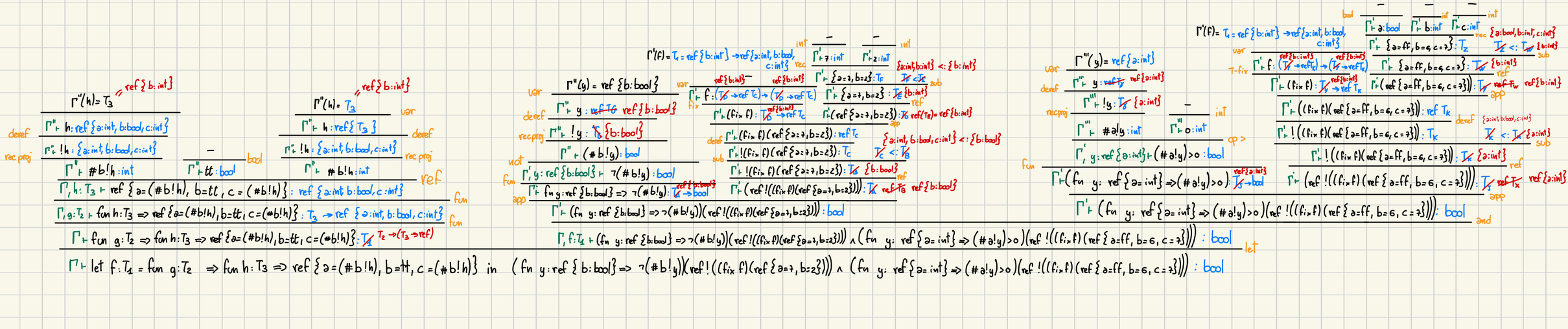
27/12/23 pomeriggio

$T_1 : T_2 \to T_3 \to \text{ref}\{a{:}int, b{:}bool, c{:}int\}$

$T_2 : T_3 \to \text{ref}\{a{:}int, b{:}bool, c{:}int\}$

$T_3 : \text{ref}\{b{:}int\}$

$T_K = T_C = \{a{:}int, b{:}bool, c{:}int\}$



Primo es risolto da solo

28/12/23 mattina

$T_1 = ref(T_2 \to \{a:bool, b:bool, c:bool, d:bool\})$

$T_2 = \{a:int\}$

$T_x = \{a:bool, b:bool, c:bool, d:bool\}$

$$\dfrac{\dfrac{\dfrac{

$T_1 = T_2 \to T_3 \to \text{ref}\, T_3$

$T_2 = \{\underbrace{a: \text{bool}}_{\text{z prende solo a:bool}}, b: \text{int}\}$

$T_4 = \{a: \text{bool}\}$

T_3

$T_2 \to T_3 \to \text{ref}\, T_3$

$\{a: \text{bool}, b: \text{int}\} \to T_4 \to \{a: \text{bool}, b: \text{int}\} \to T_4$

$T_B = \{a: \text{bool}, b: \text{int}\}$

$T_C = T_4$

$$
\dfrac{
\dfrac{
\dfrac{
\dfrac{
\dfrac{
\dfrac{\Gamma''(y) = T_3}{\Gamma'' \vdash y : \cancel{T_A}\ T_3}
}{\Gamma'' = \Gamma', y: T_3 \vdash \text{ref}\, y : \cancel{\text{ref}\, T_A}\ T_3}\ \text{ref}
}{\Gamma' = \Gamma, x: T_2 \vdash \text{fun}(y: T_3) \Rightarrow \text{ref}\, y : T_3 \to \cancel{\text{ref}\, T_A}\ T_3}\ \text{fun}
}{\Gamma \vdash \text{fun}(z: T_2) \Rightarrow \text{fun}(y: T_3) \Rightarrow \text{ref}\, y : T_2 \to T_3 \to \text{ref}\, T_3}\ \text{fun}
\quad
\dfrac{
\dfrac{
\dfrac{
\dfrac{\Gamma'''(x) = T_1}{\Gamma''' \vdash x : (T_B \to T_4) \to (T_B \to T_4)}\ \text{var}
\quad
\dfrac{\Gamma''' \vdash \text{true} : \text{bool} \quad \Gamma''' \vdash 5 : \text{int}}{\Gamma''' \vdash \{a = \text{true}, b = 5\} : T_B}\ \text{record}
}{\Gamma''' \vdash (\text{fix}\ x) : T_B \to T_4}\ \text{T-fix}
\quad
\dfrac{
\dfrac{
\dfrac{
\dfrac{\Gamma^{V}}{\Gamma^{IV} \vdash z : \text{ref}\{a: \text{bool}\} \vdash (\text{fix}\ x)\{a = \text{false}, b = 6, c = 7\} : T_R}\ \text{app}
}{\Gamma^{IV} \vdash (\text{fun}\ z: \text{ref}\{a: \text{bool}\} \Rightarrow (\text{fix}\ x)\{a = \text{false}, b = 6, c = 7\}) : \text{ref}\{a: \text{bool}\}}\ \text{fun}
\quad
\dfrac{\Gamma^{IV}(w) = T_4}{\Gamma^{IV} \vdash w : T_C}
}{\Gamma^{IV} = \Gamma''', w: T_4 \vdash (\text{fun}\ z: \text{ref}\{a: \text{bool}\} \Rightarrow (\text{fix}\ x)\{a = \text{false}, b = 6, c = 7\})\, w : T_R}\ \text{app}
}{\Gamma''' \vdash (\text{fix}\ x)\{a = \text{true}, b = 5\} : T_4}\ \text{app}
}{\Gamma''' = \Gamma, x: T_4 \vdash \text{let}\ w: T_4 = (\text{fix}\ x)\{a = \text{true}, b = 5\}\ \text{in}\ (\text{fun}\ z: \text{ref}\{a: \text{bool}\} \Rightarrow (\text{fix}\ x)\{a = \text{false}, b = 6, c = 7\})\, w : T_R}\ \text{let}
}{\Gamma \vdash \text{let}\ x: T_1 = \text{fun}(z: T_2) \Rightarrow \text{fun}(y: T_3) \Rightarrow \text{ref}\, y\ \text{in}\ \text{let}\ w: T_4 = (\text{fix}\ x)\{a = \text{true}, b = 5\}\ \text{in}\ (\text{fun}\ z: \text{ref}\{a: \text{bool}\} \Rightarrow (\text{fix}\ x)\{a = \text{false}, b = 6, c = 7\})\, w : T_R}\ \text{let}
$$

$fun = T_1 \to bool$

$T_1 = ref\{a:bool\} \to \{a:bool, b:int\}$

$T_C = \{a:bool, b:int\}$

$T_D = ref\{a:bool\}$

$T_B = \{b:int\}$

$T_M = \{a:bool, b:int\}$

$T_O = \{a:bool, b:int, c:int\}$

$T_N = \{a:bool\}$

$$\dfrac{\dfrac{\dfrac{

$T_2 = T_3 \to \text{ref}\, T_3$

$T_4 = \text{ref}\, T_3$

$T_3 = \{a{:}\text{bool}, b{:}\text{int}\}$

$T_1 = T_2 \to T_3 \to \text{ref}\, T_3$

$\overset{!}{=} \underbrace{(T_B \to T_4)}_{T_2} \to (\underbrace{T_B}_{T_3} \to \underbrace{T_4}_{\text{ref}\, T_3})$

$T_1 = \underbrace{(\underbrace{\{a{:}\text{bool}, b{:}\text{int}\}}_{T_3} \to T_4)}_{T_2} \to (\{a{:}\text{bool}, b{:}\text{int}\} \to T_4)$

$T_J = T_B = T_3$

$T_R = T_4$

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\Gamma''(y) = T_3}{\Gamma'' \vdash y : \cancel{T_B}\ T_3}\,\text{var}}{\Gamma'' = \Gamma', y{:}T_2 \vdash \text{ref}\, y : \text{ref}\ \cancel{T_B}\ T_3}\,\text{ref}}{\Gamma' = \Gamma, z{:}T_2 \vdash \text{fun}\,(y{:}T_3) \Rightarrow \text{ref}\, y : T_3 \to \text{ref}\ \cancel{T_B}}\,\text{fun}}{\Gamma \vdash \text{fun}\,(z{:}T_2) \Rightarrow \text{fun}\,(y{:}T_3) \Rightarrow \text{ref}\, y : T_2 \to \cancel{T_A}}\,\text{fun} \quad \Gamma''' = \Gamma, x{:}T_1 \vdash \text{let}\ w{:}\ T_4 = (\text{fix}\, x)\{a = \text{true}, b = 5\}\ \text{in}\ (\text{fun}\ z{:}\ \text{ref}\{a{:}\text{bool}\} \Rightarrow (\text{fix}\, x)\{a = \text{false}, b = 6, c = 7\})\, w : T_R}{\Gamma \vdash \text{let}\, x : T_1 = \text{fun}\,(z{:}T_2) \Rightarrow \text{fun}\,(y{:}T_3) \Rightarrow \text{ref}\, y\ \text{in}\ \text{let}\ w{:}\ T_4 = (\text{fix}\, x)\{a = \text{true}, b = 5\}\ \text{in}\ (\text{fun}\ z{:}\ \text{ref}\{a{:}\text{bool}\} \Rightarrow (\text{fix}\, x)\{a = \text{false}, b = 6, c = 7\})\, w : T_R = \text{ref}\{a{:}\text{bool}, b{:}\text{int}\}}\,\text{let}}$$

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\Gamma''(x) = T_1}{\Gamma''' \vdash x : \underbrace{(\cancel{T_B} \to T_4)}_{T_2} \to (\cancel{T_B} \to T_4)}\,\text{var} \quad \dfrac{}{\Gamma''' \vdash \text{true} : \text{bool}}\,\text{bool} \quad \dfrac{}{\Gamma''' \vdash 5 : \text{int}}\,\text{int}}{\Gamma''' \vdash (\text{fix}\, x) : \cancel{T_B} \to T_4 \quad \Gamma''' \vdash \{a = \text{true}, b = 5\} : \cancel{T_B}}\,\text{app}}{\Gamma''' \vdash (\text{fix}\, x)\{a = \text{true}, b = 5\} : T_4}\,\text{app}}{\Gamma'' = \Gamma', x{:}T_1 \vdash \text{let}\ w{:}\ T_4 = (\text{fix}\, x)\{a = \text{true}, b = 5\}\ \text{in}\ (\text{fun}\ z{:}\ \text{ref}\{a{:}\text{bool}\} \Rightarrow (\text{fix}\, x)\{a = \text{false}, b = 6, c = 7\})\, w : T_R}\,\text{let}}$$

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\Gamma^{V}(x) = T_1}{\Gamma^{V} \vdash x : \cancel{(T_3 \to T_4)} \to \cancel{(T_3 \to T_4)}}\,\text{var}}{\Gamma^{V} \vdash (\text{fix}\, x) : \cancel{T_J} \to \cancel{T_R}}\,\text{T-fix} \quad \dfrac{\dfrac{\Gamma^{V} \vdash ff : \text{int} \quad \Gamma^{V} \vdash 6 : \text{int} \quad \Gamma^{V} \vdash 7 : \text{int}}{\Gamma^{V} \vdash \{a = \text{false}, b = 6, c = 7\} : \cancel{T_K}}\,\text{record} \quad \cancel{T_K} < \cancel{T_B}}{\Gamma^{V} \vdash \{a = \text{false}, b = 6, c = 7\} : \cancel{T_J}}\,\text{sub}}{\Gamma^{V'} = \Gamma^{V}, z{:}\text{ref}\{a{:}\text{bool}\} \vdash (\text{fix}\, x)\{a = \text{false}, b = 6, c = 7\} : \cancel{T_R}}\,\text{app} \quad \dfrac{\Gamma^{V}(w) = \cancel{T_A}}{\Gamma^{V} \vdash w : \cancel{T_B}}\,\text{var}}{\Gamma'' = \Gamma'', w{:}T_4 \vdash (\text{fun}\ z{:}\ \text{ref}\{a{:}\text{bool}\} \Rightarrow (\text{fix}\, x)\{a = \text{false}, b = 6, c = 7\})\, w : T_R}\,\text{let}$$

2.a

$$\begin{tikzcd}
\alpha_j(\alpha_j E_1 \| \alpha_j E_2) \arrow[d, "\alpha"] & \approx_r & \alpha_j \alpha_j E_1 \| \alpha_j \alpha_j E_2 \arrow[d, "\alpha"] \\
(\alpha_j E_1 \| \alpha_j E_2) & \overset{id}{\Longleftrightarrow} & \alpha_j E_1 \| \alpha_j \alpha_j E_2 \arrow[d, "\alpha"] \\
 & & \alpha_j E_1 \| a_j E_2
\end{tikzcd}$$

$$\begin{tikzcd}
\alpha_j(\alpha_j E_1 \| \alpha_j E_2) \arrow[d, "\alpha"] & \approx_r & \alpha_j \alpha_j E_1 \| \alpha_j \alpha_j E_2 \arrow[d, "\alpha"] \\
(\alpha_j E_1 \| \alpha_j E_2) \arrow[d, "\alpha"] & & \alpha_j E_1 \| \alpha_j \alpha_j E_2 \arrow[d, "\alpha"] \\
E_1 \| \alpha_j E_2 \arrow[d, "skip"] & & E_1 \| \alpha_j \alpha_j E_2 \arrow[d, "\alpha"] \\
E_1 \| \alpha_j E_2 \arrow[d, "E_1"] & & E_1 \| \alpha_j E_2 \arrow[d, "E_1"] \\
\alpha_j E_2 \arrow[d, "a"] & & \alpha_j E_2 \arrow[d, "a"] \\
\times & & E_2 \\
a &
\end{tikzcd}$$

a: $|t = 1$

$E_1 =$

Lecture 16 min 48

b

$P \sqsubseteq Q$? P é simulato da Q?

$$\overset{P}{\overset{\shortparallel}{\text{await true protect } (\alpha;E_1 \oplus \beta;E_2);\ l:=1 \text{ end}}} \approx_r \overset{Q}{\overset{\shortparallel}{(\text{await } l=0 \text{ protect } \alpha;E_1;\ l:=1 \text{ end}) \parallel (\text{await } l:=0 \text{ protect } \beta;E_2;\ l:=1 \text{ end})}}$$

$\mathcal{R} = \{(P,Q), (P',Q'), (Q'', P'')\} \cup Id$

1 Esaminiamo la prima coppia (P,Q) da sinistra verso destra

1.1 $P \xrightarrow{choice\ L} P' \equiv skip$ allora $\exists Q'.\ Q \xrightarrow{choice\ L} Q' \equiv skip \parallel D$, purche $(P',Q') \in \mathcal{R}$

1.2 $P \xrightarrow{choice\ R} P'' \equiv skip$ allora $\exists Q''.\ Q \xrightarrow{choice\ L} Q' \equiv C \parallel skip$, purche $(P'',Q'') \in \mathcal{R}$

2 Esaminiamo la seconda coppia (P,Q) sinistra verso destra

Siccome stiamo dimostrando $P' \sqsubseteq Q'$ ed essendo $P' \equiv skip$ non ho azioni da eseguire, perciò il caso é banalmente risolto

3. Esaminiamo la terza coppia (P'',Q'') da sinistra verso destra

Siccome stiamo dimostrando $P'' \sqsubseteq Q''$ ed essendo $P'' \equiv skip$ non ho azioni da eseguire, perciò il caso é banalmente risolto

$Q \sqsubseteq P$? Q é simulato da P?

$$\overset{P}{\overset{\shortparallel}{\text{await true protect } (\alpha;E_1 \oplus \beta;E_2);\ l:=1 \text{ end}}} \approx_r \overset{Q}{\overset{\shortparallel}{\underbrace{(\text{await } l=0 \text{ protect } \alpha;E_1;\ l:=1 \text{ end})}_{C} \parallel \underbrace{(\text{await } l:=0 \text{ protect } \beta;E_2;\ l:=1 \text{ end})}_{D}}}$$

$\mathcal{R} = \{(P,Q), (P',Q'), (P'',Q'')\} \cup Id$

1 Esaminiamo la prima coppia (P,Q) da destra verso sinistra

1.1 $Q \xrightarrow{choice\ L} Q' \equiv skip \parallel D$ allora $\exists P'.\ P \xrightarrow{choice\ L} P' \equiv skip$, purche $(P',Q') \in \mathcal{R}$

1.2 $Q \xrightarrow{choice\ R} Q'' \equiv C \parallel skip$ allora $\exists P''.\ P \xrightarrow{choice\ R} P'' \equiv skip$, purche $(P'',Q'') \in \mathcal{R}$

2. Esaminiamo la seconda coppia (P',Q') da destra verso sinistra

2.1 ma la guardia dell' await presente in D é falsa perció $Q' \longrightarrow skip$ finiamo dunque in identità

3. Esaminiamo la seconda coppia (P'',Q'') da destra verso sinistra

3.1 ma la guardia dell' await presente in D é falsa perció $Q'' \longrightarrow skip$ finiamo dunque in identità

Dimostriamo bisimulazione:

$$\overset{P}{\overset{\shortparallel}{\text{await true protect } (\alpha;E_1 \oplus \beta;E_2);\ l:=1 \text{ end}}} \approx_r \overset{Q}{\overset{\shortparallel}{(\text{await } l=0 \text{ protect } \alpha;E_1;\ l:=1 \text{ end}) \parallel (\text{await } l:=0 \text{ protect } \beta;E_2;\ l:=1 \text{ end})}}$$

$\mathcal{R} = \{(P,Q), (P',Q'), (Q'', P'')\} \cup Id$

1 Esaminiamo la prima coppia (P,Q) da sinistra verso destra

1.1 $P \xrightarrow{choice\ L} P' \equiv skip$ allora $\exists Q'.\ Q \xrightarrow{choice\ L} Q' \equiv skip \parallel D$ (lo stato si é allineato), purche $(P',Q') \in \mathcal{R}$

1.2 $P \xrightarrow{choice\ R} P'' \equiv skip$ allora $\exists Q''.\ Q \xrightarrow{choice\ L} Q'' \equiv C \parallel skip$ (lo stato si é allineato), purche $(P'',Q'') \in \mathcal{R}$

1 Esaminiamo la prima coppia (P,Q) da destra verso sinistra

1.1 $Q \xrightarrow{choice\ L} Q' \equiv skip \parallel D$ allora $\exists P'.\ P \xrightarrow{choice\ L} P' \equiv skip$ (lo stato si é allineato), purche $(P',Q') \in \mathcal{R}$

1.2 $Q \xrightarrow{choice\ R} Q'' \equiv C \parallel skip$ allora $\exists P''.\ P \xrightarrow{choice\ R} P'' \equiv skip$ (lo stato si é allineato), purche $(P'',Q'') \in \mathcal{R}$

2 Esaminiamo la seconda coppia (P,Q) sinistra verso destra

Siccome stiamo dimostrando $P' \sqsubseteq Q'$ ed essendo $P' \equiv skip$ non ho azioni da eseguire, perciò il caso é banalmente risolto

2. Esaminiamo la seconda coppia (P',Q') da destra verso sinistra

2.1 ma la guardia dell' await presente in D é falsa perció Q' é in deadlock e non presenta problemi

3. Esaminiamo la terza coppia (P'',Q'') da sinistra verso destra

Siccome stiamo dimostrando $P'' \sqsubseteq Q''$ ed essendo $P'' \equiv skip$ non ho azioni da eseguire, perciò il caso é banalmente risolto

3. Esaminiamo la seconda coppia (P,Q'') da destra verso sinistra

3.1 ma la guardia dell' await presente in D é falsa perció Q'' é in deadlock e non presenta problemi

In teoria 30.01.2020

$$\overset{P}{\overset{\shortparallel}{(\text{await true protect } (\alpha;E_1 \oplus \beta;E_2) \text{ end})}} \simeq_r \overset{Q}{\overset{\shortparallel}{(\beta;\text{await true protect } E_1 \text{ end}) \parallel (\alpha;\text{await true protect } E_2 \text{ end})}}$$

Guardandolo così pare che non valga né $P \sqsubseteq Q$ né $Q \sqsubseteq P$

Cerchiamo dunque dei contromodelli

Contromodello
α: $l := 1$
b: $m := 6$
E_1: $K := 9$
E_2: $j := 8$

→ $P \sqsubseteq Q$? P è simulato da Q?

$$(\text{await true protect } (\alpha;E_1 \oplus \beta;E_2) \text{ end}) \simeq_r (\beta;\text{await true protect } E_1 \text{ end}) \parallel (\alpha;\text{await true protect } E_2 \text{ end})$$

$\downarrow \alpha;E_1$ $\qquad$ $\downarrow \alpha$

$l := 1, K := 9$ $\qquad$ $l := 1$

stati diversi $\not\sqsubseteq$ $\qquad$ $\downarrow \beta$

$l := 1, m = 6$

→ $Q \sqsubseteq P$? Q è simulato da P?

$$(\text{await true protect } (\alpha;E_1 \oplus \beta;E_2) \text{ end}) \simeq_r (\beta;\text{await true protect } E_1 \text{ end}) \parallel (\alpha;\text{await true protect } E_2 \text{ end})$$

$\downarrow \beta;E_2$ $\qquad$ $\downarrow \beta$

$\times$ $\qquad$ $(\text{skip};\text{await true protect } E_1 \text{ end}) \parallel (\alpha;\text{await true protect } E_2 \text{ end})$

Non è in grado di mettersi al passo perché esegue pure E_2 visto che l'esecuzione avviene atomicamente

Ci accorgiamo però che se α e β lavorano sulla stessa locazione $P \sqsubseteq Q$ ma continua a valere $Q \not\sqsubseteq P$

→ $Q \sqsubseteq P$? Q è simulato da P?

$$(\text{await true protect } (\alpha;E_1 \oplus \beta;E_2) \text{ end}) \simeq_r (\beta;\text{await true protect } E_1 \text{ end}) \parallel (\alpha;\text{await true protect } E_2 \text{ end})$$

$\downarrow \beta;E_2$ $\qquad$ $\downarrow \beta$

$\times$ $\qquad$ $(\text{skip};\text{await true protect } E_1 \text{ end}) \parallel (\alpha;\text{await true protect } E_2 \text{ end})$

Non è in grado di mettersi al passo perché esegue pure E_2 visto che l'esecuzione avviene atomicamente

Modello e contromodello
α: $l := 1$
b: $l := 6$

E_1: $K := 9$
E_2: $j := 8$

→ $P \sqsubseteq Q$? P è simulato da Q?

$$\overset{P}{\overset{\shortparallel}{(\text{await true protect } (\alpha;E_1 \oplus \beta;E_2) \text{ end})}} \simeq_r \overset{Q}{\overset{\shortparallel}{\overset{C}{\overset{\shortparallel}{(\beta;\text{await true protect } E_1 \text{ end})}} \parallel \overset{D}{\overset{\shortparallel}{(\alpha;\text{await true protect } E_2 \text{ end})}}}}$$

$Q = C \parallel D$

$\mathcal{R} = \{(P,Q), (P',Q'), (P'',Q'')\} \cup Id$

1. Provo la prima coppia (P,Q) da sinistra verso destra
   1.1 Se $P \xrightarrow{\alpha;E_2} P' \equiv skip$ allora $\exists Q'$ $Q \rightarrow^* Q' \equiv skip \parallel (\text{await true protect } E_2 \text{ end})$, purché $(P',Q') \in \mathcal{R}$
   1.2 Se $P \xrightarrow{\beta,E_2} P' \equiv skip$ allora $\exists Q'$ $Q \rightarrow^* Q'' \equiv (\text{await true protect } E_1 \text{ end}) \parallel skip$, purché $(P'',Q'') \in \mathcal{R}$
2. Provo la seconda coppia (P',Q') da sinistra verso destra
   Siccome stiamo dimostrando $P' \sqsubseteq Q'$ ed essendo $P' \equiv skip$ non ho azioni da eseguire, perciò il caso è banalmente risolto
3. Provo la terza coppia (P'',Q'') da sinistra verso destra
   Siccome stiamo dimostrando $P'' \sqsubseteq Q''$ ed essendo $P'' \equiv skip$ non ho azioni da eseguire, perciò il caso è banalmente risolto

$P \sqsubseteq Q$ P é simulato da Q?

$\mathcal{R} = \{(P,Q)\} \in Id$

$$\alpha_j(\alpha_j E_1 \| a_j E_2) \approx_r \alpha_j \alpha_j E_1 \| \alpha_j \alpha_j E_2$$

$$\downarrow \qquad\qquad \downarrow$$

$$a_j E_1 \| a_j E_2 \qquad\qquad a_j E_1 \| a_j E_2$$

1. Provo la prima coppia (P,Q) da sinistra a destra

1 Se $P \to P' \equiv \alpha_j E_1 \| a_j E_2$ allora $\exists Q'$ $Q \xrightarrow{a} Q' \equiv a_j E_1 \| a_j E_2$ ma $(P',Q') \in Id$

$Q \sqsubseteq P$? q é simulato da P

$$\alpha_j(\alpha_j E_1 \| a_j E_2) \approx_r \alpha_j \alpha_j E_1 \| \alpha_j \alpha_j E_2$$

$\alpha = l=1$

$E_1 = l:=5;\ l=6$

$E_2 = skip$

$$\begin{array}{ccc}
\downarrow & & \downarrow \\
\alpha_j E_1 \| a_j E_2 & & \alpha_j E_1 \| \alpha, \alpha_j E_2 \\
\downarrow & & \downarrow \\
E_1 \| a_j E_2 & & E_1 \| \alpha_j \alpha_j E_2 \\
\downarrow l:=5 & & \downarrow l:=5 \\
l:=6 \| a_j E_2 & & l:=6 \| a_j a_j E_2 \\
\downarrow \alpha: l:=7 & & \downarrow l:=7 \\
l:=6 \| E_2 & & l:=6 \| a_j E_2 \\
\downarrow & & \downarrow \\
skip \| E_2 & & skip \| a_j E_2 \\
\downarrow & & \downarrow a \\
\times & &
\end{array}$$

Ho trovato un controesempio $Q \sqsubseteq P$

$(\text{await true protect } (E_1 + E_2) \text{ end}) \approx_p \text{skip}_; (\text{await true protect } E_2 \text{ end}) + \text{skip}_; (\text{await true protect } E_1 \text{ end})$

- entrambe le simulazioni sembrano valere
- la bisimulazione vale ?

Proviamola

$(\text{await true protect } (E_1 + E_2) \text{ end}) \approx_p \text{skip}_; (\text{await true protect } E_2 \text{ end}) + \text{skip}_; (\text{await true protect } E_1 \text{ end})$

$\downarrow_{*}$ ②

$\text{await true protect } (E_1 + E_2) \text{ end}$

③ $\downarrow E_1$

$\downarrow \text{skip}$ ①

$(\text{await true protect } E_2 \text{ end})$

④ $\downarrow$

$\times$

Controesempio per bisimulazione

E_1: $\ell = 1$

E_2: $\ell = 3$

Non vale perciò la bisimulazione

i passi vanno separati

PS NON è bisimu

$P \sim_p Q$? P simula Q ?

$$\underbrace{(\text{await true protect } (E_1 + E_2) \text{ end})}_{P} \approx_p \underbrace{\text{skip}_; (\text{await true protect } E_2 \text{ end}) + \text{skip}_; (\text{await true protect } E_1 \text{ end})}_{Q}$$

$\mathcal{R} = \{(P,Q)\} \cup Id$

1 Provo la prima coppia (P,Q) da sinistra verso destra

1.1 Se $P \xrightarrow{\text{await choiceL}} P' \equiv \text{skip}$ allora $\exists Q'$. $Q \xrightarrow{\text{choiceR skip}} \xrightarrow{E_2} Q' \equiv \text{skip}$ $(P', Q') \in Id$

1.2 Se $P \xrightarrow{\text{await choiceR}} P'' \equiv \text{skip}$ allora $\exists Q''$. $Q \xrightarrow{\text{choiceL skip}} \xrightarrow{E_2} Q'' \equiv \text{skip}$ $(P'', Q'') \in Id$

↳ l'await porta subito a skip perché l'esecuzione avviene in maniera atomica, internamente all'esecuzione atomica svolgo la choice R/L

$Q \sim_p P$? Q simula P ?

$$\underbrace{(\text{await true protect } (E_1 + E_2) \text{ end})}_{P} \approx_p \underbrace{\text{skip}_; (\text{await true protect } E_2 \text{ end}) + \text{skip}_; (\text{await true protect } E_1 \text{ end})}_{Q}$$

$\mathcal{R} = \{(P,Q), (P', Q'), (P'', Q'')\} \cup Id$

1 Provo la prima coppia (P,Q) da destra verso sinistra

1.1 Se $Q \xrightarrow{\text{choice } L} Q' \equiv \text{skip}_; (\text{await true protect } E_2 \text{ end})$ allora $\exists P'$. $P \xrightarrow{\text{skip}} P'$ con $(P', Q') \in \mathcal{R}$

1.2 Se $Q \xrightarrow{\text{choice } R} Q'' \equiv \text{skip}_; (\text{await true protect } E_1 \text{ end})$ allora $\exists P'$. $P \xrightarrow{\text{skip}} P''$ con $(P'', Q'') \in \mathcal{R}$

2 Analizziamo la seconda coppia (P', Q') da destra verso sinistra

2.1 Se $Q' \xrightarrow{\text{skip}_; \text{await } E_2} Q''' \equiv \text{skip}$ allora $\exists P'''$. $P \xrightarrow{\text{await choiceR}} P''' \equiv \text{skip}$ con $(P''', Q''') \in Id$

2.2 Se $Q'' \xrightarrow{\text{skip}_; \text{await } E_1} Q^{IV} \equiv \text{skip}$ allora $\exists P^{IV}$. $P \xrightarrow{\text{await choiceR}} P^{IV} \equiv \text{skip}$ con $(P^{IV}, Q^{IV}) \in Id$

$T_1 = T_2 \to ref\{a:int, b:bool, c:bool\}$

$T_2 = ref\ \{a:int, b:bool\}$

Non tipabile?

$T_2 = ref\ \{a:int, b:bool\}$

$T_1 = T_2 \to ref\{a:int, b:bool, c:bool\}$

$(T_F \to T_E) \to (T_F \to T_E)$ non può essere uguale a $ref\ \{a:int, b:bool\} \to ref\{a:int, b:bool, c:bool\}$

non tipabile

$$
\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\Gamma'(x) = T_x^{\,ref\{a:int,b:bool\}}}{\Gamma' \vdash x : ref\{a:int,b:bool\}}\,var}{\Gamma' \vdash !x : \{a:int,b:bool\}}\,deref}{\Gamma' \vdash \#a!x : T_A^{\,int}}\,recproj \quad \dfrac{\dfrac{\dfrac{\Gamma'(x) = T_x^{\,ref\{a:int,b:bool\}}}{\Gamma' \vdash x : ref\{a:int,b:bool\}}\,var}{\Gamma' \vdash !x : \{a:int,b:bool\}}\,deref}{\Gamma' \vdash \#a!x : int}\,recproj \quad \dfrac{}{\Gamma' \vdash 0 : int}\,int}{\Gamma' \vdash (\#a!x) \geq 0 : T_C^{\,bool}}\,op \geq \quad \dfrac{\dfrac{\dfrac{\Gamma'(x) = T_x^{\,ref\{a:int,b:bool\}}}{\Gamma' \vdash x : ref\{a:int,b:bool\}}\,var}{\Gamma' \vdash !x : \{a:int,b:bool\}}\,deref}{\Gamma' \vdash c = \#b!x : T_B^{\,bool}}\,recproj}{\Gamma' \vdash \{a = \#a!x,\ b = (\#a!x) \geq 0,\ c = \#b!x\} : T_A^{\,\{a:int,\,b:bool,\,c:bool\}}}\,record}{\Gamma' = \Gamma, x{:}T_2 \vdash ref\ \{a = \#a!x,\ b = (\#a!x) \geq 0,\ c = \#b!x\} : \cancel{ref\,T_A}\ ref\{a:int,b:bool,c:bool\}}\,ref}{\Gamma \vdash (fun(x{:}T_2) \Rightarrow ref\ \{a = \#a!x,\ b = (\#a!x) \geq 0,\ c = \#b!x\}) : \cancel{T_1}\ T_2 \to \cancel{ref\,T_A}\ ref\{a:int,b:bool,c:bool\}}\,fun}{\Gamma \vdash let\ f{:}T_1 = (fun(x{:}T_2) \Rightarrow ref\ \{a = \#a!x,\ b = (\#a!x) \geq 0,\ c = \#b!x\})\ in\ ((fun(g{:}\{b:bool, a:int\}) \Rightarrow \#bg)(f\,ix\,f)(ref\ \{a=5, b=True, c=false\})) \wedge (fun(g{:}\{a:int, b:bool\}) \Rightarrow ((\#a\,g) \geq 0) \vee (\#bg))((f\,ix\,f)(ref\ \{a=7, b=false, c=8\})) : bool}\,let
$$

$$
\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\Gamma''(g) = \{b:bool, a:int\}}{\Gamma'' \vdash g : \{b:bool, a:int\}}}{\Gamma'', g:\{b:bool, a:int\} \vdash \#bg : bool}\,recproj}{\Gamma'' \vdash fun(g{:}\{b:bool, a:int\}) \Rightarrow \#bg : T_E \to bool}\,fun \quad \dfrac{\dfrac{\Gamma''(f) = T_f}{\Gamma'' \vdash f : (T_F \to T_E) \to (T_F \to T_E)}}{\Gamma'' \vdash f\,ix\,f : T_F \to T_E}}{\Gamma'' \vdash (f\,ix\,f)(ref\ \{a=5, b=True, c=false\}) : T_E^{\,\{b:bool,\,a:int\}}}\,app \quad \dfrac{}{\Gamma'' \vdash ref\ \{a=5, b=True, c=false\} : T_F}}{\Gamma'' \vdash (fun(g{:}\{b:bool, a:int\}) \Rightarrow \#bg)((f\,ix\,f)(ref\ \{a=5, b=True, c=false\})) : bool}\,app \quad \dfrac{}{\Gamma'' \vdash (fun(g{:}\{a:int, b:bool\}) \Rightarrow ((\#a\,g) \geq 0) \vee (\#bg))((f\,ix\,f)(ref\ \{a=7, b=false, c=8\})) : bool}}{\Gamma'' = \Gamma, f{:}T_f \vdash (fun(g{:}\{b:bool, a:int\}) \Rightarrow \#bg)((f\,ix\,f)(ref\ \{a=5, b=True, c=false\})) \wedge (fun(g{:}\{a:int, b:bool\}) \Rightarrow ((\#a\,g) \geq 0) \vee (\#bg))((f\,ix\,f)(ref\ \{a=7, b=false, c=8\})) : bool}\,and
$$

$T_1 = \text{ref}\,(T_2 \to \text{ref}\,\{a:\text{int}, b:\text{bool}\})$

$T_2 = \text{ref}\,\{a:\text{bool}, c:\text{int}\}$

Fatto in classe 16/01/24

Non tipabile?

$T_0 = \text{ref}(\{a:\text{bool}, c:\text{int}\})$ $\qquad$ $T_3 = \text{ref}(\{a:\text{int}, b:\text{bool}\})$

var: $\Gamma'(y) = T_2 \;\vdash\; \Gamma' \vdash y : \text{ref}\{a:\text{bool}, c:\text{int}\}$

deref: $\Gamma' \vdash !y : \{a:\text{bool}, c:\text{int}\}$

recordproj: $\Gamma' \vdash \#c!y : T_6$ int

var: $\Gamma'(y) = T_2 \;\vdash\; \Gamma' \vdash y : \text{ref}\{a:\text{bool}, c:\text{int}\}$

deref: $\Gamma' \vdash !y : \{a:\text{bool}, c:\text{int}\}$

recordproj: $\Gamma' \vdash \#a!y : T_7$ bool

record: $\Gamma' \vdash \{a = \#c!y, b = \#a!y\} : T_4$ $\{a:\text{int}, b:\text{bool}\}$

ref: $\Gamma' \leq \Gamma, y:T_2 \vdash \text{ref}\{a = \#c!y, b = \#a!y\} : \text{ref}\,T_4$ ref $\{a:\text{int}, b:\text{bool}\}$

fun: $\Gamma \vdash \text{fun}(y:\text{ref}\,T_2) \Rightarrow \text{ref}\{a = \#c!y, b = \#a!y\} : T_2 \to \text{ref}\,T_4$ ref $\{a:\text{bool}, c:\text{int}\} \to$ ref $\{a:\text{int}, b:\text{bool}\}$

ref: $\Gamma \vdash \text{ref}(\text{fun}(y:\text{ref}\,T_2) \Rightarrow \text{ref}\{a = \#c!y, b = \#a!y\}) : T_1$ ref(ref$\{a:\text{bool}, c:\text{int}\} \to$ ref$\{a:\text{int}, b:\text{bool}\}$)

int: $\Gamma'' \vdash 3 : \text{int}$

int: $\Gamma'' \vdash 5 : \text{int}$ $\qquad$ int: $\Gamma'' \vdash 0 : \text{int}$

op>: $\Gamma'' \vdash 5 > 0 : \text{bool}$

bool: $\Gamma'' \vdash \text{tt} : \text{bool}$

vec: $\Gamma'' \vdash \{c = 3, b = 5 > 0, a = \text{true}\} : T_F$ $\{c:\text{int}, b:\text{bool}, a:\text{bool}\}$

sub: $T_F <: T_E$ $\{c:\text{int}, a:\text{bool}\}$

$\Gamma'' \vdash \{c = 3, b = 5 > 0, a = \text{true}\} : T_E$ $\{c:\text{int}, a:\text{bool}\}$

Ti una perm. di 1,2

sub: $T_E <: T_G$ $\{c:\text{int}, a:\text{bool}\}$ $\qquad$ $\{a:\text{bool}, c:\text{int}\}$

$\Gamma''(x) = T_1$

deref: $\Gamma'' \vdash x : \text{ref}\,(T_D \to T_3)$

$\Gamma'' \vdash !x : T_D \to T_3$

$\Gamma'' \vdash \{c = 3, b = 5 > 0, a = \text{true}\} : T_G$ $\{a:\text{bool}, c:\text{int}\}$

ref: $\Gamma'' \vdash \text{ref}\{c = 3, b = 5 > 0, a = \text{true}\} : T_D$ ref $\{a:\text{bool}, c:\text{int}\}$

app: $\Gamma'' \vdash !x(\text{ref}\{c = 3, b = 5 > 0, a = \text{true}\}) : T_3$

non tipabile

ref($\{a:\text{int}, b:\text{bool}\}$)

$\Gamma'''(w) = \text{ref}(\{b:\text{bool}\})$

$\Gamma''' \vdash w : T_3$

$\Gamma''' \leq \Gamma, w:T_3 \vdash \text{let } z:\text{ref}\{b:\text{bool}\} = w \text{ in } !x(\text{ref}\{a = \text{false}, b = 5, c = 1\}) : T_2$ let

let: $\Gamma'' \leq \Gamma, x:T_1 \vdash \text{let } w:T_3 = !x(\text{ref}\{c = 3, b = 5 > 0, a = \text{true}\}) \text{ in } \text{let } z:\text{ref}\{b:\text{bool}\} = w \text{ in } !x(\text{ref}\{a = \text{false}, b = 5, c = 1\}) : T_2$

let: $\text{let } x:T_1 = \text{ref}(\text{fun}(y:\text{ref}\,T_2) \Rightarrow \text{ref}\{a = \#c!y, b = \#a!y\}) \text{ in } \text{let } w:T_3 = !x(\text{ref}\{c = 3, b = 5 > 0, a = \text{true}\}) \text{ in } \text{let } z:\text{ref}\{b:\text{bool}\} = w \text{ in } !x(\text{ref}\{a = \text{false}, b = 5, c = 1\}) : T_2$

$T_1 = ref(T_2 \to ref\{a:int, b:bool\})$
$T_2 = ref(\{a:bool, c:int\})$

$T_1 = T_2 \to \{a:int, b:bool\}$
$T_A = \{a:int, b:bool\}$
$T_C = bool$
$T_B = int$
$T_D = T_E = \{a:bool, c:int\}$

$ref\ T_E = ref\ T_D = T_2$

$$T_1 = ref\left(ref(\{a:bool, c:int\}) \to ref(\{a:bool, b:bool\})\right)$$

Non tipabile perché se T_1 è così per soddisfare il ramo di sinistra, a destra uscirebbe che

$$T_F \to T_E = ref\{a:bool, c:int\} = ref\{a:bool, b:bool\}$$

non solo non sono uguali ma non sono nemmeno funzioni

Viceversa se avessi $T_F \to T_E$ per soddisfare il ramo di destra $T_1 = ref((T_F \to T_E) \to (T_F \to T_E))$

T_2

ma se T_2 è una funzione non posso tipare il ramo di sinistra

$T_H = \{c:int, b:bool, a:bool\}$
$T_G = \{a:bool, c:int\}$
$T_F = ref\{a:bool, c:int\}$

$T_3 = ref\{b:bool\}$

$$\dfrac{\dfrac{\dfrac{

$T_1 = T_2 \to T_3 \to ref\{b:bool\}$ solo $b:bool$ perché il w che è risultato di fix viene dato in input a z che prende $ref\{b:bool\}$

$T_2 = T_3 \to ref\{b:bool\}$ perché c'è fix

$T_3 =$

$T_4 = ref\{b:bool\}$

$$
\dfrac{
\dfrac{
\dfrac{

c) $a_j(E_1 \oplus skip_i E_2) \oplus (skip_j E_1) \approx_\Gamma \alpha_j(skip_j(E_1 \oplus \alpha_j E_2)) \oplus (a_j E_2)$

$P \not\subseteq Q$?

$$\underbrace{a_j(E_1 \oplus skip_i E_2) \oplus (skip_j E_1)}_{P} \approx_\Gamma \underbrace{\alpha_j(skip_j(E_1 \oplus \alpha_j E_2)) \oplus (a_j E_2)}_{Q}$$

$P \to skip \to E_1$

$Q \to \alpha_j \to \times$

Contromodello

α : $m=2$

E_1 : $n=3$

E_2 : $g=7$

$Q \subseteq P$?

$$\underbrace{a_j(E_1 \oplus skip_i E_2) \oplus (skip_j E_1)}_{P} \approx_\Gamma \underbrace{\alpha_j(skip_j(E_1 \oplus \alpha_j E_2)) \oplus (a_j E_2)}_{Q}$$

1 Provo la prima coppia (P,Q) da destra a sinistra

1.1

Esame 4.2.19

ⓐ $P \sqsubseteq Q$?

$$\alpha;(\beta;E_1 \parallel \beta;E_2) \approx skip;\alpha;\beta;(E_1 \parallel E_2)$$

$$\downarrow \alpha \qquad\qquad \downarrow \alpha$$

$$\beta;E_1 \parallel \beta;E_2 \ (m=2) \qquad \beta;(E_1 \parallel E_2) \ (m=2)$$

$$\downarrow \beta \qquad\qquad \downarrow \beta$$

$$E_1 \parallel \beta;E_2 \ (m=2, l=3) \qquad E_1 \parallel E_2 \ (m=2, l=3)$$

$$\downarrow E_1 \qquad\qquad \downarrow E_1$$

$$skip \parallel \beta;E_2 \ (m=2, l=4) \qquad skip \parallel E_2 \ (m=2, l=4)$$

$$\downarrow \beta \qquad\qquad \downarrow$$

$$E_2 \ (m=2, l=4) \qquad \times$$

impossibile allineare lo stato

α: $m=2$
β: $l=3$
E_1: $l=4$
E_2: $c=8$

$Q \sqsubseteq P$

$$\overset{P}{\alpha;(\beta;E_1 \parallel \beta;E_2)} \approx \overset{Q}{skip;\alpha;\beta;(E_1 \parallel E_2)}$$

$$\downarrow \alpha \qquad\qquad \downarrow skip;\alpha$$

$$\beta;E_1 \parallel \beta;E_2 \qquad \beta;(E_1 \parallel E_2)$$

$$\downarrow \beta,\beta \qquad\qquad \downarrow \beta$$

$$E_1 \parallel E_2 \qquad E_1 \parallel E_2$$

$\mathcal{R} = \{(P,Q), (P',Q') \dots\} \cup Id$

1 Analizziamo la prima coppia (P,Q) da destra a sinistra

1. Se $Q \xrightarrow{skip} Q' \equiv \beta;(E_1 \parallel E_2)$ allora $\exists P'$. $P \rightarrow^{\alpha} P' \equiv \beta;E_1 \parallel \beta;E_2$, purché $(P',Q') \in \mathcal{R}$

2 Analizziamo la seconda coppia (P',Q') da destra a sinistra

1. Se $Q' \rightarrow^{\beta} Q'' \equiv (E_1 \parallel E_2)$ allora $\exists P''$. $P' \rightarrow^{\beta} P'' \equiv E_1 \parallel E_2$, ma $(P'',Q'') \in Id$ quindi non vi è alcun problema

ⓑ (await true protect ($E_1 \| E_2$) end) $\approx_r$ (await true protect E_2 end) || (await true protect E_1 end) $\mathcal{R} = \{(P,Q)\} \cup Id$

skip $\ell = 1$

Possibile perche ho || dentro l'await
$\ell := !\ell+1$;
$\ell := \phi$;
$! \ell := \ell+1$;

×

o ho $\ell = \phi$ o
$\ell = 2$

$E_1 \equiv \ell := !\ell+1 \,;\, ! \ell := \ell+1 \,;$
$E_2 = \ell := \phi \,;$

$Q \subseteq P$?

(await true protect ($E_1 \| E_2$) end) $\approx_r$ (await true protect E_2 end) || (await true protect E_1 end)

choice L / choice R } await

skip $\ell = 5$

×

nessuna mossa per allineare lo stato

skip || (await true protect E_1 end) $\ell = 5$

skip || skip $\ell = 4$

E_1: $\ell = 4$
E_2: $\ell = 5$

$T_1 = \text{ref}\{T_3 \to \text{ref}\{a = T_3,\ b = \text{true}\}$

$T_3 = \text{int}$

$T_4 = \text{ref}\{a: \text{int},\ b: \text{bool}\}$

Nel secondo caso sarebbe tipabile (necessaria solo una permutazione

$$T_1 = \text{ref}\{\overset{\text{int}}{T_3} \to \text{ref}\{a = \overset{\text{int}}{T_3},\ b = \text{bool}\}\} = \text{ref}\{\text{int} \to \text{ref}\{a: \text{int},\ b: \text{bool}\}\}$$

$\text{ref}(T_C \to T_4)$

$T_4 = \text{ref}\{a: \text{int},\ b: \text{bool}\}$

$T_1 = \text{ref}\, T_A$

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\Gamma'(y) = T_3}{\Gamma' \vdash y : \cancel{T_B}\ \text{int}} \qquad \Gamma' \vdash \text{true} : \text{bool}}{\Gamma' \vdash \{a = y,\ b = \text{true}\} : T_B\ \{a: \text{int},\ b: \text{bool}\}}}{\Gamma' = \Gamma, y: T_3 \vdash \text{ref}\{a = y,\ b = \text{true}\} : \text{ref}\ T_B\ \{a: \text{int},\ b: \text{bool}\}}}{\Gamma \vdash \text{fun}\,(y: T_3) \to \text{ref}\{a = y,\ b = \text{true}\} : \cancel{T_1}\ T_3 \to \text{ref}\ T_B}\ \text{fun}}{\Gamma \vdash \text{ref}\,(\text{fun}\,(y: T_3) \to \text{ref}\{a = y,\ b = \text{true}\}) : \text{ref}\ T_A}}{\Gamma \vdash \text{let}\ x: T_1 = \text{ref}\,(\text{fun}\,(y: T_3) \to \text{ref}\{a = y,\ b = \text{true}\})\ \text{in}\ \text{let}\ w: T_4 = (!x)\,2\ \text{in}\ (\text{fun}\ z: \text{ref}\{a: \text{int}\} \to \{c = \#b!z,\ d = \#b!z\})\,w : \{c: \text{bool}, d: \text{bool}\}}\ \text{let}$$

$$\dfrac{\dfrac{\dfrac{\dfrac{\Gamma''(x) = T_1}{\Gamma'' \vdash x : \text{ref}(\overset{\text{int}}{\cancel{T_C}} \to T_4)}\ \text{var}}{\Gamma'' \vdash !x : \overset{\text{int}}{\cancel{T_C}} \to T_4}\ \text{deref} \qquad \dfrac{-}{\Gamma'' \vdash 2 : \overset{\text{int}}{\cancel{T_C}}}}{\Gamma'' \vdash (!x)\,2 : T_4}\ \text{app}}{\Gamma'' = \Gamma, x: T_1 \vdash \text{let}\ w: T_4 = (!x)\,2\ \text{in}\ (\text{fun}\ z: \text{ref}\{a: \text{int}\} \to \{c = \#b!z,\ d = \#b!z\})\,w : \{c: \text{bool}, d: \text{bool}\}}\ \text{let}$$

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\Gamma''(z) = \text{ref}\{a: \text{int}\}}{\Gamma'' \vdash z : \text{ref}\{a: \text{int}\}} \qquad \dfrac{\Gamma''(z) = \text{ref}\{a: \text{int}\}}{\Gamma'' \vdash z : \text{ref}\{a: \text{int}\}}}{\Gamma'' \vdash !z : T_C}\ \text{recproj}\ \text{non tipabile} \qquad \dfrac{\Gamma''(z) = \text{ref}\{a: \text{int}\}}{\Gamma'' \vdash z : \text{ref}\{a: \text{int}\}} \qquad \dfrac{\Gamma''(z) = \text{ref}\{a: \text{int}\}}{\Gamma'' \vdash z : \text{ref}\{a: \text{int}\}}}{\Gamma'' \vdash !z : T_C}\ \text{recproj}\ \text{non tipabile}}{\Gamma'' \vdash \#b!z : T_C} \qquad \dfrac{\Gamma''(z) = \text{ref}\{a: \text{int}\}}{\Gamma'' \vdash !z : T_C}\ \text{recproj} \qquad \dfrac{\Gamma''(z) = \text{ref}\{a: \text{int}\}}{\Gamma'' \vdash z : \text{ref}\{a: \text{int}\}}}{\Gamma''' = \Gamma'', z: \text{ref}\{a: \text{int}\} \vdash \{c = \#b!z,\ d = \#b!z\} : \{c: \text{bool}, d: \text{bool}\}}\ \text{record}}{\Gamma''' \vdash \text{fun}\ z: \text{ref}\{a: \text{int}\} \to \{c = \#b!z,\ d = \#b!z\} : T_D \to \{c: \text{bool}, d: \text{bool}\}}\ \text{fun} \qquad \dfrac{\Gamma'''(w) = T_4}{\Gamma''' \vdash w : T_D}\ \text{app}}{\Gamma''' = \Gamma'', w: T_4 \vdash (\text{fun}\ z: \text{ref}\{a: \text{int}\} \to \{c = \#b!z,\ d = \#b!z\})\,w : \{c: \text{bool}, d: \text{bool}\}}\ \text{let}$$

$= \text{ref}\{a: \text{int}, b: \text{bool}\}$

non è possibile fare subtyping sui ref e a sinistra voglio un ref{a:int}

$\alpha;(skip;E_1 \parallel \alpha;E_2) \approx_r skip;\alpha;(\alpha;E_1 \parallel skip;E_2)$

$\downarrow \alpha$ $\qquad\qquad$ $\downarrow skip;\alpha$

$skip;E_1 \parallel \alpha;E_2 \quad l:=5$ $\qquad\qquad$ $\alpha;E_1 \parallel skip;E_2 \quad l:=5$

$\downarrow E_1$ $\qquad\qquad$ $\downarrow \alpha;E_1$

$skip \parallel \alpha;E_2 \quad l:=6$ $\qquad\qquad$ $skip \parallel skip;E_2 \quad l:=6$

$\downarrow \alpha$ $\qquad\qquad$ $\downarrow$

$skip \parallel E_2 \quad l:=5$ $\qquad\qquad$ $\times$ non posso riallineare lo stato

$\alpha: l:=5$
$E_1: l:=6$
$E_2: m:=9$

$P \not\approx Q$

$\alpha;(skip;E_1 \parallel \alpha;E_2) \approx_r skip;\alpha;(\alpha;E_1 \parallel skip;E_2)$

$\downarrow \alpha \quad l=5$ $\qquad\qquad$ $\downarrow skip;\alpha; \quad l=5$

$skip;E_1 \parallel \alpha;E_2$ $\qquad\qquad$ $\alpha;E_1 \parallel skip;E_2$

$\downarrow \alpha;E_2 \quad l=6$ $\qquad\qquad$ $\downarrow E_2 \quad l=6$

$E_1 \parallel skip$ $\qquad\qquad$ $\alpha;E_1 \parallel skip$

$\downarrow$ $\qquad\qquad$ $\downarrow \alpha \quad l=5$

$\times$ $\qquad\qquad$ $E_1 \parallel skip$

$\alpha: l=5$
$E_2: l=6$
$E_1: m=3$

$f : T_1 \to int$

$T_1 : \{a{:}int, b{:}int\}$

$T_1 \equiv \{a{:}int\} \to \{c{:}int, d{:}int\} \to \{a{:}int\} \to \{c{:}int, d{:}int\}$

$\{a{:}int\} \to \{c{:}int, d{:}int\} \to \{a{:}int\} \to \{c{:}int, d{:}int\}$

$T_C =$

$T_D = \{a{:}int\}$

$$
\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\Gamma''(g) = int}{\Gamma'' \vdash g : \{c{:}int, d{:}int\}}\,var}{\Gamma'' \vdash g : \{c{:}int, d{:}int\} \vdash \#dg : int}\,\Gamma'' \equiv \text{scopi}}{\Gamma' \vdash fun(g : \{c{:}int, d{:}int\}) \to \#dg : \cancel{T_A} \to int}\,\{c{:}int, d{:}int\}\,fun \qquad \dfrac{\dfrac{\dfrac{\Gamma'(f) = T_1}{\Gamma' \vdash f : (T_B \Rightarrow T_X) \to (T_B \Rightarrow T_A)}\,var}{\Gamma' \vdash fix.f : \cancel{T_B \Rightarrow T_A}\ \{a{:}int, b{:}int\} \to \{c{:}int, d{:}int\}}\,T\text{-}fix \qquad \dfrac{\dfrac{\dfrac{}{\Gamma' \vdash 7 : int}\,int \qquad \dfrac{}{\Gamma' \vdash 6 : int}\,int}{\Gamma' \vdash \{a = 5, b = 6\} : \cancel{T_E}\ \{a{:}int, b{:}int\}} \qquad \cancel{T_E} < \cancel{T_A}\ \{a{:}int\}}{\Gamma' \vdash \{a = 5, b = 6\} : \cancel{T_A}\ \{a{:}int\}}\,subpag}{\Gamma' \vdash (fix.f)(\{a = 5, b = 6\}) : \cancel{T_A}\ \{c{:}int, d{:}int\}}\,app}{\Gamma' \vdash (fun(g : \{c{:}int, d{:}int\}) \to \#dg)((fix.f)(\{a = 5, b = 6\})) : int}\,app \qquad \dfrac{\dfrac{\dfrac{\dfrac{\Gamma'''(g) = int}{\Gamma''' \vdash g : \{d{:}int\}}\,var}{\Gamma'', g : \{d{:}int\} \vdash \#dg : int}}{\Gamma' \vdash (fun(g : \{d{:}int\}) \Rightarrow \#dg) : \cancel{T_F} \to int}\,\{d{:}int\}\,fun \qquad \dfrac{\dfrac{\dfrac{\Gamma'(f) = T_1}{\Gamma' \vdash f : (T_E \Rightarrow T_D) \to (T_E \Rightarrow T_D)}\,T\text{-}fix \qquad \dfrac{\dfrac{}{\Gamma' \vdash 7 : int}\,int}{\Gamma' \vdash \{a = 7\} : \cancel{T_E}\ \{a{:}int\}}\,rec}{\Gamma' \vdash fix.f : \cancel{T_E \Rightarrow T_D}\ \{a{:}int\} \to \{c{:}int, d{:}int\}}\,app}{\Gamma' \vdash (fix.f)(\{a = 7\}) : \cancel{T_D}\ \{c{:}int, d{:}int\}} \qquad \cancel{T_D} < \cancel{T_F}\ \{d{:}int\}}{\Gamma' \vdash (fun(g : \{d{:}int\}) \Rightarrow \#dg)((fix.f)(\{a = 7\})) : \cancel{T_F}\ \{d{:}int\}}\,app}{\Gamma' \vdash (fun(g : \{c{:}int, d{:}int\}) \to \#dg)((fix.f)(\{a = 5, b = 6\})) : int}\,\Gamma' \equiv \text{opt}}{\Gamma', f : T_1 \vdash (fun(g : \{c{:}int, d{:}int\}) \to \#dg)((fix.f)(\{a = 5, b = 6\})) + (fun(g : \{d{:}int\}) \Rightarrow \#dg)((fix.f)(\{a = 7\})) : int}
$$

$$
\Gamma \vdash fun(f : T_1) \to (fun(g : \{c{:}int, d{:}int\}) \to \#dg)((fix.f)(\{a = 5, b = 6\})) + (fun(g : \{d{:}int\}) \Rightarrow \#dg)((fix.f)(\{a = 7\})) : int \quad fun
$$

$$\alpha_j(E_1 \oplus \mathrm{skip}_j E_2) \oplus \alpha_j E_1 \approx_r \alpha_j(\mathrm{skip}_j(E_1 \oplus \mathrm{skip}_j E_2)) \oplus (\alpha_j E_2)$$

$$\begin{tikzcd}
\alpha_j(E_1 \oplus \mathrm{skip}_j E_2) \arrow[d, "\alpha"] \arrow[rr, phantom, "\approx_r"] & & \alpha_j(\mathrm{skip}_j(E_1 \oplus \mathrm{skip}_j E_2)) \arrow[d, "\alpha"] \arrow[r, phantom, "\oplus"] & (\alpha_j E_2) \\
E_1 \arrow[d] & & \mathrm{skip}_j(E_1 \oplus \mathrm{skip}_j E_2) \arrow[d, "\mathrm{skip}"] & \\
\times & & E_2 &
\end{tikzcd}$$

1 2 3

$$T_1 : \{a{:}int\} \to \{c{:}int, d{:}int\} \to \{a{:}int\} \to \{c{:}int, d{:}int\}$$

$$T_1 = \{a{:}int\} \to \{c{:}int, d{:}int\} \to \{a{:}int\} \to \{c{:}int, d{:}int\}$$

$T_D = T_E$ $\qquad$ $T_A = T_D$

$$\dfrac{\dfrac{\dfrac{

$E \in Lang ::= x \mid n \mid E_1 + E_2 \mid \text{let } x = E_1 \text{ in } E_2$

Teorema $\forall E \in Lang$, if $E \Downarrow v$ then $E \to^* v$

Big-step

(B-loc) $\cancel{\dfrac{-}{(x,s) \Downarrow s(x)}}$

(B-num) $\cancel{\dfrac{-}{\langle n, s \rangle \Downarrow n}}$

(B-Add) $\dfrac{\langle E_1, s \rangle \Downarrow n_1 \qquad \langle E_2, s \rangle \Downarrow n_2}{\langle E_1 + E_2, s \rangle \Downarrow N_3}$

(B-let-CBV) $\dfrac{\langle P \Downarrow m, s \rangle \quad \langle E\{^m/_x\} \Downarrow n, s \rangle}{\langle \text{let } x = P \text{ in } E \Downarrow n, s \rangle}$

Small step

(B-num) $\cancel{\dfrac{-}{\langle n, s \rangle \Downarrow n}}$

(B-loc) $\cancel{\dfrac{-}{(x,s) \Downarrow s(x)}}$

(S-Left) $\dfrac{\langle E_1, s \rangle \to \langle E_1', s \rangle}{\langle E_1 + E_2, s \rangle \to \langle E_1' + E_2, s \rangle}$

(S-Right) $\dfrac{\langle E_2, s \rangle \to \langle E_2', s \rangle}{\langle n_1 + E_2, s \rangle \to \langle n + E_2', s \rangle}$

(CBV-let1) $\dfrac{\langle e_1, s \rangle \to \langle e_1', s' \rangle}{\langle \text{let } x{:}T = e_1 \text{ in } e_2, s \rangle \to \langle \text{let } x{:}T = e_1' \text{ in } e_2, s' \rangle}$

(CBV-let2) $\dfrac{-}{\langle \text{let } x{:}T = v \text{ in } e_2, s \rangle \to \langle e_2\{^v/_x\}, s \rangle}$

Lemma $\forall E \in Lang$, if $E \to F$ and $E \to G$ the $F = G$

Teorema $\forall E \in Lang$, if $E \to^* v$ and $E \to^* w$ then $v = w$

$T_1 = T_2 \to T_3 \to \text{ref}\{a:\text{int}, b:\text{bool}, \cancel{c}\}$

$\quad\quad \parallel$

$T_2 = \text{ref}\{b:\text{int}\} \to \{a:\text{int}, b:\text{bool}\}$

$T_3 = \text{ref}\{b:\text{int}\}$

$T_D = \{b:\text{int}\}$

$T_C = \{a:\text{int}, b:\text{bool}, c:\text{int}\}$

$T_G = \{a:\text{int}, b:\text{int}\}$

$T_E = \{a:\text{int}, b:\text{bool}\}$

$T_F = \{b:\text{int}\}$

$T_4 = \text{ref}\{b:\text{int}\} \to \text{ref}\{a:\text{int}, b:\text{bool}\} \to \text{ref}\{b:\text{int}\} \to \text{ref}\{a:\text{int}, b:\text{bool}\}$

$$\dfrac{\dfrac{\dfrac{

Dimostrazione Esercizio

Si consideri

$$E := \quad x \mid n \mid E_1 + E_2 \mid let\ x = E_1\ in\ E_2$$

Semantica operazionale CBV

Teorema

$\forall E \in Lang$, if $E \Downarrow v$ and $E \Downarrow w$ then $v = w$

Caso Base 1

$E \equiv x$ non si applica perché $\nexists v$ t.c $E \equiv x \Downarrow v$ → Il teorema è vero perché l'ipotesi è falsa. $\nexists v$ t.c $x \Downarrow v$

↳ La big step semantic è definita solo su termini chiusi. x è solo placeholder

Caso Base 2

$E \equiv n$ Utilizzo l'assioma $\dfrac{-}{n \Downarrow n}$. Ne consegue che $v = n = w$

Caso induttivo 1

$E \equiv E_1 + E_2$ utilizzo la regola sum $\dfrac{E_1 \Downarrow n_1 \qquad E_2 \Downarrow n_2}{E_1 + E_2 \Downarrow n}$ $n = add(n_1, n_2)$ per derivare $E \Downarrow v$

$$\dfrac{\triangle\ E_1 \Downarrow v_1 \qquad \triangle\ E_2 \Downarrow v_2}{E_1 + E_2 \Downarrow v} \quad v = add(v_1, v_2)$$

Per ipotesi induttiva ho che $E_1 \Downarrow v_1$ e $E_1 \Downarrow w_1$ e che $E_2 \Downarrow v_2$ e $E_2 \Downarrow w_2$ ne consegue che $v_1 = w_1$ e che $v_2 = w_2$ di conseguenza $v = add(v_1, v_2) = add(w_1, w_2) = w$

Caso induttivo 2

$E \equiv let\ x = E_1\ in\ E_2$ utilizzo la regola let $\dfrac{E_1 \downarrow n_1 \qquad E_2\{^{n_1}/_x\} \Downarrow n}{let\ x = E_1\ in\ E_2 \downarrow n}$ per derivare $E \Downarrow v$

$$\dfrac{\triangle\ E_1 \downarrow v_1 \qquad \triangle\ E_2\{^{v_1}/_x\} \Downarrow n}{let\ x = E_1\ in\ E_2 \downarrow v}$$

Ne consegue che $E \equiv let\ x = E_1\ in\ E_2$, per qualche E_1 e E_2. Con $E_1 \Downarrow v_1$ e $E_2\{^{v_1}/_x\} \Downarrow$. Così se $E \Downarrow w$ avrò che $\exists w_1$ t.c. $\dfrac{E_1 \downarrow w_1 \qquad E_2\{^{w_1}/_x\} \Downarrow w}{let\ x = E_1\ in\ E_2 \downarrow w}$

Per (IH) poiché l'albero di derivazione di $E_1 \Downarrow v_1$ ha profondità inferiore rispetto a $E \Downarrow v$. Considerando che ho anche $E_1 \Downarrow w_1$ ne derivo che $v_1 = w_1$

Da $v_1 = w_1$ ne derivo che $E_1\{^{v_1}/_x\} \equiv E_2\{^{v_2}/_x\} \equiv E\{^{h}/_x\}$. Io so che $E_2\{^{v_1}/_x\}$ con un albero di profondità inferiore rispetto a $E \Downarrow v$, so pure che $E_2\{^{h}/_x\} \Downarrow w$. Per (IH) $v = w$

Esame 30.1.23

$E \in Lang ::= x \mid n \mid E_1 + E_2 \mid let\ x = E_1\ in\ E_2$

Teorema

$\forall E \in Lang$ if $E \Downarrow v$ then $E \rightarrow^* v$

Semantica Big step → Evaluation semantics

$$(B\text{-}Num)\ \frac{-}{\langle n, s\rangle \Downarrow n}$$

x non ha regole in quanto place holder ? Corretto?

$$(B\text{-}Add)\ \frac{\langle E_1, s\rangle \Downarrow n_1 \qquad \langle E_2, s\rangle \Downarrow n_2}{\langle E_1 + E_2, s\rangle \Downarrow n_3} \quad n_3 = add(n_1, n_2)$$

$$(B\text{-}Let\ CBV)\ \frac{E_1 \Downarrow n \qquad E_2\{^n/_x\} \Downarrow v}{let\ x = E_1\ in\ E_2 \Downarrow v}$$

Semantica small step → semantica operazionale

per la n la regola non c'è ? ? O è la stessa della big step

x è sempre un place holder

Non c'è la regola (B-num) in small step Perché non esiste

↳ Divergerei continuando a valutare n se n andasse in sé stesso

$$(S\text{-}Left)\ \frac{E_1 \rightarrow E_1'}{E_1 + E_2 \rightarrow E_1' + E_2}$$

$$(S\text{-}Right)\ \frac{E_2 \rightarrow E_2'}{n_1 + E_2 \rightarrow n_1 + E_2'}$$

$$(S\text{-}Add)\ \frac{-}{n_1 + n_2 \rightarrow_{ch} n_3} \quad n_3 = add(n_1, n_2)$$

Bisogna aggiungere store? ? non serve lo store

Se possibile controllare la dimostrazione ?

$$(CBV\text{-}let1)\ \frac{\langle e_1, s\rangle \rightarrow \langle e_1', s'\rangle}{\langle let\ x{:}T = e_1\ in\ e_2, s\rangle \rightarrow \langle let\ x{:}T = e_1'\ in\ e_2, s'\rangle} \qquad (CBV\text{-}let2)\ \frac{-}{\langle let\ x{:}T = v\ in\ e_2, s\rangle \rightarrow \langle e_2\{^v/_x\}, s\rangle}$$

Proof ?

By rule induction su come ho derivato che $E \Downarrow v$

Caso Base 1

Sia $E \Downarrow v$ derivato dall'applicazione di (B-num) si deriva che $E \equiv n$ per qualche n, quindi $E = n \rightarrow^* n$

Caso base 2

$E \equiv x$ non si applica perché $\nexists v$ t.c $E \equiv x \Downarrow v$ → Il teorema è vero perché l'ipotesi è falsa. $\nexists v$ t.c $x \Downarrow v$

↳ La big step semantic è definita solo su termini chiusi. x è solo placeholder

Caso induttivo 1

Sia $E \Downarrow v$ derivato dall'applicazione di (B-add) $\frac{\langle E_1, s\rangle \Downarrow v_1 \qquad \langle E_2, s\rangle \Downarrow v_2}{\langle E_1 + E_2, s\rangle \Downarrow v}$ $v = add(v_1, v_2)$

(Consideriamo $E \equiv E_1$) (Consideriamo $E \equiv E_2$)

ma allora guardando la profondità dell'albero di derivazione per (IH) abbiamo che $E_1 \Downarrow v_1$ ma allora $E_1 \rightarrow^* v_1$, e che $E_2 \Downarrow v_2$ ma allora $E_2 \rightarrow^* v_2$

Ma allora $E \equiv E_1 + E_2 \xrightarrow{(s\text{-}left)}{}^* v_1 + E_2 \xrightarrow{(s\text{-}right)}{}^* v_1 + v_2 \xrightarrow{(s\text{-}add)} v = add(v_1, v_2)$

Caso induttivo 2

Sia $E \Downarrow v$ derivato dall'applicazione di (B-Let CBV) $\frac{E_1 \Downarrow v_1 \qquad E_2\{^{v_1}/_x\} \Downarrow v}{let\ x = E_1\ in\ E_2 \Downarrow v}$

(Consideriamo $E \equiv E_1$) (Consideriamo $E \equiv E_2$)

ma allora guardando la profondità dell'albero di derivazione per (IH) abbiamo che $E_1 \Downarrow v_1$ ma allora $E_1 \rightarrow^* v_1$, e che $E_2\{^{v_1}/_x\} \Downarrow v$ ma allora $E_2\{^{v_1}/_x\} \rightarrow v$

Ma allora $E \equiv let\ x = E_1\ in\ E_2 \xrightarrow{(CBV\text{-}let2)} let\ x = v_1\ in\ E_2 \rightarrow E_2\{^{v_1}/_x\} \xrightarrow{Per\ IH}{}^* v$

Esercitazione

$T_1 = ref(T_2 \to ref\ T_{REC})$ $\{b:bool\}$

$T_2 = \{a:bool,\ c:int\}$

$T_{REC} = T_3 = \{\cancel{a:int},\ b:bool\}$

sub su input funzione a dx

sub su output a sx

$$\dfrac{\dfrac{\dfrac{

(a)

$$a;(skip;E_1 \parallel a;E_2) \approx skip;a;(\alpha;E_1 \parallel skip;E_2)$$

$\downarrow a$ $\qquad\qquad$ $\downarrow a$

$(skip;E_1 \parallel a;E_2)$ $\ell=5$ $\qquad$ $(\alpha;E_1 \parallel skip;E_2)$ $\ell=3$

$\downarrow skip$ $\qquad\qquad$ $\downarrow a$

$E_1 \parallel a;E_2$ $\qquad$ $E_1 \parallel skip;E_2$

$\downarrow E_1$ $\qquad\qquad$ $\downarrow E_1$ $\ell=7$

$a;E_2$ $\ell=7$ $\qquad$ E_2

$\downarrow a$ $\qquad\qquad$ $\downarrow$

E_2 $\ell=5$ $\qquad$ $\times$

a: $\ell=5$

E_1: $\ell=7$

E_2: $m=2$

(b) $a;(skip;E_1 \parallel a;E_2) \approx skip;a;(\alpha;E_1 \parallel skip;E_2)$

$\downarrow * \ \ell=3$ $\qquad\qquad$ $\downarrow skip$

$\downarrow a$ $\qquad\qquad$ $a;(\alpha;E_1 \parallel skip;E_2)$

$skip;E_1 \parallel a;E_2$ $\qquad\qquad$ $\downarrow a$ $\ell=5$

$\downarrow a$ $\qquad\qquad$ $\alpha;E_1 \parallel skip;E_2$

$skip;E_1 \parallel E_2$ $\qquad\qquad$ $\downarrow skip$ $\ell=5$

$\downarrow E_2$ $\qquad\qquad$ $\alpha;E_1 \parallel E_2$

E_1 $\qquad\qquad$ $\downarrow E_2$

$\downarrow$ $\qquad\qquad$ $a;E_1$ $\ell=7$

$\times$ $\qquad\qquad$ $\downarrow a$

(c) $a;(a;a;E_1 \parallel a;E_2) \simeq a;a;E_1 \parallel a;a;E_2$

$\downarrow a$ $\qquad\qquad$ $\downarrow a$

$a;a;E_1 \parallel a;E_2$ $\qquad\qquad$ $a;a;E_1 \parallel a;E_2$

$|$ $\qquad\qquad$ $|$

ID $\qquad\qquad$ ID

$a;(a;a;E_1 \parallel a;E_2) \simeq a;a;E_1 \parallel a;a;E_2$

$\downarrow a$ $\ell=7$ $\qquad\qquad$ $\downarrow a$ $\ell=7$

$a;a;E_1 \parallel a;E_2$ $\qquad\qquad$ $a;E_1 \parallel a;a;E_2$

$\downarrow a$ $\ell=7$ $\qquad\qquad$ $\downarrow a$ $\ell=7$

$E_1 \parallel a;E_2$ $\qquad\qquad$ $E_1 \parallel a;a;E_2$

$\downarrow \ell=3$ $\ell=3$ $\qquad\qquad$ $\downarrow \ell=3$ $\ell=3$

$\ell=5 \parallel a;E_2$ $\qquad\qquad$ $\ell=3 \parallel a;a;E_2$

$\downarrow a$ $\ell=7$ $\qquad\qquad$ $\downarrow a$ $\ell=7$

$\ell=5 \parallel E_2$ $\qquad\qquad$ $\ell=5 \parallel a;E_2$

$\downarrow \ell=5$ $\ell=5$ $\qquad\qquad$ $\downarrow \ell=5$ $\ell=5$

E_2 $\qquad\qquad$ $a;E_2$

$\downarrow$ $\qquad\qquad$ $\downarrow \ell=7$

$\times$ $\qquad\qquad$ E_2

$a=\ell=7$

E_1 $\ell=3$; $\ell=5$

$\underline{\subseteq}$

$\underline{\exists}$

$\not\approx_r$

await true protect $(E_1 + E_2)$ end $\approx$ skip ; await true protect (E_1) end + skip ; await true protect (E_1) end

1. Controesempio bisimulazione
2. Dim $\underline{\subseteq}$
3. Dim $\underline{\exists}$

$E \equiv n \mid x \mid E + E \mid \text{let } x = E \text{ in } E$

- Semantica BS e CBV

(B-Num) $\dfrac{-}{\langle n, s\rangle \Downarrow n}$

x non ha regole è un placeholder

(B-Add) $\dfrac{\langle E_1, s\rangle \Downarrow n_1 \qquad \langle E_2, s\rangle \Downarrow n_2}{\langle E_1 + E_2, s\rangle \Downarrow n_3}$ $\quad n_3 = add(n_1, n_2)$

(B-Let CBV) $\dfrac{E_1 \Downarrow n \qquad E_2\{^n/_x\} \Downarrow v}{\text{let } x = E_1 \text{ in } E_2 \Downarrow v}$

- Semanti SS CBV

(CBV-let1) $\dfrac{e_1 \to e_1'}{\text{let } x:T = e_1 \text{ in } e_2 \to \text{let } x:T = e_1' \text{ in } e_2}$ $\qquad$ (CBV-let2) $\dfrac{-}{\langle \text{let } x:T = v \text{ in } e_2, s\rangle \to \langle e_2\{^v/_x\}, s\rangle}$

Theorem:

$\forall E \in Exp \quad E \Downarrow v$ e $E \to^* w$ allora $v = w$

Proof:

By structural induction on E

Sia $E \equiv \text{let } x = E_1 \text{ in } E_2$ supponiamo $E \Downarrow v$ allora abbiamo usato la regola let

$\dfrac{E_1 \Downarrow v_1 \qquad E_2\{^{v_1}/_x\} \Downarrow v}{\text{let } x = E_1 \text{ in } E \Downarrow v}$ Ma allora per (IH) $E_1 \to^* v_1$. $\boxed{\text{Ma allora } E_2\{^{v_1}/_x\} \to v}$

↳ Non siamo in grado di dimostrarlo perché non è una sottostruttura

Proof:

By rule induction su come deriviamo E

Caso Base 1

Sia $E \Downarrow v$ derivato dall'applicazione di (B-num) si deriva che $E \equiv n$ per qualche n, quindi $E = n \to n^*$

Caso base 2

$E \equiv x$ non si applica perché $\nexists v$ t.c $E \equiv x \Downarrow v$ → Il teorema è vero perché l'ipotesi è falsa. $\nexists v$ t.c $x \Downarrow v$

↳ La big step semantic è definita solo su termini chiusi. x è solo placehoder

Caso induttivo 1

Sia $E \Downarrow v$ derivato dall'applicazione di (B-add) $\dfrac{\langle E_1, s\rangle \Downarrow v_1 \qquad \langle E_2, s\rangle \Downarrow v_2}{\langle E_1 + E_2, s\rangle \Downarrow v}$ $\quad v = add(v_1, v_2)$

(Consideriamo $E \equiv E_1$) $\qquad$ (Consideriamo $E \equiv E_2$)

ma allora guardando la profondità dell'albero di derivazione per (IH) abbiamo che $E_1 \Downarrow v_1$ ma allora $E_1 \to^* v_1$, e che $E_2 \Downarrow v_2$ ma allora $E_2 \to^* v_2$

Ma allora $E \equiv E_1 + E_2 \xrightarrow{(s\text{-left})}{}^* v_1 + E_2 \xrightarrow{(s\text{-right})}{}^* v_1 + v_2 \xrightarrow{(s\text{-add})} v = add(v_1, v_2)$

Caso induttivo 2

Sia $E \Downarrow v$ derivato dall'applicazione di (B-Let CBV) $\dfrac{E_1 \Downarrow v_1 \qquad E_2\{^{v_1}/_x\} \Downarrow v}{\text{let } x = E_1 \text{ in } E_2 \Downarrow v}$

(Consideriamo $E \equiv E_1$) $\qquad$ (Consideriamo $E \equiv E_2$)

ma allora guardando la profondità dell'albero di derivazione per (IH) abbiamo che $E_1 \Downarrow v_1$ ma allora $E_1 \to^* v_1$, e che $E_2\{^{v_1}/_x\} \Downarrow v$ ma allora $E_2\{^{v_1}/_x\} \to v$

Ma allora $E \equiv \text{let } x = E_1 \text{ in } E_2 \xrightarrow{(CBV\text{-let1})} \text{let } x = v_1 \text{ in } E_2 \xrightarrow{(CBV\text{-let2})} E_2\{^{v_1}/_x\} \xrightarrow{\text{Per IH}}{}^* v$

$$\alpha;(E_1 + skip;E_2) + \alpha;E_1 \simeq_r \alpha;(skip;(E_1 + skip;E_2)) + (\alpha;E_2)$$

$$\begin{array}{ccc} \downarrow \alpha \quad (1) & & \downarrow a \quad (2) \\ E_1 & & E_1 + skip;E_2 \\ \downarrow * & & \downarrow skip \quad (3) \\ (4)\ E_1 & & E_2 \\ \not\downarrow & & \downarrow E_2 \ (5) \end{array}$$

$E_1 \equiv n = 2$

$E_2 \equiv n = 17$

$$\alpha;(E_1 + skip;E_2) + \alpha;E_1 \overset{?}{\equiv} \alpha;(skip;(E_1 + skip;E_2)) + (\alpha;E_2)$$

Sembra valere

$$\overbrace{\alpha;(E_1 + skip;E_2) + \alpha;E_1}^{P} \overset{?}{\equiv} \overbrace{\alpha;(skip;(E_1 + skip;E_2)) + (\alpha;E_2)}^{Q}$$

$$R = \{(P,Q), (E_1, E_1 + skip; E_2), \dots\} \cup Id$$

Sembra valere

① Analizzo la prima coppia

1.1 Sia $P \xrightarrow{a} (E_1 + skip;E_2) \equiv P'$, allora $Q \xrightarrow{a} \xrightarrow{skip} (E_1 + skip;E_2) \equiv Q'$ con (P', Q') con $(P', Q') \in Id$

1.2 Sia $P \xrightarrow{a} E_1$ allora $\exists Q \xrightarrow{a} \xrightarrow{skip} (E_1 + skip;E_2)$ con $(E_1, E_1 + skip;E_2) \in R$

② Analizzo la seconda coppia

2.1 Sia $E_1 \xrightarrow{\beta} E_1'$ allora $E_1 + skip;E_2 \xrightarrow{\beta} E_1'$, con $(E_1', E_1') \in Id \subseteq R$

$T_1 = T_2 \to T_3 \to \text{ref}\ \{b:\text{bool}\}$

$\star\ T_2 = \cancel{T_3}\ \text{int} \to \text{ref}\ \{b:\text{bool}\}$

$\star\ T_3 = \text{int}$

$\star\ T_1 = \text{int} \to \text{ref}\{b:\text{bool}\} \to \text{int} \to \text{ref}\ \{b:\text{bool}\}$

$\star\ T_4 = \text{ref}\{b:\text{bool}\}$

var $\dfrac{\Gamma''(z) = T_2 \to T_3}{\Gamma'' \vdash z : \cancel{T_2}\ \text{int} \to \cancel{T_3}\ \text{ref}\{b:\text{bool}\}}$ var $\dfrac{\Gamma''(y) = T_3}{\Gamma'' \vdash y : \cancel{T_3}\ \text{int}}$

app $\dfrac{\Gamma'' \vdash z : \cancel{T_2} \to \cancel{T_3}\ \text{ref}\{b:\text{bool}\} \qquad \Gamma'' \vdash y : \cancel{T_3}\ \text{int}}{\Gamma'' \vdash z\,y : \cancel{\text{ref}\ T_3}\ \text{ref}\{b:\text{bool}\}}$

deref $\dfrac{\Gamma'' \vdash z\,y : \cancel{\text{ref}\ T_3}\ \text{ref}\{b:\text{bool}\}}{\Gamma'' \vdash !(z\,y) : \{b:\text{bool}\}}$

recproj $\dfrac{\Gamma'' \vdash !(z\,y) : \{b:\text{bool}\}}{\Gamma'' \vdash \#b(!(z\,y)) : \cancel{T_C}\ \text{bool}}$ $\qquad$ $\dfrac{-}{\Gamma'' \vdash \text{true} : \text{bool}}$ bool

record $\dfrac{\Gamma'' \vdash \#b(!(z\,y)) : \cancel{T_C}\ \text{bool} \qquad \Gamma'' \vdash \text{true} : \text{bool}}{\Gamma'' \vdash \{a = \#b(!(z\,y)),\ b = \text{true}\} : \cancel{T_4}\ \{a:\text{bool}, b:\text{bool}\}}$

sub $\dfrac{\Gamma'' \vdash \{a = \#b(!(z\,y)),\ b = \text{true}\} : \cancel{T_4}\ \{a:\text{bool}, b:\text{bool}\} \qquad \cancel{T_4} <: \cancel{T_B}\ \{b:\text{bool}\}}{\Gamma'' \vdash \{a = \#b(!(z\,y)),\ b = \text{true}\} : \cancel{T_B}\ \{b:\text{bool}\}}$

ref $\dfrac{\Gamma'' \vdash \{a = \#b(!(z\,y)),\ b = \text{true}\} : \cancel{T_B}\ \{b:\text{bool}\}}{\Gamma'', y:T_3 \vdash \text{ref}\ \{a = \#b(!(z\,y)),\ b = \text{true}\} : \cancel{\text{ref}\ T_B}\ \text{ref}\{b:\text{bool}\}}$ Γ''

fun $\dfrac{\Gamma'', y:T_3 \vdash \text{ref}\ \{a = \#b(!(z\,y)),\ b = \text{true}\} : \cancel{\text{ref}\ T_B}\ \text{ref}\{b:\text{bool}\}}{\Gamma'', z:T_2 \vdash \text{fun}\ (y:T_3) \Rightarrow \text{ref}\ \{a = \#b(!(z\,y)),\ b = \text{true}\} : \cancel{T_B}\ \{a:\text{bool}, b:\text{bool}\} \to \cancel{\text{ref}\ T_B}\ \text{ref}\{b:\text{bool}\}}$ Γ''

fun $\dfrac{\Gamma'', z:T_2 \vdash \text{fun}\ (y:T_3) \Rightarrow \text{ref}\ \{a = \#b(!(z\,y)),\ b = \text{true}\} : \cancel{T_1}\ T_2 \to T_3 \to \text{ref}\ T_B}{\Gamma \vdash \text{fun}\ (z:T_2) \Rightarrow \text{fun}\ (y:T_3) \Rightarrow \text{ref}\ \{a = \#b(!(z\,y)),\ b = \text{true}\} : \cancel{T_1}\ T_2 \to T_3 \to \text{ref}\ T_B}$

var $\dfrac{\Gamma'''(x) = T_1}{\Gamma''' \vdash x : \text{int} \to \cancel{T_4}\ \text{ref}\{b:\text{bool}\} \to \text{int} \to \cancel{T_4}\ \text{ref}\{b:\text{bool}\}}$

fix $\dfrac{\Gamma''' \vdash x : \text{int} \to \cancel{T_4}\ \text{ref}\{b:\text{bool}\} \to \text{int} \to \cancel{T_4}\ \text{ref}\{b:\text{bool}\} \qquad -}{\Gamma''' \vdash \text{fix}\ x : \text{int} \to \cancel{T_4}\ \text{ref}\{b:\text{bool}\}}$ int

app $\dfrac{\Gamma''' \vdash \text{fix}\ x : \text{int} \to \cancel{T_4}\ \text{ref}\{b:\text{bool}\} \qquad \Gamma''' \vdash 10 : \text{int}}{\Gamma''' \vdash (\text{fix}\ x)\ 10 : T_4}$

$\Gamma''' = \Gamma, x:T_1$

$\Gamma''(z) = \text{ref}\{b:\text{bool}\}$

deref $\dfrac{\Gamma''(z) = \text{ref}\{b:\text{bool}\}}{\Gamma'' \vdash z : \cancel{\text{ref}\ T_A}\ \text{ref}\{b:\text{bool}\}}$

recproj $\dfrac{\Gamma'' \vdash z : \cancel{\text{ref}\ T_A}\ \text{ref}\{b:\text{bool}\}}{\Gamma'' \vdash !z : \cancel{T_A}\ \{b:\text{bool}\}}$

$\Gamma''(z) = \text{ref}\{b:\text{bool}\}$

deref $\dfrac{\Gamma''(z) = \text{ref}\{b:\text{bool}\}}{\Gamma'' \vdash z : \cancel{\text{ref}\ T_A}\ \text{ref}\{b:\text{bool}\}}$

recproj $\dfrac{\Gamma'' \vdash z : \cancel{\text{ref}\ T_A}\ \text{ref}\{b:\text{bool}\}}{\Gamma'' \vdash !z : \cancel{T_A}\ \{b:\text{bool}\}}$

$\Gamma'' \vdash \#b!z : \cancel{T_A}\ \text{bool} \qquad - \qquad \Gamma'' \vdash \#b!z : \cancel{T_A}\ \text{bool}$

rec $\dfrac{\Gamma'' \vdash \#b!z : \cancel{T_A}\ \text{bool} \qquad \Gamma'' \vdash \#b!z : \cancel{T_A}\ \text{bool}}{\Gamma'' \vdash \{c = \#b!z \quad d = \#b!z\} : \cancel{T_A}\ \{a:\text{bool}, b:\text{bool}\}}$

fun $\dfrac{\Gamma'' \vdash \{c = \#b!z \quad d = \#b!z\} : \cancel{T_A}\ \{a:\text{bool}, b:\text{bool}\}}{\Gamma'' \vdash (\text{fun}\ z : \text{ref}\{b:\text{bool}\} \Rightarrow \{c = \#b!z \quad d = \#b!z\}) : \cancel{T_A}\ \text{ref}\{b:\text{bool}\} \to T_A}$

$\Gamma''(w) = T_4 = \text{ref}\{b:\text{bool}\}$

app $\dfrac{\Gamma''(w) = T_4 = \text{ref}\{b:\text{bool}\} \qquad \Gamma'' \vdash (\text{fun}\ z : \text{ref}\{b:\text{bool}\} \Rightarrow \{c = \#b!z \quad d = \#b!z\}) : \cancel{T_A}\ \text{ref}\{b:\text{bool}\} \to T_A}{\Gamma'' \vdash w : \cancel{T_A}\ \text{ref}\{b:\text{bool}\}}$

$\Gamma'' = \Gamma, w:T_4$

let $\dfrac{\Gamma'' \vdash w : \cancel{T_A}\ \text{ref}\{b:\text{bool}\} \qquad \Gamma'' \vdash (\text{fun}\ z : \text{ref}\{b:\text{bool}\} \Rightarrow \{c = \#b!z \quad d = \#b!z\}) : \cancel{T_A}\ \text{ref}\{b:\text{bool}\} \to T_A}{\Gamma'' \vdash w : T_A}$

let $\dfrac{\Gamma''' \vdash (\text{fix}\ x)\ 10 : T_4 \qquad \Gamma'' \vdash w : T_A}{\Gamma \vdash \text{let}\ w : T_4 = (\text{fix}\ x)\ 10\ \text{in}\ (\text{fun}\ z : \text{ref}\{b:\text{bool}\} \Rightarrow \{c = \#b!z \quad d = \#b!z\})\ w : T_A}$

let $\dfrac{\Gamma \vdash \text{fun}\ (z:T_2) \Rightarrow \text{fun}\ (y:T_3) \Rightarrow \text{ref}\ \{a = \#b(!(z\,y)),\ b = \text{true}\} : \cancel{T_1}\ T_2 \to T_3 \to \text{ref}\ T_B \qquad \Gamma \vdash \text{let}\ w : T_4 = (\text{fix}\ x)\ 10\ \text{in}\ (\text{fun}\ z : \text{ref}\{b:\text{bool}\} \Rightarrow \{c = \#b!z \quad d = \#b!z\})\ w : T_A}{\Gamma \vdash \text{let}\ x : T_1 = \text{fun}\ (z:T_2) \Rightarrow \text{fun}\ (y:T_3) \Rightarrow \text{ref}\ \{a = \#b(!(z\,y)),\ b = \text{true}\}\ \text{in}\ \text{let}\ w : T_4 = (\text{fix}\ x)\ 10\ \text{in}\ (\text{fun}\ z : \text{ref}\{b:\text{bool}\} \Rightarrow \{c = \#b!z \quad d = \#b!z\})\ w : T_A}$

Esercitazione

18.07.2018

$$\mathrm{Fib} = \lambda f.\lambda z.\ \mathrm{if}\ z=0 \,||\, z=1\ \mathrm{then}\ 1\ \mathrm{else}\ f(z-1)+f(z-2)$$

$$= \lambda f.\lambda z.\ F$$

$$\mathrm{fib} \stackrel{def}{=} \mathrm{fix}\ \mathrm{Fib}$$

$$= \mathrm{fix}\,(\lambda f.\lambda z.F)$$

$$= \mathrm{rec}\ f.\ F$$

$$\Gamma : \underset{T_{fib}}{\mathrm{fib}} : \mathrm{int} \to \mathrm{int} \to \mathrm{int} \to \mathrm{int}$$

$$\Gamma \vdash \mathrm{fix}.\ \mathrm{fib} : \mathrm{int} \to \mathrm{int}$$

$T_1 \equiv T_2 \to T_C$

$T_2 \equiv T_{fib}$

$T_3 \equiv T_{fib}$

$T_C \equiv \mathrm{int} \to \mathrm{int}$

$$\Gamma'(g) = T_2 = \underbrace{\mathrm{int} \to \mathrm{int}}_{T_F} \to \mathrm{int} \to \mathrm{int}$$

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\Gamma'(f) \equiv T_1 = T_{fib} \to T}{\Gamma' \vdash f : T_{fib} \to T_B}\,\mathrm{var} \quad \dfrac{\Gamma'(\mathrm{fib}) = \mathrm{int} \to \mathrm{int} \to \mathrm{int} \to \mathrm{int}}{\Gamma' \vdash \mathrm{fib} : T_{fib}}\,\mathrm{var}}{\Gamma'' = \Gamma', f : T_1 \vdash f\ \mathrm{fib} : T_B}\,\mathrm{app}}{\Gamma \vdash \mathrm{fun}\,(f : T_1) \Rightarrow f\ \mathrm{fib} : T_1 \to T_B}\,\mathrm{fun} \quad \dfrac{\dfrac{\dfrac{\dfrac{\Gamma''(g) = T_2 \equiv T_F \to T_C \equiv T_{fib}}{\Gamma'' \vdash g : T_F \to \cancel{T_C}\ \mathrm{int} \to \mathrm{int}}\,\mathrm{var} \quad \dfrac{\dfrac{\Gamma''' h : T_3 = T_F \to T_F \equiv T_{fib}}{\Gamma''' \vdash h : T_F \to T_F}\,\mathrm{var}}{\Gamma''' \vdash \mathrm{fix}\ h : \cancel{T_F}\ \mathrm{int} \to \mathrm{int}}\,\mathrm{fix}}{\Gamma'' = \Gamma'', h : T_3 \vdash g(\mathrm{fix}\ h) : \cancel{T_C}\ \mathrm{int} \to \mathrm{int}}\,\mathrm{app}}{\Gamma'' \vdash \mathrm{fun}\,(h : T_3) \Rightarrow g(\mathrm{fix}\ h) : \cancel{T_3}\ T_{fib} \to T_C}\,\mathrm{fun} \quad \dfrac{\Gamma''(\mathrm{fib}) = T_3}{\Gamma'' \vdash \mathrm{fib} : T_3}\,\mathrm{var}}{\Gamma'' = \Gamma', g : T_2 \vdash ((\mathrm{fun}\,(h : T_3) \Rightarrow g(\mathrm{fix}\ h))\ \mathrm{fib} : T_C}\,\mathrm{app}}{\Gamma \vdash \mathrm{fun}\,(g : T_2) \Rightarrow ((\mathrm{fun}\,(h : T_3) \Rightarrow g(\mathrm{fix}\ h))\ \mathrm{fib} : T_1 = T_2 \to T_C}\,\mathrm{fun}}{\Gamma \vdash (\mathrm{fun}\,(f : T_1) \Rightarrow f\ \mathrm{fib})(\mathrm{fun}\,(g : T_2) \Rightarrow ((\mathrm{fun}\,(h : T_3) \Rightarrow g(\mathrm{fix}\ h))\ \mathrm{fib})) : T_B}\,\mathrm{app}$$

$T_3 = T_{fib}$

Esercizio 2

dovrei tagliare la g

let $f:T_1 = fun(x:T_2) \Rightarrow ref\{p = \#ax,\ q = (\#bx) \geq 0,\ r = \#cx\}$ in $((fun(g:ref\{p:int, r:int\}) \Rightarrow \#p!g)(f\{a=5, b=6, c=7\}) \wedge (fun(g:ref\{p:int, r:int\}) \Rightarrow (\#p!g) \leq (\#r!g))(f\{a=7, b=8\})$

NON É QUINDI TIPABILE

Scrivere l'albero mostrando la contraddizione

Semantica CBN del LET

$$\frac{-}{let\ x = E_1\ in\ E_2 \longrightarrow E_2\{E_1/x\}}$$

Hp

LEMMA for any $E \in Lang$, if $E \to F$ and $E \to G$ then $F = G$

Proof By structure induction

$E \equiv let\ x = E_1\ in\ E_2$

Sia $E \to F$ allora $E \equiv let\ x = E_1\ in\ E_2$ e $E_2\{E_1/x\} \to F$. Sia inoltre $E \to G$ ma allora $E_2\{E_1/x\} \to G$. Come faccio a dire che $F = G$?
Sia $E \equiv let\ x = E_1\ in\ E_2$, sia $E \to F$ ma allora ho usato la regola (let CBN) dove $E \to E_2\{E_1/x\} \equiv F$. Sia $E \to G$ ma allora usando la regola (let CBN) $E \to E_2\{E_1/x\} \equiv G$

Applichiamo rule induction

Proof by rule induction su come ho derivato $E \to F$

Casi base

let $E = E_1 + E_2$. Sia $E \to F$

- E_1 non valore $\Rightarrow$ uso s-left
- E_1 valore, E_2 non valore $\Rightarrow$ uso la s-right
- Entrambi valori $\to$ uso la S-add

Una volta dimostrato il lemma, dimostriamo ora il teorema

TEOREMA $\forall E$, $E \to^* v$ and $E \to^* w$, then $v = w$

Proof:

Sia $E \to^* v$ allora $\exists n \geq \phi$ $E \equiv E^0 \to E^1 \to \ \to E^n \equiv v$
Sia $E \to^* w$ allora $\exists m \geq \phi$ $E \equiv E^0 \to E^1 \to \ \to E^m \equiv w$

Supponiamo ($n \leq m$, l'altro caso sarebbe simile)

Ci sono due casi

- $m = \phi$ cioé $E \equiv h$ é possibile che $n > \phi$? No, $n = \phi$ e quindi $E \equiv F \equiv G \equiv h$
- $m > \phi$ primo passo $E \equiv E^0 \to E^1 \to E^2 \to \ \to E^m \to v$ quindi $m = n$, $v \equiv w$

$$\begin{array}{ccccccccc} E \equiv E^0 & \to & E^1 & \to & E^2 & \to & \ \to & E^n & \to w \\ \| ? & & \| ? & & & & & & \end{array}$$

so l'uguaglianza del lemma precedente

Esercizio 5

siamo in una semantica left to right call by name

① $fix(E) \equiv E(fix.E)$? $\mathcal{R} = \{(P,Q)\} \cup Id$

$\downarrow P$ ① $\downarrow Q$ ②

$E(fixE) \equiv E(fix\ E)$ guardo la regola CBN di fix

cado in identità

② $fix(E) \exists E(fix.E)$? La relazione è la stessa di prima

La simulazione vale, ricado nell'Identità

$E;fixE \longrightarrow fix\ E$

$E;fixE \xleftrightarrow{Id} E;fix\ E$

Esercizio 3 [22/06/2016]

3c $\alpha;(skip;E_1 \oplus skip;E_2) \approx_T skip;a;(E_1 \oplus E_2)$

$P = \alpha;(skip;E_1 \oplus skip;E_2)$ $Q = skip;a;(E_1 \oplus E_2)$

① $\downarrow \alpha$ ② $\downarrow skip;a$

$skip;E_1 \oplus skip;E_2$ $E_1 \oplus E_2$

③ $\downarrow skip$ ④ $\downarrow *$

E_1 $E_1 \oplus E_2$

⑥ $\downarrow$ ⑤ $\downarrow E_2$

x $skip$

Cade la bisimulazione, dimostrare le due simulazioni

$\mathcal{R} = \{(P,Q),(P',Q'),(P'',Q'),(P''',Q')\} \cup Id$

Dimostrazione $P \equiv Q$

1. Analizziamo la prima coppia da destra a sinistra (P,Q)

   1.1 Se $P \xrightarrow{\alpha} P' \equiv skip;E_1 \oplus skip;E_2$ allora $\exists Q.\ Q \xrightarrow{skip} Q' \equiv E_1 \oplus E_2$, purché $(P',Q') \in \mathcal{R}$

   rispondo con più passi per allineare lo stato

2. Analizziamo la seconda coppia da destra a sinistra (P',Q')

   2.1 Se $P' \xrightarrow{skip} P'' \equiv E_1$ allora $Q' \xrightarrow{*} Q'$, purché $(P'',Q') \in \mathcal{R}$

   2.2 Se $P' \xrightarrow{skip} P''' = E_2$ allora $Q' \xrightarrow{*} Q'$, purché $(P''',Q') \in \mathcal{R}$

   Qui devo stare per forza fermo perché in small step con choice comincio comunque a valutare ciò che scelgo

3. Analizziamo la terza coppia (P'',Q')

   3.1 Se $P'' \xrightarrow{E_1} skip$ allora $\exists Q''.\ Q' \xrightarrow{E_1} Q'' \equiv skip$, ma $(skip,skip) \in Id$ quindi il caso è banalmente risolto

4. Analizziamo la quarta coppia (P''',Q')

   4.1 Se $P''' \xrightarrow{E_2} skip$ allora $\exists Q''.\ Q' \xrightarrow{E_2} Q'' \equiv skip$, ma $(skip,skip) \in Id$ quindi il caso è banalmente risolto

---

Dimostrazione $P \sqsupseteq Q$ $\mathcal{R} = \{(P,Q),(P,Q'),(P',Q'')\} \cup Id$

1 Analizziamo la prima coppia da destra a sinistra (P,Q)

   1.1 Se $Q \xrightarrow{skip} Q' \equiv \alpha;(E_1 \oplus E_2)$ allora $P \xrightarrow{*} P$, purché $(P,Q') \in \mathcal{R}$

   sto fermo per rispondere a skip

2 Analizziamo la seconda coppia da destra a sinistra (P,Q')

   2.1 Se $Q' \xrightarrow{\alpha} Q'' \equiv E_1 \oplus E_2$ allora $\exists P'.\ P \xrightarrow{\alpha} P'$, purché $(P',Q'') \in \mathcal{R}$

3. Analizziamo la terza coppia da destra a sinistra (P',Q'')

   3.1 Se $Q'' \xrightarrow{E_1} Q''' \equiv skip$ allora $\exists P''.\ P' \xrightarrow{skip;E_1} P'' \equiv skip$, ma $(skip,skip) \in Id$ ed allora il caso è banalmente risolto

   3.2 Se $Q'' \xrightarrow{E_2} Q''' \equiv skip$ allora $\exists P''.\ P' \xrightarrow{skip;E_2} P'' \equiv skip$, ma $(skip,skip) \in Id$ ed allora il caso è banalmente risolto

Esercizio

$\alpha;(skip;E_1 \oplus skip;E_2) \approx_T \alpha;(skip;E_1 \oplus E_2)$

Esercizi 24/01/24

2.

ma la c mi serve, non tipabile

devo tagliare la c

$$\Gamma \vdash \text{let } f:T_1 = \text{fun}\,(x:T_2) \Rightarrow \text{ref}\{p=\#ax,\ q=(\#bx)\geq 0,\ r=\#cx\} \text{ in } \big((\text{fun}(g:\text{ref}\{p:\text{int}, r:\text{int}\}) \Rightarrow \#qg)\,(f\{a=5, b=6, c=7\}) \wedge (\text{fun}(g:\text{ref}\{p:\text{int}, r:\text{int}\}) \Rightarrow (\#pg) \leq (\#rg))\,(f\{a=7, b=8\})\big) :$$

$T_1 = T_2 \rightarrow \text{ref } T_{rec}$

$T_2 = \{a:\text{int}, b:\text{int}\} = T_{2\,dx}$
per ramo destro

$T_2 = \{a:\text{int}, b:\text{int}\} = T_{2\,sx}$
per ramo sinistro

Ecco qui evidenziati i problemi
il tipo T_2 non combacia poiché
è necessario il campo g per
fare r = #cx, invece devo
tagliarlo per far combaciare
gli input di f

Inoltre manca il campo q in
fun g per eseguire #qg

$$\Gamma' \vdash \{p=\#ax,\ q=(\#bx)\geq 0,\ r=\#cx\} : T_{rec}$$

$$\Gamma'_1 x:T_2 \vdash \text{ref}\{p=\#ax,\ q=(\#bx)\geq 0,\ r=\#cx\} : \text{ref } T_{rec}$$

$$\Gamma \vdash \text{fun}\,(x:T_2) \Rightarrow \text{ref}\{p=\#ax,\ q=(\#bx)\geq 0,\ r=\#cx\} : T_1 \cancel{T_2} \rightarrow T_{rec}$$

$$\Gamma \vdash \text{let } f:T_1 = \text{fun}\,(x:T_2) \Rightarrow \text{ref}\{p=\#ax,\ q=(\#bx)\geq 0,\ r=\#cx\} \text{ in } \big((\text{fun}(g:\text{ref}\{p:\text{int}, r:\text{int}\}) \Rightarrow \#qg)\,(f\{a=5, b=6, c=7\}) \wedge (\text{fun}(g:\text{ref}\{p:\text{int}, r:\text{int}\}) \Rightarrow (\#pg) \leq (\#rg))\,(f\{a=7, b=8\})\big) : \text{bool}$$

$$\Gamma'(x) = T_{2\,sx} \quad \Gamma'(x) = T_{2\,sx} \quad \Gamma'(x) = T_{2\,sx}$$

var, recproj, op≥, recproj, var, rec proj, record, ref, fun

$$\Gamma'''(g) : \text{ref}\{p:\text{int}, r:\text{int}\}$$

$$\Gamma'''_1 g:\text{ref}\{p:\text{int}, r:\text{int}\} \vdash \#qg : T_B$$

rec proj, fun, app

$$\Gamma'' \vdash 5:\text{int} \quad \Gamma'' \vdash 5:\text{int} \quad \Gamma'' \vdash 5:\text{int}$$

int, int, int

$$\Gamma''(f) = T_2 \quad \Gamma'' \vdash \{a=5, b=6, c=7\} : T_D \quad T_D \leq T_C$$

rec, sub

$$\Gamma'' \vdash f : T_C \rightarrow T_A \quad \Gamma'' \vdash \{a=5, b=6, c=7\} : T_C$$

var, app

$$\Gamma'' \vdash f\{a=5, b=6, c=7\} : T_A$$

$$\Gamma''_1 g:\text{ref}\{p:\text{int}, r:\text{int}\} \vdash (\#pg) \leq (\#rg) : \text{bool}$$

fun

$$\Gamma'' \vdash \text{fun}\,(g:\text{ref}\{p:\text{int}, r:\text{int}\}) \Rightarrow (\#pg) \leq (\#rg) : T_D \rightarrow \text{bool}$$

$$\Gamma'' \vdash (\text{fun}(g:\text{ref}\{p:\text{int}, r:\text{int}\}) \Rightarrow \#qg)\,(f\{a=5, b=6, c=7\}) : \text{bool}$$

$$\Gamma'' \vdash (\text{fun}(g:\text{ref}\{p:\text{int}, r:\text{int}\}) \Rightarrow (\#pg) \leq (\#rg))\,(f\{a=7, b=8\}) : \text{bool}$$

app, app∧

$$\Gamma''(f) = T_1 \quad \Gamma'' \vdash 7:\text{int} \quad \Gamma'' \vdash 8:\text{int}$$

record

$$\Gamma'' \vdash f : T_{2\,dx} \rightarrow T_D \quad \Gamma'' \vdash \{a=7, b=8\} : T_{2\,dx}$$

app, app

$$\Gamma'' \vdash f\{a=7, b=8\} : T_D$$

$$\Gamma'_1 f:T_2 \vdash (\text{fun}(g:\text{ref}\{p:\text{int}, r:\text{int}\}) \Rightarrow \#qg)\,(f\{a=5, b=6, c=7\}) \wedge (\text{fun}(g:\text{ref}\{p:\text{int}, r:\text{int}\}) \Rightarrow (\#pg) \leq (\#rg))\,(f\{a=7, b=8\}) : \text{bool}$$

let

Esercizio 3:

(1) $\alpha_3(\beta_3 E_1 \parallel \beta_3 E_2) \approx_p skip_3\, \alpha_3 \beta_3 (E_1 \parallel E_2)$

$\downarrow a \quad r=9$ $\qquad$ $\downarrow skip_3; a \quad r=9$

$\cancel{P \sqsubseteq Q}$

$\beta_3 E_1 \parallel \beta_3 E_2$ $\qquad$ $\beta_3(E_1 \parallel E_2)$

$\downarrow \beta \quad r=9, m=8$ $\qquad$ $\downarrow \beta \quad r=9; m=8$

$E_1 \parallel \beta_3 E_2$ $\qquad$ $E_1 \parallel E_2$

$\downarrow E_1 \quad r=9; m=7$ $\qquad$ $\downarrow E_1 \quad r=9; m=7$

$skip \parallel \beta_3 E_2$ $\qquad$ $skip \parallel E_2$

$\downarrow \beta \quad r=9; m=8$ $\qquad$ $\downarrow$ $\times$ non posso allineare lo stato

$skip \parallel E_2$

$E_1: \; m:=7$

$E_2: \; l:=8$

$\alpha: \; r=9$

$\beta: \; m=8$

$P \sqsupseteq Q$

$\alpha_3(\beta_3 E_1 \parallel \beta_3 E_2) \approx_p skip_3\, \alpha_3 \beta_3 (E_1 \parallel E_2)$

Vale $P \sqsupseteq Q$

Non vale la bisimulazione poiché cade $P \sqsubseteq Q$

$\mathcal{R} = \{ (P,Q), (P,Q'), (P',Q'') \} \in Id$

Proof $P \sqsupseteq Q$

1 Analizzo la 1° coppia (P,Q) da dx a sx

1.1 Se $Q \xrightarrow{skip} Q' \equiv \alpha_3 \beta_3 (E_1 \parallel E_2)$ allora $P \xrightarrow{\alpha} P \equiv \alpha_3(\beta_3 E_1 \parallel \beta_3 E_2)$, purché $(P,Q') \in \mathcal{R}$

2 Analizzo la 2 coppia (P,Q') da dx a sx

2.1 Se $Q' \xrightarrow{\alpha} Q'' \equiv \beta_3(E_1 \parallel E_2)$ allora $\exists P'.\ P \xrightarrow{\alpha} P' \equiv \beta_3 E_1 \parallel \beta_3 E_2$, purché $(P', Q'') \in Id$

3 Analizzo la 3 coppia (P',Q'') da dx a sx

3.1 Se $Q'' \xrightarrow{\beta} Q''' \equiv E_1 \parallel E_2$ allora $\exists P''.\ P' \xrightarrow{\beta} P'' \equiv E_1 \parallel E_2$, ma Q''' e $P'' \in Id$ quindi il caso è banalmente risolto

(B) await true protect $(E_1 \parallel E_2)$ end $\approx_p$ await true protect (E_2) end $\parallel$ await true protect (E_1) end

$\downarrow$ skip $\quad m=2$, $m=3$ $\qquad$ $\downarrow E_2 \quad m=3$

$\downarrow E_1$

$\times$

$E_1: \; m=2$

$E_2: \; m=3$

$\cancel{P \sqsubseteq Q}$

await true protect $(E_1 \parallel E_2)$ end $\approx_p$ await true protect (E_2) end $\parallel$ await true protect (E_1) end

$P \sqsubseteq Q$

La bisimulazione non vale perché $P \sqsupseteq Q$ non vale

1 Analizzo la prima coppia da sx a dx

1.1 Se $P \xrightarrow{await} P' \equiv skip$ allora $\exists Q.\ Q \xrightarrow{await\, E_1\ \ await\, E_2} Q' \equiv skip$, ma $(P',Q') \in Id$ perciò il caso è banalmente risolto

Posso scambiare l'ordine in base a cosa P consuma prima

Proposta 1

ⓒ $\alpha;(E_1 \oplus \alpha;E_2) \approx_r \alpha;(skip;(E_1 \oplus skip;E_2)) \oplus \alpha;E_2$

$P \sqsubseteq Q$ ✓

$Q \sqsubseteq P$ ✓

entrambe le simulazioni sembrano valere, valuto la bisimulazione

$\alpha;(E_1 \oplus \alpha;E_2) \approx_r \alpha;(skip;(E_1 \oplus skip;E_2)) \oplus \alpha;E_2$

② $\downarrow \alpha$ $\qquad$ $\downarrow$ ① α

$E_1 \oplus \alpha;E_2$ $\qquad$ E_2

③ $\downarrow E_1$ $\qquad$ $\downarrow$

$skip$

$\mathcal{R} = \{(P,Q),(P',Q'),(P'',Q'')\dots\} \cup Id$

Caso bisimulazione

1 Analizziamo la prima coppia (P,Q) da sinistra verso destra

1.1 Se $P \xrightarrow{\alpha} P' \equiv E_1 \oplus \alpha;E_2$ allora $\exists Q'$. $Q \xrightarrow{\alpha} Q' \equiv skip;(E_1 \oplus skip;E_2)$, purché $(P',Q') \in \mathcal{R}$

2 Analizziamo la seconda coppia (P',Q') da sinistra verso destra

2.1 Se $P' \xrightarrow{E_1} P'' \equiv skip$ allora $\exists Q''$. $Q' \xrightarrow{skip\ E_1} Q'' \equiv skip$, ma $(P'',Q'') \in Id$ quindi il caso è banalmente risolto

2.2 Se $P' \xrightarrow{\alpha} P''' \equiv E_2$ allora $\exists Q'''$. $Q' \xrightarrow{skip} Q''' \equiv E_2$, ma $(P''',Q''') \in Id$ quindi il caso è banalmente risolto

Dimostriamo $P \sqsupseteq Q$

$\mathcal{R} = \{(P,Q),(P',Q'),(P'',Q''),(P',Q''')\} \cup Id$

1. Analizziamo la prima coppia (P,Q) da destra verso sinistra

1.1 Se $Q \xrightarrow{\alpha} Q' \equiv skip;(E_1 \oplus skip;E_2)$ allora $\exists P'$. $P \to P' \equiv E_1 \oplus \alpha;E_2$, purché $(P',Q') \in \mathcal{R}$

1.2 Se $Q \xrightarrow{\alpha} Q'' \equiv \alpha;E_2$ allora $\exists P''$. $P \xrightarrow{\alpha} P'' \equiv E_1 \oplus \alpha;E_2$, purché $(P'',Q'') \in \mathcal{R}$

2. Analizziamo la seconda coppia (P',Q') da destra verso sinistra

2.1 Se $Q' \xrightarrow{skip} Q''' \equiv E_1 \oplus skip;E_2$ allora $P' \xrightarrow{\ } P'$, purché $(P',Q''') \in \mathcal{R}$

3. Analizziamo la terza coppia (P'',Q'') da destra verso sinistra

3.1 Se $Q'' \xrightarrow{\alpha} Q^{IV} \equiv E_2$ allora $\exists P^{IV}$. $P'' \xrightarrow{\alpha} P^{IV} \equiv E_2$, $(P^{IV},Q^{IV}) \in Id$ quindi il caso è banalmente risolto

4. Analizziamo la quarta coppia (P',Q''') da destra verso sinistra

4.1 Se $Q''' \xrightarrow{E_1} Q^{IV} \equiv skip$ allora $\exists P^{IV}$. $P \xrightarrow{E_1} P^{IV} \equiv skip$, ma $(P^{IV},Q^{IV}) \in Id$ quindi il caso è banalmente risolto

$$\text{Term} = T_1 \to \text{bool}$$

$$T_1 = \text{ref}\{a:\text{bool}\} \to \{a:\text{bool}, b:\text{int}\}$$

Non tipabile manca !

$$\dfrac{\dfrac{\dfrac{\dfrac{\Gamma''(y) = \text{ref}\{b:\text{int}\}}{\Gamma'' \vdash y : \text{ref}\{b:\text{int}\}}\ \text{recordproj} \qquad \dfrac{}{\Gamma'' \vdash 0 : \text{int}}\ \text{int}}{\Gamma'' \vdash \#by : \text{int}}\ \text{bool}}{\Gamma'' = \Gamma', y:\text{ref}\{b:\text{int}\} \vdash \#by > 0 : \text{bool}}\ \text{fun}}{\Gamma' \vdash \text{fn } y:\text{ref}\{b:\text{int}\} \Rightarrow \#by > 0 : \cancel{T_A} \to \text{bool}}\ \text{ref}\{b:\text{int}\}$$

$$\dfrac{\dfrac{\dfrac{\dfrac{\Gamma'(f) = T_1}{\Gamma' \vdash f : \text{ref}\,T_C \to T_B}\ \text{var} \qquad \dfrac{\dfrac{\dfrac{\Gamma' \vdash \text{true} : \text{bool} \quad \Gamma' \vdash \text{false} : \text{bool}}{\Gamma' \vdash \{a=\text{true}, b=\text{false}\} : \cancel{T_B}\ \{a:\text{bool}, b:\text{bool}\}}\ \text{record} \quad \cancel{T_B} <: \cancel{T_C}\ \{a:\text{bool}\}}{\Gamma' \vdash \{a=\text{true}, b=\text{false}\} : \cancel{T_C}\ \{a:\text{bool}\}}\ \text{sub}}{\Gamma' \vdash \text{ref}\{a=\text{true}, b=\text{false}\} : \cancel{T_C}\ \text{ref}\{a:\text{bool}\}}\ \text{ref}}{\Gamma' \vdash f(\text{ref}\{a=\text{true}, b=\text{false}\}) : \cancel{T_B}\ \{a:\text{bool}, b:\text{int}\}}\ \text{app} \quad \cancel{T_B} <: \cancel{T_A}\ \{b:\text{int}\}}{\Gamma' \vdash f(\text{ref}\{a=\text{true}, b=\text{false}\}) : \cancel{T_A}\ \{b:\text{int}\}}\ \text{sub}}{\Gamma' \vdash \text{ref}(f(\text{ref}\{a=\text{true}, b=\text{false}\})) : \text{ref}\,\cancel{T_A}\ \text{ref}\{b:\text{int}\}}\ \text{ref}$$

$$\dfrac{\dfrac{\dfrac{\dfrac{\Gamma''(a) = \text{ref}\{a:\text{bool}\}}{\Gamma'' \vdash a : \text{ref}\{a:\text{bool}\}}\ \text{var}}{\Gamma'' = \Gamma', z:\text{ref}\{a:\text{bool}\} \vdash \#az : \text{bool}}\ \text{recordproj}}{\Gamma' \vdash \text{fn } z:\text{ref}\{a:\text{bool}\} \Rightarrow \#az : \cancel{T_E} \to \cancel{T_F}\ \text{bool}}\ \text{fun} \quad \cancel{T_E} \to \cancel{T_F}\ \{a:\text{bool}\}}{\Gamma' \vdash (\text{fn } z:\text{ref}\{a:\text{bool}\} \Rightarrow \#az)(\text{ref}(f(\text{ref}\{a=\text{false}, b=6, c=7\})) : \text{bool}}\ \text{app}$$

$$\dfrac{\dfrac{\dfrac{\dfrac{\Gamma'(f) = T_1}{\Gamma' \vdash f : T_H \to T_G}\ \text{var} \qquad \dfrac{\dfrac{\dfrac{\dfrac{}{\Gamma' \vdash \text{ff} : \text{bool}}\ \text{bool} \quad \dfrac{}{\Gamma' \vdash 6 : \text{int}}\ \text{int} \quad \dfrac{}{\Gamma' \vdash 7 : \text{int}}\ \text{int}}{\Gamma' \vdash \{a=\text{false}, b=6, c=7\} : \cancel{T_H}\ \{a:\text{bool}, b:\text{int}, c:\text{int}\}}\ \text{sub} \quad \cancel{T_H} <: \cancel{T_I}\ \{a:\text{bool}\}}{\Gamma' \vdash \{a=\text{false}, b=6, c=7\} : \cancel{T_I}\ \{a:\text{bool}\}}\ \text{sub}}{\Gamma' \vdash \text{ref}\{a=\text{false}, b=6, c=7\} : \cancel{T_G}\ \text{ref}\{a:\text{bool}\}}\ \text{ref}}{\Gamma' \vdash f(\text{ref}\{a=\text{false}, b=6, c=7\}) : \cancel{T_G}\ \{a:\text{bool}, b:\text{int}\}}\ \text{app} \quad \cancel{T_G} <: \cancel{T_F}\ \{a:\text{bool}\}}{\Gamma' \vdash f(\text{ref}\{a=\text{false}, b=6, c=7\}) : \cancel{T_F}\ \{a:\text{bool}\}}\ \text{sub}}{\Gamma' \vdash \text{ref}(f(\text{ref}\{a=\text{false}, b=6, c=7\})) : \cancel{T_E}\ \text{ref}\{a:\text{bool}\}}\ \text{ref}$$

$$\text{ref}\{a:\text{bool}\} \to \{a:\text{bool}, b:\text{int}\} = T_H \to T_G$$

$$\dfrac{\dfrac{\dfrac{\Gamma' \vdash (\text{fn } y:\text{ref}\{b:\text{int}\} \Rightarrow \#by > 0)(\text{ref}(f(\text{ref}\{a=\text{true}, b=\text{false}\})) : \text{bool} \qquad \Gamma' \vdash (\text{fn } z:\text{ref}\{a:\text{bool}\} \Rightarrow \#az)(\text{ref}(f(\text{ref}\{a=\text{false}, b=6, c=7\})) : \text{bool}}{\Gamma' = \Gamma, f:T_1 \vdash (\text{fn } y:\text{ref}\{b:\text{int}\} \Rightarrow \#by > 0)(\text{ref}(f(\text{ref}\{a=\text{true}, b=\text{false}\})) \wedge (\text{fn } z:\text{ref}\{a:\text{bool}\} \Rightarrow \#az)(\text{ref}(f(\text{ref}\{a=\text{false}, b=6, c=7\})) : \text{bool}}\ \text{op} \wedge}{\Gamma \vdash \text{fun } (f:T_1) \Rightarrow (\text{fn } y:\text{ref}\{b:\text{int}\} \Rightarrow \#by > 0)(\text{ref}(f(\text{ref}\{a=\text{true}, b=\text{false}\})) \wedge (\text{fn } z:\text{ref}\{a:\text{bool}\} \Rightarrow \#az)(\text{ref}(f(\text{ref}\{a=\text{false}, b=6, c=7\})) : T_1 \to \text{bool}}\ \text{fun}$$

$\alpha ; (\alpha ; E_1 \oplus skip ; E_2) \oplus (\alpha ; E_1) \approx_r \alpha ; (skip ; (skip ; E_1 \oplus \alpha ; E_2)) \oplus (\alpha ; E_2)$

② $\xrightarrow{\text{choice } L \ \alpha}$ $\alpha ; E_1 \oplus skip ; E_2$

③ $\xrightarrow{\text{choice } L \ \alpha}$ E_1

stesso notazione di dx

① $\xrightarrow{\text{choice } R \ \alpha}$ E_2

La regola choice R è

$$\frac{e_2 \to e_2'}{e_1 \oplus e_2 \to e_2'}$$

quindi esegue pure la α

2.

$E \in Lang ::= x \mid n \mid E_1 + E_2 \mid \text{let } x = E_1 \text{ in } E_2$

Semantica big step

(B-Num) $\dfrac{-}{n \Downarrow n}$

x non ha regole perché è placeholder

(B-Add) $\dfrac{e_1 \Downarrow n_1 \quad e_2 \Downarrow n_2}{E_1 + E_2 \Downarrow n_3}$ $n_3 = add(n_1, n_2)$

(B-let CBV) $\dfrac{P \Downarrow m \quad E\{^m/_x\} \Downarrow n}{\text{let } x = P \text{ in } E \Downarrow n}$

Semantica small-step

(S Left) $\dfrac{E_1 \to E_1'}{E_1 + E_2 \to E_1' + E_2}$

(S-N. Right) $\dfrac{E_2 \to E_2'}{n_1 + E_2 \to n_1 + E_2'}$

(S-Add) $\dfrac{-}{n_1 + n_2 \to_{ch} n_3}$ $n_3 = add(n_1, n_2)$

(CBV-let 1) $\dfrac{e_1 \to e_1'}{\text{let } x:T = e_1 \text{ in } e_2 \to \text{let } x:T = e_1' \text{ in } e_2}$

(CBV-let 2) $\dfrac{}{\text{let } x:T = v \text{ in } e_2, s \to e_2\{^v/_x\}}$

Proof by rule induction su come ho derivato $E \Downarrow v$

Caso base 1

1 Se $E \equiv n$ allora $E \Downarrow n$ usando la regola (B-num) e ho $E \equiv n \to^* n$

Caso base 2

2 Se $E \equiv x$ non esiste una regola big step tale che E

Essendo la premessa falsa il caso è per forza vero $\to$ la big step semantic è definita solo su termini chiusi

Caso base 3

3 Se $E \equiv E_1 + E_2$ usando la regola (B-Add) $\dfrac{E_1 \Downarrow n_1 \quad E_2 \Downarrow n_2}{E_1 + E_2 \Downarrow n_3}$ $n_3 = add(n_1, n_2)$

ma allora per ipotesi induttiva sulla profondità dell'albero $E_1 \Downarrow v_1$ ma allora $E_1 \to^* v_1$ e $E_2 \Downarrow v_2$ ma allora $E_2 \to^* v_2$

ma allora $E \equiv E_1 + E_2 \underset{\text{S-left}}{\to^*} v_1 + E_2 \underset{\text{S-right}}{\to^*} v_1 + v_2 \underset{\text{S-add}}{\to} v_3 = add(v_1 + v_2)$

Caso base 4

4. Se $E \equiv \text{let } x = E_1 \text{ in } E_2$ usando la regola (B-let CBV) $\dfrac{E_1 \Downarrow m \quad E_2\{^m/_x\} \Downarrow n}{\text{let } x = E_1 \text{ in } E_2 \Downarrow n}$

ma allora per ipotesi induttiva sulla profondità dell'albero di derivazione $E_1 \Downarrow m$ ma allora $E_1 \to^* m$ e $E_2\{^m/_x\} \Downarrow n$ ma allora $E_2\{^m/_x\} \to^* n$

ma allora $E \equiv \text{let } x = E_1 \text{ in } E_2 \to^* \text{let } x = m \text{ in } E_2 \to^* E_2\{^m/_x\} \to n$

① $\alpha;(skip;E_1 \oplus skip;E_2) \oplus \alpha;E_1 \approx_F \alpha;(skip;(\alpha;E_1 \oplus \alpha;E_2)) \oplus (\alpha;\alpha;E_2)$

$P \sqsubseteq Q$

$P \sqsupseteq Q$

$P \not\sqsubseteq Q$

② $\downarrow \alpha$

$skip;E_1 \oplus skip;E_2$

③ $\downarrow skip$

E_1

⑥ $\downarrow$

$\times$

$E_1 : m := 4$

$E_2 : m := 5$

$\alpha : m := 6$

Cade Bisimulazione

① $\downarrow \alpha$

$\alpha;E_2$

④ $\downarrow \alpha$

E_2

⑤ $\downarrow E_2$

$skip$

Dimostriamo $P \sqsubseteq Q$

$\mathcal{R} = \{(P,Q),(P',Q')\} \cup Id$

1 Analizziamo la prima coppia da sinistra a destra (P,Q)

1.1 Se $P \xrightarrow{choiceL} P' \equiv skip;E_1 \oplus skip;E_2$ allora $\exists Q'.\ Q \xrightarrow{\alpha}\xrightarrow{skip} Q' \equiv \alpha;E_1 \oplus \alpha;E_2$, perché $(P',Q') \in \mathcal{R}$

1.2 Se $P \xrightarrow{choiceR} P'' \equiv E_1$ allora $\exists Q''.\ Q \xrightarrow{choiceL}\xrightarrow{skip}\xrightarrow{choiceL} Q'' \equiv E_1$, ma $(P'',Q'') \in Id$

2 Analizziamo la seconda coppia da sinistra a destra (P',Q')

2.1 Se $P' \xrightarrow{choiceL} P''' \equiv E_1$ allora $\exists Q'''.\ Q' \xrightarrow{choiceL} Q''' \equiv E_1$, ma $(P''',Q''') \in Id$ quindi il caso é banalmente risolto

2.2 Se $P' \xrightarrow{choiceR} P^{iv} \equiv E_2$ allora $\exists Q^{iv}.\ Q' \xrightarrow{choiceR} Q^{iv} \equiv E_2$, ma $(P^{iv},Q^{iv}) \in Id$ quindi il caso é banalmente risolto

Dimostriamo $P \sqsupseteq Q$

$\mathcal{R} = \{(P,Q),(P',Q'),(P'',Q''),(P',Q''')\} \cup Id$

1 Analizziamo la prima coppia da destra a sinistra (P,Q)

1.1 Se $Q \xrightarrow{choiceL} Q' \equiv skip;(\alpha;E_1 \oplus \alpha;E_2)$ allora $\exists P'.\ P \xrightarrow{\alpha} P' \equiv (skip;E_1 \oplus skip;E_2)$, perché $(P',Q') \in \mathcal{R}$

1.2 Se $Q \xrightarrow{choiceR} Q'' \equiv \alpha;E_2$ allora $\exists P''.\ P \xrightarrow{\alpha} P'' \equiv (skip;E_1 \oplus skip;E_2)$, perché $(P'',Q'') \in \mathcal{R}$

2 Analizziamo la seconda coppia da destra a sinistra (P',Q')

2.1 Se $Q' \xrightarrow{skip} Q''' \equiv (\alpha;E_1 \oplus \alpha;E_2)$ allora $P' \xrightarrow{\alpha} P' \equiv (skip;E_1 \oplus skip;E_2)$, perché $(P',Q''') \in \mathcal{R}$

3 Analizziamo la terza coppia da destra a sinistra (P'',Q'')

3.1 Se $Q \xrightarrow{choiceR} Q^{iv} \equiv E_2$ allora $\exists P^{iv}.\ P'' \xrightarrow{choiceR} P^{iv} \equiv E_2$, ma $(Q^{iv},P^{iv}) \in Id$ allora il caso é banalmente risolto

4 Analizziamo la quarta coppia da destra a sinistra (P',Q''')

4.1 Se $Q''' \xrightarrow{choiceL} Q^{v} \equiv E_1$ allora $\exists P^{v}.\ P' \xrightarrow{choiceL} P^{v} \equiv E_1$ ma $(Q^{v},P^{v}) \in Id$ allora il caso é banalmente risolto

4.2 Se $Q''' \xrightarrow{choiceR} Q^{vi} \equiv E_2$ allora $\exists P^{vi}.\ P' \xrightarrow{choiceR} P^{vi} \equiv E_2$ ma allora $(Q^{vi},P^{vi}) \in Id$ allora il caso e banalmente risolto

$$\alpha;(\alpha;E_1 \parallel \alpha;E_2) \simeq \alpha;\alpha;E_1 \parallel \alpha;\alpha;E_2$$

$$\downarrow \alpha \qquad\qquad \downarrow \alpha$$

$P \not\simeq Q$ $\quad \alpha;E_1 \parallel \alpha;E_2 \qquad \downarrow \alpha$

$\qquad\qquad\qquad\qquad$ Identità $\quad \alpha;E_1 \parallel \alpha;E_2$

$P \simeq Q$

$$\alpha;(\alpha;E_1 \parallel \alpha;E_2) \simeq \alpha;\alpha;E_1 \parallel \alpha;\alpha;E_2$$

$$\downarrow \alpha \qquad\qquad \downarrow \alpha$$

$$\alpha;E_1 \parallel \alpha;E_2 \qquad\qquad \alpha;\alpha;E_1 \parallel \alpha;E_2$$

$$\downarrow \alpha \qquad\qquad \downarrow \alpha$$

$$\alpha;E_1 \parallel E_2 \qquad\qquad \alpha;\alpha;E_1 \parallel E_2$$

$$\downarrow \ell:=1 \qquad\qquad \downarrow$$

$$\alpha;E_1 \parallel \ell:=2 \qquad\qquad \alpha;\alpha;E_1 \parallel \ell=2$$

$$\downarrow \alpha \qquad\qquad \downarrow \alpha$$

$$E_1 \parallel \ell:=2 \qquad\qquad \alpha;E_1 \parallel \ell:=2$$

$$\downarrow \ell:=2 \qquad\qquad \downarrow \ell:=2$$

$$E_1 \qquad\qquad \alpha;E_1$$

$$\downarrow \qquad\qquad \downarrow \alpha$$

$$\times$$

$\alpha: \quad \ell=3$

$E_2: \quad \ell:=1;\ \ell:=2$

$E_1: \quad \ell:=0;$

$E \in Cang ::= x \mid n \mid E_1 + E_2 \mid fn\ x \Rightarrow E \mid E_1\ E_2$

Semantica small step

x non ha regola è un place holder

(S Left) $$\frac{E_1 \to E_1''}{E_1 + E_2 \to E_1' + E_2}$$

(S-N. Right) $$\frac{E_2 \to E_2'}{n_1 + E_2 \to n_1 + E_2'}$$

(S-Add) $$\frac{-}{n_1 + n_2 \to_{ch} n_3} \quad n_3 = add(n_1, n_2)$$

(CBV fn) $$\frac{-}{(fn\ x{:}T \to e)\,v \to e\{v/x\}}$$

(CBV app 1) $$\frac{e_1 \to e_1'}{e_1\ e_2 \to e_1'\ e_2}$$

(CBV app 2) $$\frac{e_2 \to e_2'}{v e_2 \to v e_2'}$$

$T_1 \equiv T_2 \to \text{ref } T_{REC}$

$T_2 = \text{ref}\{a:\text{int}, b:\text{bool}\}$

Non tipabile

T_2 deve essere un record

$T_2 \to \text{ref}$

T_2 deve essere una funzione

$\text{ref}\{a:\text{int}, b:\text{bool}\}$

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\Gamma'(x)=T_2}{\Gamma' \vdash x : \text{ref}\{a:\text{int}, b:\text{bool}\}}\,\text{var}}{\Gamma' \vdash !x : \{a:\text{int}, b:\text{bool}\}}\,\text{deref}}{\Gamma' \vdash \#a!x : \text{int}}\,\text{recproj} \quad \dfrac{\dfrac{\dfrac{\Gamma'(x)=T_2}{\Gamma' \vdash x : \text{ref}\{a:\text{int}, b:\text{bool}\}}\,\text{var}}{\Gamma' \vdash !x : \{a:\text{int}, b:\text{bool}\}}\,\text{deref}}{\Gamma' \vdash \#a!x : \text{int}}\,\text{recproj} \quad \dfrac{}{\Gamma' \vdash 0 : \text{int}}}{\Gamma' \vdash \#a!x \geq 0 : \text{bool}}\,\text{op}_{\geq} \quad \dfrac{\dfrac{\dfrac{\Gamma'(x)=T_2}{\Gamma' \vdash x : \text{ref}\{a:\text{int}, b:\text{bool}\}}\,\text{var}}{\Gamma' \vdash !x : \{a:\text{int}, b:\text{bool}\}}\,\text{deref}}{\Gamma' \vdash \#b!x : \text{int}}\,\text{recproj}}{\Gamma' \vdash \{a = \#a!x,\ b = (\#a!x) \geq 0,\ c = \#b!x\} : T_A}\,\text{rec} \qquad T_A <: T_{REF}}{\Gamma' \vdash \{a = \#a!x,\ b = (\#a!x) \geq 0,\ c = \#b!x\} : T_{REF}}$$

$$\dfrac{\dfrac{\Gamma'' = \Gamma', x:T_2 \vdash \text{ref}\{a = \#a!x,\ b = (\#a!x) \geq 0,\ c = \#b!x\} : \text{ref } T_{REF}}{\Gamma \vdash \text{fun}(x:T_2) \Rightarrow \text{ref}\{a = \#a!x,\ b = (\#a!x) \geq 0,\ c = \#b!x\} : T_1}\,\text{ref}}{}\,\text{fun}$$

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\Gamma''(g) = \{b:\text{bool}, a:\text{int}\}}{\Gamma''' \vdash g : \{b:\text{bool}, a:\text{int}\}}\,\text{var}}{\Gamma''' = \Gamma'', g:\{b:\text{bool}, a:\text{int}\} \vdash \#bg : \text{bool}}\,\text{recproj}}{\Gamma'' \vdash \text{fun}(g:\{b:\text{bool}, a:\text{int}\}) = \#bg : T_B \to \text{bool}}\,\text{fun} \quad \dfrac{\dfrac{\dfrac{\dfrac{\Gamma''(f) = T_1}{\Gamma'' \vdash f : T_C \to \text{ref } T_B \to T_C \to \text{ref } T_B}\,\text{fix} \quad \Gamma'' \vdash \text{ref}\{a=5, b=\text{true}, c=\text{false}\} : T_C}{\Gamma'' \vdash (\text{fix } f) : T_C \to \text{ref } T_B}\,\text{app}}{\Gamma'' \vdash (\text{fix } f)(\text{ref}\{a=5, b=\text{true}, c=\text{false}\}) : \text{ref } T_B}\,\text{app}}{\Gamma'' \vdash !(\text{fix } f)(\text{ref}\{a=5, b=\text{true}, c=\text{false}\}) : T_B}\,\text{deref}}{\Gamma'' \vdash (\text{fun}(g:\{b:\text{bool}, a:\text{int}\}) = \#bg)\ (!(\text{fix } f)(\text{ref}\{a=5, b=\text{true}, c=\text{false}\})) : \text{bool}}\,\text{app}}{\Gamma'' = \Gamma, f:T_1 \vdash ((\text{fun}(g:\{b:\text{bool}, a:\text{int}\}) = \#bg)\ (!(\text{fix } f)(\text{ref}\{a=5, b=\text{true}, c=\text{false}\})) \wedge (\text{fun}(g:\text{ref}\{a:\text{int}, b:\text{bool}\}) \Rightarrow ((\#ag) \geq 0) \vee (\#bg))\ (!(\text{fix } f)(\text{ref}\{a=7, b=\text{false}, c=8\}))) : \text{bool}}\,\text{op}_{\wedge}$$

$$\text{let } f:T_1 = (\text{fun}(x:T_2) \Rightarrow \text{ref}\{a = \#a!x,\ b = (\#a!x) \geq 0,\ c = \#b!x\}) \text{ in } ((\text{fun}(g:\{b:\text{bool}, a:\text{int}\}) = \#bg)\ (!(\text{fix } f)(\text{ref}\{a=5, b=\text{true}, c=\text{false}\})) \wedge (\text{fun}(g:\text{ref}\{a:\text{int}, b:\text{bool}\}) \Rightarrow ((\#ag) \geq 0) \vee (\#bg))\ (!(\text{fix } f)(\text{ref}\{a=7, b=\text{false}, c=8\}))) : \text{bool}$$

let

1

$\Gamma \vdash$ let f : T_1 = fun (g : T_2) => fun (h : T_3) => ref { a = (#b!h), b = true, c = (#b!h) }

Se $\forall E$

$\exists n \geq 0 \; . \quad E^0 \to E^1 \to \dots \to E^n = v$

$\exists v \geq 0 \; . \quad E^0 \to E^1 \to \dots \to E^m = w$

Supponiamo $m < n$ (l'altro caso sarebbe simmetrico)

CASO BASE

$n = \emptyset$ ma è possibile che $m > 0$? No perciò $E = v = w = h$

$n > \emptyset$ allora

$$\begin{array}{ccccccc} E^0 & \to & E^1 & \to & \dots\dots & \to & E^n = v \\ \| & & \| & & & & \| \\ E^0 & \to & E^1 & \to & \quad \| \quad & \to & \quad \| \quad \to E^m = w \end{array}$$

ho quindi $m = n$ e $v = w$

Esercizi

Mila

Esercizio 1

while b^{P_1} do c ~ if b then $c;w^{P_2}$ else skip

Dimostrazione con semantica operazionale

$$\frac{\langle b,\sigma\rangle \to false}{\langle while\ b\ do\ c,\sigma\rangle \to \sigma} \quad \Longleftrightarrow \quad \frac{\langle b,\sigma\rangle \to false \qquad \langle skip,\sigma\rangle \to \sigma}{\langle if\ b\ then\ c;w\ else\ skip\rangle \to \sigma}$$

$$\frac{\langle b,\sigma\rangle \to true \qquad \langle c,\sigma\rangle \to \sigma'' \qquad \langle while\ b\ do\ c,\sigma''\rangle \to \sigma'}{\langle while\ b\ do\ c,\sigma\rangle \to \sigma'} \quad \Longleftrightarrow \quad \frac{\langle b,\sigma\rangle \to true \qquad \dfrac{\langle c,\sigma\rangle \to \sigma'' \qquad \langle w,\sigma''\rangle \to \sigma'}{\langle c;w,\sigma\rangle \to \sigma'}}{\langle if\ b\ then\ c;w\ else\ skip\rangle \to \sigma'}$$

Esercizio 2

if b then c_1 else c_2 ~ if $\neg b$ then c_2 else c_1

Dimostrazione con semantica operazionale

$$\frac{\langle b,\sigma\rangle \to true \qquad \langle c_1,\sigma\rangle \to \sigma'}{\langle if\ b\ then\ c_1\ else\ c_2,\sigma\rangle \to \sigma'} \qquad \frac{\dfrac{\langle b,\sigma\rangle \to true}{\langle \neg b,\sigma\rangle \to false} \qquad \langle c_1,\sigma\rangle \to \sigma'}{\langle if\ \neg b\ then\ c_2\ else\ c_1,\sigma\rangle \to \sigma'}$$

$$\frac{\langle b,\sigma\rangle \to false \qquad \langle c_2,\sigma\rangle \to \sigma'}{\langle if\ b\ then\ c_1\ else\ c_2,\sigma\rangle \to \sigma'} \qquad \frac{\dfrac{\langle b,\sigma\rangle \to false}{\langle \neg b,\sigma\rangle \to true} \qquad \langle c_2,\sigma\rangle \to \sigma'}{\langle if\ \neg b\ then\ c_2\ else\ c_1,\sigma\rangle \to \sigma'}$$

Dimostrazione con semantica denotazionale

$$\mathcal{C}[\![if\ b\ then\ c_1\ else\ c_2]\!] = \{(\sigma,\sigma') \mid \mathcal{B}[\![b]\!]\sigma = true,\ \mathcal{C}[\![c_1]\!]\sigma = \sigma'\} \cup \{(\sigma,\sigma') \mid \mathcal{B}[\![b]\!]\sigma = false,\ \mathcal{C}[\![c_2]\!] = \sigma'\}$$

$$\mathcal{C}[\![if\ \neg b\ the\ c_2\ else\ c_1]\!] = \{(\sigma,\sigma') \mid \mathcal{B}[\![\neg b]\!]\sigma = true,\ \mathcal{C}[\![c_2]\!]\sigma = \sigma'\} \cup \{(\sigma,\sigma') \mid \mathcal{B}[\![\neg b]\!] = false,\ \mathcal{C}[\![c_1]\!]\sigma = \sigma'\}$$

$$= \{(\sigma,\sigma') \mid \mathcal{B}[\![b]\!] = false,\ \mathcal{C}[\![c_2]\!]\sigma = \sigma'\} \cup \{(\sigma,\sigma') \mid \mathcal{B}[\![c_1]\!]\sigma = true,\ \mathcal{C}[\![c_1]\!]\sigma = \sigma'\}$$

$$\mathcal{C}[\![if\ b\ then\ c_1\ else\ c_2]\!] =$$

Esercizio 3

Dimostrazione con semantica operazionale

$$\frac{\langle b,\sigma\rangle \to true \qquad \dfrac{\langle c_1,\sigma\rangle \to \sigma'' \qquad \langle c,\sigma''\rangle \to \sigma'}{\langle c_1;c,\sigma\rangle \to \sigma'}}{\langle if\ b\ then\ c_1;c\ else\ c_2;c,\sigma\rangle \to \sigma'} \quad \Longleftrightarrow \quad \frac{\dfrac{\langle b,\sigma\rangle \to true \qquad \langle c_1,\sigma\rangle \to \sigma''}{\langle if\ b\ then\ c_1\ else\ c_2,\sigma\rangle \to \sigma''} \qquad \langle c,\sigma''\rangle \to \sigma'}{\langle if\ b\ then\ c_1\ else\ c_2;c,\sigma\rangle \to \sigma'}$$

$$\frac{\langle b,\sigma\rangle \to false \qquad \dfrac{\langle c_2,\sigma\rangle \to \sigma'' \qquad \langle c,\sigma''\rangle \to \sigma'}{\langle c_2;c,\sigma\rangle \to \sigma'}}{\langle if\ b\ then\ c_1;c\ else\ c_2;c,\sigma\rangle \to \sigma'} \quad \Longleftrightarrow \quad \frac{\dfrac{\langle b,\sigma\rangle \to false \qquad \langle c_2,\sigma\rangle \to \sigma''}{\langle if\ b\ then\ c_1\ else\ c_2,\sigma\rangle \to \sigma''} \qquad \langle c,\sigma''\rangle \to \sigma'}{\langle if\ b\ then\ c_1\ else\ c_2;c,\sigma\rangle \to \sigma'}$$

Dimostrazione con semantica operazionale

$$\mathcal{C}[\![\text{if } b \text{ then } c_1;c \text{ else } c_2;c]\!] = \{(\sigma,\sigma') \mid \mathcal{B}[\![b]\!]\sigma = \text{true},\ \mathcal{C}[\![c_1;c]\!]\sigma = \sigma'\} \cup \{(\sigma,\sigma') \mid \mathcal{B}[\![b]\!]\sigma = \text{false},\ \mathcal{C}[\![c_2;c]\!]\sigma = \sigma'\}$$

$$= \{(\sigma,\sigma') \mid \mathcal{B}[\![b]\!]\sigma = \text{true},\ \mathcal{C}[\![c_1]\!]\sigma = \sigma'',\ \mathcal{C}[\![c]\!]\sigma'' = \sigma'\} \cup \{(\sigma,\sigma') \mid \mathcal{B}[\![b]\!]\sigma = \text{false},\ \mathcal{C}[\![c_2]\!]\sigma = \sigma'',\ \mathcal{C}[\![c]\!]\sigma'' = \sigma'\}$$

$$= \{(\sigma'',\sigma') \mid \mathcal{B}[\![b]\!]\sigma = \text{true},\ \mathcal{C}[\![c_1]\!]\sigma = \sigma''\} \cup \{(\sigma,\sigma'') \mid \mathcal{B}[\![b]\!]\sigma = \text{false},\ \mathcal{C}[\![c_2]\!]\sigma = \sigma''\}$$

$$= \mathcal{C}[\![c]\!]\sigma''$$

$$= \mathcal{C}[\![\text{if } b \text{ then } c_1 \text{ else } c_2;c,\sigma]\!]$$

$$\text{if } b_1 \text{ then } c_1 \text{ else } (\text{if } \neg b_2 \text{ then } c_2 \text{ else } c_1) \equiv \text{if } b_1 \vee b_2 \text{ then } c_1 \text{ else } c_2$$

Dimostrazione con semantica operazionale

$$\frac{\langle b_1,\sigma\rangle \to \text{true} \qquad \langle c_1,\sigma\rangle \to \sigma'}{\langle \text{if } b_1 \text{ then } c_1 \text{ else } (\text{if } \neg b_2 \text{ then } c_2 \text{ else } c_1)\rangle \to \sigma'}$$

$$\frac{\langle b_1,\sigma\rangle \to \text{true}}{\langle b_1 \vee b_2,\sigma\rangle \to \text{true}}\ - \qquad \langle c_1,\sigma\rangle \to \sigma'$$

vale per $b_2 = \text{True}$

vale per $b_2 = \text{false}$

$$\frac{\langle b_1 \vee b_2,\sigma\rangle \to \text{true} \qquad \langle c_1,\sigma\rangle \to \sigma'}{\langle \text{if } b_1 \vee b_2 \text{ then } c_1 \text{ else } c_2,\sigma\rangle \to \sigma'}$$

$$\frac{\langle b_1,\sigma\rangle \to \text{false} \qquad \dfrac{\dfrac{\langle b_2,\sigma\rangle \to \text{false}}{\langle \neg b_2,\sigma\rangle \to \text{true}} \qquad \langle c_2,\sigma\rangle \to \sigma'}{\langle \text{if } \neg b_2 \text{ then } c_2 \text{ else } c_1,\sigma\rangle \to \sigma'}}{\langle \text{if } b_1 \text{ then } c_1 \text{ else } (\text{if } \neg b_2 \text{ then } c_2 \text{ else } c_1)\rangle \to \sigma'}$$

$$\frac{\langle b_1,\sigma\rangle \to \text{false} \qquad \langle b_2,\sigma\rangle \to \text{false}}{\langle b_1 \vee b_2,\sigma\rangle \to \text{false}} \qquad \langle c_2,\sigma\rangle \to \sigma'$$

$$\frac{\langle b_1 \vee b_2,\sigma\rangle \to \text{false} \qquad \langle c_2,\sigma\rangle \to \sigma'}{\langle \text{if } b_1 \vee b_2 \text{ then } c_1 \text{ else } c_2,\sigma\rangle \to \sigma'}$$

$$\frac{\langle b_1,\sigma\rangle \to \text{false} \qquad \dfrac{\dfrac{\langle b_2,\sigma\rangle \to \text{true}}{\langle \neg b_2,\sigma\rangle \to \text{false}} \qquad \langle c_1,\sigma\rangle \to \sigma'}{\langle \text{if } \neg b_2 \text{ then } c_2 \text{ else } c_1,\sigma\rangle \to \sigma'}}{\langle \text{if } b_1 \text{ then } c_1 \text{ else } (\text{if } \neg b_2 \text{ then } c_2 \text{ else } c_1)\rangle \to \sigma'}$$

$$\frac{\langle b_1,\sigma\rangle \to \text{false} \qquad \langle b_2,\sigma\rangle \to \text{true}}{\langle b_1 \vee b_2,\sigma\rangle \to \text{false}} \qquad \langle c_1,\sigma\rangle \to \sigma'$$

$$\frac{\langle b_1 \vee b_2,\sigma\rangle \to \text{false} \qquad \langle c_1,\sigma\rangle \to \sigma'}{\langle \text{if } b_1 \vee b_2 \text{ then } c_1 \text{ else } c_2,\sigma\rangle \to \sigma'}$$

Dimostrazione con semantica operazionale

$$= \mathcal{C}[\![\text{if } b_1 \text{ then } c_1 \text{ else } (\text{if } \neg b_2 \text{ then } c_2 \text{ else } c_1)]\!]$$

$$= \{(\sigma,\sigma') \mid \mathcal{B}[\![b_1]\!]\sigma = \text{true},\ \mathcal{C}[\![c_1]\!]\sigma = \sigma'\} \cup \{(\sigma,\sigma') \mid \mathcal{B}[\![b_1]\!]\sigma = \text{false},\ \mathcal{C}[\![\text{if } \neg b_2 \text{ then } c_2 \text{ else } c_1]\!]\}$$

$$= \{(\sigma,\sigma') \mid \mathcal{B}[\![b_1]\!]\sigma = \text{true},\ \mathcal{C}[\![c_1]\!]\sigma = \sigma'\} \cup \{(\sigma,\sigma') \mid \mathcal{B}[\![b_1]\!]\sigma = \text{false},\ \mathcal{B}[\![\neg b_2]\!] = \text{true},\ \mathcal{C}[\![c_2]\!]\sigma = \sigma'\} \cup \{(\sigma,\sigma') \mid \mathcal{B}[\![b_1]\!]\sigma = \text{false},\ \mathcal{B}[\![\neg b_2]\!] = \text{false},\ \mathcal{C}[\![c_1]\!]\sigma = \sigma'\}$$

$$= \{(\sigma,\sigma') \mid \mathcal{B}[\![b_1]\!]\sigma = \text{true},\ \mathcal{C}[\![c_1]\!]\sigma = \sigma'\} \cup \{(\sigma,\sigma') \mid \mathcal{B}[\![b_1]\!]\sigma = \text{false},\ \mathcal{B}[\![b_2]\!] = \text{false},\ \mathcal{C}[\![c_2]\!]\sigma = \sigma'\} \cup \{(\sigma,\sigma') \mid \mathcal{B}[\![b_1]\!]\sigma = \text{false},\ \mathcal{B}[\![b_2]\!] = \text{true},\ \mathcal{C}[\![c_1]\!]\sigma = \sigma'\}$$

$$= \{(\sigma,\sigma') \mid \mathcal{B}[\![b_1]\!]\sigma = \text{true} \vee \mathcal{B}[\![b_2]\!] = \text{true},\ \mathcal{C}[\![c_1]\!]\sigma = \sigma'\} \cup \{(\sigma,\sigma') \mid \mathcal{B}[\![b_1]\!]\sigma = \text{false} \wedge \mathcal{B}[\![b_2]\!] = \text{false},\ \mathcal{C}[\![c_2]\!]\sigma = \sigma'\}$$

$$= \{(\sigma,\sigma') \mid \mathcal{B}[\![b_1 \vee b_2]\!] = \text{true},\ \mathcal{C}[\![c_1]\!]\sigma = \sigma'\} \cup \{(\sigma,\sigma') \mid \mathcal{B}[\![b_1]\!]\sigma = \text{false} \wedge \mathcal{B}[\![b_2]\!] = \text{false},\ \mathcal{C}[\![c_2]\!]\sigma = \sigma'\}$$

$$= \mathcal{C}[\![\text{if } b_1 \vee b_2 \text{ then } c_1 \text{ else } c_2]\!]$$

Semantica operazionale

$$\frac{\langle b_1,\sigma\rangle \to \text{false}}{\langle \text{while } b_1 \text{ do } c \text{ exit } b_2,\sigma\rangle \to \sigma}$$

$$\frac{\langle b_1,\sigma\rangle \to \text{true} \quad \langle c,\sigma\rangle \to \sigma'' \quad \langle b_2,\sigma\rangle \to \text{true} \quad \langle \text{while } b_1 \text{ do } c \text{ exit } b_2,\sigma''\rangle \to \sigma'}{\langle \text{while } b_1 \text{ do } c \text{ exit } b_2,\sigma\rangle \to \sigma'}$$

$$\frac{\langle b_1,\sigma\rangle \to \text{true} \quad \langle c,\sigma\rangle \to \sigma' \quad \langle b_2,\sigma\rangle \to \text{false}}{\langle \text{while } b_1 \text{ do } c \text{ exit } b_2,\sigma\rangle \to \sigma'}$$

Semantica denotazionale

$$\mathcal{C}[\![\text{while } b \text{ do } c]\!] = \mathcal{C}[\![\text{if } b \text{ then } c; w \text{ else skip}]\!]$$

$$= \mathcal{C}[\![\text{if } b_1 \text{ then } c; \text{if } b \text{ then } w \text{ else skip else skip}]\!]$$

$n:=n+1;$ if $n\neq 0$ then c_1 else c_2 $\equiv$ if $n+1\neq 0$ then $n:=n+1; c_1$ else $n:=n+1; c_2$

Semantica operazionale

$$\frac{\dfrac{\langle n,\sigma\rangle \to \sigma(n) \quad \langle 1,\sigma\rangle \to 1}{\langle n+1,\sigma\rangle \to m}}{\langle n:=n+1,\sigma\rangle \to \sigma''} \qquad \frac{\dfrac{\langle n,\sigma''\rangle \to \sigma(n) \quad \langle 0,\sigma''\rangle \to 0}{\langle n\neq 0,\sigma''\rangle \to \text{True}} \quad \langle c_1,\sigma''\rangle \to \sigma'}{\langle \text{if } n\neq 0 \text{ then } c_1 \text{ else } c_1,\sigma''\rangle \to \sigma'}$$

$$\frac{\langle n:=n+1;\text{if } n\neq 0 \text{ then } c_1 \text{ else } c_2,\sigma\rangle \to \sigma'}{}$$

$$\frac{\dfrac{\dfrac{\langle n,\sigma\rangle \to \sigma(n) \quad \langle 1,\sigma\rangle \to 1}{\langle n+1,\sigma\rangle \to K} \quad \langle 0,\sigma\rangle \to 0}{\langle n+1\neq 0,\sigma\rangle \to \text{True}} \quad \dfrac{\dfrac{\langle n,\sigma\rangle \to \sigma(n) \quad \langle 1,\sigma\rangle \to 1}{\langle n+1,\sigma\rangle \to l}}{\langle n:=n+1,\sigma\rangle \to \sigma''} \quad \langle c_1,\sigma''\rangle \to \sigma'}{\langle n:=n+1; c_1,\sigma\rangle \to \sigma'}$$

$$\frac{\langle \text{if } n+1\neq 0 \text{ then } n:=n+1; c_1 \text{ else } n:=n+1; c_2,\sigma\rangle \to \sigma'}{}$$

$$\begin{aligned}
&\mathcal{C}[\![n:=n+1;\text{if } n\neq 0 \text{ then } c_1 \text{ else } c_2]\!]\\
&= \mathcal{C}[\![\text{if } n\neq 0 \text{ then } c_1 \text{ else } c_2]\!]_{\sigma[\sigma(n)+1/n]}\\
&= \{(\sigma[\sigma(n)+1/n],\sigma') \mid \mathcal{B}[\![n\neq 0]\!]_{\sigma[\sigma(n)+1/n]} = \text{true},\ \mathcal{C}[\![c_1]\!]_{\sigma[\sigma(n)+1/n]} = \sigma'\} \cup \{(\sigma[\sigma(n)+1/n],\sigma') \mid \mathcal{B}[\![n\neq 0]\!]_{\sigma[\sigma(n)+1/n]} = \text{false},\ \mathcal{C}[\![c_2]\!]_{\sigma[\sigma(n)+1/n]} = \sigma'\}\\
&= \{(\sigma,\sigma') \mid \mathcal{B}[\![n+1\neq 0]\!]_\sigma = \text{true},\ \mathcal{C}[\![n:=n+1]\!]_\sigma = \sigma[\sigma(n)+1/n],\ \mathcal{C}[\![c_1]\!]_{\sigma[\sigma(n)+1/n]} = \sigma'\} \cup \{(\sigma,\sigma') \mid \mathcal{B}[\![n+1\neq 0]\!]_\sigma = \text{false},\ \mathcal{C}[\![n:=n+1]\!]_\sigma = \sigma[\sigma(n)+1/n],\ \mathcal{C}[\![c_2]\!]_{\sigma[\sigma(n)+1/n]} = \sigma'\}\\
&= \mathcal{C}[\![\text{if } n+1\neq 0 \text{ then } c_1; n:=n+1 \text{ else } n:=n+1; c_2]\!]
\end{aligned}$$

① Trovo l'invariante del ciclo applicando le iterazioni

$\{x=n \quad n\geq 0 \quad y=1\}$

while $x>0$

$\quad y:=x\cdot y$

$\quad x:=x-1$

$\{y=n!\}$

$x=N \qquad y=x$

$x=N-1 \qquad y=x(x-1)$

$x=N-2 \qquad y=x(x-1)(x-2)$

$\vdots$

$I = n! = y\cdot x! \wedge x\geq 0$

② Dimostro che I è invariante per farlo applico la weakest precondition e dimostro $I\wedge b \to \hat{I}$

per farlo applico ad I le interpretazioni dei comandi del corpo del while e trovo $\hat{I}$, così facendo devo dimostrare che $I\wedge b \equiv \hat{I}$

$$\begin{aligned}
&= I[^{x-1}/_x][^{x\cdot y}/_y]\\
&= n! = y\cdot(x-1)! \wedge (x-1)\geq 0\\
&= n! = x\cdot y\cdot(x-1)! \wedge (x-1)\geq 0
\end{aligned}$$

$$\begin{aligned}
I\wedge b &= n! = y\cdot x! \wedge x\geq 0 \wedge x>0\\
&= n! = y\cdot x\cdot(x-1)! \wedge x>0\\
&= n! = y\cdot x\cdot(x-1)! \wedge x-1\geq 0
\end{aligned}$$

$\equiv$

$I\wedge b \to \hat{I}$ ✓

per la consequence rule I è invariante, infatti

$$\frac{\{\hat{I}\}\ \text{corpo}\ \{I\}^{\checkmark} \quad \{I\wedge b\}\to\hat{I}^{\checkmark}}{\{I\wedge b\}\ c\ \{I\}}$$

possiamo applicare la consequence rule e la regola del while

$$\frac{\dfrac{\{\hat{I}\}\ \text{corpo}\ \{I\} \quad \{I\wedge b\}\to\hat{I}}{\{I\wedge b\}\ c\ \{I\}}}{\{I\}\ \text{while } b \text{ do } c\ \{I\wedge \neg b\}}$$

ma allora devo vedere se $I \wedge \neg b \equiv$ post condizione del programma

$I \wedge \neg b \equiv n! = y \cdot x! \wedge x \geq 0 \wedge x \leq 0$

$\equiv n! = y \cdot x! \wedge x = 0$

$\equiv n! = y$ = post condizione del programma ✓ per la consequence rule

(5) Per concludere bisogna vedere se le precondizioni soddisfano l'invariante

$I = n! = y \cdot x! \wedge x \geq 0$

$\{x = n \quad n \geq 0 \quad y = 1\}$

$x \geq 0$ ✓ $x = n$ con $n \geq 0$

$n! = y \cdot x!$

$n! = 1 \cdot n!$

$n! = n!$ con $n \geq 0$ ✓

$\{0 \le N\}$
$P := 0$
$C := 1$
while $C \le N$
$\quad P := P + M$
$\quad C := C + 1$
$\{P = M \cdot N\}$

- Analizziamo le iterazioni e troviamo l'invariante

$P = M \quad C = 2$
$P = 2M \quad C = 3$
$P = 3M \quad C = 4$

$$I = P = (C-1) \cdot M \ \wedge \ C \le N+1$$

Per dimostrare che I è invariante dobbiamo dimostrare che $I \wedge b \to \hat{I}$
$\hat{I}$ lo troviamo applicando le interpretazioni dei comandi del corpo del while ad I

$$\hat{I} = I[^{C+1}/_C][^{P+M}/_P] = P = (C) \cdot M \ \wedge \ C+1 \le N+1 = P + M = C \cdot M \ \wedge \ C+1 \le N+1$$

$$I \wedge b \equiv P = (C-1) \cdot M \ \wedge \ C \le N+1 \ \wedge \ C \le N \equiv P + M = (C-1) \cdot M + M \ \wedge \ C \le N = P + M = C \cdot M \ \wedge \ C+1 \le N+1$$

$=$

Se ho $I \wedge b = \hat{I}$ allora ho che $I \wedge b \to \hat{I}$, posso applicare la consequence rule e so che I è invariante

$$\frac{\{\hat{I}\}\ \text{corpo}\ \{I\} \qquad I \wedge b \to \hat{I}}{\{I \wedge b\}\ \text{corpo}\ \{I\}}$$

posso allora applicare la regola del while

$$\frac{\dfrac{\{\hat{I}\}\ \text{corpo}\ \{I\} \qquad I \wedge b \to \hat{I}}{\{I \wedge b\}\ \text{corpo}\ \{I\}}}{\{I\}\ \text{while } b \text{ do corpo}\ \{I \wedge \neg b\}}$$

Ma allora devo avere $I \wedge \neg b$ = post condizione del programma

$$I \wedge \neg b \equiv P = (C-1) \cdot M \ \wedge \ C \le N+1 \ \wedge \ C > N \equiv P = (C-1) \cdot M \ \wedge \ C = N+1 = P = N \cdot M \ \wedge \ C = N+1$$

= post condizione ✓ consequence rule

Mostriamo che l'invariante è soddisfatto dalle precondizioni e prima del ciclo

$\{0 \le N\}$
$P := 0$
$C := 1$
$\Rightarrow I$

$$I = P = (C-1) \cdot M \ \wedge \ C \le N+1$$

$P = 0$, $C = 1$ $\quad\downarrow\quad$ $0 = 0 \cdot M$ ✓

$C = 1$, $N \ge 0$ $\quad\searrow\quad$ $1 \le N+1$ ✓ con $N \ge 0$

$\{N \geq 0\}$

```
a=0
b=1
while n>0 do
    b:=a+b
    a:=b-a
    n:=n-1
```

$\{a = fib(n)\}$

Troviamo l'invariante I osservando le iterazioni

$a=1$	$b=1$	$n=N-1$
$a=1$	$b=2$	$n=N-2$
$a=2$	$b=3$	$n=N-3$
$a=3$	$b=5$	$n=N-4$
$a=5$	$b=8$	$n=N-5$
$a=8$	$b=13$	$n=N-6$

$$I = a = fib(N-n) \wedge b = fib(N-n+1) \wedge n \geq 0$$

Adesso per mostrare che I è invariante devo dimostrare che $I \wedge b \rightarrow \hat{I}$ in modo tale da poter applicare la consequence rule. Per calcolare $\hat{I}$ che altro non è che la weakest precondition devo applicare ad I le interpretazioni dei comandi.

$$\hat{I} = I[^{n-1}/_n][^{b-a}/_a][^{b+a}/_b]$$

$$= a = fib(N-n+1) \wedge b = fib(N-n+2) \wedge n-1 \geq 0$$

$$= b-a = fib(N-n+1) \wedge b = fib(N-n+2) \wedge n-1 \geq 0$$

$$= b = fib(N-n+1) \wedge b+a = fib(N-n+2) \wedge n-1 \geq 0$$

$$I \wedge b = a = fib(N-n) \wedge b = fib(N-n+1) \wedge n \geq 0 \wedge n > 0$$

$$= b = fib(N-n+1) \wedge a+b = fib(N-n) + fib(N-n+1) \wedge n > 0$$

$$= b = fib(N-n+1) \wedge b+a = fib(N-n+2) \wedge n-1 \geq 0$$

Avendo trovato $\hat{I} \equiv I \wedge b$ ho dimostrato $I \wedge b \rightarrow \hat{I}$ e allora posso applicare la consequence rule

$$\frac{\{\hat{I}\}\ corpo\ \{I\}^{\checkmark} \qquad I \wedge b \rightarrow \hat{I}^{\checkmark}}{\{I \wedge b\}\ corpo\ \{I\}}$$

ma allora posso applicare la regola del while

$$\frac{\dfrac{\{\hat{I}\}\ corpo\ \{I\}^{\checkmark} \qquad I \wedge b \rightarrow \hat{I}^{\checkmark}}{\{I \wedge b\}\ corpo\ \{I\}}}{\{I\}\ while\ b\ do\ corpo\ \{I \wedge \neg b\}}$$

ma allora $I \wedge \neg b$ deve corrispondere con la post condizione del programma

$$I \wedge \neg b = a = fib(N-n) \wedge b = fib(N-n+1) \wedge n \geq 0 \wedge n \leq 0$$

$$= n = 0 \wedge a = fib(N) \wedge b = (N+1)$$

!!! post condizione ✓ per la consequence rule

$N \geq 0$
$a=0$
$b=1$
$\Longrightarrow I$

$$I = a = fib(N-n) \wedge b = fib(N-n+1) \wedge n \geq 0$$

$$0 = fib(N-n) \wedge 1 = fib(N-n+1)$$

$n = N = 0$ ✓

$0 = fib(0) \wedge 1 = fib(1)$ ✓

$\{n \geq 0\}$

$s = 1$

$x = 1$

while $x \leq n$ do

$\hat{I} \rightarrow$ if $f(x) \geq 0$ then $\overset{A_T}{s := s * f(x)}$

else $\overset{A_F}{s := s * (-f(x))}$

$x := x + 1$

$\{S = \prod_{i=1}^{n} |f(i)|\}$

① Calcoliamo l'invariante

$x = 2 \qquad s = f(1)$

$x = 3 \qquad s = f(1) * f(2)$

$x = 4 \qquad s = |f(1)| * |f(2)| * |f(3)|$

$$I = x \leq n+1 \wedge s = \prod_{i=1}^{x-1} |f(i)|$$

Dimostriamo che I é invariante e per farlo dobbiamo applicare la consequence rule per farlo dobbiamo avere che $I \wedge b \rightarrow \hat{I}$ per calcolare $\hat{I}$ calcolo la wp utilizzo le interpretazioni dei comandi nel while

$$\hat{I} = I[^{x+1}/_x] = x+1 \leq n+1 \wedge s = \prod_{i=1}^{x} |f(i)|$$

$$\overset{!}{=} x \leq n \wedge s = \prod_{i=1}^{x} |f(i)|$$

Calcolo la wp pure sui rami dell' if trovando così A_T e A_F

$$\frac{\overbrace{\{A \wedge b\}}^{A_T} c_0 \overbrace{\{B\}}^{\hat{I}} \qquad \overbrace{\{A \wedge \neg b\}}^{A_F} c_1 \overbrace{\{B\}}^{\hat{I}}}{\{A\} \text{ if } b \text{ then } c_0 \text{ else } c_1 \{B\}}$$

$$A_T = \hat{I}[^{s * f(x)}/_s]$$

$$\overset{!}{=} x \leq n \wedge s * f(x) = \prod_{i=1}^{x} |f(i)|$$

$$A_F = \hat{I}[^{s*(-f(x))}/_s]$$

$$\overset{!}{=} x \leq n \wedge s * (-f(x)) = \prod_{i=1}^{x} |f(i)|$$

Dimostro che

$$A_T \overset{?}{=} I \wedge b_w \wedge bif$$

$$= \underline{x \leq n+1} \wedge s = \prod_{i=1}^{x-1} |f(i)| \wedge \underline{x \leq n} \wedge f(x) \geq 0$$

$$= x \leq n \wedge s = \prod_{i=1}^{x-1} |f(i)| \wedge f(x) \geq 0$$

$$= x \leq n \wedge s * f(x) = \prod_{i=1}^{x} |f(i)| \wedge f(x) \geq 0 \equiv A_T \checkmark$$

Dimostro che:

$$A_F \overset{?}{=} I \wedge b_w \wedge \neg bif$$

$$= \underline{x \leq n+1} \wedge s = \prod_{i=1}^{x-1} |f(i)| \wedge \underline{x \leq n} \wedge f(x) < 0$$

$$= x \leq n \wedge s * (-f(x)) = \prod_{i=1}^{x} |f(i)|| \wedge f(x) < 0 \equiv A_F \checkmark$$

Avendo dimostrato che $A_T \equiv I \wedge bw \wedge bif$ e quindi $I \wedge bw \wedge bif \rightarrow A_T$

e che $A_F \equiv I \wedge bw \wedge \neg bif$ e quindi $I \wedge bw \wedge bif \rightarrow A_F$

per la consequence rule I é invariante infatti posso applicare la regola del while

$$\frac{\dfrac{\overbrace{\{A \wedge b\}}^{A_T} c_0 \overbrace{\{B\}}^{\hat{I}} \qquad \overbrace{\{A \wedge \neg b\}}^{A_F} c_1 \overbrace{\{B\}}^{\hat{I}}}{\{I \wedge b_w\} \text{ if } b \text{ then } c_0 \text{ else } c_1 \{\hat{I}\}} \qquad \{\hat{I}\}\ x := x+1\ \{I\}}{\dfrac{\{I \wedge b_w\} \text{ corpo } \{I\}}{\{I\} \text{ while } b_w \text{ do corpo } \{I \wedge \neg b_w\}}}$$

Dunque $I \wedge \neg b_w \equiv$ post condizione programma

$$I \wedge \neg b_w = \underline{x \leq n+1} \wedge s = \prod_{i=1}^{x-1} |f(i)| \wedge \underline{x > n}$$

$$= x = n+1 \wedge s = \prod_{i=1}^{n} |f(i)| \equiv \text{post condizione programma} \checkmark$$

Verifichiamo ora che l'invariante valga prima del ciclo

$\{n \geq 0\}$

$s = 1 \longrightarrow I$

$x = 1$

$$x \leq n+1 \wedge s = \prod_{i=1}^{x-1} |f(i)|$$

$$\downarrow \qquad\qquad \downarrow$$

$$1 \leq n+1 \qquad\qquad 1 = 1 * \prod_{i=1}^{0} |f(i)|$$

$$\downarrow \qquad\qquad \downarrow$$

OK ho $n \geq 0$ $\qquad\qquad 1 = 1 * 1$

$\{A\}$ while b_w do if b_{if} then c_1 else c_2 ; c_3 $\{B\}$

A_T A_F

$$\dfrac{A \to I \qquad \dfrac{\dfrac{I \wedge b_w \wedge b_{if} \to A_T}{\{I \wedge b_w \wedge b_{if}\}\, c_1\, \{\hat{I}\}} \qquad \dfrac{I \wedge b_w \wedge \neg b_{if} \to A_F}{\{I \wedge b_w \wedge \neg b_{if}\}\, c_2\, \{\hat{I}\}}}{\dfrac{\{I \wedge b_w\}\ \text{if } b_{if} \text{ then } c_1 \text{ else } c_2;\ \{\hat{I}\} \qquad \{\hat{I}\}\, c_3\, \{I\}}{\dfrac{\{I \wedge b_w\}\ \text{if } b_{if} \text{ then } c_1 \text{ else } c_2;\ c_3\ \{I\}}{\{A\}\ \text{while } b_w \text{ do if } b_{if} \text{ then } c_1 \text{ else } c_2;\ c_3\ \{B\}}}} \qquad B \to I \wedge \neg b}{\{A\}\ \text{while } b_w \text{ do if } b_{if} \text{ then } c_1 \text{ else } c_2;\ c_3\ \{B\}}$$

$a[0,n)$ array di interi

$\{n > 0\}$
$m := a[0]$
$x := 1$
while $(x < n)$ do
$\quad m := \max(m, a[x])$
$\quad x := x+1$

$\{\overset{I}{m = \max\{a[x] \mid x \in [0,n)\}}\}$

① Troviamo l'invariante

$m = \max(a[0], a[1])$ $\qquad x = 2$

$m = \max((a[0], a[1]), a[2])$ $\qquad x = 3$

$m = \max(\max((a[0], a[1]), a[2]), a[3])$ $\qquad x = 4$

$I \equiv m = \max\{a[k] \mid k \in (0, x-1)\} \wedge x \leq n$

① Per dimostrare che I è invariante dobbiamo dimostrare che $I \wedge b \to \hat{I}$
per calcolare $\hat{I}$ dobbiamo utilizzare la weakest precondition e allora applichiamo ad I le interpretazioni dei comandi del while

$\hat{I} = I\,[{}^{x+1}/_{x}][{}^{\max(m, a[x])}/_{m}] \equiv m = \max\{a[k] \mid k \in (0, x)\} \wedge x+1 \leq n$
$\equiv \max(m, a[x]) = \max\{a[k] \mid k \in (0, x)\} \wedge x < n$

$I \wedge b_w = m = \max\{a[k] \mid k \in (0, x-1)\} \wedge x \leq n \wedge x < n$
$\equiv \max(m, a[x]) = \max\{a[k] \mid k \in (0, x)\} \wedge x < n$

$I \wedge b_w \equiv \hat{I}$ allora ho dimostrato $I \wedge b_w \to \hat{I}$ possiamo allora applicare la consequence rule

$$\dfrac{\{\hat{I}\}\ \text{corpo}\ \{I\} \qquad I \wedge b \to \hat{I}}{\{I \wedge b\}\ \text{corpo}\ \{I\}}$$

abbiamo dunque dimostrato che I è invariante e infatti vale la regola del while

$$\dfrac{\dfrac{\{\hat{I}\}\ \text{corpo}\ \{I\} \qquad I \wedge b \to \hat{I}}{\{I \wedge b\}\ \text{corpo}\ \{I\}}}{\{I\}\ \text{while } b_w \text{ do } c\ \{I \wedge \neg b\}}$$

Così facendo però $I \wedge \neg b$ deve essere congruo alla post condizione del programma, verifichiamolo.

$I \wedge \neg b_w = m = \max\{a[k] \mid k \in (0, x-1)\} \wedge x \leq n \wedge x \geq n$

$I \wedge \neg b_w = m = \max\{a[k] \mid k \in (0, x-1)\} \wedge x = n$
$\equiv m = \max\{a[k] \mid k \in (0, n-1)\} \wedge x = n$
$\equiv$ post condizione ✓

Dimostriamo ora che I è invariante per le precondizioni

$\{n > 0\}$
$m := a[0]$ $\longrightarrow I$
$x := 1$

$I \equiv m = \max\{a[k] \mid k \in [0, x-1)\} \wedge x \leq n$
$\equiv a[0] = \max\{a[k] \mid k = 0\} \wedge 1 \leq n$

vero con $n > 0$

$\{x=0, count=0\}$

while $(x<n)$ do

  if $a[x] > A$ then $count = count+1$

  else skip

$\hat{I}$ $\quad x = x+1$

$\{count = |\{y \mid a[y] > A,\ y \in [0,n)\}|\}$

Calcoliamo l'invariante

$x=1 \qquad count = |\{k \mid a[k] > A,\ k \in [0,1)\}|$

$x=2 \qquad count = |\{k \mid a[k] > A,\ k \in [0,2)\}|$

$x=3 \qquad count = |\{k \mid a[k] > A,\ k \in [0,3)\}|$

$$I \equiv count = |\{k \mid a[k] > A,\ k \in [0,x)\}| \wedge x \le n$$

Verifichiamo se I è invariante per farlo dobbiamo dimostrare $I \wedge b_w \wedge b_{if} \to A_T$ e $I \wedge b_w \wedge \neg b_{if} \to A_F$ per poter applicare la consequence rule.
Per trovare A_T e A_F applichiamo le interpretazioni dei comandi nel corpo del while.

$$\hat{I} = I[^{x+1}/_x] = count = |\{k \mid a[k] > A,\ k \in [0,x+1)\}| \wedge x+1 \le n$$

$$A_T = \hat{I}[^{count+1}/_{count}] = count+1 = |\{k \mid a[k] > A,\ k \in [0,x+1)\}| \wedge x+1 \le n$$

$$A_F = \hat{I}$$

$$I \wedge b_w \wedge b_{if} \to A_T \quad ?$$

$$I \wedge b_w \wedge b_{if} = count = |\{k \mid a[k] > A,\ k \in [0,x)\}| \wedge x \le n \wedge x < n \wedge a[x] > A$$

$$\overset{!}{=} count = |\{k \mid a[k] > A,\ k \in [0,x)\}| \wedge x < n \wedge a[x] > A$$

$$\overset{!}{=} count+1 = |\{k \mid a[k] > A,\ k \in [0,x+1)\}| \wedge x+1 \le n \wedge a[x] > A \equiv A_T$$

Sì per questo motivo

$$I \wedge b_w \wedge \neg b_{if} = count = |\{k \mid a[k] > A,\ k \in [0,x)\}| \wedge x \le n \wedge x < n \wedge a[x] < A$$

$$\overset{!}{=} count = |\{k \mid a[k] > A,\ k \in [0,x)\}| \wedge x < n \wedge a[x] < A$$

$$\overset{!}{=} count = |\{k \mid a[k] > A,\ k \in [0,x+1)\}| \wedge x+1 \le n \wedge a[x] < A \equiv A_F$$

Avendo trovato che I è invariante so che la regola è

$$\dfrac{\dfrac{\dfrac{\overbrace{\{I \wedge b_w \wedge b_{if}\}}^{A_T}\ c_1\ \overbrace{\{\hat{I}\}}^{\hat{I}} \qquad \overbrace{\{I \wedge b_w \wedge \neg b_{if}\}}^{A_F}\ c_1\ \overbrace{\{\hat{I}\}}^{\hat{I}}}{\{I \wedge b_w\}\ \text{if } b \text{ then } c_1 \text{ else } c_2\ \{\hat{I}\}} \qquad \{\hat{I}\}\ x = x+1\ \{I\}}{\{I \wedge b_w\}\ \text{corpo}\ \{I\}}}{\{I\}\ \text{while } b \text{ do corpo}\ \{I \wedge \neg b\}}$$

Valida per la consequence rule

Dimostriamo ora che $I \wedge \neg b \equiv$ post condizione

$$I \wedge \neg b \equiv count = |\{k \mid a[k] > A,\ k \in [0,x)\}| \wedge x \le n \wedge x \ge n$$

$$\overset{!}{\equiv} x = n \wedge count = |\{k \mid a[k] > A,\ k \in [0,n)\}| \equiv \text{post condizione}$$

Dimostriamo che I vale prima del ciclo

$\{x=0, count=0\} \to I$

$$count = |\{k \mid a[k] > A,\ k \in [0,x)\}| \wedge x \le n$$

$$0 = |\{k \mid a[k] > A,\ k \in \emptyset\}| \wedge 0 \le n$$

Alternate(b, c_1, c_2)

Operazionale

$$\frac{\langle b, \sigma\rangle \to \text{false}}{\langle A(b, c_1, c_2), \sigma\rangle \to \sigma}$$

$$\frac{\langle b, \sigma\rangle \to \text{true} \qquad \langle c_1, \sigma\rangle \to \sigma'' \qquad \langle A(b, c_2, c_1), \sigma''\rangle \to \sigma'}{\langle A(b, c_1, c_2), \sigma\rangle \to \sigma'}$$

Denotazionale

$$\mathcal{C}[\![\text{Alternate}(b, c_1, c_2)]\!] = \mathcal{C}[\![\text{while } b \text{ do } c_1 \text{ and } c_2]\!]$$

$$= \{(\sigma, \sigma') \mid \mathcal{B}[\![b]\!]_\sigma = \text{false}\} \cup$$

$$\{(\sigma, \sigma') \mid \mathcal{B}[\![b]\!]_\sigma = \text{true} \quad \mathcal{C}[\![c_1]\!]_\sigma = \sigma'' \wedge$$

$$\mathcal{C}[\![\text{while } b \text{ alternate } c_2 \text{ and } c_1]\!]_{\sigma''} = \sigma'\}$$

$$\frac{\langle b, \sigma\rangle \to \text{false}}{\langle A(b, c_1, c_2), \sigma\rangle \to \sigma} \iff \frac{\langle b, \sigma\rangle \to \text{false}}{\langle \text{while } b \text{ do } c_1; \text{if } b \text{ then } c_2 \text{ else skip}, \sigma\rangle \to \sigma}$$

$$\frac{\langle b, \sigma\rangle \to \text{true} \qquad \langle c_1, \sigma\rangle \to \sigma' \qquad \dfrac{\langle b, \sigma'\rangle \to \text{false}}{\langle A(b, c_2, c_1), \sigma'\rangle \to \sigma'}}{\langle A(b, c_1, c_2), \sigma\rangle \to \sigma'} \iff \frac{\langle b, \sigma\rangle \to \text{true} \qquad \langle c_1, \sigma\rangle \to \sigma' \qquad \dfrac{\langle b, \sigma'\rangle \to \text{false} \qquad \langle \text{skip}, \sigma'\rangle \to \sigma'}{\langle \text{if } b \text{ then } c_2 \text{ else skip}, \sigma'\rangle \to \sigma'}}{\langle \text{while } b \text{ do } c_1; \text{if } b \text{ then } c_2 \text{ else skip}, \sigma\rangle \to \sigma'}$$

$$\frac{\langle b, \sigma\rangle \to \text{true} \qquad \langle c_1, \sigma\rangle \to \sigma'' \qquad \dfrac{\langle b, \sigma''\rangle \to \text{true} \qquad \langle c_2, \sigma''\rangle \to \sigma'}{\langle A(b, c_2, c_1), \sigma'\rangle \to \sigma'}}{\langle A(b, c_1, c_2), \sigma\rangle \to \sigma'} \iff \frac{\langle b, \sigma\rangle \to \text{true} \qquad \langle c_1, \sigma\rangle \to \sigma'' \qquad \dfrac{\langle b, \sigma''\rangle \to \text{true} \qquad \langle c_2, \sigma''\rangle \to \sigma'}{\langle \text{if } b \text{ then } c_2 \text{ else skip}, \sigma'\rangle \to \sigma'}}{\langle \text{while } b \text{ do } c_1; \text{if } b \text{ then } c_2 \text{ else skip}, \sigma\rangle \to \sigma'}$$

$$\text{if } b_1 \text{ then } (\text{if } b_2 \text{ then } c_1 \text{ else } c_2) \text{ else } c_2 \equiv \text{if } b_1 \wedge b_2 \text{ then } c_1 \text{ else } c_2$$

Dimostriamo l'uguaglianza con la semantica operazionale

Don't care

$$\frac{\langle b_1, \sigma\rangle \to \text{false} \qquad \langle c_2, \sigma\rangle \to \sigma'}{\langle \text{if } b_1 \text{ then } (\text{if } b_2 \text{ then } c_1 \text{ else } c_2) \text{ else } c_2, \sigma\rangle \to \sigma'} \iff \frac{\dfrac{\langle b_1, \sigma\rangle \to \text{false}}{\langle b_1 \wedge b_2, \sigma\rangle \to \text{false}} \qquad \langle c_2, \sigma\rangle \to \sigma'}{\langle \text{if } b_1 \wedge b_2 \text{ then } c_1 \text{ else } c_2, \sigma\rangle \to \sigma'}$$

$$\frac{\langle b_1, \sigma\rangle \to \text{true} \qquad \dfrac{\langle b_2, \sigma\rangle \to \text{false} \qquad \langle c_2, \sigma\rangle \to \sigma'}{\langle \text{if } b_2 \text{ then } c_1 \text{ else } c_2, \sigma\rangle \to \sigma'}}{\langle \text{if } b_1 \text{ then } (\text{if } b_2 \text{ then } c_1 \text{ else } c_2) \text{ else } c_2, \sigma\rangle \to \sigma'} \iff \frac{\dfrac{\langle b_1, \sigma\rangle \to \text{true} \qquad \langle b_2, \sigma\rangle \to \text{false}}{\langle b_1 \wedge b_2, \sigma\rangle \to \text{false}} \qquad \langle c_2, \sigma\rangle \to \sigma'}{\langle \text{if } b_1 \wedge b_2 \text{ then } c_1 \text{ else } c_2, \sigma\rangle \to \sigma'}$$

$$\frac{\langle b_1, \sigma\rangle \to \text{true} \qquad \dfrac{\langle b_2, \sigma\rangle \to \text{true} \qquad \langle c_1, \sigma\rangle \to \sigma'}{\langle \text{if } b_2 \text{ then } c_1 \text{ else } c_2, \sigma\rangle \to \sigma'}}{\langle \text{if } b_1 \text{ then } (\text{if } b_2 \text{ then } c_1 \text{ else } c_2) \text{ else } c_2, \sigma\rangle \to \sigma'} \iff \frac{\dfrac{\langle b_1, \sigma\rangle \to \text{true} \qquad \langle b_2, \sigma\rangle \to \text{true}}{\langle b_1 \wedge b_2, \sigma\rangle \to \text{true}} \qquad \langle c_1, \sigma\rangle \to \sigma'}{\langle \text{if } b_1 \wedge b_2 \text{ then } c_1 \text{ else } c_2, \sigma\rangle \to \sigma'}$$

$\{n \geq 0\}$

$x := 0$

$s := 0$

while $x \leq n$

if $x \bmod K = 0$ then $s = s + x$ (A_T)

else skip (A_F)

$\hat{I}$

$x := x + 1$

$\{s = \sum_{y=0}^{n} \{y \mid y \bmod K = 0\}\}$

Troviamo l'invariante svolgendo alcune iterazioni del ciclo

$$s = \sum_{y=0}^{0} \{y \mid y \bmod K = 0\} \wedge x = 1$$

$$s = \sum_{y=0}^{1} \{y \mid y \bmod K = 0\} \wedge x = 2$$

$$s = \sum_{y=0}^{2} \{y \mid y \bmod K = 0\} \wedge x = 3$$

$$I \equiv s = \sum_{y=0}^{x-1} \{y \mid y \bmod K = 0\} \wedge x \leq n+1$$

Dimostriamo ora che I è invariante, per farlo devo mostrare che $I \wedge b_w \wedge b_{if} \rightarrow A_T$ e che $I \wedge b_w \wedge \neg b_{if} \rightarrow A_F$

per farlo calcoliamo $\hat{I}$ applicando ad I le interpretazioni dei comandi nel while

$$\hat{I} = I[^{x+1}/_x] = s = \sum_{y=0}^{x-1} \{y \mid y \bmod K = 0\} \wedge x \leq n+1 \; [^{x+1}/_x]$$

$$= s = \sum_{y=0}^{x} \{y \mid y \bmod K = 0\} \wedge x \leq n$$

$$A_T = \hat{I}[^{s+x}/_s] = s = \sum_{y=0}^{x} \{y \mid y \bmod K = 0\} \wedge x+1 \leq n+1 \; [^{s+x}/_x]$$

$$= s+x = \sum_{y=0}^{x} \{y \mid y \bmod K = 0\} \wedge x \leq n$$

$$A_F = \hat{I}$$

Mostriamo $I \wedge b_w \wedge b_{if} \rightarrow A_T$

$$I \wedge b_w \wedge b_{if} \equiv s = \sum_{y=0}^{x-1} \{y \mid y \bmod K = 0\} \wedge \underline{x \leq n+1} \wedge \underline{x \leq n} \wedge x \bmod k = 0$$

$$= s = \sum_{y=0}^{x-1} \{y \mid y \bmod K = 0\} \wedge x \leq n \wedge x \bmod k = 0$$

$$= s+x = \sum_{y=0}^{x} \{y \mid y \bmod k = 0\} \wedge x \leq n \wedge x \bmod k = 0 \equiv A_T$$

faccio un passo in più e incremento perché il valore è multiplo di K

Mostriamo $I \wedge b_w \wedge \neg b_{if} \rightarrow A_F$

$$I \wedge b_w \wedge \neg b_{if} \equiv s = \sum_{y=0}^{x-1} \{y \mid y \bmod K = 0\} \wedge \underline{x \leq n+1} \wedge \underline{x \leq n} \wedge x \bmod k \neq 0$$

$$\equiv s = \sum_{y=0}^{x-1} \{y \mid y \bmod K = 0\} \wedge \underline{x \leq n} \wedge x \bmod k \neq 0$$

$$\equiv s = \sum_{y=0}^{x} \{y \mid y \bmod K = 0\} \wedge x \leq n \wedge x \bmod k \neq 0 \equiv A_F \equiv \hat{I}$$

faccio un passo in più e non incremento perché il valore non è multiplo di K

Avendo mostrato $I \wedge b_w \wedge b_{if} \equiv A_T$ e che $I \wedge b_w \wedge \neg b_{if} \equiv A_F$ so che I è invariante posso dunque derivare per la consequence rule

$$\dfrac{\dfrac{\dfrac{\overbrace{\{I \wedge b_w \wedge b_{if}\}}^{A_T} c_1 \{\hat{I}\} \quad \overbrace{\{I \wedge b_w \wedge \neg b_{if}\}}^{A_F} c_2 \{\hat{I}\}}{\{I \wedge b_w\} \text{if} \ldots \{\hat{I}\}} \quad \{\hat{I}\} c_3 \{I\}}{\{I \wedge b_w\} c \{I\}}}{\{I\} \text{ while } b_w \text{ do } c \{I \wedge \neg b_w\}}$$

So dunque che $I \wedge \neg b_w$ deve essere uguale alla post condizione

$$I \wedge \neg b_w \equiv I \equiv s = \sum_{y=0}^{x-1} \{y \mid y \bmod K = 0\} \wedge x \leq n+1 \wedge x > n$$

$$= x = n+1 \wedge s = \sum_{y=0}^{n} \{y \mid y \bmod K = 0\} \equiv \text{post condizione QED}$$

Applichiamo ora le precondizioni per vedere se l'invariante le soddisfa

$\{n \geq 0\}$

$x := 0$ $\rightarrow I$

$s := 0$

$$I \equiv s = \sum_{y=0}^{x-1} \{y \mid y \bmod K = 0\} \wedge x \leq n+1$$

$0 = 0$ ✓ $\qquad 0 \leq n+1$ ✓ con $n \geq 0$

Troviamo l'invariante eseguendo alcune iterazioni

$s = \prod_{i=1}^{1} |f(i)|$ $\qquad x = 2$

$s = \prod_{i=1}^{2} |f(i)|$ $\qquad x = 3$

$s = \prod_{i=1}^{3} |f(i)|$ $\qquad x = 4$

$\vdots$

$$I = s = \prod_{i=1}^{x-1} |f(i)| \quad \wedge \quad x \leq n+1$$

Dimostriamo che I è invariante per farlo dobbiamo poter applicare la regola del while e per farlo dobbiamo dimostrare che
$I \wedge b_w \wedge b_{if} \rightarrow A_T$ e $I \wedge b_w \wedge \neg b_{if} \rightarrow A_F$

Per farlo dobbiamo trovare $\hat{I}$ applicando ad I l'interpretazione di $x := x+1$

$$\hat{I} = I[^{x+1}/_x] = s = \prod_{i=1}^{x} |f(i)| \quad \wedge \quad x+1 \leq n+1$$

Troviamo ora A_T

$$A_T = \hat{I}[^{s*f(x)}/_s] = s * f(x) = \prod_{i=1}^{x} |f(i)| \quad \wedge \quad x+\cancel{1} \leq n+\cancel{1}$$

$$A_F = \hat{I}[^{s*(-f(x))}/_s] = s * (-f(x)) = \prod_{i=1}^{x} |f(i)| \quad \wedge \quad x+\cancel{1} \leq n+\cancel{1}$$

$$I \wedge b_w \wedge b_{if} = s = \prod_{i=1}^{x-1} |f(i)| \quad \wedge \quad x \leq n+1 \quad \wedge \quad x \leq n \quad \wedge \quad f(x) \geq 0$$

$$= s = \prod_{i=1}^{x-1} |f(i)| \quad \wedge \quad x \leq n \quad \wedge \quad f(x) \geq 0$$

$$= s * \underbrace{f(x) = \prod_{i=1}^{x} |f(i)| \quad \wedge \quad x \leq n \quad \wedge \quad f(x) \geq 0}_{\text{è positivo}} = A_T \checkmark$$

$$I \wedge b_w \wedge \neg b_{if} = s = \prod_{i=1}^{x-1} |f(i)| \quad \wedge \quad x \leq n+1 \quad \wedge \quad x \leq n \quad \wedge \quad f(x) < 0$$

$$= s = \prod_{i=1}^{x-1} |f(i)| \quad \wedge \quad x \leq n \quad \wedge \quad f(x) < 0$$

$$= s * \underbrace{(-f(x)) = \prod_{i=1}^{x} |f(i)| \quad \wedge \quad x \leq n \quad \wedge \quad f(x) < 0}_{\text{essendo negativo cambio segno}} = A_F$$

Avendo dimostrato $I \wedge b_w \wedge b_{if} \rightarrow A_T$ perché sono congruenti e $I \wedge b_w \wedge \neg b_{if} \rightarrow A_F$ posso derivare che I è invariante

$$\dfrac{\dfrac{\dfrac{\overbrace{\{I \wedge b_w \wedge b_{if}\}\, c_1\, \{\hat{I}\}}^{A_T} \quad \overbrace{\{I \wedge b_w \wedge \neg b_{if}\}\, c_1\, \{\hat{I}\}}^{A_F}}{\{I \wedge b_w\}\ \text{if } b \text{ do } c_1 \text{ else } c_2\ \{\hat{I}\}} \quad \{\hat{I}\}\, c_3\, \{I\}}{\{I \wedge b_w\}\ c\ \{I\}}}{\{I\}\ \text{while } b \text{ do } c\ \{I \wedge \neg b_w\}}$$

Avendo dimostrato ciò devo far vedere che $I \wedge \neg b_w \equiv$ post condizione

$$I \wedge \neg b_w \equiv s = \prod_{i=1}^{x-1} |f(i)| \quad \wedge \quad x \leq n+1 \quad \wedge \quad x > n$$

$$\equiv x = n+1 \quad \wedge \quad s = \prod_{i=1}^{n} |f(i)| \equiv \text{postcondizione}$$

Dimostriamo che I vale anche prima del ciclo e per le precondizioni

$$\begin{cases} n \geq 0 \\ s := 1 \\ x := 1 \end{cases} \rightarrow I$$

$$I = s = \prod_{i=1}^{x-1} |f(i)| \quad \wedge \quad x \leq n+1$$

$1 \leq n+1$ ✓ perché $n \geq 0$

$$1 = \prod_{i=1}^{0} |f(i)|$$

→ 1 per ipotesi del testo

$1 = 1$ ✓

$\{x=N,\ y=M,\ p=0,\ d=0\}$

while $x \le y$

if $x \bmod 2 = 0$ then $\overset{A_T}{p := p+x}$

else $\overset{A_F}{d := d+x}$

$\hat{I} \longrightarrow$ $x := x+1$

$\{p = \sum_{i=N}^{M} \{i \mid i \text{ pari}\},\ d = \sum_{i=N}^{M} \{i \mid i \text{ dispari}\}\}$ $\swarrow I$

Troviamo l'invariante svolgendo alcune iterazioni

$$p = \sum_{i=N}^{N} \{i \mid i \text{ pari}\} \qquad d = \sum_{i=N}^{N} \{i \mid i \text{ dispari}\} \qquad x = N+1$$

$$p = \sum_{i=N}^{N+1} \{i \mid i \text{ pari}\} \qquad d = \sum_{i=N}^{N+1} \{i \mid i \text{ dispari}\} \qquad x = N+2$$

$$p = \sum_{i=N}^{N+2} \{i \mid i \text{ pari}\} \qquad d = \sum_{i=N}^{N+2} \{i \mid i \text{ dispari}\} \qquad x = N+3$$

$$\vdots$$

$$I \equiv p = \sum_{i=N}^{x-1} \{i \mid i \text{ pari}\} \ \wedge\ d = \sum_{i=N}^{x-1} \{i \mid i \text{ dispari}\} \ \wedge\ x \le y+1$$

Per dimostrare che I é invariante dobbiamo mostrare che $I \wedge b_w \wedge b_{if} \to A_T$ e che $I \wedge b_w \wedge \neg b_{if} \to A_F$

per trovare $\hat{I}$ applichiamo ad I le interpretazioni dei comandi nel while

$$\hat{I} = I[^{x+1}/_x] = p = \sum_{i=N}^{x} \{i \mid i \text{ pari}\} \ \wedge\ d = \sum_{i=N}^{x} \{i \mid i \text{ dispari}\} \ \wedge\ x + \cancel{1} \le y + \cancel{1}$$

Dimostriamo $I \wedge b_w \wedge b_{if} \to A_T$

$$A_T = \hat{I}[^{p+x}/_x] = p+x = \sum_{i=N}^{x} \{i \mid i \text{ pari}\} \ \wedge\ d = \sum_{i=N}^{x} \{i \mid i \text{ dispari}\} \ \wedge\ x \le y$$

$$I \wedge b_w \wedge b_{if} \equiv p = \sum_{i=N}^{x-1} \{i \mid i \text{ pari}\} \ \wedge\ d = \sum_{i=N}^{x-1} \{i \mid i \text{ dispari}\} \ \wedge\ x \le y+1 \ \wedge\ x \le y \ \wedge\ x \bmod 2 = 0$$

$$\equiv p = \sum_{i=N}^{x-1} \{i \mid i \text{ pari}\} \ \wedge\ d = \sum_{i=N}^{x-1} \{i \mid i \text{ dispari}\} \ \wedge\ x \le y \ \wedge\ x \bmod 2 = 0$$

$$\equiv p+x = \sum_{i=N}^{x} \{i \mid i \text{ pari}\} \ \wedge\ d = \sum_{i=N}^{x} \{i \mid i \text{ dispari}\} \ \wedge\ x \le y \ \wedge\ x \bmod 2 = 0 \equiv A_T$$

essendo divisibile per due il valore del passo successivo é pari e viene quindi sommato a p

Dimostriamo $I \wedge b_w \wedge \neg b_{if} \to A_F$

$$A_F = \hat{I}[^{d+x}/_x] = p = \sum_{i=N}^{x} \{i \mid i \text{ pari}\} \ \wedge\ d+x = \sum_{i=N}^{x} \{i \mid i \text{ dispari}\} \ \wedge\ x \le y$$

$$I \wedge b_w \wedge \neg b_{if} \equiv p = \sum_{i=N}^{x-1} \{i \mid i \text{ pari}\} \ \wedge\ d = \sum_{i=N}^{x-1} \{i \mid i \text{ dispari}\} \ \wedge\ x \le y+1 \ \wedge\ x \le y \ \wedge\ x \bmod 2 \ne 0$$

$$\equiv p = \sum_{i=N}^{x-1} \{i \mid i \text{ pari}\} \ \wedge\ d = \sum_{i-N}^{x-1} \{i \mid i \text{ dispari}\} \ \wedge\ x \le y \ \wedge\ x \bmod 2 \ne 0$$

$$\equiv p = \sum_{i=N}^{x} \{i \mid i \text{ pari}\} \ \wedge\ d+x = \sum_{i-N}^{x} \{i \mid i \text{ dispari}\} \ \wedge\ x \le y \ \wedge\ x \bmod 2 \ne 0 \equiv A_T$$

Avendo dimostrato che $I \wedge b_w \wedge b_{if} \to A_T$ e che $I \wedge b_w \wedge \neg b_{if} \to A_F$ so che I é invariante posso dunque derivare:

per consequence rule

$$\dfrac{\dfrac{\dfrac{\overbrace{\{I \wedge b_w \wedge b_{if}\}\, c_1\, \{\hat{I}\}}^{A_T} \qquad \overbrace{\{I \wedge b_w \wedge \neg b_{if}\}\, c_2\, \{\hat{I}\}}^{A_F}}{\{I \wedge b_w\}\ \text{if } b_{if} \text{ do } c_1 \text{ else } c_2\ \{\hat{I}\}} \qquad \{\hat{I}\}\, c_3\, \{I\}}{\{I \wedge b_w\}\, c\, \{I\}}}{\{I\}\ \text{while } b_w \text{ do } c\ \{I \wedge \neg b_w\}}$$

Di conseguenza $I \wedge \neg b_w \equiv$ postcondizioni del programma

$$I \wedge \neg b_w \equiv p = \sum_{i=N}^{x-1} \{i \mid i \text{ pari}\} \ \wedge\ d = \sum_{i=N}^{x-1} \{i \mid i \text{ dispari}\} \ \wedge\ x \le y+1 \ \wedge\ x > y$$

$$\equiv x = y+1 \ \wedge\ p = \sum_{i=N}^{y} \{i \mid i \text{ pari}\} \ \wedge\ d = \sum_{i=N}^{y} \{i \mid i \text{ dispari}\}$$

$$\equiv x = y+1 \ \wedge\ p = \sum_{i=N}^{M} \{i \mid i \text{ pari}\} \ \wedge\ d = \sum_{i=N}^{M} \{i \mid i \text{ dispari}\} \equiv \text{postcondizioni programma} \ \checkmark$$

Mostriamo ora che I é valido per le precondizioni

$\{x=N,\ y=M,\ p=0,\ d=0\}$

$$I \equiv p = \sum_{i=N}^{N-1} \{i \mid i \text{ pari}\} \ \wedge\ d = \sum_{i=N}^{N-1} \{i \mid i \text{ dispari}\} \ \wedge\ x \le y+1$$

$$\equiv 0 = 0 \ \wedge\ 0 = 0 \ \wedge\ N \le M+1$$

$\checkmark \qquad \checkmark \qquad \checkmark$ per $M > N$

Semantica operazionale

$$\frac{\langle b,\sigma\rangle \to false}{\langle if\ b\ then\ c,\sigma\rangle \to \sigma}$$

$$\frac{\langle b,\sigma\rangle \to true \qquad \langle c,\sigma\rangle \to \sigma'}{\langle if\ b\ then\ c,\sigma\rangle \to \sigma'}$$

Semantica denotazionale

$$\mathcal{C}[\![if\ b\ then\ c]\!] = \{(\sigma,\sigma) \mid \mathcal{B}[\![b]\!]_\sigma = false\} \cup \{(\sigma,\sigma') \mid \mathcal{B}[\![b]\!]_\sigma = true,\ \mathcal{C}[\![c]\!]_\sigma = \sigma'\}$$

Dimostriamo l'equivalenza tra

$$if\ b\ then\ c\ else\ skip \equiv if\ b\ then\ c$$

Semantica operazionale

$$\frac{\langle b,\sigma\rangle \to false \qquad \langle skip,\sigma\rangle \to \sigma}{\langle if\ b\ then\ c\ else\ skip,\sigma\rangle \to \sigma} \iff \frac{\langle b,\sigma\rangle \to false}{\langle if\ b\ then\ c,\sigma\rangle \to \sigma}$$

$$\frac{\langle b,\sigma\rangle \to true \qquad \langle c,\sigma\rangle \to \sigma'}{\langle if\ b\ then\ c\ else\ skip,\sigma\rangle \to \sigma'} \iff \frac{\langle b,\sigma\rangle \to true \qquad \langle c,\sigma\rangle \to \sigma'}{\langle if\ b\ then\ c,\sigma\rangle \to \sigma'}$$

Dimostriamo l'uguaglianza con la semantica denotazionale

$$\mathcal{C}[\![if\ b\ then\ c\ else\ skip]\!] = \{(\sigma,\sigma') \mid \mathcal{B}[\![b]\!]_\sigma = true,\ \mathcal{C}[\![c]\!]_\sigma = \sigma'\} \cup \{(\sigma,\sigma) \mid \mathcal{B}[\![b]\!]_\sigma = false,\ \mathcal{C}[\![skip]\!]_\sigma = \sigma\}$$

$$= \{(\sigma,\sigma') \mid \mathcal{B}[\![b]\!]_\sigma = true,\ \mathcal{C}[\![c]\!]_\sigma = \sigma'\} \cup \{(\sigma,\sigma) \mid \mathcal{B}[\![b]\!]_\sigma = false\}$$

$$\mathcal{C}[\![if\ b\ then\ c]\!] =$$

Equivalenza con semantica operazionale tra: P_1 if b_1 then c_1 else if b_2 then c_1 else c_2
P_2 if $b_1 \vee b_2$ then c_1 else c_2

Don't care sono comunque true

$$\frac{\langle b_1,\sigma\rangle \to true \qquad \langle c_1,\sigma\rangle \to \sigma'}{\langle if\ b_1\ then\ c_1\ else\ if\ b_2\ then\ c_1\ else\ c_2,\sigma\rangle \to \sigma'} \iff \frac{\dfrac{\langle b_1,\sigma\rangle \to true}{\langle b_1 \vee b_2\rangle \to true} \qquad \langle c_1,\sigma\rangle \to \sigma'}{\langle if\ b_1 \vee b_2\ then\ c_1\ else\ c_2,\sigma\rangle \to \sigma'}$$

$$\frac{\langle b_1,\sigma\rangle \to false \qquad \dfrac{\langle b_2,\sigma\rangle \to true \qquad \langle c_1,\sigma\rangle \to \sigma'}{\langle if\ b_2\ then\ c_1\ else\ c_2,\sigma\rangle \to \sigma'}}{\langle if\ b_1\ then\ c_1\ else\ if\ b_2\ then\ c_1\ else\ c_2,\sigma\rangle \to \sigma'} \iff \frac{\dfrac{\langle b_1,\sigma\rangle \to false \qquad \langle b_2,\sigma\rangle \to true}{\langle b_1 \vee b_2\rangle \to true} \qquad \langle c_1,\sigma\rangle \to \sigma'}{\langle if\ b_1 \vee b_2\ then\ c_1\ else\ c_2,\sigma\rangle \to \sigma'}$$

$$\frac{\langle b_1,\sigma\rangle \to false \qquad \dfrac{\langle b_2,\sigma\rangle \to false \qquad \langle c_2,\sigma\rangle \to \sigma'}{\langle if\ b_2\ then\ c_1\ else\ c_2,\sigma\rangle \to \sigma'}}{\langle if\ b_1\ then\ c_1\ else\ if\ b_2\ then\ c_1\ else\ c_2,\sigma\rangle \to \sigma'} \iff \frac{\dfrac{\langle b_1,\sigma\rangle \to false \qquad \langle b_2,\sigma\rangle \to false}{\langle b_1 \vee b_2\rangle \to false} \qquad \langle c_2,\sigma\rangle \to \sigma'}{\langle if\ b_1 \vee b_2\ then\ c_1\ else\ c_2,\sigma\rangle \to \sigma'}$$

Dimostrazione equivalenza con semantica denotazionale

$$\mathcal{C}[\![if\ b_1\ then\ c_1\ else\ if\ b_2\ then\ c_1\ else\ c_2]\!]_\sigma = \{(\sigma,\sigma') \mid \mathcal{B}[\![b_1]\!]_\sigma = true,\ \mathcal{C}[\![c_1]\!]_\sigma = \sigma'\} \cup \{(\sigma,\sigma') \mid \mathcal{B}[\![b_1]\!]_\sigma = false,\ \mathcal{C}[\![if\ b_2\ then\ c_1\ else\ c_2]\!]_\sigma = \sigma'\}$$

$$= \{(\sigma,\sigma') \mid \mathcal{B}[\![b_1]\!]_\sigma = true,\ \mathcal{C}[\![c_1]\!]_\sigma = \sigma'\} \cup \{(\sigma,\sigma') \mid \mathcal{B}[\![b_1]\!]_\sigma = false,\ \mathcal{B}[\![b_2]\!]_\sigma = true,\ \mathcal{C}[\![c_1]\!]_\sigma = \sigma'\} \cup$$

$$\cup \{(\sigma,\sigma') \mid \mathcal{B}[\![b_1]\!]_\sigma = false,\ \mathcal{B}[\![b_2]\!]_\sigma = false,\ \mathcal{C}[\![c_2]\!]_\sigma = \sigma'\}$$

$$= \{(\sigma,\sigma') \mid \mathcal{B}[\![b_1]\!]_\sigma = true \vee \mathcal{B}[\![b_2]\!]_\sigma = true,\ \mathcal{C}[\![c_1]\!]_\sigma = \sigma'\} \cup \{(\sigma,\sigma') \mid \mathcal{B}[\![b_1]\!]_\sigma = false \wedge \mathcal{B}[\![b_2]\!]_\sigma = false,\ \mathcal{C}[\![c_2]\!]_\sigma = \sigma'\}$$

$$= \{(\sigma,\sigma') \mid \mathcal{B}[\![b_1 \vee b_2]\!]_\sigma = true,\ \mathcal{C}[\![c_1]\!]_\sigma = \sigma'\} \cup \{(\sigma,\sigma') \mid \mathcal{B}[\![b_1 \wedge b_2]\!]_\sigma = false,\ \mathcal{C}[\![c_2]\!]_\sigma = \sigma'\}$$

$$= \mathcal{C}[\![if\ b_1 \vee b_2\ then\ c_1\ else\ c_2]\!]$$

while b_1 do c exit b_2

Operazionale

1

$$\frac{\langle b_1, \sigma\rangle \to false}{\langle \text{while } b_1 \text{ do } c \text{ exit } b_2, \sigma\rangle \to \sigma}$$

2

$$\frac{\langle b_1, \sigma\rangle \to true \qquad \langle c, \sigma\rangle \to \sigma' \qquad \langle b_2, \sigma'\rangle \to true}{\langle \text{while } b_1 \text{ do } c \text{ exit } b_2, \sigma\rangle \to \sigma'}$$

3

$$\frac{\langle b_1, \sigma\rangle \to true \qquad \langle c, \sigma\rangle \to \sigma'' \qquad \langle b_2, \sigma''\rangle \to false \qquad \langle \text{while } b_1 \text{ do } c \text{ exit } b_2, \sigma''\rangle \to \sigma'}{\langle \text{while } b_1 \text{ do } c \text{ exit } b_2, \sigma\rangle \to \sigma'}$$

Denotazionale

$$\mathcal{C}[\![\text{while } b \text{ do } c \text{ exit } b_2]\!] = \mathcal{C}[\![\text{if } b_1 \text{ then } \{ c ; \text{if } b_2 \text{ then skip else } w \} \text{ else skip}]\!]$$

$$\stackrel{!}{=} \{(\sigma, \sigma') \mid \beta[\![b_1]\!]_\sigma = true, \mathcal{C}[\![c]\!]_\sigma = \sigma'', \mathcal{C}[\![\text{if } b_2 \text{ then } w \text{ else skip}]\!]_{\sigma''} = \sigma'\} \cup \{(\sigma, \sigma) \mid \beta[\![b_1]\!]_\sigma = false\}$$

$$\stackrel{!}{=} \{(\sigma, \sigma') \ \beta[\![b_1]\!]_\sigma = true, \mathcal{C}[\![c]\!]_\sigma = \sigma'', \beta[\![b_2]\!]_{\sigma''} = false, \mathcal{C}[\![w]\!]_{\sigma''} = \sigma'\} \cup$$

$$\{(\sigma, \sigma'') \ \beta[\![b_1]\!]_\sigma = true, \mathcal{C}[\![c]\!]_\sigma = \sigma'', \beta[\![b_2]\!]_{\sigma''} = true\} \cup \{(\sigma, \sigma) \mid \beta[\![b_1]\!]_\sigma = false\}$$

$\{ n = N > 0 \}$

$a := 0$
$b := 1$
while $n > 0$
$\quad b := a + b$
$\quad a := b - a$
$\quad n := n - 1$
$\{ a = Fib(N) \}$

Troviamo l'invarianti osservando alcune iterazioni del ciclo

$a = 1$ (fib1) $\quad b = 1$ (fib2) $\quad n = N-1$

$a = 1$ (fib2) $\quad b = 2$ (fib3) $\quad n = N-2$

$a = 2$ (fib3) $\quad b = 3$ (fib4) $\quad n = N-3$

$a = 3$ (fib4) $\quad b = 5$ (fib5) $\quad n = N-4$

$a = fib(N-n) = fib(N-N+3) = fib(3)$
$b = fib(N-n+1) = fib(N-N+3+1) = fib(4)$

$$I \equiv a = fib(N-n) \wedge b = fib(N-n+1) \wedge n \geq 0$$

Per dimostrare che I è invariante dobbiamo mostrare che $I \wedge b \rightarrow \hat{I}$ per farlo dobbiamo applicare ad I le interpretazioni dei comandi per trovare $\hat{I}$

$$I \wedge b \equiv a = fib(N-n) \wedge b = fib(N-n+1) \wedge n \geq 0 \wedge n > 0$$
$$\equiv a = fib(N-n) \wedge b = fib(N-n+1) \wedge n > 0$$

$$\hat{I} = I[^{n-1}/_n][^{b-a}/_a][^{a+b}/_b]$$
$$= a = fib(N-n+1) \wedge b = fib(N-n+2) \wedge n-1 \geq 0$$
$$= b-a = fib(N-n+1) \wedge b = fib(N-n+2) \wedge n-1 \geq 0$$
$$= b = fib(N-n+1) \wedge b+a = fib(N-n+2) \wedge n > 0$$
$$= b = fib(N-n+1) \wedge \cancel{fib(N-n+1)} + a = \cancel{fib(N-n+1)} + fib(N-n) \wedge n > 0$$
$$= a = fib(N-n) \wedge b = fib(N-n+1) \wedge n > 0 \equiv I \wedge b$$

$I \wedge b \equiv \hat{I}$

Avendo dimostrato che $I \wedge b \rightarrow \hat{I}$ so che I è invariante per la conseguenza e posso inoltre applicare la regola del while

$$\frac{\dfrac{\{\hat{I}\}\ c\ \{I\} \qquad I \wedge b \rightarrow \hat{I}}{\{I \wedge b\}\ c\ \{I\}}}{\{I\}\ \text{while } b \text{ do } c\ \{I \wedge \neg b\}}$$

posso allora mostrare che $I \wedge \neg b \equiv$ post condizioni del programma se I è invariante

$$I \wedge \neg b \equiv a = fib(N-n) \wedge b = fib(N-n+1) \wedge n \geq 0 \wedge n \leq 0$$
$$\equiv n = 0 \wedge a = fib(N) \wedge b = fib(N+1) \equiv \text{post condizione programma} \ \checkmark$$

Verifichiamo ora che le precondizioni siano valide per I

$\{ n = N > 0 \}$
$a := 0$
$b := 1$
$\rightarrow I$

$I \equiv a = fib(N-n) \wedge b = fib(N-n+1) \wedge n \geq 0$

$0 = fib(0) \qquad 1 = fib(1) \qquad n = N \geq 0$

$\{ n \geq 0 \}$

$x := 0$

$s := 0$

while $x \leq n$

  if $x \bmod k = 0$ then $\overset{A_T}{s := s + x}$

  else $\overset{A_F}{\text{skip}}$

$\hat{I} \rightarrow x := x + 1$

$\{ s = \sum_{y=0}^{n} \{ y \mid y \bmod k = 0 \} \}$

1 Troviamo l'invariante eseguendo alcune iterazioni del ciclo

$$s = \sum_{i=0}^{0} \{ i \mid i \bmod k = 0 \} \qquad x = 1$$

$$s = \sum_{i=0}^{1} \{ i \mid i \bmod k = 0 \} \qquad x = 2$$

$$s = \sum_{i=0}^{2} \{ i \mid i \bmod k = 0 \} \qquad x = 3$$

$$s = \sum_{i=0}^{3} \{ i \mid i \bmod k = 0 \} \qquad x = 4$$

$$I \equiv s = \sum_{i=0}^{x-1} \{ i \mid i \bmod k = 0 \} \wedge x \leq n+1$$

Verifichiamo ora che I sia invariante, per farlo dobbiamo poter applicare la consequence rule e per farlo dobbiamo dimostrare che $I \wedge b_w \wedge b_{if} \rightarrow A_T$ e che $I \wedge b_w \wedge \neg b_{if} \rightarrow A_F$ per trovare A_T e A_F applichiamo le interpretazioni dei comandi nel while calcolando dunque le opportune wp

$$\hat{I} = I[^{x+1}/_x] = s = \sum_{i=0}^{x} \{ i \mid i \bmod k = 0 \} \wedge x+1 \leq n+1$$

$$A_T = \hat{I}[^{s+x}/_x] = s + x = \sum_{i=0}^{x} \{ i \mid i \bmod k = 0 \} \wedge x+1 \leq n+1$$

$$A_F = \hat{I}$$

$$I \wedge b_w \wedge b_{if} = s = \sum_{i=0}^{x-1} \{ i \mid i \bmod k = 0 \} \wedge x \leq n+1 \wedge x \leq n \wedge x \bmod k = 0$$

$$= s = \sum_{i=0}^{x-1} \{ i \mid i \bmod k = 0 \} \wedge x \leq n \wedge x \bmod k = 0$$

$$= s + x = \sum_{i=0}^{x} \{ i \mid i \bmod k = 0 \} \wedge x \leq n \wedge x \bmod k = 0 \equiv A_T$$

facciamo un altro passo e sommo il valore perché la condizione dice il valore è un multiplo di k

$$I \wedge b_w \wedge \neg b_{if} = s = \sum_{i=0}^{x-1} \{ i \mid i \bmod k = 0 \} \wedge x \leq n+1 \wedge x \leq n \wedge x \bmod k \neq 0$$

$$= s = \sum_{i=0}^{x-1} \{ i \mid i \bmod k = 0 \} \wedge x \leq n \wedge x \bmod k \neq 0$$

$$= s = \sum_{i=0}^{x} \{ i \mid i \bmod k = 0 \} \wedge x \leq n \wedge x \bmod k \neq 0 \equiv A_F$$

facciamo un altro passo e non sommo il valore perché la condizione dice il valore è non un multiplo di k

Avendo mostrato che $I \wedge b_w \wedge b_{if} \equiv A_T$ e che $I \wedge b_w \wedge \neg b_{if} \equiv A_F$ ho mostrato che $I \wedge b_w \wedge b_{if} \rightarrow A_T$ e che $I \wedge b_w \wedge \neg b_{if} \rightarrow A_F$ posso applicare la consequence rule e le seguenti regole nella derivare

$$\frac{\dfrac{\overbrace{\{ I \wedge b_w \wedge b_{if} \}}^{A_T} c_1 \{ \hat{I} \} \qquad \overbrace{\{ I \wedge b_w \wedge \neg b_{if} \}}^{A_F} c_2 \{ \hat{I} \}}{\{ I \wedge b_w \} \text{ if } b_{if} \text{ then } c_1 \text{ else } c_2 \{ \hat{I} \}} \qquad \{ \hat{I} \} c_3 \{ I \}}{\{ I \wedge b_w \} \; c \; \{ I \}}$$

$$\{ I \} \text{ while } b_w \text{ do } c \; \{ I \wedge \neg b_w \}$$

Dobbiamo dunque dimostrare avendo applicato tale regola che se I è invariante allora per la regola del while devono valere che $I \wedge \neg b$ deve corrispondere con le post condizioni del programma e che I deve corrispondere con le precondizioni del programma

$$I \wedge \neg b_w \equiv s = \sum_{i=0}^{x-1} \{ i \mid i \bmod k = 0 \} \wedge x \leq n+1 \wedge x > n$$

$$= x = n+1 \wedge s = \sum_{i=0}^{n} \{ i \mid i \bmod k = 0 \} \equiv \text{post condizioni } \checkmark$$

$$I = s = \sum_{i=0}^{x-1} \{ i \mid i \bmod k = 0 \} \wedge x \leq n+1$$

$\{ n \geq 0 \}$

$x := 0$

$s := 0$

$$0 = \sum_{i=0}^{-1} \{ \emptyset \}$$

$$0 = 0$$

$0 \leq n+1$ ✓ perché $n \geq 0$

Dimostriamo con semantica operazionale l'equivalenza

$x := x+1 ; \text{if } x \neq 0 \text{ then } c_1 \text{ else } c_2$ $\qquad$ $\text{if } x+1 \neq 0 \text{ then } x := x+1 ; c_1 \text{ else } x := x+1 ; c_2$

$$\dfrac{\dfrac{\dfrac{\langle x,\sigma\rangle \to \sigma(x) \quad \langle 1,\sigma\rangle \to 1}{\langle x+1,\sigma\rangle \to \sigma(x)+1} \quad \dfrac{\langle x,\sigma[\sigma(x)+1/x]\rangle \to \sigma(x)+1 \quad \langle 0,\sigma[\sigma(x)+1/x]\rangle \to 0}{\langle x\neq 0,\sigma[\sigma(x)+1/x]\rangle \to \text{true}} \quad \langle c_1,\sigma[\sigma(x)+1/x]\rangle \to \sigma'}{\langle x:=x+1,\sigma\rangle \to \sigma[\sigma(x)+1/x] \quad \langle \text{if } x\neq 0 \text{ then } c_1 \text{ else } c_2,\sigma[\sigma(x)+1/x]\rangle \to \sigma'}}{\langle x:=x+1 ; \text{if } x\neq 0 \text{ then } c_1 \text{ else } c_2,\sigma\rangle \to \sigma'}$$

$$\Longleftrightarrow$$

$$\dfrac{\dfrac{\langle x,\sigma\rangle \to \sigma(x) \quad \langle 1,\sigma\rangle \to 1}{\langle x+1,\sigma\rangle \to \sigma(x)+1} \quad \langle 0,\sigma\rangle \to 0}{\langle x+1\neq 0,\sigma\rangle \to \text{true}} \quad \dfrac{\dfrac{\dfrac{\langle x,\sigma\rangle \to \sigma(x) \quad \langle 1,\sigma\rangle \to 1}{\langle x+1,\sigma\rangle \to n}}{\langle x:=x+1,\sigma\rangle \to \sigma[\sigma(x)+1/x]} \quad \langle c_1,\sigma[\sigma(x)+1/x]\rangle \to \sigma'}{\langle x:=x+1 ; c_1,\sigma\rangle \to \sigma'}$$
$$\langle \text{if } x+1\neq 0 \text{ then } x:=x+1 ; c_1 \text{ else } x:=x+1 ; c_2,\sigma\rangle \to \sigma'$$

$$\dfrac{\dfrac{\dfrac{\langle x,\sigma\rangle \to \sigma(x) \quad \langle 1,\sigma\rangle \to 1}{\langle x+1,\sigma\rangle \to \sigma(x)+1} \quad \dfrac{\langle x,\sigma[x+1/x]\rangle \to \sigma(x)+1 \quad \langle 0,\sigma[x+1/x]\rangle \to 0}{\langle x\neq 0,\sigma[x+1/x]\rangle \to \text{false}} \quad \langle c_2,\sigma[x+1/x]\rangle \to \sigma'}{\langle x:=x+1,\sigma\rangle \to \sigma[x+1/x] \quad \langle \text{if } x\neq 0 \text{ then } c_1 \text{ else } c_2,\sigma[x+1/x]\rangle \to \sigma'}}{\langle x:=x+1 ; \text{if } x\neq 0 \text{ then } c_1 \text{ else } c_2,\sigma\rangle \to \sigma'}$$

$$\Longleftrightarrow$$

$$\dfrac{\dfrac{\langle x,\sigma\rangle \to \sigma(x) \quad \langle 1,\sigma\rangle \to 1}{\langle x+1,\sigma\rangle \to n} \quad \langle 0,\sigma\rangle \to 0}{\langle x+1\neq 0,\sigma\rangle \to \text{false}} \quad \dfrac{\dfrac{\dfrac{\langle x,\sigma\rangle \to \sigma(x) \quad \langle 1,\sigma\rangle \to 1}{\langle x+1,\sigma\rangle \to n}}{\langle x:=x+1,\sigma\rangle \to \sigma[n/x]} \quad \langle c_2,\sigma[n/x]\rangle \to \sigma'}{\langle x:=x+1 ; c_2,\sigma\rangle \to \sigma'}$$
$$\langle \text{if } x+1\neq 0 \text{ then } x:=x+1 ; c_1 \text{ else } x:=x+1 ; c_2,\sigma\rangle \to \sigma'$$

$$\begin{aligned}
\mathcal{C}[\![x:=x+1 ; \text{if } x\neq 0 \text{ then } c_1 \text{ else } c_2]\!] &= \mathcal{C}[\![\text{if } x\neq 0 \text{ then } c_1 \text{ else } c_2]\!]_{\sigma[x+1/x]} \\
&= \{(\sigma[x+1/x],\sigma') \mid \beta[\![x\neq 0]\!]_{\sigma[x+1/x]} = \text{true},\ \mathcal{C}[\![c_1]\!]_{\sigma[x+1/x]} = \sigma'\} \\
&\quad \cup \{(\sigma[x+1/x],\sigma') \mid \beta[\![x\neq 0]\!]_{\sigma[x+1/x]} = \text{false},\ \mathcal{C}[\![c_2]\!]_{\sigma[x+1/x]} = \sigma'\} \\
&= \{(\sigma,\sigma') \mid \beta[\![x+1\neq 0]\!]_{\sigma} = \text{true},\ \mathcal{C}[\![x:=x+1]\!]_{\sigma} = \sigma[x+1/x],\ \mathcal{C}[\![c_1]\!]_{\sigma[x+1/x]} = \sigma'\} \\
&\quad \cup \{(\sigma,\sigma') \mid \beta[\![x+1\neq 0]\!]_{\sigma} = \text{false},\ \mathcal{C}[\![x:=x+1]\!]_{\sigma} = \sigma[x+1/x],\ \mathcal{C}[\![c_2]\!]_{\sigma[x+1/x]} = \sigma'\} \\
&= \mathcal{C}[\![\text{if } x+1\neq 0 \text{ then } x:=x+1 ; c_1 \text{ else } x:=x+1 ; c_2]\!]
\end{aligned}$$

$\{ n \geq 0 \}$

$s := 0$
$x := 0$
while $x \leq n$
  if $f(x) \neq 0$ then $\overset{A_T}{s := s + f(x)}$;
  else $\overset{A_F}{skip}$;
$\hat{I} \rightarrow$ $x := x + 1$

$\{ s = \sum_{i=0}^{n} f(i) \}$

1 Troviamo l'invariante applicando alcune iterazioni del ciclo

$s := f(0)$ $\quad x := 1$
$s := f(0) + f(1)$ $\quad x := 2$
$s := f(0) + f(1) + f(2)$ $\quad x := 2$

$$I = s = \sum_{i=0}^{x-1} f(i) \quad \wedge \quad x \leq n+1$$

Dimostriamo ora che I è invariante, per farlo dobbiamo poter applicare la regola del while e per farlo dobbiamo poter applicare la consequence rule e per farlo dobbiamo mostrare che $I \wedge b_w \wedge b_{if} \rightarrow A_T$ e che $I \wedge b_w \wedge \neg b_{if} \rightarrow A_F$ per calcolare A_T e A_F applichiamo le interpretazioni dei comandi dei corrispettivi rami dell'if ad $\hat{I}$. Calcoliamo dunque le wp $\hat{I}$ viene calcolato applicando ad I le interpretazioni dei comandi del while

$$\hat{I} = I[^{x+1}/_x] = s = \sum_{i=0}^{x} f(i) \quad \wedge \quad x+1 \leq n+1$$
$$= s = \sum_{i=0}^{x} f(i) \quad \wedge \quad x \leq n$$

$$A_T = \hat{I}[^{s+f(x)}/_s] = s + f(x) = \sum_{i=0}^{x} f(i) \quad \wedge \quad x \leq n$$

$$A_F = \hat{I}$$

$$I \wedge b_w \wedge b_{if} \equiv s = \sum_{i=0}^{x-1} f(i) \quad \wedge \quad x \leq n+1 \quad \wedge \quad x \leq n \quad \wedge \quad f(x) \neq 0$$
$$= s = \sum_{i=0}^{x-1} f(i) \quad \wedge \quad x \leq n \quad \wedge \quad f(x) \neq 0$$
$$= s + f(x) = \sum_{i=0}^{x} f(i) \wedge \quad x \leq n \quad \wedge \quad f(x) \neq 0 \equiv A_T$$

posso fare un ulteriore passo e il valore si somma perché è diverso da zero

$$I \wedge b_w \wedge \neg b_{if} \equiv s = \sum_{i=0}^{x-1} f(i) \quad \wedge \quad x \leq n+1 \quad \wedge \quad x \leq n \quad \wedge \quad f(x) = 0$$
$$= s = \sum_{i=0}^{x-1} f(i) \quad \wedge \quad x \leq n \quad \wedge \quad f(x) = 0$$
$$= s = \sum_{i=0}^{x} f(i) \wedge \quad x \leq n \quad \wedge \quad \underline{f(x) = 0}$$

Il valore non si somma perché vale zero, facciamo dunque un passo in più ma s non varia

Avendo mostrato che $I \wedge b_w \wedge b_{if} \rightarrow A_T$ facendo vedere $I \wedge b_w \wedge \neg b_{if} \rightarrow A_T$ e che $I \wedge b_w \wedge b_{if} \rightarrow A_F$ facendo vedere $I \wedge b_w \wedge \neg b_{if} \rightarrow A_F$ possiamo costruire la seguente derivazione, applicare la regola while e di conseguenza mostrare che I è invariante

$$\dfrac{\dfrac{\overbrace{\{I \wedge b_w \wedge b_{if}\} c_1 \{\hat{I}\}}^{A_T \ \checkmark \text{ per consequence rule}} \quad \overbrace{\{I \wedge b_w \wedge \neg b_{if}\} c_2 \{\hat{I}\}}^{A_F \ \checkmark \text{ per consequence rule}}}{\{I \wedge b_w\} \text{ if } b_{if} \text{ then } c_1 \text{ else } c_2 \{\hat{I}\}} \quad \{\hat{I}\} c_3 \{I\}}{\dfrac{\{I \wedge b_w\} \ c \ \{I\}}{\{I\} \text{ while } b_w \text{ do } c \ \{I \wedge \neg b\}}}$$

Avendo derivato ciò so che I è invariante, pertanto dobbiamo mostrare che
I vale per le precondizioni
$I \wedge \neg b$ corrisponde con le post condizioni

$$I \wedge \neg b \equiv \quad s = \sum_{i=0}^{x-1} f(i) \quad \wedge \quad x \leq n+1 \quad \wedge \quad x > n$$

$$x = n+1 \quad \wedge \quad s = \sum_{i=0}^{n} f(i) \equiv \text{post condizioni} \ \checkmark \text{ CVD}$$

$\{ n \geq 0 \}$
$s := 0$
$x := 0$
$\rightarrow I$

$$s = \sum_{i=0}^{x-1} f(i) \quad \wedge \quad x \leq n+1$$

somma di zero elementi

$$0 = \sum_{i=0}^{-1} f(i) \quad \wedge \quad 0 \leq n+1$$

$\hookrightarrow$ OK ho $n \geq 0$

$0 = 0$

$\checkmark$ CVD

$\{ n \geq 0 \}$

$x := 0$

$s := 0$

while $x \leq n$

  if $x \bmod k = 0$ then $\overset{A_T}{s := s+x}$

  else $\overset{A_F}{skip}$

$\hat{I} \longrightarrow x := x+1$

$\{ s = \sum_{y=0}^{n} \{ y \mid y \bmod k = 0 \} \}$

1 Troviamo l'invariante eseguendo il programma e osservando alcune iterazioni del ciclo

$$s = \sum_{i=0}^{0} \{ i \mid i \bmod k = 0 \} \qquad x = 1$$

$$s = \sum_{i=0}^{1} \{ i \mid i \bmod k = 0 \} \qquad x = 2$$

$$s = \sum_{i=0}^{2} \{ i \mid i \bmod k = 0 \} \qquad x = 3$$

$$\vdots$$

$$I \equiv s = \sum_{i=0}^{x-1} \{ i \mid i \bmod k = 0 \} \wedge x \leq n+1$$

Per dimostrare che I è effettivamente poter applicare la regola della semantica assiomatica per il while. Per farlo dobbiamo mostrare che $I \wedge b_w \wedge b_{if} \rightarrow A_T$ e che $I \wedge b_w \wedge \neg b_{if} \rightarrow A_F$. Per farlo dobbiamo calcolare le relative wp

Calcoliamo $\hat{I}$ applicando ad I l'interpretazione di $x := x+1$

$$I[^{x+1}/_x] = s = \sum_{i=0}^{x} \{ i \mid i \bmod k = 0 \} \wedge x + \cancel{1} \leq n + \cancel{1}$$

Calcoliamo ora A_T e A_F applicando le interpretazioni dei comandi nei relativi rami dell'if a $\hat{I}$

$$A_T = \hat{I}[^{s+x}/_s] = s + x = \sum_{i=0}^{x} \{ i \mid i \bmod k = 0 \} \wedge x \leq n$$

$$A_F = \hat{I} = s = \sum_{i=0}^{x} \{ i \mid i \bmod k = 0 \} \wedge x \leq n$$

$$I \wedge b_w \wedge b_{if} \equiv s = \sum_{i=0}^{x-1} \{ i \mid i \bmod k = 0 \} \wedge x \leq n+1 \wedge x \leq n \wedge x \bmod k = 0$$

$$\equiv s = \sum_{i=0}^{x-1} \{ i \mid i \bmod k = 0 \} \wedge x \leq n \wedge x \bmod k = 0$$

$$\equiv s + x = \sum_{i=0}^{x} \{ i \mid i \bmod k = 0 \} \wedge x \leq n \wedge x \bmod k = 0 \equiv A_T$$

facciamo un passo e aggiungiamo il valore perché soddisfa il valore

$$I \wedge b_w \wedge \neg b_{if} \equiv s = \sum_{i=0}^{x+1} \{ i \mid i \bmod k = 0 \} \wedge x \leq n+1 \wedge x \leq n \wedge x \bmod k \neq 0$$

$$\equiv s = \sum_{i=0}^{x-1} \{ i \mid i \bmod k = 0 \} \wedge x \leq n \wedge x \bmod k \neq 0$$

$$\equiv s = \sum_{i=0}^{x} \{ i \mid i \bmod k = 0 \} \wedge x \leq n \wedge x \bmod k \neq 0 \equiv A_F$$

facciamo un passo ma non aggiungiamo il valore poiché non soddisfa la condizione

Avendo mostrato che $I \wedge b_w \wedge b_{if} \rightarrow A_T$ facendo vedere l'equivalenza, e idem per $I \wedge b_w \wedge \neg b_{if}$, possiamo dire che I è invariante, possiamo infatti derivare che:

$$\dfrac{\dfrac{\dfrac{\overbrace{\{ I \wedge b_w \wedge b_{if} \}}^{A_T \ \vee \ \text{per consequence rule}} c_1 \{ \hat{I} \} \qquad \overbrace{\{ I \wedge b_w \wedge \neg b_{if} \}}^{A_F \ \vee \ \text{per consequence rule}} c_2 \{ \hat{I} \}}{\{ I \wedge b_w \} \text{if } b_{if} \text{ then } c_1 \text{ else } c_2 \{ \hat{I} \} \qquad \{ \hat{I} \} c_3 \{ I \}}}{\{ I \wedge b_w \} \ c \ \{ I \}}}{\{ I \} \text{ while } b_w \text{ do } c \ \{ I \wedge \neg b_w \}}$$

Consequence rule

$$\dfrac{\{ D \} \ c \ \{ B \} \qquad A \rightarrow D}{\{ A \} \ c \ \{ B \}}$$

Avendo derivato ciò so che I è invariante pertanto dobbiamo verificare che

I sia concorde alle precondizioni

I sia concorde alle postcondizioni

$I \wedge \neg bw \equiv s = \sum_{i=0}^{x-1} \{i \mid i \bmod K = 0\} \wedge x \leq n+1 \wedge x > n$

$\equiv x = n+1 \wedge s = \sum_{i=0}^{n} \{i \mid i \bmod K = 0\}$ ✓ concorde alle post condizioni

$\{n \geq 0\}$

$x := 0$

$s := 0$

$\rightarrow I$?

$s = \sum_{i=0}^{x-1} \{i \mid i \bmod K = 0\} \wedge x \leq n+1$

$0 = \sum_{i=0}^{-1} \{i \mid i \bmod K = 0\}$ $\quad 0 \leq n+1$ ✓ perché ho $n \geq 0$

$0 = \sum_{i=0}^{-1} \{\emptyset\}$

$0 = 0$ ✓

$\{ n = N \geq 0 \}$

$a := 0$

$b := 1$

while $n \geq 0$

$\hat{I}$ $\quad b := a + b$

$\quad a := b - a$

I $\quad n := n - 1$

$\{ a = fib(N) \}$

① Trovare l'invariante eseguendo il programma e alcune invarianti di ciclo

fib 2	fib 1	
$b := 1$	$a := 1$	$n := N-1$
fib 3	fib 2	
$b := 2$	$a := 1$	$n := N-2$
fib 4	fib 3	
$b := 3$	$a := 2$	$n := N-3$
fib 5	fib 4	
$b := 5$	$a := 3$	$n := N-4$

$$I \equiv b = fib(N-n+1) \wedge a = fib(N-n) \wedge n \geq 0$$

Dimostriamo che I è invariante, per farlo dobbiamo applicare la regola del while e possiamo farlo solo se possiamo applicare la consequence rule, ma per farlo devo mostrare che $I \wedge b \rightarrow \hat{I}$, per calcolare $\hat{I}$ calcoliamo la weakest precondition applicando ad I le interpretazioni dei comandi nel while

$$\hat{I} = I[^{n-1}/_n][^{b-a}/_a][^{b+a}/_b]$$

$$= b = fib(N-n+2) \wedge a = fib(N-n+1) \wedge n-1 \geq 0 \quad [^{b-a}/_a][^{b+a}/_b]$$

$$= b = fib(N-n+2) \wedge b-a = fib(N-n+1) \wedge n-1 \geq 0 \quad [^{b+a}/_b]$$

$$= b+a = fib(N-n+2) \wedge b = fib(N-n+1) \wedge n-1 \geq 0$$

$$I \wedge b_w \equiv b = fib(N-n+1) \wedge a = fib(N-n) \wedge n \geq 0 \wedge n > 0$$

$$\equiv b = fib(N-n+1) \wedge a = fib(N-n) \wedge n > 0$$

$$\equiv b = fib(N-n+1) \wedge b+a = fib(N-n) + fib(N-n+1) \wedge n > 0$$

$$\equiv b+a = fib(N-n+2) \wedge b = fib(N-n+1) \wedge n > 0 \equiv \hat{I} \quad ✓$$

Ho dimostrato che I è invariante, vale infatti la consequence rule

$$\frac{\{\hat{I}\}\ c\ \{I\} \qquad I \wedge b \rightarrow \hat{I}}{\{I \wedge b\}\ c\ \{I\}}$$

posso dunque applicare la regola del while

$$\frac{\dfrac{\{\hat{I}\}\ c\ \{I\} \qquad I \wedge b \rightarrow \hat{I}}{\{I \wedge b\}\ c\ \{I\}}}{\{I\}\ \text{while } b \text{ do } c\ \{I \wedge \neg b\}}$$

so dunque che I è invariante, ma allora I deve essere congruo alle precondizioni

$I \wedge \neg b$ deve essere congruo alle postcondizioni

$$I \wedge \neg b_w \equiv b = fib(N-n+1) \wedge a = fib(N-n) \wedge n \geq 0 \wedge n \leq 0$$

$$\equiv n = 0 \wedge b = fib(N+1) \wedge a = fib(N) \equiv \text{post condizioni} \ ✓$$

$$\{n = N \geq 0\} \to I \quad b = fib(N-n+1) \wedge a = fib(N-n) \wedge n \geq 0$$

$a := 0$

$b := 1$

$$1 = fib(0+1) \checkmark \qquad 0 = fib(0) \checkmark \qquad n \geq 0 \checkmark$$

---

if b_1 then c_1 else if b_2 then c_1 else c_2

$$\frac{\langle b_1, \sigma \rangle \to true \qquad \langle c_1, \sigma \rangle \to \sigma'}{\langle \text{if } b_1 \text{ then } c_1 \text{ else if } b_2 \text{ then } c_1 \text{ else } c_2, \sigma \rangle \to \sigma'} \iff$$

if $b_1 \vee b_2$ then c_1 else c_2

qualsiasi valore non cambia la valutazione

$$\frac{\dfrac{\langle b_1, \sigma \rangle \to true}{\langle b_1 \vee b_2, \sigma \rangle \to true} \qquad \langle c_1, \sigma \rangle \to \sigma'}{\langle \text{if } b_1 \vee b_2 \text{ then } c_1 \text{ else } c_2, \sigma \rangle \to \sigma'}$$

$$\frac{\langle b_1, \sigma \rangle \to false \qquad \dfrac{\langle b_2, \sigma \rangle \to true \qquad \langle c_1, \sigma \rangle \to \sigma'}{\langle \text{if } b_2 \text{ then } c_1 \text{ else } c_2, \sigma \rangle \to \sigma'}}{\langle \text{if } b_1 \text{ then } c_1 \text{ else if } b_2 \text{ then } c_1 \text{ else } c_2, \sigma \rangle \to \sigma'} \iff$$

$$\frac{\dfrac{\langle b_1, \sigma \rangle \to false \qquad \langle b_2, \sigma \rangle \to true}{\langle b_1 \vee b_2 \rangle \to true} \qquad \langle c_1, \sigma \rangle \to \sigma'}{\langle \text{if } b_1 \vee b_2 \text{ then } c_1 \text{ else } c_2, \sigma \rangle \to \sigma'}$$

$$\frac{\langle b_1, \sigma \rangle \to false \qquad \dfrac{\langle b_2, \sigma \rangle \to false \qquad \langle c_2, \sigma \rangle \to \sigma'}{\langle \text{if } b_2 \text{ then } c_1 \text{ else } c_2, \sigma \rangle \to \sigma'}}{\langle \text{if } b_1 \text{ then } c_1 \text{ else if } b_2 \text{ then } c_1 \text{ else } c_2, \sigma \rangle \to \sigma'}$$

$$\frac{\dfrac{\langle b_1, \sigma \rangle \to false \qquad \langle b_2, \sigma \rangle \to false}{\langle b_1 \vee b_2, \sigma \rangle \to false} \qquad \langle c_2, \sigma \rangle \to \sigma'}{\langle \text{if } b_1 \vee b_2 \text{ then } c_1 \text{ else } c_2, \sigma \rangle \to \sigma'}$$

$$\begin{aligned}
\mathcal{C}[\![\text{if } b_1 \text{ then } c_1 \text{ else if } b_2 \text{ then } c_1 \text{ else } c_2]\!] &= \{(\sigma, \sigma') \mid \beta[\![b_1]\!]_\sigma = true, \mathcal{C}[\![c_1]\!]_\sigma = \sigma'\} \cup \\
&\quad \{(\sigma, \sigma') \mid \beta[\![b_1]\!]_\sigma = false, \beta[\![b_2]\!]_\sigma = true, \mathcal{C}[\![c_1]\!]_\sigma = \sigma'\} \cup \\
&\quad \{(\sigma, \sigma') \mid \beta[\![b_1]\!] = false, \beta[\![b_2]\!]_\sigma = false, \mathcal{C}[\![c_2]\!]_\sigma = \sigma'\} \\
&= \{(\sigma, \sigma') \mid \beta[\![b_1 \vee b_2]\!]_\sigma = true, \mathcal{C}[\![c_1]\!]_\sigma = \sigma'\} \cup \\
&\quad \{(\sigma, \sigma') \mid \beta[\![b_1 \wedge b_2]\!]_\sigma = false, \mathcal{C}[\![c_2]\!]_\sigma = \sigma'\} \\
&= \mathcal{C}[\![\text{if } b_1 \vee b_2 \text{ then } c_1 \text{ else } c_2]\!]
\end{aligned}$$

while b_1 do c exit b_2

$$\frac{\langle b_1, \sigma \rangle \to false}{\text{while } b_1 \text{ do } c \text{ exit } b_2}$$

$$\frac{\langle b_1, \sigma \rangle \to true \qquad \langle c, \sigma \rangle \to \sigma' \qquad \langle b_2, \sigma \rangle \to true}{\langle \text{while } b_1 \text{ do } c \text{ exit } b_2, \sigma \rangle \to \sigma'}$$

$$\frac{\langle b_1, \sigma \rangle \to true \qquad \langle c, \sigma \rangle \to \sigma'' \qquad \langle b_2, \sigma \rangle \to false \qquad \langle \text{while } b_1 \text{ do } c \text{ exit } b_2, \sigma'' \rangle \to \sigma'}{\langle \text{while } b_1 \text{ do } c \text{ exit } b_2, \sigma \rangle \to \sigma'}$$

$$\mathcal{C}[\![\underbrace{\text{while } b_1 \text{ do } c \text{ exit } b_2}_{\varphi}]\!] = \mathcal{C}[\![\text{if } b_1 \text{ then } \{c; \text{ if } b_2 \text{ then skip else } w\} \text{ else skip}]\!]$$

$$= \{(\sigma,\sigma') \mid \mathcal{B}[\![b_1]\!]_\sigma = \text{true},\ \mathcal{C}[\![c; \text{ if } b_2 \text{ then skip else } w]\!]_\sigma = \sigma'\} \cup \{(\sigma,\sigma) \mid \mathcal{B}[\![b_1]\!] = \text{false}\}$$

$$= \{(\sigma,\sigma') \mid \mathcal{B}[\![b_1]\!]_\sigma = \text{true},\ \mathcal{C}[\![c_1]\!]_\sigma = \sigma'',\ \mathcal{C}[\![\text{if } b_2 \text{ then skip else } w]\!]_{\sigma''} = \sigma'\}$$

$$\cup \{(\sigma,\sigma) \mid \mathcal{B}[\![b_1]\!] = \text{false}\}$$

$$= \{(\sigma,\sigma'') \mid \mathcal{B}[\![b_1]\!]_\sigma = \text{true},\ \mathcal{C}[\![c_1]\!]_\sigma = \sigma'',\ \mathcal{B}[\![b_2]\!]_{\sigma''} = \text{true}\}$$

$$\cup \{(\sigma,\sigma') \mid \mathcal{B}[\![b_1]\!]_\sigma = \text{true},\ \mathcal{C}[\![c_1]\!]_\sigma = \sigma'',\ \mathcal{B}[\![b_2]\!]_{\sigma''} = \text{false},\ \mathcal{C}[\![w]\!]_{\sigma''} = \sigma'\}$$

$$\cup \{(\sigma,\sigma) \mid \mathcal{B}[\![b_1]\!]_\sigma = \text{false}\} = \Gamma(\varphi)$$

$\{ n \geq 0 \}$

```
x := 0
s := 0
while x ≤ n
    if mod K = 0 then s := s+x   (A_T)
                 else skip       (A_F)
    x := x+1                     (I)
```

$$\{ s = \sum_{y=0}^{n} \{ y \mid y \bmod K = 0 \} \}$$

Troviamo l'invariante eseguendo il programma e svolgendo alcune iterazioni del programma

$$s := \sum_{i=0}^{0} \{ i \mid i \bmod K = 0 \} \qquad x = 1$$

$$s := \sum_{i=0}^{1} \{ i \mid i \bmod K = 0 \} \qquad x = 2$$

$$s := \sum_{i=0}^{2} \{ i \mid i \bmod K = 0 \} \qquad x = 3$$

$$s := \sum_{i=0}^{3} \{ i \mid i \bmod K = 0 \} \qquad x = 4$$

$$\vdots \qquad\qquad \vdots$$

$$I \equiv s := \sum_{i=0}^{x-1} \{ i \mid i \bmod K = 0 \} \quad \wedge \quad x \leq n+1$$

Dimostriamo che I è invariante per farlo dobbiamo poter applicare la regola del while e per farlo dobbiamo mostrare che vale la consequence rule, ovvero che vale $I \wedge b_w \wedge b_{if} \rightarrow A_T$ e che $I \wedge b_w \wedge \neg b_{if} \rightarrow A_F$

Per calcolare A_T e A_F dobbiamo prima passare da $\hat{I}$
$\hat{I}$ viene calcolato applicando le interpretazioni dei comandi nel corpo del while

$$\hat{I} = I[^{x+1}/_x] = s := \sum_{i=0}^{x} \{ i \mid i \bmod K = 0 \} \quad \wedge \quad x + \cancel{1} \leq n + \cancel{1}$$

Per calcolare A_T e A_F applichiamo ad $\hat{I}$ le interpretazioni dei comandi nei rispettivi rami dell' if, calcolando dunque le weakest precondition

$$A_T = \hat{I}[^{s+x}/_s] = s+x := \sum_{i=0}^{x} \{ i \mid i \bmod K = 0 \} \quad \wedge \quad x \leq n$$

$$A_F = \hat{I}$$

$$I \wedge b_w \wedge b_{if} \equiv s := \sum_{i=0}^{x-1} \{ i \mid i \bmod K = 0 \} \quad \wedge \quad \underline{x \leq n+1} \quad \wedge \quad \underline{x \leq n} \quad \wedge \quad x \bmod K = 0$$

$$\equiv s := \sum_{i=0}^{x-1} \{ i \mid i \bmod K = 0 \} \wedge x \leq n \wedge x \bmod K = 0$$

$$\equiv s+x = \sum_{i=0}^{x} \{ i \mid i \bmod K = 0 \} \wedge x \leq n \wedge x \bmod K = 0 \equiv A_T$$

facciamo un passo oltre e incrementiamo il valore perché soddisfa la condizione

$$I \wedge b_w \wedge b_{if} \equiv s := \sum_{i=0}^{x-1} \{ i \mid i \bmod K = 0 \} \quad \wedge \quad \underline{x \leq n+1} \quad \wedge \quad \underline{x \leq n} \quad \wedge \quad x \bmod K \neq 0$$

$$\equiv s := \sum_{i=0}^{x-1} \{ i \mid i \bmod K = 0 \} \wedge \underline{x \leq n} \wedge x \bmod K \neq 0$$

$$\equiv s = \sum_{i=0}^{x} \{ i \mid i \bmod K = 0 \} \wedge x \leq n \wedge x \bmod K \neq 0 \equiv A_F$$

facciamo un passo oltre e non incrementiamo il valore perché non soddisfa la condizione

Posso dunque applicare la regola del while, vale infatti la consequence rule e vale dunque che I è invariante

Consequence rule — Consequence rule

$$\dfrac{\dfrac{\dfrac{\dfrac{\{A_T\}\, c_1\, \{\hat{I}\} \quad I \wedge b_w \wedge b_{if} \rightarrow A_T}{\{I \wedge b_w \wedge b_{if}\}\, c_1\, \{\hat{I}\}} \quad \dfrac{\{A_F\}\, c_2\, \{\hat{I}\} \quad I \wedge b_w \wedge \neg b_{if} \rightarrow A_F}{\{I \wedge b_w \wedge \neg b_{if}\}\, c_1\, \{\hat{I}\}}}{\{I \wedge b_w\}\ \text{if } b_{if} \text{ then } c_1 \text{ else } c_2\ \{\hat{I}\}} \quad \{\hat{I}\}\, c_3\, \{I\}}{\{I \wedge b_w\}\ c\ \{I\}}}{\{I\}\ \text{while } b_w \text{ do } c\ \{I \wedge \neg b_w\}}$$

Avendo derivato ciò so che I è invariante nel programma

perciò I sarà concorde con le precondizioni

$I \wedge \neg b_w$ sarà concorde con le postcondizioni

$$I \wedge \neg b_w \equiv s := \sum_{i=0}^{k-1} \{i \mid i \bmod K = 0\} \wedge x \leq n+1 \wedge x > n$$

$$\equiv x = n \wedge s := \sum_{i=0}^{n} \{i \mid i \bmod K = 0\} \equiv \text{postcondizioni} \quad \text{CVD}$$

$\{n \geq 0\}$

$x := 0$

$s := 0$

$$\longrightarrow I \equiv s := \sum_{i=0}^{k-1} \{i \mid i \bmod K = 0\} \wedge x \leq n+1$$

$$0 = \sum_{i=0}^{-1} \{i \mid i \bmod K = 0\} \wedge 0 \leq n+1$$

$$0 = \sum_{i=0}^{-1} \{\phi\} \wedge 0 \leq n+1$$

$$0 = 0 \wedge 0 \leq n+1 \quad \checkmark$$

per $n \geq 0$ ✓ CVD

$\{n \geq 0\}$

$s := 0$

$x := 0$

while $x \leq n$

  if $f(x) \neq 0$ then ^{A_T} $s := s + f(x)$

  else ^{A_F} skip

$\hat{I} \rightarrow$ $x := x + 1$

$\{s = \sum_{i=0}^{n} f(i)\}$

Troviamo l'invariante eseguendo il programma e alcune iterazioni del ciclo

$s = \sum_{i=0}^{0} f(i)$ $\quad x = 1$

$s = \sum_{i=0}^{1} f(i)$ $\quad x = 2$

$s = \sum_{i=0}^{2} f(i)$ $\quad x = 3$

$s = \sum_{i=0}^{3} f(i)$ $\quad x = 4$

$\vdots$

$I \equiv s = \sum_{i=0}^{x-1} f(i) \wedge x \leq n+1$

Ora che abbiamo trovato l'invariante dobbiamo mostrare che lo è
per farlo dobbiamo applicare le regole del while e per farlo
dobbiamo poter applicare la consequence rule.
Per farlo dobbiamo mostrare $I \wedge b_w \wedge b_{if} \rightarrow A_T$ e che $I \wedge b_{if} \wedge b_w \rightarrow A_F$
Per calcolare A_T e A_F ovvero le wp dobbiamo prima calcolare $\hat{I}$
Per trovare $\hat{I}$ dobbiamo applicare ad I l'interpretazione del comando $x = x+1$

$$\hat{I} = I[^{x+1}/_x] = s = \sum_{i=0}^{x} f(i) \wedge x+1 \leq n+1$$

Calcoliamo ora A_T e A_F in maniera del tutto analoga

$$A_T = \hat{I}[^{s+f(x)}/_s] = s + f(x) = \sum_{i=0}^{x} f(i) \wedge x+1 \leq n+1$$

$$A_F = \hat{I}$$

Mostriamo $I \wedge b_w \wedge b_{if} \rightarrow A_T$ e che $I \wedge b_w \wedge \neg b_{if} \rightarrow A_F$

$$I \wedge b_w \wedge b_{if} \equiv s = \sum_{i=0}^{x-1} f(i) \wedge x \leq n+1 \wedge x \leq n \wedge f(x) \neq 0$$

$$\equiv s + f(x) = \sum_{i=0}^{x} f(i) \wedge x \leq n \wedge f(x) \neq 0 \equiv A_T$$

facciamo un altro passo
e il valore viene aggiunto
perché soddisfa la condizione

$$I \wedge b_w \wedge \neg b_{if} \equiv s = \sum_{i=0}^{x-1} f(i) \wedge x \leq n+1 \wedge x \leq n \wedge f(x) = 0$$

$$\equiv s = \sum_{i=0}^{x} f(i) \wedge x \leq n \wedge f(x) = 0 \equiv A_F$$

facciamo un altro passo
e non aggiungiamo nulla
perché $f(x)$ è zero

Avendo dimostrato che $I \wedge b_w \wedge b_{if} \rightarrow A_T$ e che $I \wedge b_w \wedge \neg b_{if} \rightarrow A_F$
tramite le equivalenze, so che vale la consequence rule e che quindi posso
applicare la regola del while che ci dice che I è effettivamente invariante

consequence rule $\dfrac{\{A_T\}\, c_1\, \{\hat{I}\} \quad I \wedge b_w \wedge b_{if} \rightarrow \hat{I}}{\{I \wedge b_w \wedge b_{if}\}\, c_1\, \{\hat{I}\}}$ $\quad$ $\dfrac{\{A_F\}\, c\, \{\hat{I}\} \quad I \wedge b_w \wedge \neg b_{if} \rightarrow A_F}{\{I \wedge b_w \wedge \neg b_{if}\}\, c_2\, \{\hat{I}\}}$ consequence rule

$$\dfrac{\{I \wedge b_w\}\ \text{if } b \text{ then } c_1 \text{ else } c_2\ \{\hat{I}\} \quad \{\hat{I}\}\, c_3\, \{I\}}{\{I \wedge b_w\}\, c\, \{I\}}$$

$$\dfrac{\{I \wedge b_w\}\, c\, \{I\}}{\{I\}\ \text{while } b_w \text{ do } c\ \{I \wedge \neg b_w\}}$$

Sapendo che I è invariante devo verificare che $I \wedge \neg b$ vale per le postcondizioni e che I vale per le precondizioni

$$I \wedge \neg b_w = s = \sum_{i=0}^{x-1} f(i) \wedge x \leq n+1 \wedge x > n$$

$$= x = n+1 \wedge s = \sum_{i=0}^{n} f(i) \checkmark$$

concorde alle post condizioni

$\{n \geq 0\}$

$s := 0$

$x := 0$

$$I \equiv s = \sum_{i=0}^{x-1} f(i) \wedge x \leq n+1$$

$$0 = \sum_{i=0}^{-1} f(i) \wedge 0 \leq n+1$$

$$0 = \underbrace{\sum \{\phi\}}_{\text{sommatoria di zero}} \qquad 0 \leq n+1 \checkmark$$

perché $n \geq 0$

$$0 = 0 \checkmark$$

---

save x in c

$\ell = \phi$

$x = 5$

$\ell = 5$

$x = 5$

semantica operazionale

$$\frac{\langle c, \sigma \rangle \to \sigma'}{\langle \text{save } x \text{ in } c \rangle \to \sigma'[\sigma(x)/x]}$$

semantica denotazionale

$$\mathcal{C}[\![\text{save } x \text{ in } c]\!] = \{(\sigma, \sigma'[\sigma(x)/x]) \mid \mathcal{C}[\![c]\!]\sigma = \sigma'\}$$

IntDiv2(a)

$$\dfrac{\dfrac{\dfrac{\langle a,\sigma\rangle \to n \quad \langle 0,\sigma\rangle \to 0}{\langle a \bmod 2,\sigma\rangle \to m} \quad \langle 0,\sigma\rangle \to 0}{\langle a \bmod 2 \neq 0,\sigma\rangle \to \text{true}}}{\text{IntDiv}(a) \to a/2}$$

$$\dfrac{\dfrac{\dfrac{\langle a,\sigma\rangle \to n \quad \langle 0,\sigma\rangle \to 0}{\langle a \bmod 2,\sigma\rangle \to m} \quad \langle 0,\sigma\rangle \to 0}{\langle a \bmod 2 \neq 0,\sigma\rangle \to \text{false}}}{\text{IntDiv}(a) \to (a+1)/2}$$

$$\mathcal{A}[\![\text{IntDiv}(a)]\!] = \{(\sigma, a/2) \mid \mathcal{B}[\![a \bmod 2 = 0]\!]_\sigma = \text{true}\} \cup \{(\sigma, (a+1)/2) \mid \mathcal{B}[\![a \bmod 2 = 0]\!]_\sigma = \text{false}\}$$

$$\overset{!}{=} \{(\sigma, a/2) \mid \mathcal{A}[\![a \bmod 2]\!]_\sigma = \mathcal{A}[\![0]\!]_\sigma\} \cup \{(\sigma, (a+1)/2) \mid \mathcal{A}[\![a \bmod 2]\!]_\sigma \neq \mathcal{A}[\![0]\!]_\sigma\}$$

$\{ n \geq 0 \}$

$p := 0$

$c := 1$

while $c \leq n$ do

$\quad p := p + m$

$\quad c := c + 1$

$\{ p = n \cdot m \}$

Troviamo l'invariante eseguendo il programma scegliendo alcune iterazioni del ciclo

$$p = \underbrace{m}_{1} \qquad c = 2$$
$$p = \underbrace{m + m}_{2} \qquad c = 3$$
$$p = \underbrace{m + m + m}_{3} \qquad c = 4$$
$$\vdots \qquad \vdots$$

$$I = p = (c-1) \cdot m \ \wedge \ c \leq n+1$$

Dimostriamo che I è invariante, per farlo dobbiamo poter applicare la consequence rule e la regola del while.
Per applicare la regola del while dobbiamo mostrare $I \wedge b \rightarrow \hat{I}$
$\hat{I}$ lo calcoliamo calcolando la weakest precondition, applicando dunque ad I le interpretazioni dei comandi nel corpo del while

$$\hat{I} = I[^{c+1}/_{c}][^{p+m}/_{p}] \equiv p = (c-1) \cdot m \ \wedge \ c \leq n+1$$
$$\equiv p = c \cdot m \ \wedge \ c+1 \leq n+1$$
$$\equiv p + m = c \cdot m \ \wedge \ c \leq n$$

$$I \wedge b_w \equiv p = (c-1) \cdot m \ \wedge \ c \leq n+1 \ \wedge \ c \leq n$$
$$\equiv p + m = (c-1) \cdot m + m \ \wedge \ c \leq n$$
$$\equiv p + m = m(c - 1 + 1) \ \wedge \ c \leq n$$
$$\equiv p + m = m \cdot c \ \wedge \ c \leq n \equiv \hat{I}$$

Avendo mostrato che $I \wedge b \rightarrow \hat{I}$ posso applicare la consequence rule e ciò mi permette di poter applicare la regola del while, vediamo così che I è veramente invariante

$$\frac{\dfrac{\{\hat{I}\}\, c\, \{I\} \qquad I \wedge b \rightarrow \hat{I}}{\{I \wedge b\}\, c\, \{I\}}}{\{I\} \text{ while } b \text{ do } c \ \{I \wedge \neg b\}}$$

Essendo I invariante mostro che è I concorde alle precondizioni e $I \wedge \neg b$ concorde alle postcondizioni

$$I \wedge \neg b = p = (c-1) \cdot m \ \wedge \ c \leq n+1 \ \wedge \ c > n$$
$$= c = n+1 \ \wedge \ p = n \cdot m \ \checkmark$$

Concordanza con post condizioni

$\{ n \geq 0 \}$

$p := 0$ $\longrightarrow I$ $\quad p = (c-1) \cdot m \ \wedge \ c \leq n+1$

$c := 1$ $\quad 0 = 0 \cdot m \ \checkmark \qquad 1 \leq n+1 \ \checkmark$ perche $n \geq 0$ per le precondizioni

if b_1 then c_1 else if b_2 then c_1 else c_2 — if $b_1 \vee b_2$ then c_1 else c_2

$c_1 \sim c_2$ iff $\forall \sigma, \sigma' \in \Sigma \quad \langle c_1, \sigma \rangle \to \sigma' \iff \langle c_2, \sigma \rangle \to \sigma'$

$$\frac{\langle b_1, \sigma \rangle \to true \qquad \langle c_1, \sigma \rangle \to \sigma'}{\langle \text{if } b_1 \text{ then } c_1 \text{ else if } b_2 \text{ then } c_1 \text{ else } c_2, \sigma \rangle \to \sigma'} \iff \frac{\dfrac{\langle b_1, \sigma \rangle \to true}{\langle b_1 \vee b_2, \sigma \rangle \to true} \quad \langle c_1, \sigma \rangle \to \sigma'}{\text{if } b_1 \vee b_2 \text{ then } c_1 \text{ else } c_2, \sigma \rangle \Rightarrow \sigma'}$$

La condizione sarebbe comunque vera a prescindendo b_2

$$\frac{\langle b_2, \sigma \rangle \to false \qquad \dfrac{\langle b_2, \sigma \rangle \to true \qquad \langle c_1, \sigma \rangle \to \sigma'}{\langle \text{if } b_2 \text{ then } c_1 \text{ else } c_2, \sigma \rangle \to \sigma'}}{\langle \text{if } b_1 \text{ then } c_1 \text{ else if } b_2 \text{ then } c_1 \text{ else } c_2, \sigma \rangle \to \sigma'} \iff \frac{\dfrac{\langle b_1, \sigma \rangle \to false \qquad \langle b_2, \sigma \rangle \to true}{\langle b_1 \vee b_2, \sigma \rangle \to true} \quad \langle c_1, \sigma \rangle \to \sigma'}{\text{if } b_1 \vee b_2 \text{ then } c_1 \text{ else } c_2, \sigma \rangle \Rightarrow \sigma'}$$

$$\frac{\langle b_2, \sigma \rangle \to false \qquad \dfrac{\langle b_2, \sigma \rangle \to false \qquad \langle c_2, \sigma \rangle \to \sigma'}{\langle \text{if } b_2 \text{ then } c_1 \text{ else } c_2, \sigma \rangle \to \sigma'}}{\langle \text{if } b_1 \text{ then } c_1 \text{ else if } b_2 \text{ then } c_1 \text{ else } c_2, \sigma \rangle \to \sigma'} \iff \frac{\dfrac{\langle b_1, \sigma \rangle \to false \qquad \langle b_2, \sigma \rangle \to false}{\langle b_1 \vee b_2, \sigma \rangle \to true} \quad \langle c_2, \sigma \rangle \to \sigma'}{\text{if } b_1 \vee b_2 \text{ then } c_1 \text{ else } c_2, \sigma \rangle \Rightarrow \sigma'}$$

$\{ n = N \geq 0 \}$

$a := 0$

$b := 1$

while $n > 0$

$\hat{I}$ $\quad b := a + b$

$\quad a := b - a$

I $\quad n := n - 1$

$\{ a = fib(N) \}$

1 Troviamo l'invariante eseguendo il programma con alcune iterazioni del ciclo while

$b: 1$ fib 2	$a: 1$ fib 1	$n: N-1$
$b = 2$ fib 3	$a = 1$ fib 2	$n: N-2$
$b: 3$ fib 4	$a: 2$ fib 3	$n: N-3$
$b: 5$	$a: 3$	$n: N-4$

pag 12 $\quad I \equiv a = fib(N-n) \wedge b = fib(N-n+1) \wedge n \geq 0$

Per dimostrare che I è invariante dobbiamo poter applicare la regola del while nella derivazione, per farlo dobbiamo mostrare che $I \wedge b_w \rightarrow \hat{I}$ dobbiamo calcolare la wp $\hat{I}$ e vedere che l'implicazione vale.

$$\hat{I} = I[^{n-1}/_n][^{b-a}/_a][^{a+b}/_b]$$

$$= a = fib(N-n+1) \wedge b = fib(N-n+2) \wedge n-1 \geq 0$$

$$= b-a = fib(N-n+1) \wedge b = fib(N-n+2) \wedge n-1 \geq 0$$

$$= b = fib(N-n+1) \wedge b+a = fib(N-n+2) \wedge n-1 \geq 0$$

$$= b = fib(N-n+1) \wedge b+a = fib(N-n+2) \wedge n-1 \geq 0$$

$$I \wedge b_w = a = fib(N-n) \wedge b = fib(N-n+1) \wedge n \geq 0 \wedge n > 0$$

$$= a = fib(N-n) \wedge b = fib(N-n+1) \wedge n > 0$$

$$= b+a = fib(N-n) + fib(N-n+1) \wedge b = fib(N-n+1) \wedge n > 0$$

$$= b+a = fib(N-n+2) \wedge b = fib(N-n+1) \wedge n > 0$$

Abbiamo mostrato che $I \wedge b \rightarrow \hat{I}$ quindi vale la consequence rule, possiamo dunque applicare la regola del while e mostrare che I è invariante

$$\frac{\dfrac{\{\hat{I}\}\, c\, \{I\} \qquad I \wedge b \Rightarrow \hat{I}}{\{I \wedge b\}\, c\, \{I\}}\ \text{consequence rule}}{\{I\}\ \text{while } b \text{ do } c\ \{I \wedge \neg b\}}$$

Avendo mostrato che I è invariante devo verificare che
- I sia concorde alle precondizioni
- $I \wedge b$ sia concorde alle postcondizioni

$$I \wedge \neg b_w = a = fib(N-n) \wedge b = fib(N-n+1) \wedge n \geq 0 \wedge n \leq 0$$

$$= n = 0 \wedge a = fib(N) \wedge b = fib(N+1) \ \checkmark$$

$\{ n = N \geq 0 \}$

$a := 0$

$b := 1$

$a = fib(N-n) \wedge b = fib(N-n+1) \wedge n \geq 0$

$0 = fib(0) \wedge 1 = fib(1) \wedge n \geq 0$

✓ ✓ ✓

$\{x=0,\ count=0\}$

while $(x<n)$ do

  if $a[x]>A$ then $\overset{A_T}{count = count+1}$

  else $\overset{A_F}{skip}$

  $x=x+1$

$\{count = |\{y \mid a[y]>A,\ y\in[0,n)\}|\}$

Troviamo l'invariante eseguendo il programma e alcune iterazioni del ciclo

$count = |\{y \mid a[y]>A,\ y\in[0,1)\}|$ $\quad x=1$

$count = |\{y \mid a[y]>A,\ y\in[0,2)\}|$ $\quad x=2$

$count = |\{y \mid a[y]>A,\ y\in[0,3)\}|$ $\quad x=3$

$\vdots \qquad\qquad \vdots$

$I \equiv count = |\{y \mid a[y]>A,\ y\in[0,x-1]\}| \ \wedge\ x\le n$

Per dimostrare che I é invariante dobbiamo mostrare che $I\wedge b_w \wedge b_{if} \to A_T$ e che $I\wedge b_w\wedge \neg b_{if} \to A_F$ in modo da poter applicare la consequence rule e quindi poter applicare la regola del while che ci dice che I é invariante

A_T e A_F vengono calcolate calcolando le wp ovvero applicando ad I le interpretazioni dei comandi del while e dei rispettivi rami dell'if

Calcoliamo A_T e A_F passando da $\hat{I}$

$\hat{I} = I[x+1/x] \equiv count = |\{y \mid a[y]>A,\ y\in[0,x-1]\}| \ \wedge\ x\le n$

$\equiv count = |\{y \mid a[y]>A,\ y\in[0,x]\}| \ \wedge\ x+1\le n$

Calcoliamo ora A_T e A_F

$A_T \equiv I[count+1/count] \equiv count+1 = |\{y \mid a[y]>A,\ y\in[0,x]\}| \ \wedge\ x+1\le n$

$A_F \equiv \hat{I}$

$I\wedge b_w\wedge b_{if} \equiv count = |\{y \mid a[y]>A,\ y\in[0,x-1]\}| \ \wedge\ x\le n \ \wedge\ x<n \ \wedge\ a[x]>A$

$\equiv count = |\{y \mid a[y]>A,\ y\in[0,x-1]\}| \ \wedge\ x+1<n \ \wedge\ a[x]>A$

$\equiv count+1 = |\{y \mid a[y]>A,\ y\in[0,x]\}| \ \wedge\ x+1<n \ \wedge\ a[x]>A \equiv A_T$

facciamo un altro passo e aggiungiamo count perché il valore lo permette

$I\wedge b_w\wedge \neg b_{if} \equiv count = |\{y \mid a[y]>A,\ y\in[0,x-1]\}| \ \wedge\ x\le n \ \wedge\ x<n \ \wedge\ a[x]\le A$

$\equiv count = |\{y \mid a[y]>A,\ y\in[0,x-1]\}| \ \wedge\ x+1<n \ \wedge\ a[x]\le A$

$\equiv count = |\{y \mid a[y]>A,\ y\in[0,x]\}| \ \wedge\ x+1<n \ \wedge\ a[x]\le A \equiv A_F$

Avendo mostrato $I\wedge b_w\wedge b_{if} \to A_T$ e che $I\wedge b_w\wedge \neg b_{if} \to A_F$ ho quindi che posso applicare la consequence rule e quindi posso applicare la regola del while mostrando che I é invariante

$$\dfrac{\dfrac{\dfrac{\overset{A_T}{\overbrace{\{I\wedge b_w\wedge b_{if}\}}^{\text{consequence rule}}}\ c_1\ \{\hat{I}\}}{\{I\wedge b_w\}\ \text{if } b \text{ do } c_1 \text{ else } c_2\ \{\hat{I}\}} \qquad \dfrac{\overset{A_F}{\overbrace{\{I\wedge b_w\wedge \neg b_{if}\}}^{\text{consequence rule}}}\ c_2\ \{\hat{I}\}}{\{\hat{I}\}\ c_2\ \{I\}}}{\{I\wedge b_w\}\ c\ \{I\}}}{\{I\}\ \text{while } b_w \text{ do } c\ \{I\wedge \neg b_w\}}$$

$$I \wedge \neg b_w \equiv count = |\{y \mid a[y] > A,\ y \in [0, x-1]\}| \wedge x \leq n \wedge x \geq n$$

$$\equiv count = |\{y \mid a[y] > A,\ y \in [0, n-1]\}| \wedge x = n \quad \checkmark \text{ post condizioni}$$

$$\{x = 0, count = 0\} \rightarrow I \equiv count = |\{y \mid a[y] > A,\ y \in [0, x-1]\}| \wedge x \leq n$$

$$\equiv 0 = |\{y \mid a[y] > A,\ y \in [0, 0)\}| \wedge 0 \leq n$$

$$0 = |\{\emptyset\}| \wedge n \geq 0$$

$$\checkmark \qquad \checkmark \rightarrow N \geq 0$$

$\{n > 1\}$

$s = n$

$x = n - 1$

while $x > 0$ do

$\quad S := s * x$

$\quad x := x - 1$

$\{S = \prod_{x=1}^{n} x\}$

1 Troviamo l'invariante eseguendo il programma e alcune iterazioni del ciclo

$s = n \qquad x = N - 1$

$s = (n) \cdot (n-1) \qquad x = N - 2$

$s = (n) \cdot (n-1) \cdot (n-2) \qquad x = (n-3)$ $+1$ $n-2$

$s = (n) \cdot (n-1)(n-2)(n-3) \qquad x = (n-4)$ $+1$ $n-3$

$$I \equiv s = \sum_{i=x+1}^{n} i \quad \wedge \quad x \geq 0$$

$$\hat{I} = I[^{x-1}/_{x}][^{s*x}/_{x}] = s = \sum_{i=x}^{n} i \quad \wedge \quad x - 1 \geq 0$$

$$\equiv s * x = \sum_{i=x}^{n} i \quad \wedge \quad x - 1 \geq 0$$

$$I \wedge b_w \equiv s = \sum_{i=x+1}^{n} i \quad \wedge \quad x \geq 0 \quad \wedge \quad x > 0$$

$$\equiv s = \sum_{i=x+1}^{n} i \quad \wedge \quad x \geq 0$$

$$\equiv s * x = \sum_{i=x}^{n} \quad \wedge \quad x \geq 0$$

facciamo un ulteriore passo

$I \wedge b_w \rightarrow \hat{I}$

I è invariante

$$I \wedge \neg b_w \equiv s = \sum_{i=x+1}^{n} i \quad \wedge \quad x \geq 0 \quad \wedge \quad x > 0$$

pag 19

$$\equiv s = \sum_{i=x+1}^{n} i \quad \wedge \quad x \geq 0 \quad \wedge \quad x > 0$$

$$\equiv s = \sum_{i=x+1}^{n} i \quad \wedge \quad x \geq 0 \quad \wedge \quad x > 0$$

$$\equiv s * x = \sum_{i=x}^{n} i \quad \wedge \quad x > 0$$

$\{n > 1\}$

$s = n \quad \rightarrow I \qquad n = \sum_{i=n}^{n} i \quad \wedge \quad n - 1 \geq 0$

$x = n - 1$

$n = n \quad \wedge \quad n \geq 1$ ✓

$$\text{if } x \neq 0 \text{ then } x := 1 \text{ else } x := x+1 \quad \equiv \quad x := 1$$

$$P_1 \sim P_2 \iff \forall \sigma, \sigma' \in \Sigma \quad \langle P_1, \sigma \rangle \to \sigma' \iff \langle P_2, \sigma \rangle \to \sigma'$$

Per ipotesi è zero

Per ipotesi è zero

$$\dfrac{\dfrac{\langle x, \sigma \rangle \to \sigma(x) \quad \langle 0, \sigma \rangle \to 0 \;\checkmark}{\langle x \neq 0, \sigma \rangle \to \text{false}} \qquad \dfrac{\dfrac{\langle x, \sigma \rangle \to \sigma(x) \quad \langle 1, \sigma \rangle \to 1}{\langle x+1, \sigma \rangle \to \sigma(x)+1}}{\langle x := x+1, \sigma \rangle \to \sigma[\sigma(x)+1/x]}}{\langle \text{if } x \neq 0 \text{ then } x := 1 \text{ else } x := x+1, \sigma \rangle \to \sigma[\sigma(x)+1/x]} \iff \dfrac{\langle 1, \sigma \rangle \to 1}{\langle x := 1, \sigma \rangle \to \sigma[1/x]}$$

so che $\sigma(x)$ è 0 quindi otterrò il valore finale di 1

infatti ho lo stesso stato finale

$$\dfrac{\dfrac{\langle x, \sigma \rangle \to \sigma(x) \quad \langle 0, \sigma \rangle \to 0 \;\checkmark}{\langle x \neq 0, \sigma \rangle \to \text{true}} \qquad \dfrac{\langle 1, \sigma \rangle \to 1}{\langle x := 1, \sigma \rangle \to \sigma[1/x]}}{\langle \text{if } x \neq 0 \text{ then } x := 1 \text{ else } x := x+1, \sigma \rangle \to \sigma[1/x]} \iff \dfrac{\langle 1, \sigma \rangle \to 1}{\langle x := 1, \sigma \rangle \to \sigma[1/x]}$$

ho lo stesso stato finale

$$\begin{aligned}
\mathcal{C}[\![\text{if } x \neq 0 \text{ then } x := 1 \text{ else } x := x+1]\!] &= \{(\sigma, \sigma[1/x]) \mid \mathcal{B}[\![x \neq 0]\!]_\sigma = \text{true},\ \mathcal{C}[\![x := 1]\!]_\sigma = \sigma[1/x]\} \\
&\quad \cup \{(\sigma, \sigma[\sigma(x)+1/x]) \mid \mathcal{B}[\![x \neq 0]\!]_\sigma = \text{false},\ \mathcal{C}[\![x := x+1]\!]_\sigma = \sigma[x+1/x]\} \\
&= \{(\sigma, \sigma[1/x] \mid \underbrace{\mathcal{B}[\![x \neq 0]\!]_\sigma = \text{true} \vee \mathcal{B}[\![x \neq 0]\!]_\sigma = \text{false}}_{\text{ho } \sigma(x) = 1 \text{ valore finale per ogni valore di } x}\} \\
&= \{(\sigma, \sigma[1/x] \mid A[\![1]\!] = 1\} \;\checkmark \\
&= \mathcal{C}[\![x := 1]\!]
\end{aligned}$$

ma per la condizione di $\sigma(x)$ è 0

Alternate(b, c_1, c_2)

Denotazionale

$$\frac{\langle b, \sigma \rangle \to \text{false}}{\langle \text{Alternate}(b, c_1, c_2), \sigma \rangle \to \sigma}$$

$$\frac{\langle b, \sigma \rangle \to \text{true} \qquad \langle c_1; \text{Alternate}(b, c_2, c_1), \sigma \rangle \to \sigma'}{\langle \text{Alternate}(b, c_1, c_2), \sigma \rangle \to \sigma'}$$

Operazionale

$$\mathcal{C}[\![\text{Alternate}(b, c_1, c_2)]\!] \equiv \{(\sigma, \sigma) \mid \mathcal{B}[\![b]\!]_\sigma = \text{false}\} \cup \{(\sigma, \sigma') \mid \mathcal{B}[\![b]\!]_\sigma = \text{true},\ \mathcal{C}[\![c_1; \text{Alternate}(b, c_2, c_1)]\!]_\sigma = \sigma'\}$$

$$\equiv \{(\sigma, \sigma) \mid \mathcal{B}[\![b]\!]_\sigma = \text{false}\} \cup \{(\sigma, \sigma') \mid \mathcal{B}[\![b]\!]_\sigma = \text{true},\ \mathcal{C}[\![c_1]\!]_\sigma = \sigma'',\ \mathcal{C}[\![\text{Alternate}(b, c_2, c_1)]\!]_{\sigma''} = \sigma'\}$$

while b do c if b then c_2 else skip

$P_1 \sim P_2$ iff $\forall \sigma, \sigma' \in \Sigma$

$$\frac{\langle b, \sigma \rangle \to \text{false}}{\langle \text{Alternate}(b, c_1, c_2), \sigma \rangle \to \sigma} \iff \frac{\langle b, \sigma \rangle \to \text{false}}{\langle \text{while } b \text{ do } c \text{ if } b \text{ then } c_2 \text{ else skip}, \sigma \rangle \to \sigma}$$

$$\frac{\langle b, \sigma \rangle \to \text{true} \qquad \dfrac{\langle c_1, \sigma \rangle \to \sigma'' \qquad \dfrac{\langle b, \sigma'' \rangle \to \text{false}}{\langle \text{Alternate}(b, c_2, c_1), \sigma'' \rangle \to \sigma''}}{\langle c_1; \text{Alternate}(b, c_2, c_1), \sigma \rangle \to \sigma'}}{\langle \text{Alternate}(b, c_1, c_2), \sigma \rangle \to \sigma''} \iff$$

$$\frac{\langle b, \sigma \rangle \to \text{true} \qquad \dfrac{\langle c_1, \sigma \rangle \to \sigma'' \qquad \dfrac{\langle b, \sigma'' \rangle \to \text{false}}{\langle \text{if } b \text{ then } c_2 \text{ else skip}, \sigma'' \rangle \to \sigma''}}{\langle c_1; \text{if } b \text{ then } c_2 \text{ else skip}, \sigma \rangle \to \sigma''}}{\langle \text{while } b \text{ do } c_1; \text{if } b \text{ then } c_2 \text{ else skip}, \sigma \rangle \to \sigma''} \qquad \frac{\langle b, \sigma'' \rangle \to \text{false}}{\langle \text{while } b \text{ do } c_1; \text{if } b \text{ then } c_2 \text{ else skip}, \sigma'' \rangle \to \sigma''}$$

$$\frac{\langle b,\sigma\rangle \to \text{true} \quad \langle c_1;\text{Alternate}(b,c_2,c_1),\sigma\rangle \to \sigma'}{\langle \text{Alternate}(b,c_1,c_2),\sigma\rangle \to \sigma'}$$

$$\frac{\langle c_1,\sigma\rangle \to \sigma'' \quad \langle \text{Alternate}(b,c_2,c_1),\sigma''\rangle \to \sigma'}{\langle c_1;\text{Alternate}(b,c_2,c_1),\sigma\rangle \to \sigma'}$$

$$\frac{\langle b,\sigma''\rangle \to \text{true} \quad \langle c_2;\text{Alternate}(b,c_2,c_1),\sigma''\rangle \to \sigma'}{\langle \text{Alternate}(b,c_2,c_1),\sigma''\rangle \to \sigma'}$$

$$\frac{\langle c_2,\sigma''\rangle \to \sigma'}{\langle c_2;\text{Alternate}(b,c_2,c_1),\sigma''\rangle \to \sigma'}$$

$$\frac{\langle b,\sigma'\rangle \to \text{false}}{\text{Alternate}(b,c_2,c_1),\sigma'\rangle \to \sigma'}$$

$$\Updownarrow$$

$$\frac{\langle b,\sigma\rangle \to \text{true} \quad \langle c_1;\text{if } b \text{ then } c_2 \text{ else skip},\sigma\rangle \to \sigma'}{\langle \text{while } b \text{ do } c_1;\text{if } b \text{ then } c_2 \text{ else skip},\sigma\rangle \to \sigma'}$$

$$\frac{\langle c_1,\sigma\rangle \to \sigma'' \quad \langle \text{if } b \text{ then } c_2 \text{ else skip},\sigma''\rangle \to \sigma'}{\langle c_1;\text{if } b \text{ then } c_2 \text{ else skip},\sigma\rangle \to \sigma'}$$

$$\frac{\langle b,\sigma''\rangle \to \text{true} \quad \langle c_2,\sigma''\rangle \to \sigma'}{\langle \text{if } b \text{ then } c_2 \text{ else skip},\sigma''\rangle \to \sigma'}$$

$$\frac{\langle b,\sigma'\rangle \to \text{false}}{\langle \text{while } b \text{ do } c_1;\text{if } b \text{ then } c_2 \text{ else skip},\sigma'\rangle \to \sigma'}$$

$\{y=0,\ n\geq 0,\ x=1\}$
while $y<n$ do:
$y := y+1;$
$x := 3\cdot x;$
$\{x=3^n\}$

1 Troviamo l'invariante I eseguendo alcune iterazioni del programma e svolgendo le iterazioni del ciclo

$y=1 \qquad x=3$

$y=2 \qquad x=3\cdot 3$

$y=3 \qquad x=3\cdot 3\cdot 3$

$\vdots \qquad \vdots$

$$I \equiv x = 3^y \wedge y \leq n$$

Dimostriamo che I è invariante applicando la regola del while ma per poterlo fare devo applicare la consequence rule, devo dunque dimostrare $I \wedge b \rightarrow \hat{I}$

$\hat{I}$ lo calcolo applicando ad I le interpretazioni dei comandi e calcolando dunque la wp

$$\hat{I} = I[3\cdot x/x][y+1/y] = 3x = 3^{y+1} \wedge y+1 \leq n$$

$$I \wedge b_\omega \equiv x = 3^y \wedge y \leq n \wedge y < n$$
$$\equiv 3x = 3^y \cdot 3 \wedge y < n$$
$$\equiv 3x = 3^{y+1} \wedge y+1 \leq n \equiv \hat{I}$$

Avendo mostrato che $I \wedge b \rightarrow \hat{I}$ posso applicare la consequence rule, posso dunque applicare la regola del while e da ciò deriva che I è invariante

$$\dfrac{\dfrac{\{\hat{I}\} c \{I\} \quad I \wedge b \rightarrow \hat{I}}{\{I \wedge b\} c \{I\}}\ \text{consequence rule}}{\{I\}\ \text{while } b \text{ do } c\ \{I \wedge \neg b\}}$$

Verifichiamo che I sia concorde alle precondizioni e che $I \wedge \neg b$ sia concorde alle postcondizioni

$$I \wedge \neg b \equiv x = 3^y \wedge y \leq n \wedge y \geq n$$
$$\equiv x = 3^y \wedge y = n$$
$$\equiv x = 3^n \wedge y = n$$

$$\left\{\begin{array}{l} y=0 \\ n\geq 0 \\ x=1 \end{array}\right\} \rightarrow I \equiv x = 3^y \wedge y \leq n$$

$$1 = 3^0 \wedge 0 \leq n$$

$\checkmark \qquad \checkmark \rightarrow n \geq 0$

```
{y = n, n ≥ 0, x = 1}
while y ≠ 0 do :
    {[illegible]}
    y := y-1 ;
    {[illegible]}
    x := 2*x;
    {[illegible]}
{x = 2^n}
```

1 Troviamo l'invariante eseguendo il programma e alcune iterazioni del ciclo

$y = n-1 \quad x = 2$

$y = n-2 \quad x = 4$

$y = n-3 \quad x = 8$

$y = n-4 \quad x = 16$

$$I \equiv x = 2^{n-y} \wedge y \geq 0$$

Dimostriamo che I è invariante, per farlo dobbiamo applicare la regola del while e per farlo dobbiamo poter applicare la consequence rule, per applicarla dobbiamo dimostrare $I \wedge b \rightarrow \hat{I}$.

$\hat{I}$ lo calcoliamo calcolando la wp, ovvero applicando ad I le interpretazioni dei comandi nel while

$$\hat{I} = I[^{2*x}/_x][^{y-1}/_y] = 2*x = 2^{n-y} \wedge y \geq 0$$

$$\equiv 2*x = 2^{n-y+1} \wedge \underbrace{y-1 \geq 0}_{y>0}$$

$$I \wedge b_w \equiv x = 2^{n-y} \wedge y \geq 0 \wedge y \neq 0$$

$$\equiv x \cdot 2 = 2^{n-y} \cdot 2 \wedge y > 0$$

$$\equiv 2x = 2^{n-y+1} \wedge y > 0$$

$I \wedge b \rightarrow \hat{I}$ vale

Avendo mostrato che $I \wedge b \rightarrow \hat{I}$ so che posso applicare la consequence rule, posso dunque applicare la regola del while ciò mi dice che I è invariante

$$\frac{\dfrac{\{\hat{I}\}\, c\, \{I\} \qquad I \wedge b \rightarrow \hat{I}}{\{I \wedge b\}\, c\, \{I\}}}{\{I\}\ \text{while } b \text{ do } c\ \{I \wedge \neg b\}}$$

Sapendo che I è invariante devo mostrare che I è concorde alle pre condizioni

e $I \wedge \neg b$ è concorde con le post condizioni

$$I \wedge \neg b \equiv x = 2^{n-y} \wedge y \geq 0 \wedge y = 0$$

$$\equiv y = 0 \wedge x = 2^n$$

$$\left\{\begin{array}{l} y = n \\ n \geq 0 \\ x = 1 \end{array}\right\} \rightarrow I \qquad x = 2^{n-y} \wedge y \geq 0$$

$0 = 2^{n-n}$ ✓ $\qquad y \geq 0 \rightarrow y = n \geq 0$ ✓

if b then c

semantica operazionale

$$\frac{\langle b,\sigma\rangle \to false}{\langle if\ b\ then\ c,\sigma\rangle \Rightarrow \sigma}$$

$$\frac{\langle b,\sigma\rangle \to true \qquad \langle c,\sigma\rangle \to \sigma'}{\langle if\ b\ then\ c,\sigma\rangle \Rightarrow \sigma'}$$

semantica denotazionale

$$\mathcal{C}[\![if\ b\ then\ c]\!] = \{(\sigma,\sigma) \mid \mathcal{B}[\![b]\!]_\sigma = false\} \cup \{(\sigma,\sigma') \mid \mathcal{B}[\![b]\!]_\sigma = true,\ \mathcal{C}[\![c]\!]_\sigma = \sigma'\}$$

Equivalenza semantica operazionale

$P_1 \sim P_2$ iff $\forall \sigma,\sigma' \in \Sigma \quad \langle P_1,\sigma\rangle \to \sigma' \Leftrightarrow \langle P_2,\sigma\rangle \to \sigma'$

$$\frac{\langle b,\sigma\rangle \to false}{\langle if\ b\ then\ c,\sigma\rangle \Rightarrow \sigma} \quad \longleftrightarrow \quad \frac{\langle b,\sigma\rangle \to false \qquad \langle skip,\sigma\rangle \Rightarrow \sigma}{\langle if\ b\ then\ c\ else\ skip,\sigma\rangle \Rightarrow \sigma}$$

$$\frac{\langle b,\sigma\rangle \to true \qquad \langle c,\sigma\rangle \to \sigma'}{\langle if\ b\ then\ c,\sigma\rangle \Rightarrow \sigma'} \quad \longleftrightarrow \quad \frac{\langle b,\sigma\rangle \to true \qquad \langle c,\sigma\rangle \to \sigma'}{\langle if\ b\ then\ c,\sigma\rangle \Rightarrow \sigma'}$$

Equivalenza semantica denotazionale

$$\begin{aligned}
\mathcal{C}[\![if\ b\ then\ c\ else\ skip]\!]_\sigma &= \{(\sigma,\sigma') \mid \mathcal{B}[\![b]\!]_\sigma = true,\ \mathcal{C}[\![c]\!]_\sigma = \sigma'\} \cup \{(\sigma,\sigma) \mid \mathcal{B}[\![b]\!]_\sigma = false,\ \mathcal{C}[\![skip]\!]_\sigma = \sigma\} \\
&\overset{!}{=} \{(\sigma,\sigma') \mid \mathcal{B}[\![b]\!]_\sigma = true,\ \mathcal{C}[\![c]\!]_\sigma = \sigma'\} \cup \{(\sigma,\sigma) \mid \mathcal{B}[\![b]\!]_\sigma = false\} \\
&\overset{!}{=} \mathcal{C}[\![if\ b\ then\ c]\!]
\end{aligned}$$

$\{x=N,\ y=M,\ p=0,\ d=0\}$

while $x \le y$

if $x \bmod 2 = 0$ then $\overset{A_T}{p := p+x}$

else $\overset{A_F}{d := d+x}$

$\hat{I} \rightarrow$ $x := x+1$ $\leftarrow I$

$\{p = \sum_{i=N}^{M} \{i \mid i \text{ pari}\},\ d = \sum_{i=N}^{M} \{i \mid i \text{ dispari}\}\}$

Troviamo l'invariante eseguendo il programma e alcune iterazioni del ciclo

$$p = \sum_{i=N}^{N} \{i \mid i \text{ è pari}\} \qquad d = \sum_{i=N}^{N} \{i \mid i \text{ è dispari}\} \qquad x = N+1$$

$$p = \sum_{i=N}^{N+1} \{i \mid i \text{ è pari}\} \qquad d = \sum_{i=N}^{N+1} \{i \mid i \text{ è dispari}\} \qquad x = N+2$$

$$p = \sum_{i=N}^{N+2} \{i \mid i \text{ è pari}\} \qquad d = \sum_{i=N}^{N+2} \{i \mid i \text{ è dispari}\} \qquad x = N+3$$

$$\vdots \qquad\qquad \vdots \qquad\qquad \vdots$$

$$I \equiv p = \sum_{i=N}^{x-1} \{i \mid i \text{ è pari}\} \wedge d = \sum_{i=N}^{x-1} \{i \mid i \text{ è dispari}\} \wedge x \le y+1$$

Abbiamo trovato l'invariante, dobbiamo però verificarlo, dobbiamo però per dire che I è invariante, poter applicare la regola del while e per farlo dobbiamo poter applicare la consequence rule. Per applicarla dobbiamo mostrare che $I \wedge b_w \wedge b_{if} \rightarrow A_T$ e che $I \wedge b_w \wedge \neg b_{if} \rightarrow A_F$

Per calcolare A_T e A_F dobbiamo calcolare la wp e per farlo dobbiamo applicare le interpretazioni dei comandi nel corpo del while e nei rispettivi rami dell'if.

Passiamo a calcolare $\hat{I}$ in maniera analoga

$$\hat{I} = I[^{x+1}/_x] = p = \sum_{i=N}^{x} \{i \mid i \text{ è pari}\} \wedge d = \sum_{i=N}^{x} \{i \mid i \text{ è dispari}\} \wedge x+1 \le y+1$$

$$A_T = \hat{I}[^{p+x}/_p] = p+x = \sum_{i=N}^{x} \{i \mid i \text{ è pari}\} \wedge d = \sum_{i=N}^{x} \{i \mid i \text{ è dispari}\} \wedge x+1 \le y+1$$

$$A_F = \hat{I}[^{d+x}/_d] = p = \sum_{i=N}^{x} \{i \mid i \text{ è pari}\} \wedge d+x = \sum_{i=N}^{x} \{i \mid i \text{ è dispari}\} \wedge x+1 \le y+1$$

$$I \wedge b_w \wedge b_{if} \equiv p = \sum_{i=N}^{x-1} \{i \mid i \text{ è pari}\} \wedge d = \sum_{i=N}^{x-1} \{i \mid i \text{ è dispari}\} \wedge x \le y+1 \wedge x \le y \wedge x \bmod 2 = 0$$

$$\equiv p = \sum_{i=N}^{x-1} \{i \mid i \text{ è pari}\} \wedge d = \sum_{i=N}^{x-1} \{i \mid i \text{ è dispari}\} \wedge x \le y \wedge x \bmod 2 = 0$$

$$\equiv p+x = \sum_{i=N}^{x} \{i \mid i \text{ è pari}\} \wedge d = \sum_{i=N}^{x} \{i \mid i \text{ è dispari}\} \wedge x \le y \wedge x \bmod 2 = 0$$

facciamo un altro passo e aggiungiamo il valore a p perché $x \bmod 2 = 0$ perché x è pari

$$I \wedge b_w \wedge \neg b_{if} \equiv p = \sum_{i=N}^{x-1} \{i \mid i \text{ è pari}\} \wedge d = \sum_{i=N}^{x-1} \{i \mid i \text{ è dispari}\} \wedge x \le y+1 \wedge x \le y \wedge x \bmod 2 \ne 0$$

$$\equiv p = \sum_{i=N}^{x-1} \{i \mid i \text{ è pari}\} \wedge d = \sum_{i=N}^{x-1} \{i \mid i \text{ è dispari}\} \wedge x \le y \wedge x \bmod 2 \ne 0$$

$$\equiv p = \sum_{i=N}^{x} \{i \mid i \text{ è pari}\} \wedge d+x = \sum_{i=N}^{x} \{i \mid i \text{ è dispari}\} \wedge x \le y \wedge x \bmod 2 \ne 0$$

facciamo un altro passo e aggiungiamo il valore a d perché $x \bmod 2 \ne 0$ perché x è dispari

Avendo mostrato che $I \wedge b_w \wedge b_{if} \rightarrow A_T$ e che $I \wedge b_w \wedge \neg b_{if} \rightarrow A_F$ so che posso applicare la consequence rule, di conseguenza posso applicare la regola del while e quindi ho che I è invariante

Conseguenza rule $\{A_T\}\, c_1\, \{\hat{I}\}$ $\quad I \wedge b_w \wedge b_{if} \Rightarrow A_T$ $\qquad$ $\{A_F\}\, c_1\, \{\hat{I}\}$ $\quad I \wedge b_w \wedge \neg b_{if} \Rightarrow A_F$ Conseguenza rule

$\{I \wedge b_w \wedge b_{if}\}\, c_2\, \{\hat{I}\}$ $\qquad$ $\{I \wedge b_w \wedge \neg b_{if}\}\, c_2\, \{\hat{I}\}$

$\{I \wedge b_w\}$ if b_{if} do c_1 else c_2 $\{\hat{I}\}$ $\qquad$ $\{\hat{I}\}\, c_3\, \{I\}$

$\{I \wedge b_w\}\, c\, \{I\}$

$\{I\}$ while b_w do c $\{I \wedge \neg b_w\}$

Avendo mostrato che I è invariante so che I deve corrispondere alle precondizioni del programma e $I \wedge \neg b_w$ con le post condizioni

$$I \wedge \neg b_w \equiv p = \sum_{i=N}^{x-1} \{i \mid i \text{ è pari}\} \wedge d = \sum_{i=N}^{x-1} \{i \mid i \text{ è dispari}\} \wedge \underline{x \leq y+1} \wedge \underline{x > y}$$

$$\equiv x = y+1 \wedge p = \sum_{i=N}^{y} \{i \mid i \text{ è pari}\} \wedge d = \sum_{i=N}^{y} \{i \mid i \text{ è dispari}\}$$

$$\equiv x = y+1 \wedge p = \sum_{i=N}^{N} \{i \mid i \text{ è pari}\} \wedge d = \sum_{i=N}^{N} \{i \mid i \text{ è dispari}\} \equiv \text{post condizioni} \checkmark$$

$$\{x = N,\ y = M,\ p = 0,\ d = 0\} \rightarrow I$$

$$I \equiv p = \sum_{i=N}^{x-1} \{i \mid i \text{ è pari}\} \wedge d = \sum_{i=N}^{x-1} \{i \mid i \text{ è dispari}\} \wedge x \leq y$$

$$\equiv 0 = \sum_{i=N}^{N-1} \{i \mid i \text{ è pari}\} \wedge 0 = \sum_{i=N}^{N-1} \{i \mid i \text{ è dispari}\} \wedge N \leq M \checkmark$$

$$\equiv 0 = \sum \{\phi\} \qquad 0 = \sum \phi$$

zero elementi $\qquad$ zero elementi

$0 = 0 \checkmark \qquad 0 = 0 \checkmark$

Alternate (b, c_1, c_2)

$$\frac{\langle b, \sigma \rangle \to \text{false}}{\langle \text{Alternate}(b, c_1, c_2), \sigma \rangle \to \sigma}$$

$$\frac{\langle b, \sigma \rangle \to \text{false} \qquad \langle c_1; \text{Alternate}(b, c_1, c_2), \sigma \rangle \to \sigma'}{\langle \text{Alternate}(b, c_1, c_2), \sigma \rangle \to \sigma'}$$

Semantica denotazionale

$$\mathcal{C}[\![\overset{\omega}{\text{Alternate}}(b, c_1, c_2)]\!] = \Gamma(\varphi)$$

$$= \{(\sigma, \sigma'') \mid \beta[\![b]\!]\sigma = \text{false}\} \cup \{(\sigma, \sigma'') \mid \beta[\![b]\!]\sigma = \text{true}, \mathcal{C}[\![c_1]\!]\sigma = \sigma'', \beta[\![b]\!]\sigma'' = \text{false}\}$$

$$\cup \{(\sigma, \sigma') \mid \beta[\![b]\!]\sigma = \text{true}, \mathcal{C}[\![c_1]\!]\sigma = \sigma'', \beta[\![b]\!]\sigma'' = \text{true}, \mathcal{C}[\![c_2; \omega]\!]\sigma'' = \sigma'\}$$

$$= \mathcal{C}[\![\underbrace{\text{if } b \ c_1; \text{if } b \text{ then } c_2; \omega \text{ else skip}}_{\omega}]\!]$$

Dimostrare equivalenza con semantica operazionale e denotazionale tra

$$\overset{P_1}{\text{Alternate}(b, c_1, c_2)} \sim \overset{P_2}{\text{while } b \text{ do } (c_1; \text{if } b \text{ then } c_2 \text{ else skip})}$$

Vogliamo mostrare che

$$P_1 \sim P_2 \quad \text{iff} \quad \forall \sigma, \sigma' \in \Sigma \quad \langle P_1, \sigma \rangle \to \sigma' \iff \langle P_2, \sigma \rangle \to \sigma'$$

$$\frac{\langle b, \sigma \rangle \to \text{false}}{\langle \text{Alternate}(b, c_1, c_2), \sigma \rangle \to \sigma} \iff \frac{\langle b, \sigma \rangle \to \text{false}}{\langle \text{while } b \text{ do } (c_1; \text{if } b \text{ then } c_2 \text{ else skip}), \sigma \rangle \to \sigma} \quad \checkmark$$

$$\frac{\langle b, \sigma \rangle \to \text{true} \qquad \dfrac{\langle c_1; \sigma \rangle \to \sigma'' \qquad \dfrac{\langle b, \sigma'' \rangle \to \text{false}}{\langle \text{Alternate}(b, c_2, c_1), \sigma'' \rangle \to \sigma''}}{\langle c_1; \text{Alternate}(b, c_2, c_1), \sigma \rangle \to \sigma''}}{\langle \text{Alternate}(b, c_1, c_2), \sigma \rangle \to \sigma''} \quad \checkmark$$

$$\Updownarrow$$

$$\frac{\langle b, \sigma \rangle \to \text{true} \qquad \dfrac{\langle c_1, \sigma \rangle \to \sigma'' \qquad \dfrac{\langle b, \sigma'' \rangle \to \text{false}}{\langle \text{if } b \text{ then } c_2 \text{ else skip}, \sigma'' \rangle \to \sigma''}}{\langle c_1; \text{if } b \text{ then } c_2 \text{ else skip}, \sigma \rangle \to \sigma''} \qquad \dfrac{\langle b, \sigma'' \rangle \to \text{false}}{\langle \text{while } b \text{ do } (c_1; \text{if } b \text{ then } c_2 \text{ else skip}), \sigma'' \rangle \to \sigma''}}{\langle \text{while } b \text{ do } (c_1; \text{if } b \text{ then } c_2 \text{ else skip}), \sigma \rangle \to \sigma''}$$

$$\dfrac{\langle b, \sigma \rangle \to \text{true} \qquad \dfrac{\langle c_1, \sigma \rangle \to \sigma'' \qquad \dfrac{\langle b, \sigma'' \rangle \to \text{true} \qquad \dfrac{\langle c_2, \sigma'' \rangle \to \sigma' \qquad \dfrac{\langle b, \sigma' \rangle \to \text{false}}{\langle \text{Alternate}(b, c_1, c_2), \sigma' \rangle \to \sigma'}}{\langle c_2; \text{Alternate}(b, c_1, c_2), \sigma'' \rangle \to \sigma'}}{\langle \text{Alternate}(b, c_2, c_1), \sigma'' \rangle \to \sigma'}}{\langle c_1; \text{Alternate}(b, c_2, c_1), \sigma \rangle \to \sigma'}}{\langle \text{Alternate}(b, c_1, c_2), \sigma \rangle \to \sigma'}$$

$$\Updownarrow$$

$$\dfrac{\langle b, \sigma \rangle \to \text{true} \qquad \dfrac{\langle c_1, \sigma \rangle \to \sigma'' \qquad \dfrac{\langle b, \sigma'' \rangle \to \text{true} \qquad \langle c_2, \sigma'' \rangle \to \sigma'}{\langle \text{if } b \text{ then } c_2 \text{ else skip}, \sigma'' \rangle \to \sigma'}}{\langle c_1; \text{if } b \text{ then } c_2 \text{ else skip}, \sigma \rangle \to \sigma'} \qquad \dfrac{\langle b, \sigma' \rangle \to \text{false}}{\langle \text{while } b \text{ do } (c_1; \text{if } b \text{ then } c_2 \text{ else skip}), \sigma' \rangle \to \sigma'}}{\langle \text{while } b \text{ do } (c_1; \text{if } b \text{ then } c_2 \text{ else skip}), \sigma \rangle \to \sigma'}$$

$\{x=n,\ n\geq 0,\ y=1\}$

while $x>0$ do:

$\hat{I} \rightarrow$ $y = x*y$

$x = x-1$

$\{y=n!\}$

Troviamo l'invariante eseguendo il programma e alcune iterazioni di ciclo

$$\begin{array}{ll} y=n & x=n-1 \\ y=n\cdot(n-1) & x=n-2 \\ y=n\cdot(n-1)\cdot(n-2) & x=n-3 \\ \vdots & \vdots \end{array}$$

$$I \equiv y = \frac{n!}{x!} \wedge x \geq 0$$

Per dimostrare che I è invariante dobbiamo poter applicare la regola del while e per farlo ci serve la consequence rule.
Per applicare la consequence rule dobbiamo dimostrare che $I \wedge b \rightarrow \hat{I}$
per calcolare $\hat{I}$ applichiamo ad I le interpretazioni dei comandi nel corpo del while

$$\hat{I} = I[x-1/x][x*y/y] = y = \frac{n!}{(x-1)!} \wedge x-1 \geq 0$$

$$= x*y = \frac{n!}{(x-1)!} \wedge x-1 \geq 0$$

$$I \wedge b_w \equiv y = \frac{n!}{x!} \wedge x \geq 0 \wedge x > 0$$

$$\equiv y = \frac{n!}{x!} \wedge x > 0$$

$$= y*x = \frac{n!}{x!}\cdot x \wedge x-1>0$$

$$= y*x = \frac{n!}{\cancel{x}(x-1)!}\cdot \cancel{x} \wedge x-1>0 \equiv \hat{I}$$

Avendo mostrato che $I \wedge b_w \rightarrow \hat{I}$ posso applicare la consequence rule, di conseguenza posso applicare la regola del while e quindi so che I è invariante.

$$\frac{\dfrac{\{\hat{I}\}\ c\ \{I\} \qquad I\wedge b \rightarrow \hat{I}}{\{I\wedge b\}\ c\ \{I\}}\ \text{consequence rule}}{\{I\}\ \text{while } b \text{ do } c\ \{I \wedge \neg b\}}$$

Sapendo che I è invariante verifichiamo che I soddisfa le precondizioni e $I \wedge \neg b$ soddisfa le postcondizioni — per la regola del while

$$I \wedge \neg b \equiv y = \frac{n!}{x!} \wedge x \geq 0 \wedge x \leq 0$$

$$= x = 0 \wedge y = \frac{n!}{0!}$$

$$= x = 0 \wedge y = n! \ \checkmark$$

$\{x=n,\ n\geq 0,\ y=1\} \rightarrow I$ $\qquad y = \frac{n!}{x!} \wedge x \geq 0$

$1 = \frac{n!}{n!} \wedge x \geq 0 \checkmark$ poiché $x=n$ con $n \geq 0$

$\max(a_1, a_2)$

Semantica operazionale

$$\frac{\dfrac{\langle a_1,\sigma\rangle \to n_1 \quad \langle a_2,\sigma\rangle \to n_2}{\langle a_1 > a_2,\sigma\rangle \to \text{true}}}{\langle \max(a_1,a_2),\sigma\rangle \to a_1}$$

se si intende il valore valutato lo scegliamo così

$$\frac{\dfrac{\langle a_1,\sigma\rangle \to n_1 \quad \langle a_2,\sigma\rangle \to n_2}{\langle a_1 > a_2,\sigma\rangle \to \text{true}} \quad \langle a_1,\sigma\rangle \to n_1}{\langle \max(a_1,a_2),\sigma\rangle \to n_1}$$

$$\frac{\dfrac{\langle a_1,\sigma\rangle \to n_1 \quad \langle a_2,\sigma\rangle \to n_2}{\langle a_1 > a_2,\sigma\rangle \to \text{false}}}{\langle \max(a_1,a_2),\sigma\rangle \to a_2}$$

$$\frac{\dfrac{\langle a_1,\sigma\rangle \to n_1 \quad \langle a_2,\sigma\rangle \to n_2}{\langle a_1 > a_2,\sigma\rangle \to \text{false}} \quad \langle a_2,\sigma\rangle \to n_2}{\langle \max(a_1,a_2),\sigma\rangle \to n_2}$$

Semantica Denotazionale

$$A[\![\max(a_1,a_2)]\!] = \{(\sigma, a_1) \mid B[\![a_1 > a_2]\!]_\sigma = \text{true}\} \cup \{(\sigma, a_2) \mid B[\![a_1 > a_2]\!] = \text{false}\}$$

$$= \{(\sigma, a_1) \mid A[\![a_1]\!]_\sigma > A[\![a_2]\!]_\sigma\} \cup \{(\sigma, a_1) \mid A[\![a_1]\!] \leq A[\![a_2]\!]\}$$

se si intende il valore valutato lo scegliamo così

Si aggiunga a IMP un'espressione aritmetica $\max(a_1, a_2)$ che, valutate a_1 e a_2 ritorna il massimo tra i due valori.

1. Estendere la semantica operazionale

$$\frac{\langle a_1,\sigma\rangle \to n_1 \quad \langle a_2,\sigma\rangle \to n_2}{\langle \max(a_1,a_2),\sigma\rangle \to n} \quad n = \max(n_1, n_2)$$

2. Estendere la semantica denotazionale

$$\mathcal{A}[\![\max(a_1,a_2)]\!]\sigma = \begin{cases} \mathcal{A}[\![a_1]\!]\sigma & \text{if } \mathcal{A}[\![a_1]\!]\sigma \geq \mathcal{A}[\![a_2]\!]\sigma \\ \mathcal{A}[\![a_2]\!]\sigma & \text{else} \end{cases}$$

$$A[\![\max(a_1,a_2)]\!] =$$

$$= \{(\sigma, n_1) \mid A[\![a_1]\!]_\sigma > A[\![a_2]\!]_\sigma,\ A[\![a_1]\!]_\sigma = n_1\} \cup \{(\sigma, n_2) \mid A[\![a_1]\!] \leq A[\![a_2]\!],\ A[\![a_2]\!]_\sigma = n_2\}$$

Dimostrazione equivalenza semantica operazionale e denotazionale

- Supponiamo che $\langle \max(a_1,a_2),\sigma\rangle \to a_1$

  ma allora ho ottenuto n_1 perché $\langle a_1 > a_2,\sigma\rangle \to \text{true}$ ma ciò è equivalente a $B[\![a_1 > a_2]\!]_\sigma = \text{true}$ come noto e sapendo ciò nella semantica denotazionale viene ritornata la coppia (σ, a_1) perciò vale l'equivalenza

- Supponiamo che $\langle \max(a_1,a_2),\sigma\rangle \to a_2$

  ma allora ho ottenuto n_1 perché $\langle a_1 > a_2,\sigma\rangle \to \text{false}$ ma ciò è equivalente a $B[\![a_1 > a_2]\!]_\sigma = \text{false}$ come noto e sapendo ciò nella semantica denotazionale viene ritornata la coppia (σ, a_2) perciò vale l'equivalenza

Semantica operazionale

$$\frac{\langle c, \sigma \rangle \to \sigma'}{\langle \text{save } x \text{ in } c, \sigma \rangle \to \sigma'[\sigma(x)/x]}$$

Semantica denotazionale

$$\mathcal{C}[\![\text{save } x \text{ in } c]\!] = \{ (\sigma, \sigma'[\sigma(x)/x]) \mid \mathcal{C}[\![c]\!]_\sigma = \sigma' \}$$

IntDiv2(a)

Semantica operazionale

$$\frac{\langle a, \sigma \rangle \to n \qquad \dfrac{\dfrac{\langle n, \sigma \rangle \to n \quad \langle 2, \sigma \rangle \to 2}{\langle n \bmod 2, \sigma \rangle \to m} \quad \langle 0, \sigma \rangle \to 0}{\langle n \bmod 2 = 0, \sigma \rangle \to \text{true}}}{\langle \text{IntDiv2}(a), \sigma \rangle \to \frac{n}{2}}$$

$$\frac{\langle a, \sigma \rangle \to n \qquad \dfrac{\dfrac{\langle n, \sigma \rangle \to n \quad \langle 2, \sigma \rangle \to 2}{\langle n \bmod 2, \sigma \rangle \to m} \quad \langle 0, \sigma \rangle \to 0}{\langle n \bmod 2 = 0, \sigma \rangle \to \text{false}}}{\langle \text{IntDiv2}(a), \sigma \rangle \to \frac{n+1}{2}}$$

Semantica denotazionale

$$\mathcal{A}[\![\text{IntDiv}(A)]\!] = \{ (\sigma, \tfrac{n}{2}) \mid \mathcal{A}[\![a]\!]_\sigma = n, \mathcal{B}[\![n \bmod 2 = 0]\!]_\sigma = \text{true} \}$$
$$\cup \{ (\sigma, \tfrac{n+1}{2}) \mid \mathcal{A}[\![a]\!]_\sigma = n, \mathcal{B}[\![n \bmod 2 = 0]\!]_\sigma = \text{false} \}$$

Dimostrazione di equivalenza tra semantica operazionale e denotazionale

op $\rightarrow$ den

Supponiamo che $\langle IntDiv2(a), \sigma \rangle \rightarrow \frac{n+1}{2}$

ma so che $\langle a, \sigma \rangle \rightarrow n \xleftrightarrow{IH} A[\![a]\!]_\sigma = n$

so che

$\langle n \bmod 2 = 0, \sigma \rangle \rightarrow false \xleftrightarrow{IH} B[\![n \bmod 2 = 0]\!]_\sigma = false$

$\langle n, \sigma \rangle \rightarrow n \xleftrightarrow{IH} A[\![n]\!]_\sigma = n$

$\downarrow$

$\forall n$

Da ciò so che $A[\![IntDiv2(a)]\!]_\sigma = \frac{n+1}{2}$, basta osservare gli insiemi della semantica denotazionale e infatti quando $B[\![n \bmod 2 = 0]\!]_\sigma = true$ e $A[\![a]\!]_\sigma = n$ ho la coppia $(\sigma, \frac{n+1}{2})$

Caso true

Supponiamo che $\langle IntDiv2(a), \sigma \rangle \rightarrow \frac{n}{2}$

ma so che $\langle a, \sigma \rangle \rightarrow n \xleftrightarrow{IH} A[\![a]\!]_\sigma = n$

so che

$\langle n \bmod 2 = 0, \sigma \rangle \rightarrow true \xleftrightarrow{IH} B[\![n \bmod 2 = 0]\!]_\sigma = true$

$\langle n, \sigma \rangle \rightarrow n \xleftrightarrow{IH} A[\![n]\!]_\sigma = n$

$\downarrow$

$\forall n$

Da ciò so che $A[\![IntDiv2(a)]\!]_\sigma = \frac{n}{2}$, basta osservare gli insiemi della semantica denotazionale e infatti quando $B[\![n \bmod 2 = 0]\!]_\sigma = true$ e $A[\![a]\!]_\sigma = n$ ho la coppia $(\sigma, \frac{n}{2})$

den $\rightarrow$ op

Supponiamo che $A[\![IntDiv2(a), \sigma]\!]_\sigma = \frac{n+1}{2}$

Ma allora so che $A[\![a]\!]_\sigma = n \xleftrightarrow{IH} \langle a, \sigma \rangle \rightarrow n$

$B[\![n \bmod 2 = 0]\!]_\sigma = false \xleftrightarrow{IH} \langle n \bmod 2 = 0, \sigma \rangle \rightarrow false$

ma ciò ci permette di applicare

$$\frac{\langle a, \sigma \rangle \rightarrow n \quad \dfrac{\langle n, \sigma \rangle \rightarrow n \quad \langle 2, \sigma \rangle \rightarrow 2}{\langle n \bmod 2, \sigma \rangle \rightarrow m} \quad \langle 0, \sigma \rangle \rightarrow 0}{\langle n \bmod 2 = 0, \sigma \rangle \rightarrow false}$$

e derivo lo stesso stato

$\langle IntDiv2(a), \sigma \rangle \rightarrow \frac{n+1}{2}$

den $\rightarrow$ op

Supponiamo che $A[\![\text{IntDiv2}(a), \sigma]\!]_\sigma = \frac{n}{2}$

Ma allora so che $A[\![a]\!]_\sigma = n \xleftrightarrow{\text{IH}} \langle a, \sigma \rangle \rightarrow n$

$B[\![n \bmod 2 = 0]\!]_\sigma = \text{true} \xleftrightarrow{\text{IH}} \langle n \bmod 2 = 0, \sigma \rangle \rightarrow \text{true}$

ma ciò ci permette di applicare

$$\frac{\langle a, \sigma \rangle \rightarrow n \qquad \dfrac{\dfrac{\langle n, \sigma \rangle \rightarrow n \quad \langle 2, \sigma \rangle \rightarrow 2}{\langle n \bmod 2, \sigma \rangle \rightarrow m} \quad \langle 0, \sigma \rangle \rightarrow 0}{\langle n \bmod 2 = 0, \sigma \rangle \rightarrow \text{true}}}{\langle \text{IntDiv2}(a), \sigma \rangle \rightarrow \frac{n}{2}}$$

e derivo lo stesso stato

$\{n > 1\}$

$s = n$

$x = n - 1$

while $x > 0$ do:

$\quad s = s * x$

$\quad x = x - 1$

$\{s = \prod_{x=1}^{n} x\}$

Troviamo l'invariante del ciclo eseguendo il programma e alcune iterazioni del ciclo

$s = \prod_{i=n-1}^{n} i \qquad x = n-2$

$s = \prod_{i=n-2}^{n} i \qquad x = n-3$

$s = \prod_{i=n-3}^{n} i \qquad x = n-4$

$\vdots \qquad \vdots$

$I \equiv s = \prod_{i=x+1}^{n} i \quad \wedge \quad x \geq 0$

Per dimostrare che I è invariante devo poter applicare la regola del while e per farlo devo poter applicare la consequence rule, devo dunque dimostrare che $I \wedge b \rightarrow \hat{I}$.
$\hat{I}$ non è altro che la wp e per calcolarla devo applicare ad I le interpretazioni dei comandi nel corpo del while

$$\hat{I} = I[x-1/x][s*x/s] = s = \prod_{i=x}^{n} i \quad \wedge \quad x-1 \geq 0$$

$$= s * x = \prod_{i=x}^{n} i \quad \wedge \quad x-1 \geq 0$$

$$I \wedge b \equiv s = \prod_{i=x+1}^{n} i \quad \wedge \quad x \geq 0 \quad \wedge \quad x > 0$$

$$\equiv s * x = \prod_{i=x}^{n} i \quad \wedge \quad \underbrace{x > 0}_{x-1 \geq 0 \checkmark} \equiv \hat{I}$$

aggiungo un elemento alla produttoria

Ho che $I \wedge b \rightarrow \hat{I}$ di conseguenza so che posso applicare la consequence rule posso allora applicare la regola del while deducendo:

$$\frac{\dfrac{\{\hat{I}\} \subset \{I\} \qquad I \wedge b \rightarrow \hat{I} \checkmark}{\{I \wedge b\} \subset \{I\}} \text{ consequence rule}}{\{I\} \text{ while } b \text{ do } c \ \{I \wedge \neg b\}}$$

Avendo dedotto che vale la regola del while so che I deve valere per le precondizioni del programma e $I \wedge \neg b$ per le postcondizioni

$$I \wedge \neg b \equiv s = \prod_{i=x+1}^{n} i \quad \wedge \quad x \geq 0 \wedge x \leq 0$$

$$= x = 0 \quad \wedge \quad s = \prod_{i=1}^{n} i \quad \equiv \text{post condizioni} \checkmark$$

$\{n > 1\}$

$s = n$

$x = n - 1$

$\longrightarrow I \quad s = \prod_{i=x+1}^{n} i \quad \wedge \quad x \geq 0$

$n = \prod_{i=n}^{n} i \checkmark \wedge \quad n-1 \geq 0 \checkmark$ perché $n > 1$

Semantica operazionale

$$\frac{\langle a_1, \sigma \rangle \to n_1 \qquad \langle a_2, \sigma \rangle \to n_2 \qquad \langle n_1 > n_2, \sigma \rangle \to true}{\langle max(a_1, a_2), \sigma \rangle \to n_1}$$

$$\frac{\langle a_1, \sigma \rangle \to n_1 \qquad \langle a_2, \sigma \rangle \to n_2 \qquad \langle n_1 > n_2, \sigma \rangle \to false}{\langle max(a_1, a_2), \sigma \rangle \to false}$$

Semantica denotazionale

$$A[\![max(a_1, a_2)]\!] = \{(\sigma, n_1) \mid A[\![a_1]\!]_\sigma = n_1,\ A[\![a_2]\!]_\sigma = n_2 \wedge B[\![n_1 > n_2]\!] = true\}$$
$$\cup \{(\sigma, n_1) \mid A[\![a_1]\!]_\sigma = n_1,\ A[\![a_2]\!]_\sigma = n_2 \wedge B[\![n_1 > n_2]\!] = true\}$$

Dimostrazione equivalenza tra semantiche

operazionale $\longrightarrow$ denotazionale

Caso 1

- Supponiamo $\langle max(a_1, a_2), \sigma \rangle \to \underline{n_1}$

ma allora so che $\langle a_1, \sigma \rangle \to n_1 \overset{\text{per IH}}{\longleftrightarrow} A[\![a_1]\!]_\sigma = n_1$

$\langle a_2, \sigma \rangle \to n_2 \overset{\text{per IH}}{\longleftrightarrow} A[\![a_2]\!]_\sigma = n_2$

$\langle n_1 > n_2, \sigma \rangle \to true \overset{\text{per IH}}{\longleftrightarrow} B[\![n_1 > n_2]\!] = true$

da ciò posso derivare che guardando gli insiemi definiti dalla semantica denotazionale $\{(\sigma, n_1) \mid A[\![a_1]\!]_\sigma = n_1,\ A[\![a_2]\!]_\sigma = n_2 \wedge B[\![n_1 > n_2]\!] = true\}$ questo è il corrispondente composto da coppie $(\sigma, \underline{n_1})$, giungo dunque alla stessa conclusione con entrambe le semantiche e perciò ho la loro equivalenza in questo caso ✓

Caso 2

- Supponiamo $\langle max(a_1, a_2), \sigma \rangle \to \underline{n_2}$

ma allora so che $\langle a_1, \sigma \rangle \to n_1 \overset{\text{per IH}}{\longleftrightarrow} A[\![a_1]\!]_\sigma = n_1$

$\langle a_2, \sigma \rangle \to n_2 \overset{\text{per IH}}{\longleftrightarrow} A[\![a_2]\!]_\sigma = n_2$

$\langle n_1 > n_2, \sigma \rangle \to false \overset{\text{per IH}}{\longleftrightarrow} B[\![n_1 > n_2]\!] = false$

da ciò posso derivare che guardando gli insiemi definiti dalla semantica denotazionale $\{(\sigma, n_2) \mid A[\![a_1]\!]_\sigma = n_1,\ A[\![a_2]\!]_\sigma = n_2 \wedge B[\![n_1 > n_2]\!] = false\}$ questo è il corrispondente composto da coppie $(\sigma, \underline{n_2})$, giungo dunque alla stessa conclusione con entrambe

denotazionale $\longrightarrow$ operazionale

- Caso 1

Supponiamo che $A[\![\max(a_1, a_2)]\!]_\sigma = n_1$

allora so che $A[\![a_1]\!]_\sigma = n_1 \overset{\text{per IH}}{\Longleftrightarrow} \langle a_1, \sigma \rangle \to n_1$

$A[\![a_2]\!]_\sigma = n_2 \overset{\text{per IH}}{\Longleftrightarrow} \langle a_2, \sigma \rangle \to n_2$

$B[\![n_1 > n_2]\!] = \text{true} \overset{\text{per IH}}{\Longleftrightarrow} \langle n_1 > n_2, \sigma \rangle \to \text{true}$

ma allora posso applicare la regola $\dfrac{\langle a_1, \sigma \rangle \to n_1 \quad \langle a_2, \sigma \rangle \to n_2 \quad \langle n_1 > n_2, \sigma \rangle \to \text{true}}{\langle \max(a_1, a_2), \sigma \rangle \to n_1}$

ottengo dunque n_1 giungo dunque alla stessa conclusione con entrambe le semantiche

- Caso 2

Supponiamo che $A[\![\max(a_1, a_2)]\!]_\sigma = n_2$

allora so che $A[\![a_1]\!]_\sigma = n_1 \overset{\text{per IH}}{\Longleftrightarrow} \langle a_1, \sigma \rangle \to n_1$

$A[\![a_2]\!]_\sigma = n_2 \overset{\text{per IH}}{\Longleftrightarrow} \langle a_2, \sigma \rangle \to n_2$

$B[\![n_1 > n_2]\!] = \text{false} \overset{\text{per IH}}{\Longleftrightarrow} \langle n_1 > n_2, \sigma \rangle \to \text{false}$

ma allora posso applicare la regola $\dfrac{\langle a_1, \sigma \rangle \to n_1 \quad \langle a_2, \sigma \rangle \to n_2 \quad \langle n_1 > n_2, \sigma \rangle \to \text{false}}{\langle \max(a_1, a_2), \sigma \rangle \to n_2}$

ottengo dunque n_2 giungo dunque alla stessa conclusione con entrambe le semantiche

$\{ n = N > 0 \}$

```
a = 0
b = 1
while n > 0 do
    b = a+b
    a = b-a
    n = n-1
```

$\hat{I}$

$\{ a = Fib(N) \}$

Troviamo l'invariante di eseguendo il programma e alcune iterazioni del ciclo

$b = 1$ (fib 2)	$a = 1$ (fib 1)	$n = N-1$
$b = 2$ (fib 3)	$a = 1$ (fib 2)	$n = N-2$
$b = 3$ (fib 4)	$a = 2$ (fib 3)	$n = N-3$
$b = 5$ (fib 5)	$a = 3$ (fib 4)	$n = N-4$

$$I \equiv b = fib(N-n+1) \wedge a = fib(N-n) \wedge n \geq 0$$

Dimostriamo dunque che I è invariante, per farlo dobbiamo applicare la regola del while e per farlo dobbiamo applicare la consegenza rule, per fare ciò dobbiamo mostrare che $I \wedge b \rightarrow \hat{I}$

Per calcolare $\hat{I}$ calcoliamo la wp applicando ad I le interpretazioni dei comandi nel ciclo while

$$I[^{n-1}/_n][^{b-a}/_a][^{a+b}/_b] \equiv b = fib(N-n+1) \wedge a = fib(N-n) \wedge n \geq 0$$
$$\equiv b = fib(N-n+2) \wedge a = fib(N-n+1) \wedge n-1 \geq 0$$
$$\equiv b = fib(N-n+2) \wedge b-a = fib(N-n+1) \wedge n-1 \geq 0$$
$$\equiv b+a = fib(N-n+2) \wedge b = fib(N-n+1) \wedge n-1 \geq 0$$

$$I \wedge b \equiv b = fib(N-n+1) \wedge a = fib(N-n) \wedge n \geq 0 \wedge n > 0$$
$$\equiv b = fib(N-n+1) \wedge a = fib(N-n) \wedge n > 0$$
$$\equiv b = fib(N-n+1) \wedge a+b = fib(N-n) + fib(N-n+1) \wedge n > 0$$
$$\equiv b = fib(N-n+1) \wedge a+b = fib(N-n+2) \wedge n-1 \geq 0 \equiv \hat{I}$$

Avendo mostrato che $I \wedge b \equiv \hat{I}$ ho mostrato che $I \wedge b \rightarrow \hat{I}$ posso allora applicare la consequence rule e quindi applicare la regola del while dalla derivazione:

$$\dfrac{\dfrac{\{\hat{I}\}\, c\, \{I\} \qquad I \wedge b \rightarrow \hat{I}}{\{I \wedge b\}\, c\, \{I\}}\ \text{consequence rule}}{\{I\}\ \text{while } b \text{ do } c\ \{I \wedge \neg b\}}\ \text{while}$$

So dunque che I è invariante devo però verificare che I sia concorde alle precondizioni e $I \wedge \neg b$ sia concorde alle postcondizioni

$$I \wedge \neg b \equiv b = fib(N-n+1) \wedge a = fib(N-n) \wedge n \geq 0 \wedge n \leq 0$$
$$\equiv n = 0 \wedge b = fib(N+1) \wedge a = fib(N) \quad \checkmark$$

concorde alle post condizioni

$\{ n = N > 0 \}$

$a = 0$

$b = 1$

$\rightarrow \quad I \equiv b = fib(N - n + 1) \wedge a = fib(N - n) \wedge n \geq 0$

$1 = fib(1) \quad \wedge \quad 0 = fib(0) \quad \wedge \quad n \geq 0 \rightarrow n = N > 0$

$\{ n \geq 0 \}$

$s = 1$

$x = 1$

while $x \leq n$ do:

if $f(x) \geq 0$ then $s = s * f(x)$

else $s = s * (-f(x))$

$x = x+1$

$\{ s = \prod_{i=1}^{n} |f(i)| \}$

Troviamo l'invariante eseguendo alcune iterazioni del ciclo

$s = \prod_{i=1}^{1} |f(i)|$ $\quad x = 2$

$s = \prod_{i=1}^{2} |f(i)|$ $\quad x = 3$

$s = \prod_{i=1}^{3} |f(i)|$ $\quad x = 4$

$\vdots \qquad \vdots$

$$I \equiv s = \prod_{i=1}^{x-1} |f(i)| \wedge x \leq n+1$$

Dobbiamo ora mostrare che I è invariante, per farlo dobbiamo applicare la regola del while e per farlo dobbiamo poter applicare la consequence rule, per poterla applicare dobbiamo mostrare che $I \wedge b_w \wedge b_{if} \Rightarrow A_T$ e che $I \wedge b_w \wedge \neg b_{if} \Rightarrow A_F$. A_T e A_F le calcoliamo calcolando la wp ovvero applicando ad I l'interpretazione del comando nel ciclo while, calcolando $\hat{I}$ e applicando ad essa l'interpretazione di ciascun comando nel corrispettivo ramo dell'if trovando A_i e A_f

$$\hat{I} = I[^{x+1}/_x] = s = \prod_{i=1}^{x} |f(i)| \wedge x \cancel{+1} \leq n \cancel{+1}$$

$$A_T = \hat{I}[^{s*f(x)}/_s] = s*f(x) = \prod_{i=1}^{x} |f(i)| \wedge x \leq n$$

$$A_F = \hat{I}[^{s*(-f(x))}/_s] = s*(-f(x)) = \prod_{i=1}^{x} |f(i)| \wedge x \leq n$$

$$I \wedge b_w \wedge b_{if} \equiv s = \prod_{i=1}^{x-1} |f(i)| \wedge x \leq n+1 \wedge x \leq n \wedge f(x) \geq 0$$

$$\equiv s = \prod_{i=1}^{x-1} |f(i)| \wedge x \leq n \wedge f(x) \geq 0$$

$$\equiv s * f(x) = \prod_{i=1}^{x} |f(i)| \wedge x \leq n \wedge f(x) \geq 0 \equiv A_T$$

moltiplichiamo s per il valore della funzione facendo un passo in più perché $f(x) \geq 0$

$$I \wedge b_w \wedge \neg b_{if} \equiv s = \prod_{i=1}^{x-1} |f(i)| \wedge x \leq n+1 \wedge x \leq n \wedge f(x) < 0$$

$$\equiv s = \prod_{i=1}^{x-1} |f(i)| \wedge x \leq n \wedge f(x) < 0$$

$$\equiv s*(-f(x)) = \prod_{i=1}^{x} |f(i)| \wedge x \leq n \wedge f(x) < 0 \equiv A_F$$

moltiplichiamo s per il valore della funzione cambiando segno facendo un passo in più perché $f(x) \geq 0$

Ho mostrato che $I \wedge b_w \wedge b_{if} \to A_T$ tramite equivalenza e che $I \wedge b_w \wedge \neg b_{if} \to A_F$ sempre tramite equivalenza, posso dunque applicare la consequence rule e applicare la regola del while deducendo:

$$
\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\{A_T\}\, c_1\, \{\hat{I}\} \quad I \wedge b_w \wedge b_{if} \to A_T}{\{I \wedge b_w \wedge b_{if}\}\, c_1\, \{\hat{I}\}}\ \text{consequence rule} \qquad \dfrac{\{A_F\}\, c_2\, \{\hat{I}\} \quad I \wedge b_w \wedge \neg b_{if} \to A_F}{\{I \wedge b_w \wedge \neg b_{if}\}\, c_2\, \{\hat{I}\}}\ \text{consequence rule}}{\{I \wedge b_w\}\ \text{if } b_{if} \text{ then } c_1 \text{ else } c_2\ \{\hat{I}\}} \qquad \{\hat{I}\}\, c_3\, \{I\}}{\{I \wedge b_w\}\ \text{corpo}\ \{I\}}}{\{I\}\ \text{while } b_w \text{ do corpo}\ \{I \wedge \neg b_w\}}
$$

Avendo applicato la regola del while so che I è invariante, devo verificare però che I sia concorde alle precondizioni e che $I \wedge \neg b_w$ sia concorde alle postcondizioni

$$I \wedge \neg b_w \equiv s = \prod_{i=1}^{x-1} |f(i)| \ \wedge \ x \leq n+1 \ \wedge \ x > n$$

$$\equiv x = n+1 \ \wedge \ s = \prod_{i=1}^{x} |f(i)|$$

$\{n \geq 0\}$

$s = 1$ $\to I$

$x = 1$

$$s = \prod_{i=1}^{x-1} |f(i)| \ \wedge \ x \leq n+1$$

$$1 = \prod_{i=1}^{0} |f(i)| \ \wedge \ 1 \leq n+1$$

$$1 = 1 \checkmark \ \wedge \ 1 \leq n+1 \checkmark \to n \geq 0$$

$\{ n = N \geq 0 \}$

```
a = 0 ;
b = 1 ;
while n > 0 do
        b = a+b
        a = b-a
        n := n-1
```

$\{ a = Fib(N) \}$

Trovo l'invariante del ciclo eseguendo il programma e alcune iterazioni di ciclo

$b = 1$ fib 2	$a = 1$ fib 1	$n = N-1$
$b = 2$ fib 3	$a = 1$ fib 2	$n = N-2$
$b = 3$ fib 4	$a = 2$ fib 3	$n = N-3$
$b = 5$ fib 5	$a = 3$ fib 4	$n = N-4$
$b = 8$ fib 6	$a = 5$ fib 5	$n = N-5$

$$I \equiv a = fib(N-n) \wedge b = fib(N-n+1) \wedge n \geq 0$$

Dimostriamo che I è invariante e per farlo dobbiamo applicare la regola del while e per farlo dobbiamo applicare la consequence rule che è applicabile se e solo se $I \wedge b \rightarrow \hat{I}$.

Dobbiamo calcolare $\hat{I}$ e per farlo calcoliamo la wp, applicando ad I le interpretazioni dei comandi nel corpo del while

$$I[^{n-1}/_n][^{b-a}/_a][^{b+a}/_b] = a = fib(N-n) \wedge b = fib(N-n+1) \wedge n \geq 0$$

$$= a = fib(N-n-1) \wedge b = fib(N-n+2) \wedge n-1 \geq 0$$

$$= b-a = fib(N-n-1) \wedge b = fib(N-n+2) \wedge n-1 \geq 0$$

$$= b = fib(N-n-1) \wedge b+a = fib(N-n+2) \wedge n-1 \geq 0$$

$$I \wedge b \equiv a = fib(N-n) \wedge b = fib(N-n+1) \wedge n \geq 0 \wedge n > 0$$

$$= a = fib(N-n) \wedge b = fib(N-n+1) \wedge n > 0$$

$$= b+a = fib(N-n) + fib(N-n+1) \wedge b = fib(N-n+1) \wedge n-1 > 0$$

$$= b+a = fib(N-n+2) \wedge = b = fib(N-n+1) \wedge n-1 > 0 \equiv \hat{I}$$

Avendo mostrato che $I \wedge b \rightarrow \hat{I}$ attraverso l'equivalenza so che posso applicare la regola del while poiché vale la consequence rule, derivo infatti:

$$\frac{\dfrac{\{\hat{I}\} \subset \{I\} \quad I \wedge b \rightarrow \hat{I}}{\{I \wedge b\} \subset \{I\}}\ \text{consequence rule}}{\{I\}\ \text{while } b \text{ do } c\ \{I \wedge \neg b\}}$$

Abbiamo dimostrato che I è invariante, dobbiamo ora mostrare, per la regola del while, che I è concorde alle precondizioni e che $I \wedge \neg b$ è concorde alle post condizioni

$$I \wedge \neg b \equiv a = fib(N-n) \wedge b = fib(N-n+1) \wedge n \geq 0 \wedge n \leq 0$$

$$\equiv a = fib(N) \wedge b = fib(N+1) \wedge n = 0 \ \checkmark$$

concorde alle postcondizioni

$$\equiv a = fib(N)$$

$\{ n = N \geq 0 \}$

$a = 0;$

$b = 1;$

$\longrightarrow$

$a = fib(N - n) \wedge b = fib(N - n + 1) \wedge n \geq 0$

$0 = fib(N - N) \wedge 1 = fib(N - N + 1) \wedge N \geq 0$ ✓

$0 = 0 \wedge 1 = 1 \wedge N \geq 0$

OK tutto concorde alle precondizioni

$$\overset{P_1}{\text{if } x \neq 0 \text{ the } x := 1 \text{ else } x := x+1} \equiv \overset{P_2}{x := 1}$$

$P_1 \sim P_2$ iff $\forall \sigma, \sigma' \in \Sigma \quad \langle P_1, \sigma \rangle \to \sigma' \iff \langle P_2, \sigma \rangle \to \sigma'$

Equivalenza con semantica operazionale

$$\dfrac{\dfrac{\langle x, \sigma \rangle \to \sigma(x) \quad \langle 0, \sigma \rangle \to 0}{\langle x \neq 0, \sigma \rangle \to \text{true}} \quad \dfrac{\langle 1, \sigma \rangle \to 1}{\langle x := 1, \sigma \rangle \to \sigma[1/x]}}{\langle \text{if } x \neq 0 \text{ the } x := 1 \text{ else } x := x+1, \sigma \rangle \to \sigma[1/x]} \iff \dfrac{\langle 1, \sigma \rangle \to \sigma}{\langle x := 1, \sigma \rangle \to \sigma[1/x]}$$

L'equivalenza vale, giungo allo stesso stato finale

$$\dfrac{\dfrac{\overset{\sigma(x) = 0 \text{ per IPOTESI}}{\langle x, \sigma \rangle \to \sigma(x)} \quad \langle 0, \sigma \rangle \to 0}{\langle x \neq 0, \sigma \rangle \to \text{false}} \quad \dfrac{\dfrac{\langle x, \sigma \rangle \to \sigma(x) \quad \langle 1, \sigma \rangle \to 1}{\langle x+1, \sigma \rangle \to \sigma(x)+1}}{\langle x := x+1, \sigma \rangle \to \sigma[\sigma(x)+1/x]}}{\langle \text{if } x \neq 0 \text{ the } x := 1 \text{ else } x := x+1, \sigma \rangle \to \underbrace{\sigma[\sigma(x)+1/x]}} \qquad \dfrac{\langle 1, \sigma \rangle \to \sigma}{\langle x := 1, \sigma \rangle \to \sigma[1/x]}$$

Per ipotesi $\sigma(x) = 0$ perche $x \neq 0$ ritorna falso quindi ho $\sigma[0+1/x] = \sigma[1/x]$ giungo dunque allo stesso stato

Equivalenza della semantica denotazionale

$$\begin{aligned}
\mathcal{C}[\![\text{if } x \neq 0 \text{ the } x := 1 \text{ else } x := x+1]\!] &= \{(\sigma, \sigma[1/x]) \mid \mathcal{B}[\![x \neq 0]\!] = \text{true}, \mathcal{C}[\![x := 1]\!]_\sigma = \sigma[1/x]\} \\
&\quad \cup \{(\sigma, \sigma[x+1/x]) \mid \mathcal{B}[\![x \neq 0]\!] = \text{false}, \mathcal{C}[\![x := x+1]\!]_\sigma = \sigma[x+1/x]\} \\
&= \{(\sigma, \sigma[1/x]) \mid \mathcal{A}[\![x]\!]_\sigma \neq 0, \mathcal{C}[\![x := 1]\!]_\sigma = \sigma[1/x]\} \\
&\quad \cup \{(\sigma, \sigma[x+1/x]) \mid \mathcal{A}[\![x]\!]_\sigma = 0, \mathcal{C}[\![x := x+1]\!]_\sigma = \sigma[\sigma(x)+1/x]\} \\
&= \{(\sigma, \sigma[1/x]) \mid \mathcal{A}[\![x]\!]_\sigma \neq 0, \mathcal{C}[\![x := 1]\!]_\sigma = \sigma[1/x]\} \\
&\quad \cup \{(\sigma, \sigma[0+1/x]) \mid \mathcal{A}[\![x]\!]_\sigma = 0, \mathcal{C}[\![x := x+1]\!]_\sigma = \sigma[0+1/x]\} \\
&= \{(\sigma, \sigma[1/x]) \mid \overbrace{\mathcal{A}[\![x]\!]_\sigma \neq 0 \vee \mathcal{A}[\![x]\!] = 0}^{\text{tutti i casi}}, \mathcal{C}[\![x := 1]\!] = \sigma[1/x]\} \\
&= \mathcal{C}[\![x := 1]\!]
\end{aligned}$$

Dimostriamo l'equivalenza tra due programmi usando semantica operazionale e denotazionale

$$\underbrace{\text{if } b_1 \text{ then } (\text{if } b_2 \text{ then } c_1 \text{ else } c_2) \text{ else } c_2}_{P_1} \equiv \underbrace{\text{if } b_1 \wedge b_2 \text{ then } c_1 \text{ else } c_2}_{P_2}$$

$$P_1 \sim P_2 \quad \text{iff} \quad \forall \sigma, \sigma' \in \Sigma \quad \langle P_1, \sigma \rangle \to \sigma' \iff \langle P_2, \sigma \rangle \to \sigma'$$

Don't care → La condizione sarebbe comunque falsa

$$\frac{\langle b_1, \sigma \rangle \to \text{false} \qquad \langle c_2, \sigma \rangle \to \sigma'}{\langle \text{if } b_1 \text{ then } (\text{if } b_2 \text{ then } c_1 \text{ else } c_2) \text{ else } c_2, \sigma \rangle \to \sigma'} \iff \frac{\dfrac{\langle b_1, \sigma \rangle \to \text{false} \qquad -}{\langle b_1 \wedge b_2, \sigma \rangle \to \text{false}} \qquad \langle c_2, \sigma \rangle \to \sigma'}{\langle \text{if } b_1 \wedge b_2 \text{ then } c_1 \text{ else } c_2, \sigma \rangle \to \sigma'}$$

$$\frac{\langle b_1, \sigma \rangle \to \text{true} \qquad \dfrac{\langle b_2, \sigma \rangle \to \text{false} \qquad \langle c_2, \sigma \rangle \to \sigma'}{\langle \text{if } b_2 \text{ then } c_1 \text{ else } c_2, \sigma \rangle \to \sigma'}}{\langle \text{if } b_1 \text{ then } (\text{if } b_2 \text{ then } c_1 \text{ else } c_2) \text{ else } c_2, \sigma \rangle \to \sigma'} \iff \frac{\dfrac{\langle b_1, \sigma \rangle \to \text{true} \qquad \langle b_2, \sigma \rangle \to \text{false}}{\langle b_1 \wedge b_2, \sigma \rangle \to \text{false}} \qquad \langle c_2, \sigma \rangle \to \sigma'}{\langle \text{if } b_1 \wedge b_2 \text{ then } c_1 \text{ else } c_2, \sigma \rangle \to \sigma'}$$

$$\frac{\langle b_1, \sigma \rangle \to \text{true} \qquad \dfrac{\langle b_2, \sigma \rangle \to \text{true} \qquad \langle c_1, \sigma \rangle \to \sigma''}{\langle \text{if } b_2 \text{ then } c_1 \text{ else } c_2, \sigma \rangle \to \sigma''}}{\langle \text{if } b_1 \text{ then } (\text{if } b_2 \text{ then } c_1 \text{ else } c_2) \text{ else } c_2, \sigma \rangle \to \sigma''} \iff \frac{\dfrac{\langle b_1, \sigma \rangle \to \text{true} \qquad \langle b_2, \sigma \rangle \to \text{true}}{\langle b_1 \wedge b_2, \sigma \rangle \to \text{false}} \qquad \langle c_1, \sigma \rangle \to \sigma''}{\langle \text{if } b_1 \wedge b_2 \text{ then } c_1 \text{ else } c_2, \sigma \rangle \to \sigma''}$$

Semantica denotazionale

$$\begin{aligned}
\mathcal{C}[\![\text{if } b_1 \text{ then } (\text{if } b_2 \text{ then } c_1 \text{ else } c_2) \text{ else } c_2]\!] &= \{(\sigma, \sigma') \mid \mathcal{B}[\![b_1]\!]_\sigma = \text{false},\ \mathcal{C}[\![c_2]\!]_\sigma = \sigma'\} \\
&\quad \cup \{(\sigma, \sigma') \mid \mathcal{B}[\![b_1]\!]_\sigma = \text{true},\ \mathcal{B}[\![b_2]\!]_\sigma = \text{false},\ \mathcal{C}[\![c_2]\!]_\sigma = \sigma'\} \\
&\quad \cup \{(\sigma, \sigma'') \mid \mathcal{B}[\![b_1]\!]_\sigma = \text{true},\ \mathcal{B}[\![b_2]\!]_\sigma = \text{true},\ \mathcal{C}[\![c_1]\!]_\sigma = \sigma''\} \\
&= \{(\sigma, \sigma') \mid \mathcal{B}[\![b_1]\!] = \text{false} \vee \mathcal{B}[\![b_2]\!] = \text{false},\ \mathcal{C}[\![c_2]\!]_\sigma = \sigma'\} \\
&\quad \cup \{(\sigma, \sigma'') \mid \mathcal{B}[\![b_1]\!] = \text{true} \wedge \mathcal{B}[\![b_2]\!] = \text{true},\ \mathcal{C}[\![c_1]\!]_\sigma = \sigma''\} \\
&= \mathcal{C}[\![\text{if } b_1 \wedge b_2 \text{ then } c_1 \text{ else } c_2]\!]
\end{aligned}$$

$max(a_1, a_2)$ che valutare a_1 e a_2 ritorna il massimo tra i due valori

semantica operazionale

$$\frac{\langle a_1, \sigma\rangle \to n_1 \quad \langle a_2, \sigma\rangle \to n_2 \quad \langle n_1 > n_2, \sigma\rangle \to true}{\langle max(a_1, a_2), \sigma\rangle \to n_1}$$

$$\frac{\langle a_1, \sigma\rangle \to n_1 \quad \langle a_2, \sigma\rangle \to n_2 \quad \langle n_1 > n_2, \sigma\rangle \to false}{\langle max(a_1, a_2), \sigma\rangle \to n_2}$$

semantica denotazionale

$$A[\![max(a_1, a_2)]\!] = \{(\sigma, n_1) \mid A[\![a_1]\!]_\sigma = n_1,\ A[\![a_2]\!]_\sigma = n_2,\ B[\![n_1 > n_2]\!]_\sigma = true\}$$
$$\cup \{(\sigma, n_2) \mid A[\![a_1]\!]_\sigma = n_1,\ A[\![a_2]\!]_\sigma = n_2,\ B[\![n_1 > n_2]\!]_\sigma = false\}$$

Dimostrazione equivalenza tra semantiche

operazionale $\to$ denotazionale

- Supponiamo che $\langle max(a_1, a_2), \sigma\rangle \to n_1$

  ma allora so che $\langle a_1, \sigma\rangle \to n_1 \overset{\text{per IH}}{\longleftrightarrow} A[\![a_1]\!]_\sigma = n_1$

  $\langle a_2, \sigma\rangle \to n_2 \overset{\text{per IH}}{\longleftrightarrow} A[\![a_2]\!]_\sigma = n_2$

  $\langle n_1 > n_2, \sigma\rangle \to true \overset{\text{per IH}}{\longleftrightarrow} B[\![n_1 > n_2]\!]_\sigma = true$

  guardando gli insiemi della semantica denotazionale so che la rispettiva valutazione sarà (σ, n_1) poiché corrispondono all'insieme $\{(\sigma, n_1) \mid A[\![a_1]\!]_\sigma = n_1,\ A[\![a_2]\!]_\sigma = n_2,\ B[\![n_1 > n_2]\!]_\sigma = true\}$

  ottengo dunque n_1 come per la operazionale e le semantiche per questo caso sono equivalenti

- Supponiamo che $\langle max(a_1, a_2), \sigma\rangle \to n_2$

  ma allora so che $\langle a_1, \sigma\rangle \to n_1 \overset{\text{per IH}}{\longleftrightarrow} A[\![a_1]\!]_\sigma = n_1$

  $\langle a_2, \sigma\rangle \to n_2 \overset{\text{per IH}}{\longleftrightarrow} A[\![a_2]\!]_\sigma = n_2$

  $\langle n_1 > n_2, \sigma\rangle \to false \overset{\text{per IH}}{\longleftrightarrow} B[\![n_1 > n_2]\!]_\sigma = false$

  guardando gli insiemi della semantica denotazionale so che la rispettiva valutazione sarà (σ, n_2) poiché corrispondono all'insieme $\{(\sigma, n_2) \mid A[\![a_1]\!]_\sigma = n_1,\ A[\![a_2]\!]_\sigma = n_2,\ B[\![n_1 > n_2]\!]_\sigma = false\}$

  ottengo dunque n_2 come per la operazionale e le semantiche per questo caso sono equivalenti

denotazionale $\longrightarrow$ operazionale

- Supponiamo che $A[\![\max(a_1, a_2)]\!]_\sigma = n_1$

ma so che: $A[\![a_1]\!]_\sigma = n_1 \overset{\text{per IH}}{\Longleftrightarrow} \langle a_1, \sigma \rangle \to n_1$

$A[\![a_2]\!]_\sigma = n_2 \overset{\text{per IH}}{\Longleftrightarrow} \langle a_2, \sigma \rangle \to n_2$

$B[\![n_1 > n_2]\!]_\sigma = \text{true} \overset{\text{per IH}}{\Longleftrightarrow} \langle n_1 > n_2, \sigma \rangle \to \text{true}$

posso allora applicare la regola $\dfrac{\langle a_1, \sigma \rangle \to n_1 \quad \langle a_2, \sigma \rangle \to n_2 \quad \langle n_1 > n_2, \sigma \rangle \to \text{true}}{\langle \max(a_1, a_2), \sigma \rangle \to n_1}$

e derivo la stessa conclusione della denotazionale, ovvero n_1

- Supponiamo che $A[\![\max(a_1, a_2)]\!]_\sigma = n_2$

ma so che: $A[\![a_1]\!]_\sigma = n_1 \overset{\text{per IH}}{\Longleftrightarrow} \langle a_1, \sigma \rangle \to n_1$

$A[\![a_2]\!]_\sigma = n_2 \overset{\text{per IH}}{\Longleftrightarrow} \langle a_2, \sigma \rangle \to n_2$

$B[\![n_1 > n_2]\!]_\sigma = \text{false} \overset{\text{per IH}}{\Longleftrightarrow} \langle n_1 > n_2, \sigma \rangle \to \text{false}$

posso allora applicare la regola $\dfrac{\langle a_1, \sigma \rangle \to n_1 \quad \langle a_2, \sigma \rangle \to n_2 \quad \langle n_1 > n_2, \sigma \rangle \to \text{false}}{\langle \max(a_1, a_2), \sigma \rangle \to n_2}$

e derivo la stessa conclusione della denotazionale, ovvero n_1

Int Div (a) $\quad \frac{a}{2}$ se pari $\quad \frac{a+1}{2}$ se dispari

Semantica operazionale

$$\frac{\langle a,\sigma\rangle \to n \qquad \dfrac{\dfrac{\langle n,\sigma\rangle \to n \quad \langle 2,\sigma\rangle \to 2}{\langle n \bmod 2,\sigma\rangle \to m} \qquad \langle 0,\sigma\rangle \to 0}{\langle n \bmod 2 \neq 0,\sigma\rangle \to false}}{\langle \text{Int Div}(a),\sigma\rangle \to \frac{n}{2}}$$

mod definito nell' insieme di operazioni

$$\frac{\langle a_1,\sigma\rangle \to n_1 \qquad \langle a_2,\sigma\rangle \to n_2}{\langle a_1 \ op \ a_2,\sigma\rangle \to n} \quad n = op(n_1,n_2)$$

$$\frac{\langle a,\sigma\rangle \to n \qquad \dfrac{\dfrac{\langle n,\sigma\rangle \to n \quad \langle 2,\sigma\rangle \to 2}{\langle n \bmod 2,\sigma\rangle \to m} \qquad \langle 0,\sigma\rangle \to 0}{\langle n \bmod 2 \neq 0,\sigma\rangle \to true}}{\langle \text{Int Div}(a),\sigma\rangle \to \frac{n+1}{2}}$$

Semantica denotazionale

$$\begin{aligned}
A[\![\text{IntDiv}(a)]\!] &= \{(\sigma, \tfrac{n}{2}) \mid A[\![a]\!]_\sigma = n,\ B[\![n \bmod 2 \neq 0]\!] = false\} \\
&\quad \cup \{(\sigma, \tfrac{n+1}{2}) \mid A[\![a]\!]_\sigma = n,\ B[\![n \bmod 2 \neq 0]\!] = true\} \\
&= \{(\sigma, \tfrac{n}{2}) \mid A[\![a]\!]_\sigma = n,\ A[\![n \bmod 2]\!]_\sigma = 0\} \\
&\quad \cup \{(\sigma, \tfrac{n+1}{2}) \mid A[\![a]\!]_\sigma = n,\ A[\![n \bmod 2]\!]_\sigma \neq 0\}
\end{aligned}$$

save x in c

semantica operazionale

$$\frac{\langle c, \sigma \rangle \to \sigma'}{\langle \text{save } x \text{ in } c, \sigma \rangle \to \sigma'[\sigma(x)/x]}$$

semantica denotazionale

$$\mathcal{C}[\![\text{save } x \text{ in } c]\!] = \{(\sigma, \sigma'[\sigma(x)/x]) \mid \mathcal{C}[\![c]\!]\sigma = \sigma'\}$$

Equivalenza fra semantica operazionale e denotazionale

- operazionale $\to$ denotazionale
  - Supponiamo che $\langle \text{save } x \text{ in } c, \sigma \rangle \to \sigma'[\sigma(x)/x]$

    ma allora so che $\langle c, \sigma \rangle \to \sigma' \overset{\text{per IH}}{\longleftrightarrow} \mathcal{C}[\![c]\!]\sigma = \sigma'$

    ma da ciò so che per la semantica denotazionale del comando ottengo

    la coppia $(\sigma, \sigma'[\sigma(x)/x])$ lo stato finale è dunque lo stesso e l'implicazione vale

denotazionale $\to$ operazionale

- Supponiamo che $\mathcal{C}[\![\text{save } x \text{ in } c]\!]_\sigma = \sigma'[\sigma(x)/x]$

  ma allora so che $\mathcal{C}[\![c]\!]\sigma = \sigma' \overset{\text{per IH}}{\longleftrightarrow} \langle c, \sigma \rangle \to \sigma'$

  posso dunque applicare la regola della semantica operazionale $\dfrac{\langle c, \sigma \rangle \to \sigma'}{\langle \text{save } x \text{ in } c, \sigma \rangle \to \sigma'[\sigma(x)/x]}$

  e ottengo dunque $\sigma'[\sigma(x)/x]$ ovvero lo stesso stato finale ✓

while b then c_1 and c_2

Semantica operazionale

$$\frac{\langle b,\sigma\rangle \to \text{false}}{\langle \text{while } b \text{ alternate } c_1 \text{ and } c_2, \sigma\rangle \to \sigma}$$

$$\frac{\langle b,\sigma\rangle \to \text{true} \qquad \langle c_1; \text{while } b \text{ alternate } c_2 \text{ and } c_1, \sigma\rangle \to \sigma'}{\langle \text{while } b \text{ alternate } c_1 \text{ and } c_2, \sigma\rangle \to \sigma'}$$

Semantica denotazionale

$$\mathcal{C}[\![\overbrace{\text{while } b \text{ alternate } c_1 \text{ and } c_2}^{\varphi}, \sigma]\!] = \Gamma(\mathcal{C}[\![\varphi]\!])$$

$$= \{(\sigma,\sigma) \mid \mathcal{B}[\![b]\!]_\sigma = \text{false}\}$$
$$\cup \{(\sigma,\sigma') \mid \mathcal{B}[\![b]\!]_\sigma = \text{true},\ \mathcal{C}[\![c_1; \text{while } b \text{ alternate } c_2 \text{ and } c_1]\!]_\sigma = \sigma'\}$$

$$= \{(\sigma,\sigma) \mid \mathcal{B}[\![b]\!]_\sigma = \text{false}\}$$
$$\cup \{(\sigma,\sigma') \mid \mathcal{B}[\![b]\!]_\sigma = \text{true},\ \mathcal{C}[\![c_1]\!]_\sigma = \sigma'',\ \mathcal{C}[\![\text{while } b \text{ alternate } c_2 \text{ and } c_1]\!]_{\sigma''} = \sigma'\}$$

while b_1 do c exit b_2

Semantica Operazionale

① $$\frac{\langle b_1, \sigma \rangle \to \text{false}}{\langle \text{while } b_1 \text{ do } c \text{ exit } b_2, \sigma \rangle \to \sigma}$$

② $$\frac{\langle b_1, \sigma \rangle \to \text{true} \quad \langle c, \sigma \rangle \to \sigma'' \quad \langle b_2, \sigma'' \rangle \to \text{true}}{\langle \text{while } b_1 \text{ do } c \text{ exit } b_2, \sigma \rangle \to \sigma''}$$

③ $$\frac{\langle b_1, \sigma \rangle \to \text{true} \quad \langle c, \sigma \rangle \to \sigma'' \quad \langle b_2, \sigma'' \rangle \to \text{false} \quad \langle \text{while } b_1 \text{ do } c \text{ exit } b_2, \sigma'' \rangle \to \sigma'}{\langle \text{while } b_1 \text{ do } c \text{ exit } b_2, \sigma \rangle \to \sigma'}$$

Semantica Denotazionale

$$\mathcal{C}[\![\text{while } b_1 \text{ do } c \text{ exit } b_2]\!] = \Gamma(\mathcal{C}[\![w]\!])$$

$$= \{(\sigma, \sigma) \mid \mathcal{B}[\![b_1]\!]_\sigma = \text{false}\}$$

$$\cup \{(\sigma, \sigma'') \mid \mathcal{B}[\![b_1]\!]_\sigma = \text{true},\ \mathcal{C}[\![c]\!]_\sigma = \sigma'',\ \mathcal{B}[\![b_2]\!]_{\sigma''} = \text{true}\}$$

$$\cup \{(\sigma, \sigma') \mid \mathcal{B}[\![b_1]\!]_\sigma = \text{true},\ \mathcal{C}[\![c]\!]_\sigma = \sigma'',\ \mathcal{B}[\![b_2]\!]_{\sigma''} = \text{false},$$

$$\mathcal{C}[\![\text{while } b \text{ do } c \text{ exit } b_2]\!]_{\sigma''} = \sigma'\}$$

if b then c

Semantica operazionale

$$\frac{\langle b, \sigma \rangle \to \text{false}}{\langle \text{if } b \text{ then } c, \sigma \rangle \to \sigma}$$

$$\frac{\langle b, \sigma \rangle \to \text{True} \quad \langle c, \sigma \rangle \to \sigma'}{\langle \text{if } b \text{ then } c, \sigma \rangle \to \sigma'}$$

Semantica denotazionale

$$\mathcal{C}[\![\text{if } b \text{ then } c]\!] = \{(\sigma, \sigma) \mid \mathcal{B}[\![b]\!]_\sigma = \text{false}\} \cup \{(\sigma, \sigma') \mid \mathcal{B}[\![b]\!]_\sigma = \text{true},\ \mathcal{C}[\![c]\!]_\sigma = \sigma'\}$$

P_1 if b then c else skip

P_2 if b then c

$P_1 \sim P_2$ iff $\forall \sigma, \sigma' \in \Sigma \ \langle P_1, \sigma \rangle \to \sigma' \iff \langle P_2, \sigma \rangle \to \sigma'$

$$\frac{\langle b, \sigma \rangle \to true \qquad \langle c, \sigma \rangle \to \sigma'}{\langle \text{if } b \text{ then } c \text{ else skip}, \sigma \rangle \to \sigma'} \iff \frac{\langle b, \sigma \rangle \to true \qquad \langle c, \sigma \rangle \to \sigma'}{\langle \text{if } b \text{ then } c, \sigma \rangle \to \sigma'}$$

$$\frac{\langle b, \sigma \rangle \to false \qquad \langle skip, \sigma \rangle \to \sigma}{\langle \text{if } b \text{ then } c \text{ else skip}, \sigma \rangle \to \sigma} \iff \frac{\langle b, \sigma \rangle \to false}{\langle \text{if } b \text{ then } c, \sigma \rangle \to \sigma}$$

$\{x = N, y = M, p = 0, d = 0\}$

while $x \leq y$

if $x \bmod 2 = 0$ then $\overset{A_T}{p := p + x}$

else $\overset{A_F}{d := d + x}$

$\hat{I} \rightarrow$

$x = x + 1$

$\{p = \sum_{i=N}^{M} \{i \mid i \text{ pari}\} \wedge d = \sum_{i=N}^{M} \{i \mid i \text{ dispari}\}\}$

1 Troviamo l'invariante eseguendo il programma e alcune iterazioni di ciclo

$p = \sum_{i=N}^{N} \{i \mid i \text{ è pari}\}$ $\quad d = \sum_{i=N}^{N} \{i \mid i \text{ è dispari}\}$ $\quad x = N+1$

$p = \sum_{i=N}^{N+1} \{i \mid i \text{ è pari}\}$ $\quad d = \sum_{i=N}^{N+1} \{i \mid i \text{ è dispari}\}$ $\quad x = N+2$

$p = \sum_{i=N}^{N+2} \{i \mid i \text{ è pari}\}$ $\quad d = \sum_{i=N}^{N+2} \{i \mid i \text{ è dispari}\}$ $\quad x = N+3$

$p = \sum_{i=N}^{N+3} \{i \mid i \text{ è pari}\}$ $\quad d = \sum_{i=N}^{N+3} \{i \mid i \text{ è dispari}\}$ $\quad x = N+4$

$\vdots \qquad \vdots$

$$I \equiv p = \sum_{i=N}^{x-1} \{i \mid i \text{ è pari}\} \wedge d = \sum_{i=N}^{x-1} \{i \mid i \text{ è dispari}\} \wedge x \leq y+1$$

Per dimostrare che I è invariante devo poter applicare la regola del while e per farlo devo applicare la consequence rule, ovvero dimostrare che $I \wedge b_w \wedge b_{if} \rightarrow A_T$ e che $I \wedge b_w \wedge \neg b_{if} \rightarrow A_F$

A_T e A_F le calcoliamo come le wp, passando per $\hat{I}$ calcolato attraverso l'applicazione delle interpretazioni dei comandi nel corpo del while e calcolando poi A_T e A_F applicando ad I le interpretazioni dei comandi nei rispettivi rami dell'if

$$\hat{I} = I[x+1/x] = p = \sum_{i=N}^{x-1} \{i \mid i \text{ è pari}\} \wedge d = \sum_{i=N}^{x-1} \{i \mid i \text{ è dispari}\} \wedge x \leq y+1$$

$$\overset{!}{=} p = \sum_{i=N}^{x} \{i \mid i \text{ è pari}\} \wedge d = \sum_{i=N}^{x} \{i \mid i \text{ è dispari}\} \wedge x+1 \leq y+1$$

$$A_T = \hat{I}[p+x/p] = p + x = \sum_{i=N}^{x} \{i \mid i \text{ è pari}\} \wedge d = \sum_{i=N}^{x} \{i \mid i \text{ è dispari}\} \wedge x \leq y$$

$$A_F = \hat{I}[d+x/d] = p = \sum_{i=N}^{x} \{i \mid i \text{ è pari}\} \wedge d + x = \sum_{i=N} \{i \mid i \text{ è dispari}\} \wedge x \leq y$$

$$I \wedge b_w \wedge b_{if} \equiv p = \sum_{i=N}^{x-1} \{i \mid i \text{ è pari}\} \wedge d = \sum_{i=N}^{x-1} \{i \mid i \text{ è dispari}\} \wedge x \leq y+1 \wedge x \leq y \wedge x \bmod 2 = 0$$

$$\equiv p = \sum_{i=N}^{x-1} \{i \mid i \text{ è pari}\} \wedge d = \sum_{i=N}^{x-1} \{i \mid i \text{ è dispari}\} \wedge x \leq y \wedge x \bmod 2 = 0$$

$$\equiv p + x = \sum_{i=N}^{x} \{i \mid i \text{ è pari}\} \wedge d = \sum_{i=N}^{x} \{i \mid i \text{ è dispari}\} \wedge x \leq y \wedge x \bmod 2 = 0 \equiv A_T$$

facciamo un altro passo e aggiungiamo il valore a p perché è pari

$$I \wedge b_w \wedge \neg b_{if} \equiv p = \sum_{i=N}^{x-1} \{i \mid i \text{ é pari}\} \wedge d = \sum_{i=N}^{x-1} \{i \mid i \text{ é dispari}\} \wedge x \leq y+1 \wedge x \leq y \wedge x \bmod 2 \neq 0$$

$$\equiv p = \sum_{i=N}^{x-1} \{i \mid i \text{ é pari}\} \wedge d = \sum_{i=N}^{x-1} \{i \mid i \text{ é dispari}\} \wedge x \leq y \wedge x \bmod 2 \neq 0$$

$$\equiv p = \sum_{i=N}^{x} \{i \mid i \text{ é pari}\} \wedge d+x = \sum_{i=N}^{x} \{i \mid i \text{ é dispari}\} \wedge x \leq y \wedge x \bmod 2 \neq 0 \equiv A_F$$

facciamo un altro passo e aggiungiamo il valore a d perché è dispari

Avendo mostrato che $I \wedge b_w \wedge b_{if} \rightarrow A_T$ per equivalenza
e che $I \wedge b_w \wedge \neg b_{if} \rightarrow A_F$ per equivalenza
So che posso applicare la conseguence rule e pertanto applicare la regola del while

$$\dfrac{\dfrac{\dfrac{\dfrac{\{A_T\}\, c\, \{\hat{I}\} \quad I \wedge b_w \wedge b_{if} \rightarrow A_T}{\{I \wedge b_w \wedge b_{if}\}\, c_1\, \{\hat{I}\}}\ \text{conseq rule} \checkmark \qquad \dfrac{\{A_F\}\, c_2\, \{\hat{I}\} \quad I \wedge b_w \wedge \neg b_{if} \rightarrow A_F}{\{I \wedge b_w \wedge \neg b_{if}\}\, c_2\, \{\hat{I}\}}\ \text{conseq rule} \checkmark}{\{I \wedge b_w\}\ \text{if } b_w \text{ then } c_1 \text{ else } c_2\ \{\hat{I}\}} \qquad \{\hat{I}\}\, c_3\, \{I\}}{\{I \wedge b_w\}\ \text{corpo}\ \{I\}}}{\{I\}\ \text{while } b_w \text{ do corpo}\ \{I \wedge \neg b_w\}}$$

Avendo mostrato ciò so che I è invariante devo però fare vedere che I è concorde alle precondizioni e che $I \wedge \neg b_w$ è concorde alle postcondizioni

$$I \wedge \neg b_w \equiv p = \sum_{i=N}^{x-1} \{i \mid i \text{ é pari}\} \wedge d = \sum_{i=N}^{x-1} \{i \mid i \text{ é dispari}\} \wedge x \leq y+1 \wedge x > y$$

$$= x = y+1 \wedge p = \sum_{i=N}^{y} \{i \mid i \text{ é pari}\} \wedge d = \sum_{i=N}^{y} \{i \mid i \text{ é dispari}\}$$

$$= x = y+1 \wedge p = \sum_{i=N}^{M} \{i \mid i \text{ é pari}\} \wedge d = \sum_{i=N}^{M} \{i \mid i \text{ é dispari}\} \equiv \text{post condizioni} \checkmark$$

$$\{x = N, y = M, p = 0, d = 0\} \rightarrow I \equiv p = \sum_{i=N}^{x-1} \{i \mid i \text{ é pari}\} \wedge d = \sum_{i=N}^{x-1} \{i \mid i \text{ é dispari}\} \wedge x \leq y+1$$

$$0 = \sum_{i=N}^{N-1} \{i \mid i \text{ é pari}\} \qquad 0 = \sum_{i=N}^{N-1} \{i \mid i \text{ é dispari}\} \qquad N \leq M+1 \checkmark$$

$$0 = 0 \checkmark \qquad 0 = 0 \checkmark$$

I valida per le precondizioni ✓