Mode System per linguaggi logici
il Mercury

Lorenzo Ceragioli

June 5, 2017

Table of Contents

Programmazione Logica

Kovalsky: programma = logica + controllo

- Semantica di un linguaggio di programmazione attraverso un sistema di deduzione logica.
- Teoremi = associazioni programma-semantica.
- Dimostrare = Calcolare.

Kovalsky: programma = logica + controllo

IDEA: Usare direttamente un sistema di deduzione come modello di calcolo.

- Sintassi più o meno quella della logica del primo ordine.
- Immediata semantica dichiarativa.
- Esecuzione come dimostrazione di un teorema.

Domini sintattici:

- $v, v', v_1, v_2 \in Var$ sono variabili logiche
- $f, f', f_1, f_2 \in FuncSym$ sono simboli di funzioni
- $p, p', p_1, p_2 \in PredName$ sono simboli di predicato

Segnatura di funzioni Σ
dove $f/n \in \Sigma$ sono coppie simbolo di funzione e relativa arietà

Segnatura di predicati Π
dove $p/n \in \Pi$ sono coppie simbolo di predicato e relativa arietà

Termini

Termini *ground* (universo di Herbrand)

$$\tau(\Sigma) = \bigcup_{f/n \in \Sigma} \{f(t_1, \ldots, t_n) \mid \{t_1, \ldots, t_n\} \subseteq \tau(\Sigma)\}$$

Termini

$$\tau(\Sigma, Var) = Var \cup \bigcup_{f/n \in \Sigma} \{f(t_1, \ldots, t_n) \mid \{t_1, \ldots, t_n\} \subseteq \tau(\Sigma, Var)\}$$

Formule

Formule *atomiche*

$$\alpha(\Sigma, Var, \Pi) = \{p(t_1, \ldots, t_n) \mid p/n \in \Pi \wedge \{t_1, \ldots, t_n\} \subseteq \tau(\Sigma, Var)\}$$

Clausole

$$C(\Sigma, Var, \Pi) = \{A \leftarrow B \mid A \in \alpha(\Sigma, Var, \Pi) \wedge B \in body(\Sigma, Var, \Pi)\}$$

- A viene chiamata *testa* della clausola
- B viene chiamata *corpo* della clausola

Formule

Clausole

$$\{A \leftarrow B \mid A \in \alpha(\Sigma, Var, \Pi) \wedge B \in body(\Sigma, Var, \Pi)\}$$

La testa di una clausola deve essere una formula atomica.

$body(\Sigma, Var, \Pi)$ dipende dal linguaggio, il caso più semplice è quello di *Prolog* nel quale contiene solo congiunzioni di letterali (clausole di Horn).

Programmi

$$P \subseteq_{fin} C(\Sigma, Var, \Pi)$$

Un programma è un insieme finito di clausole.

Semantica Dichiarativa

Significato delle clausole

Con $A \leftarrow G$ intendiamo $\forall X.\ A \leftarrow \exists Y.\ G$ dove X sono tutte e sole le variabili presenti in A e Y sono tutte e sole le variabili presenti in G e non in A.

Significato di un programma

La semantica di un programma logico può essere semplicemente ricondotto a quella dell'insieme di predicati del primo ordine che contiene. Semantica denotazionale basata sui modelli e sulla conseguenza logica.

Semantica Dichiarativa

Semantica di un programma mediante dimostrabilità.

La semantica Dichiarativa di un programma logico è l'insieme di tutte e sole le formule atomiche deducibili dalle clausole del programma (modello minimo di Herbrand).

In *Prolog* si usa una sola regola di derivazione *(risoluzione)*

$$\frac{\begin{array}{c} L_1 \vee \cdots \vee L_n \vee A \\ L'_1 \vee \cdots \vee L'_m \vee \neg A' \\ \theta = mgu(A, A') \end{array}}{\theta(L_1) \vee \cdots \vee \theta(L_n) \vee \theta(L'_1) \vee \cdots \vee \theta(L'_m)}$$

Dove mgu stà per *most general unifier*.

La *risoluzione* è completa per refutazione.

$$P, \neg A \vdash_{ris} false \iff P \vdash A$$

Le due semantiche definite sono equivalenti.

Per ogni programma P, per ogni formula atomica ground A,

$$P \models A \iff P, \neg A \vdash_{ris} false$$

Problemi

Problema di dimostrazione

Dimostrazione come **algoritmo di ricerca** su un albero infinito.

In teoria la regola di risoluzione è completa.

Occorre anche che la regola di derivazione sia applicata in una certa maniera.

Alcuni algoritmi, come una ricerca per livelli, sono completi,
altri, come la ricerca in profondità **non lo sono**.

Problema di dimostrazione

Prolog **usa una strategia di dimostrazione del secondo tipo**

- Ricerca in profondità con backtrack
- L'ordine dei rami dipende dall'ordine di definizione delle clausole
- E da quello dei letterali nelle clausole

Questo contraddice la semantica dichiarativa del programma

- l'ordine influenza la semantica
- le formule dimostrabili sono **un sottoinsieme** delle conseguenze logiche del programma

Esempio

Programma1

```
sposati(X,Y) :-
    sposati(Y,X).
sposati(abramo, sara).
```

Programma2

```
sposati(abramo, sara).
sposati(X,Y) :-
   sposati(Y,X).
```

Esempio

Programma1

```
sposati(X,Y) :-
    sposati(Y,X).
sposati(abramo, sara).
```

Semantica dichiarativa:

$$\{sposati(abramo, sara), \\ sposati(sara, abramo)\}.$$

Programma2

```
sposati(abramo, sara).
sposati(X,Y) :-
    sposati(Y,X).
```

Semantica dichiarativa:

$$\{sposati(abramo, sara), \\ sposati(sara, abramo)\}.$$

Esempio

Programma1

```
sposati(X,Y) :-
    sposati(Y,X).
sposati(abramo, sara).
```

Semantica dichiarativa:

$$\{sposati(abramo, sara), \\ sposati(sara, abramo)\}.$$

Semantica *Prolog*:

$$\{\}.$$

Programma2

```
sposati(abramo, sara).
sposati(X,Y) :-
    sposati(Y,X).
```

Semantica dichiarativa:

$$\{sposati(abramo, sara), \\ sposati(sara, abramo)\}.$$

Semantica *Prolog*:

$$\{sposati(abramo, sara), \\ sposati(sara, abramo)\}.$$

Inciso

Problema:
Ricerca di una dimostrazione per la semantica negli altri linguaggi.

Esempio (IMP)

```
x := 0;
if x = 0 then skip else (while true do skip)
```

- Sempre prima la regola relativa al ramo else.
- Sempre semantica dei comandi e poi della guardia.

Dettaglio trascurabile: semantica *small step* definisce l'ordine effettivo.

Altri problemi

- costrutti non dichiarativi per input, output
- **costrutti non dichiarativi** per influenzare la strategia di dimostrazione
  - tagli
  - predicati sulla forma di temini come atom() e gound()
- **assenza di *occur check*** per ragioni di efficienza
- costrutti builtin che **richiedono termini ground** per funzionare (errori a tempo di esecuzione)

Esempio

Programma

```
sumlist([], [], []).

sumlist([X|Xs], [Y|Ys], [Z|Zs]) :-
    Z is X+Y,
    sumlist(Xs, Ys, Zs).

goldbachlist(Ns) :-
    sumlist(Xs, Ys, Ns),
    primes(Xs),
    primes(Ys).
```

Esempio

Relativamente al nome di predicato $goldbachlist$

Ci aspetteremmo che la semantica del programma precedente contenga

$$\begin{aligned}\{goldbachlist([]),\\ goldbachlist([4]),\\ goldbachlist([5]),\\ goldbachlist([4,5]),\\ \ldots\}.\end{aligned}$$

In realtà la semantica del programma precedente contiene solo

$$\{goldbachlist([])\}.$$

Esempio

Se servono soluzioni di $goldbach(X)$ oltre alla lista vuota $(X \mapsto [])$ l'esecuzione terminerebbe lamentando un errore relativo all'istanziazione dei termini relativo all'operatore is.

is richiede che il termine alla sua destra sia *ground*.

Questo esprime un **vincolo sulla *modeness* del predicato**.

Soluzioni

- approccio dinamico per mezzo di costrutti non dichiarativi quali *ground()* ed *atom()* in *Prolog*.
- approccio statico (mode system) descrittivo.
- **approccio statico (mode system) prescrittivo.**

Bonus: come si farebbe in *Prolog*

```
sumlist([], [], []).

sumlist([X|Xs], [Y|Ys], [Z|Zs]) :-
    ground(Z),
    ground(Y),
    X is Z-Y,
    sumlist(Xs, Ys, Zs).

sumlist([X|Xs], [Y|Ys], [Z|Zs]) :-
    ground(X),
    ground(Y),
    Z is X+Y,
    sumlist(Xs, Ys, Zs).

sumlist([X|Xs], [Y|Ys], [Z|Zs]) :-
    ground(X),
    ground(Z),
    Y is Z-X,
    sumlist(Xs, Ys, Zs).
```

Mode System

Cos'è un *mode* di un predicato

Esempio: nessun vincolo sull'istanziazione delle variabili.

```
append([], Ys, Ys).

append([N|Xs], Ys, [N|Zs]) :-
    append(Xs, Ys, Zs).
```

Il predicato `append` potrà essere usato con parametri aventi diversi gradi di istanziazione.

Cos'è un *mode* di un predicato

Tutti i seguenti usi sono ragionevoli (usiamo dei valori per chiarezza dove vogliamo rappresentare un termine *ground*).

```
append([1,2,3], [1,2], [1,2,3,1,2]).

append([1,2], [1], [1,1,2]).

append([1,2,3], [1,2], X).

append(X, [1,2], X).

append(A, B, C).

append([X,Y], [1,2], [Z|Zs]).
```

Cos'è un *mode* di un predicato

Ognuna delle applicazioni precedenti corrisponde ad un *mode* diverso del predicato `append`.

Intuitivamente un `mode` è una specifica sull'istanziazione dei parametri del predicato.

Per ogni parametro del predicato definisce lo **stato di istanziazione** che un termine deve avere quando viene legato e quale sarà lo stato di istanziazione dopo il *binding*.

Un predicato può avere più *mode*.

Esempio

```
append([1,2,3], [1,2], [1,2,3,1,2]).

append([1,2], [1], [1,1,2]).
```

Nel *mode* relativo a questa occorrenza

- il primo termine è ***ground*** inizialmente e anche dopo l'applicazione
- il secondo termine è ***ground*** inizialmente e anche dopo l'applicazione
- il terzo termine è ***ground*** inizialmente e anche dopo l'applicazione

O meglio, il valore non è definito dopo l'applicazione nel secondo caso, perché il predicato è sempre falso...

Esempio

```
append([1,2,3], [1,2], X).
```

Nel *mode* relativo a questa occorrenza

- il primo termine è ***ground*** inizialmente e anche dopo l'applicazione
- il secondo termine è ***ground*** inizialmente e anche dopo l'applicazione
- il terzo termine è ***free*** inizialmente e dopo l'applicazione è ***ground*** (legato al valore `[1,2,3,1,2]`)

Esempio

```
append(A, B, C).
```

Nel *mode* relativo a questa occorrenza

- il primo termine è ***free*** inizialmente, dopo l'applicazione ...
- il secondo termine è ***free*** inizialmente, dopo l'applicazione ...
- il terzo termine è ***free*** inizialmente, dopo l'applicazione ...

Otteniamo alcuni legami fra le tre variabili libere, questo dovrà essere ricordato se una di esse dovesse essere legata a un valore in futuro.
Otteniamo anche delle informazioni riguardo allo **stato di istanziazione che avranno le variabili dopo il *binding***.

Esempio

```
append([X,Y], [1,2], [Z|Zs]).
```

?

In alcuni linguaggi i ***mode* sono dichiarati** e verificati dal compilatore o a tempo di esecuzione.

In *Mercury* ad esempio per ogni predicato avremo una dichiarazione di tipi e più dichiarazioni di *mode*.

```
:- pred append(list(T), list(T), list(T)).
:- mode append(in     , in     , out) is det.
:- mode append(out    , out    , in) is multi.
:- mode append(out    , in     , in) is semidet.
```

Mercury

Mercury è un linguaggio di programmazione

- efficiente
- funzionale e logico (puro)
- con un sistema di tipi forte e statico
- **con un sistema di modi forte e statico**
- con un sistema di determinismo forte e statico

Sintassi

Come detto precedentemente dobbiamo definire che forma può avere il corpo di una clausola $B \in body(\Sigma, Var, \Pi)$.

$$
\begin{aligned}
B ::= \; & p(t_1, \ldots, t_n) \mid \\
& v = v' \mid \\
& v = f(t_1, \ldots, t_n) \mid \\
& \neg B^1 \mid \\
& \wedge B^1, B^2, \ldots B^n \mid \\
& \vee B^1, B^2, \ldots B^n \mid \\
& if \;\; B^1 \; then \; B^2 \; else \; B^3
\end{aligned}
$$

Dove $t_1, t_2, \ldots, t_n \in \tau(\Sigma, Var)$

Forma normale per i programmi

Termini *flattered*

$$\widetilde{\tau}(\Sigma, Var) = Var \cup \bigcup_{f/n \in \Sigma} \{f(v_1, \ldots, v_n) \mid \{v_1, \ldots, v_n\} \subseteq Var\}$$

Formule *atomiche flattered*

$$\widetilde{\alpha}(\Sigma, Var, \Pi) = \{p(v_1, \ldots, v_n) \mid p/n \in \Pi \land \{v_1, \ldots, v_n\} \subseteq Var\}$$

Clausole *flattered*

$$\widetilde{C}(\Sigma, Var, \Pi) = \{A \leftarrow B \mid A \in \widetilde{\alpha}(\Sigma, Var, \Pi) \land B \in \widetilde{body}(\Sigma, Var, \Pi)\}$$

Forma normale per i programmi

Corpo di una clausola *flattered* $B \in \widetilde{body}(\Sigma, Var, \Pi)$.

$$
\begin{aligned}
B ::= \; & p(v_1, \dots, v_n) \mid \\
& v = v' \mid \\
& v = f(v_1, \dots, v_n) \mid \\
& \exists v_1, \dots v_n.\ B^1 \mid \\
& \neg B^1 \mid \\
& \wedge\, B^1, B^2, \dots B^n \mid \\
& \vee\, B^1, B^2, \dots B^n \mid \\
& if\ B^1\ then\ B^2\ else\ B^3
\end{aligned}
$$

Forma normale per i programmi

- Testa delle clausole è *flattered*.
- Predicati nel corpo della clausola sono *flattered*.
- Termini nel corpo della clausola sono *flattered*.
- Quantificazione esistenziale sulle variabili esplicitata.

$$uq : \widetilde{body}(\Sigma, Var, \Pi) \to 2^{Var}$$
$$uq(B) = \{v \in Var \mid v \text{ occorre non quantificata in } B\}$$

Esempio

$$
\begin{array}{l}
\texttt{append(Xs, Ys, Zs)} \leftarrow \\
\qquad \vee\,(\ \wedge(\texttt{Xs=[]}, \\
\qquad\qquad \texttt{Ys=Zs} \\
\qquad\qquad), \\
\qquad\quad \exists \texttt{Xs0,Zs0,X.}(\\
\qquad\qquad \wedge(\texttt{Xs = [X | Xs0]}, \\
\qquad\qquad\quad \texttt{Zs = [X | Zs0]}, \\
\qquad\qquad\quad \texttt{append(Xs0, Ys, Zs0)} \\
\qquad\qquad) \\
\qquad\quad) \\
\qquad)
\end{array}
$$

Sistema di tipi

Un tipo definisce un **insieme di termini *ground***.

Una *definizione di tipo* ha la forma

$$\texttt{:- type } T(v_1, \ldots, v_n) \texttt{ ---> } \begin{array}{l} f_1(t^1_1, \ldots, t^1_{m_1}); \\ \ldots; \\ f_k(t^k_1, \ldots, t^k_{m_k})\,\texttt{.} \end{array}$$

Dove:

- T è un costruttore di tipo;
- $v_1 \ldots v_n$ sono parametri di tipo;
- $f_1 \ldots f_k \in FuncSym$ sono simboli di funzione;
- $f_1/m_1 \ldots f_k/m_k \in \Sigma$;
- $t_1 \ldots t_m$ sono tipi (parametri $\subseteq \{v_1 \ldots v_n\}$).

Esempio

Prendiamo come esempio il tipo lista

```
:- type list(T) ---> [];
                     [T | list(T)].
```

Il termine `[1,2,3,4]` è di tipo `list(int)`.

```
[1,2,3,4] = [ 1|
               [ 2|
                  [ 3|
                     [ 4|
                        []
                     ]
                  ]
               ]
            ]
```

Stato di istanziazione

Stati di istanziazione (inst)

$$
\begin{aligned}
Inst(\Sigma) = \{free\} \cup & \\
& \{\ bound(I) \mid I \subseteq \{f(i_1, \ldots, i_n) \mid f/n \in \Sigma \wedge \\
& \qquad \{i_1, \ldots, i_n\} \subseteq Inst(\Sigma)\} \wedge \\
& \qquad \text{ogni } f \text{ compare al massimo una volta in } I\ \}.
\end{aligned}
$$

Esempio

$$\Sigma = \{[\,]/0,\ [\_|\_]/2\}$$

Sono stati di istanziazione $i_1, i_2 \in Inst(\Sigma)$

$$i_1 = bound(\{[\,],\ [free|free]\})$$
$$i_2 = bound(\{\ [\,],\ [free|bound(\{\ [\,],\ [free|free]\ \})]\ \})$$

Non è stato di istanziazione $i_3 \notin Inst(\Sigma)$

$$i_3 = bound(\{[\,],\ [free|free],\ [free|bound(\{[\,],\ [free|free]\})]\})$$

Funzione di concretizzazione γ

$$\gamma : \; Inst(\Sigma) \rightarrow 2^{\tau(\Sigma, Var)}$$

$$\gamma(free) = \{\_\}$$
$$\gamma(bound(I)) = \bigcup_{f(i_1,\ldots,i_n) \in I} \{f(t_1, \ldots, t_n) \mid \forall j \in \{1, \ldots, n\}.\ t_j \in \gamma(i_j)\}$$

Dove indichiamo con $\_ \in Var$ una variabile qualunque che non compare altrove (il nostro semplice *mode system* non tiene conto degli *alias*).

Esempio

$$
\begin{aligned}
i = bound(&\\
&\{[free|bound(\\
&\qquad \{[free|bound(\\
&\qquad\qquad \{[free|bound(\{[\ ]\})]\})]\})]\})
\end{aligned}
$$

$$
\begin{aligned}
\gamma(i) = \gamma(bound(&\\
&\{[free|bound(\\
&\qquad \{[free|bound(\\
&\qquad\qquad \{[free|bound(\{[\ ]\})]\})]\})]\}))
\end{aligned}
$$

Esempio

$$
\begin{aligned}
i = bound(&\\
&\{[free|bound(\\
&\qquad \{[free|bound(\\
&\qquad\qquad \{[free|bound(\{[\ ]\})]\})]\})]\})
\end{aligned}
$$

$$
\begin{aligned}
\gamma(i) = \{[t_1|t_2] \mid t_1 \in \gamma(free) \wedge &\\
t_2 \in \gamma(bound(&\\
&\{[free|bound(\\
&\qquad \{[free|bound(\{[\ ]\})]\})]\})\}))
\end{aligned}
$$

Esempio

$$
\begin{aligned}
i = bound(&\\
&\{[free|bound(\\
&\qquad \{[free|bound(\\
&\qquad\qquad \{[free|bound(\{[\ ]\})]\})]\})]\})
\end{aligned}
$$

$$
\begin{aligned}
\gamma(i) = \{[t_1|t_2] \mid t_1 \in \{\_\} \wedge&\\
&t_2 \in \gamma(bound(\\
&\qquad \{[free|bound(\\
&\qquad\qquad \{[free|bound(\{[\ ]\})]\})]\}))\}
\end{aligned}
$$

Esempio

$$
\begin{aligned}
i = bound(&\\
&\{[free|bound(\\
&\qquad \{[free|bound(\\
&\qquad\qquad \{[free|bound(\{[\,]\})]\})]\})]\})
\end{aligned}
$$

$$
\begin{aligned}
\gamma(i) = \{[t_1|t_2] \mid t_1 = \_ \wedge &\\
t_2 \in \gamma(bound(&\\
&\{[free|bound(\\
&\qquad \{[free|bound(\{[\,]\})]\})]\})\}))
\end{aligned}
$$

Esempio

$$
\begin{aligned}
i = bound(&\\
&\{[free|bound(\\
&\qquad \{[free|bound(\\
&\qquad\qquad \{[free|bound(\{[\ ]\})]\})]\})]\})
\end{aligned}
$$

$$
\begin{aligned}
\gamma(i) = \{[t_1|t_2] \mid t_1 = \_ \wedge &\\
&t_2 \in \{[t_3|t_4] \mid t_3 = \_ \wedge\\
&\qquad t_4 \in \gamma(bound\\
&\qquad\qquad \{[free|bound(\{[\ ]\})]\})\})\})
\end{aligned}
$$

Esempio

$$
\begin{aligned}
i = bound(&\\
&\{[free|bound(\\
&\qquad \{[free|bound(\\
&\qquad\qquad \{[free|bound(\{[\,]\})]\})]\})]\})
\end{aligned}
$$

$$
\begin{aligned}
\gamma(i) = \{[t_1|t_2] \mid t_1 = \_ \wedge&\\
&t_2 \in \{[t_3|t_4] \mid t_3 = \_ \wedge\\
&\qquad t_4 \in \{[t_5|t_6] \mid t_5 = \_ \wedge\\
&\qquad\qquad t_6 \in \gamma(bound(\{[\,]\}))\})\})\})
\end{aligned}
$$

Esempio

$$
\begin{aligned}
i = bound(&\\
&\{[free|bound(\\
&\qquad \{[free|bound(\\
&\qquad\qquad \{[free|bound(\{[\ ]\})]\})]\})]\})
\end{aligned}
$$

$$
\begin{aligned}
\gamma(i) = \{[t_1|t_2] \mid t_1 = \_ \wedge &\\
& t_2 \in \{[t_3|t_4] \mid t_3 = \_ \wedge\\
&\qquad t_4 \in \{[t_5|t_6] \mid t_5 = \_ \wedge\\
&\qquad\qquad t_6 \in \{[\ ]\}\ \}\ \}\ \}
\end{aligned}
$$

Esempio

$$
\begin{aligned}
i = bound(&\\
&\{[free|bound(\\
&\qquad\{[free|bound(\\
&\qquad\qquad\{[free|bound(\{[\ ]\})]\})]\})]\})
\end{aligned}
$$

$$
\begin{aligned}
\gamma(i) = \{[t_1|t_2] \mid t_1 = \_ \wedge&\\
&t_2 \in \{[t_3|t_4] \mid t_3 = \_ \wedge\\
&\qquad t_4 \in \{[\_|[\ ]]\}\ \}\ \}
\end{aligned}
$$

Esempio

$$
\begin{aligned}
i = bound(& \\
&\{[free|bound(\\
&\qquad \{[free|bound(\\
&\qquad\qquad \{[free|bound(\{[\ ]\})]\})]\})]\})
\end{aligned}
$$

$$
\gamma(i) = \{[t_1|t_2] \mid t_1 = \_ \wedge \\
t_2 \in \{[\_|[\_|[\ ]]]\}
$$

Esempio

$$
\begin{aligned}
i = bound(& \\
&\{[free|bound(\\
&\qquad \{[free|bound(\\
&\qquad\qquad \{[free|bound(\{[\ ]\})]\})]\})]\})
\end{aligned}
$$

$$\gamma(i) = \{[\_|[\_|[\_|[\ ]]]]\}$$

Partial Order $\preceq$

Definiamo l'ordine parziale $\preceq$ tale che $i \preceq i'$ se i è *istanziato almeno quanto* i'.

$$
\begin{aligned}
& i \preceq free \\
& bound(I) \preceq bound(I') \iff \\
& \qquad \forall f(i_1, \ldots, i_n) \in I.\ \exists f(i'_1, \ldots, i'_n) \in I'. \\
& \qquad\qquad \forall j \in \{1, \ldots, n\}.\ i_j \preceq i'_j
\end{aligned}
$$

Non vogliamo che il binding fra una variabile ed il parametro di un predicato modifichi lo stato di istanziazione della variabile in modo tale che esso sia meno istanziato di quanto non fosse prima.

Partial Order $\preceq$

Definiamo il grado di istanziazione $not\_reached$

$$not\_reached = bound(\emptyset).$$

Notiamo che

$$\gamma(not\_reached) = \emptyset.$$

Partial Order $\preceq$

Vale che, per ogni $i \in Inst(\Sigma)$

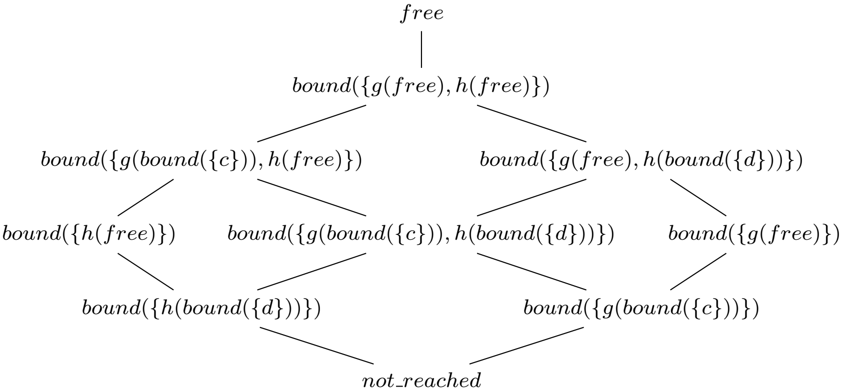
$$not\_reached \preceq i \preceq free$$

Inoltre

$$(Inst(\Sigma), \preceq) \text{ è un } \textit{Reticolo Completo} \text{ con}$$
$$\top = free \text{ e } \bot = not\_reached$$

Scriviamo $\curlywedge$ e $\curlyvee$ per gli operatori *meet* e *join*.

Herbrand Universe $= \{g(c), h(d)\}$.



ground

Dato Σ definiamo il grado di istanziazione $ground$

$$ground = bound(\{f(ground_1, \ldots, ground_n) \mid f/n \in \Sigma\}).$$

Notiamo che

$$\gamma(gound) = \tau(\Sigma).$$

Se $i \in Inst(\Sigma) \preceq ground$ allora diciamo che i è uno stato di istanziazione *ground*.

Instmap

Definiamo $Instmap(Var, \Sigma)$ **l'insieme delle funzioni**

$$Instmap(Var, \Sigma) \subseteq Var \to Inst(\Sigma)$$

tale che

$$\begin{aligned} F \in Instmap(Var, \Sigma) \iff \\ &(\exists v \in dom(F).\ F(v) = not\_reached) \\ &\quad \implies (\forall v \in dom(F). F(v) = not\_reached) \end{aligned}$$

unreachable instmap

Definiamo $Unreachable \subseteq Instmap(Var, \Sigma)$

$$
\begin{aligned}
&Unreachable(Var, \Sigma) = \\
&\qquad \{F \in Instmap(Var, \Sigma) \mid \forall v \in dom(F).\ F(v) = not\_reached\}
\end{aligned}
$$

Chiamiamo *unreachable* le instmap in $Unreachable(Var, \Sigma)$.

instmap update $\oplus$

Con il proseguire della dimostrazione le variabili diventeranno via via più istanziate.

Definiamo instmap update l'operatore $\oplus$

$\oplus : \; Instmap(Var, \Sigma) \times Instmap(Var, \Sigma) \rightarrow Instmap(Var, \Sigma)$
dove $F \oplus F'$richiede che $\forall v \in dom(F').\ v \in dom(F) \wedge F'(v) \preceq F(v)$

$$(F \oplus F')(v) = \begin{cases} not\_reached & \text{se } F \in Unreachable(Var, \Sigma) \\ & \quad \vee\ F' \in Unreachable(Var, \Sigma) \\ F'(v) & \text{se } v \in dom(F') \\ F(v) & \text{altrimenti} \end{cases}$$

Mode

Definiamo $Mode(Var, \Sigma)$

$$
\begin{aligned}
Mode(Var, \Sigma) &\subseteq Instmap(Var, \Sigma) \times Instmap(Var, \Sigma) \\
Mode(Var, \Sigma) &= \{\langle F, F' \rangle \mid \\
&\qquad dom(F) = dom(F') \wedge \\
&\qquad \forall v \in dom(F).\ F'(v) \preceq F(v)\}
\end{aligned}
$$

Notazione

$$
\begin{aligned}
M &= \langle F, F' \rangle \\
M_{init} &= F \\
M_{fin} &= F' \\
dom(M) &= dom(F) = dom(F') \\
M \oplus F &= \langle M_{init}, M_{fin} \oplus F \rangle
\end{aligned}
$$

Partial Order $\sqsubseteq$

Quando definiamo il $mode$ per un predicato stiamo richiedendo che una variabile abbia un certo stato di istanziazione perché compaia come parametro di un predicato. In realtà è necessario anche accettare stati di istanziazione più specifici di quello dichiarato.

Definiamo l'ordine parziale $\sqsubseteq$ tale che $i \sqsubseteq i'$ se i è più specifico di i' (diremo anche i è compatibile con i').

$$
\begin{aligned}
&free \sqsubseteq free \\
&not\_reached \sqsubseteq free \\
&bound(I) \sqsubseteq bound(I') \iff \\
&\qquad \forall f(i_1, \ldots, i_n) \in I.\ \exists f(i'_1, \ldots, i'_n) \in I'. \\
&\qquad\qquad \forall j \in \{1, \ldots, n\}.\ i_j \sqsubseteq i'_j
\end{aligned}
$$

Partial Order $\sqsubseteq$

$\forall i \in Inst(\Sigma).\ not\_reached \sqsubseteq i$

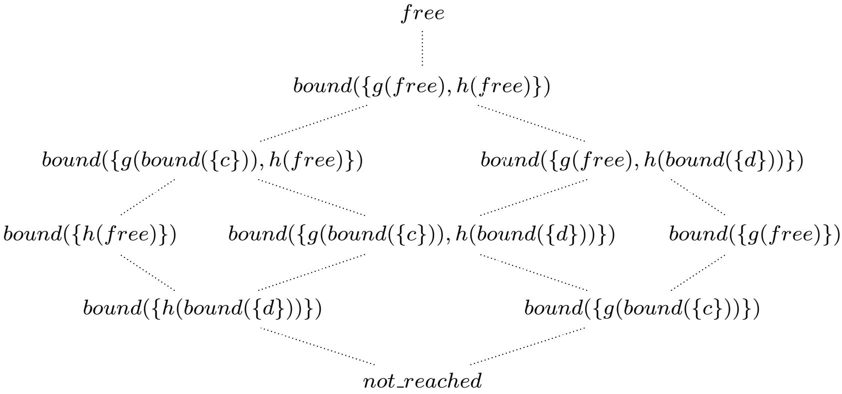
$\forall i, i' \in Inst(\Sigma). i \sqsubseteq i' \implies i \preceq i'$

$\perp_{\sqsubseteq} = not\_reached$ greatest lower bound esiste sempre

$\nexists \top_{\sqsubseteq}$ least upper bound no!

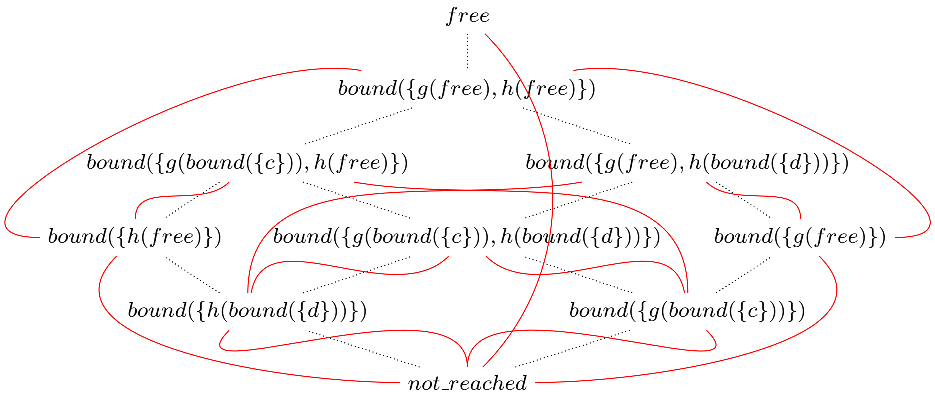
Esempio: *diagramma di Hasse*

Herbrand Universe $= \{g(c), h(d)\}$.



Esempio: *diagramma di Hasse*

Herbrand Universe $= \{g(c), h(d)\}$.



Funzione di astrazione α

$$\alpha : \; 2^{\tau(\Sigma, Var)} \to Inst(\Sigma)$$

$$\alpha(T) = \bigsqcup_{t \in T} \alpha'(t)$$

$$\alpha'(\_) = free$$
$$\alpha'(f(t_1, \ldots, t_n)) = bound(\{f(\alpha'(t_1), \ldots, \alpha'(t_n))\})$$

Funzione di astrazione α

Notiamo che α è una funzione parziale.

- Non è definita quando non esiste il least upper bound per T.
- Non è definita se sono presenti variabili con *alias* in T.

La coppia di funzioni α e γ formano una *connessione di Galois*.

$$(2^{\tau(\Sigma, Var)}, \subseteq) \underset{L}{\overset{R}{\rightleftarrows}} (Inst(\Sigma), \sqsubseteq)$$

Esempio

$\alpha'([\,]) = bound(\ \{[\,]\}\ )$
$\alpha'([A]) = bound(\{\ [\ free \mid bound(\{\ [\ ]\ \})]\ \})$
$\alpha'([A, B]) = bound(\{\ [\ free \mid bound(\{\ [\ free \mid bound(\{\ [\ ]\ \})]\ \})]\ \})$
$\alpha'([A, A])$ non è definita

$$\begin{aligned}\alpha(\{[\,], [A, B]\}) &= \alpha'([A, B]) \sqcup\ \alpha'([\,])\\ &= bound(\ \{\ [\ ],\ [free|\ bound(\{\ [free|\ bound(\{\ [\ ]\ \})]\ \})]\ \}\ )\end{aligned}$$

$\alpha(\{[A], [[\,]]\}) =$ non è definita

Possiamo dare una definizione alternativa di *ground*

$$ground = bound(\{f(ground_1, \ldots, ground_n) \mid f/n \in \Sigma\}).$$

$$ground = \alpha(\tau(\Sigma)).$$

Procedura

Una procedura è una coppia *clausola-mode* nella quale *mode* è adeguata per la clausola

$$Proc(\Sigma, Var, \Pi) \subseteq C(\Sigma, Var, \Pi) \times Mode(\Sigma, Var)$$

$$Proc(\Sigma, Var, \Pi) = \\ \{\langle p(v_1, \ldots, v_n) \leftarrow B,\ M\rangle \mid dom(M) = v_1, \ldots, v_n\}$$

Operazioni per *mode analysis*

Definiamo alcune operazioni su *mode*

- operazione di *mode sequence* $\triangleright$
- operazione di restrizione sui *mode* $\ominus$
- operazione di *mode merge* $\bowtie$

Funzioni parziali: errori di *mode* per il compilatore.

Operazione di *mode sequence* $\triangleright$

$$\triangleright : \; Mode(\Sigma, Var) \times Mode(\Sigma, Var) \rightarrow Mode(\Sigma, Var)$$

$$\langle F, F' \rangle \triangleright \langle F', F'' \rangle = \langle F, F'' \rangle$$

Operazioni per *mode analysis*: $\ominus$

Operazione di restrizione sui *mode* $\ominus$

$$\ominus : \; Mode(\Sigma, Var) \times 2^{Var} \rightarrow Mode(\Sigma, Var)$$

$$\begin{aligned} M \ominus V = \langle & \{v \mapsto M_{init}(v) \mid v \in dom(M) \backslash V\}, \\ & \{v \mapsto M_{fin}(v) \mid v \in dom(M) \backslash V\} \rangle \end{aligned}$$

Operazioni per *mode analysis*: $\bowtie$

Operazione di *mode merge* $\bowtie$

$$\bowtie\colon\ Mode(\Sigma, Var) \times Mode(\Sigma, Var) \rightarrow Mode(\Sigma, Var)$$

$$\begin{aligned}\langle F, F' \rangle \triangleright \langle F, F'' \rangle = \langle F, \{ v \mapsto i \mid v \in dom(F) \\ \wedge\, i = F'(v) \sqcup F''(v) \} \rangle\end{aligned}$$

Unificazione astratta fra due stati di istanziazione i_1 e i_2 è i

- L'unificazione deve essere più instanziata sia di i_1 sia di i_2 ($i \preceq i_1$, $i \preceq i_2$).
- L'idea è quella del *greatest lower bound* su $(Inst, \preceq)$.
- Problema dell'*aliasing* delle variabili.
- Soluzione banale: stato di istanziazione risultante deve essere *ground*.

... *Liveness information* e *alias tracking*.

Unificazione astratta fra due stati di istanziazione i_1 **e** i_2 **è** i

$$abs\_unify\_inst(i_1, i_2, i) \iff i = i_1 \curlywedge i_2 \;\wedge\; i \preceq ground$$

Esempi

$$abs\_unify\_inst(free,\ bound(\{g(bound(\{c\}))\}),\ i)$$
$$\iff \quad i = bound(\{g(bound(\{c\}))\})$$

$$\begin{aligned} abs\_unify\_inst(\ & bound(\{g(bound(\{c\})), \\ & \qquad\qquad h(bound(\{d\}))\}), \\ & bound(\{g(free)\}), \\ & i) \end{aligned}$$
$$\iff \quad i = bound(\{g(bound(\{c\}))\})$$

Esempi

$$\begin{aligned} abs\_unify\_inst(&bound(\{h(free)\}),\\ &bound(\{g(free)\}),\\ &i)\\ \iff \quad & i = not\_reached \end{aligned}$$

$$abs\_unify\_inst(free,\ bound(\{g(free)\}),\ i)$$

non è definito

Unificazione astratta fra stato istanziazione e termine

- Per come la abbiamo definita l'unificazione astratta opera solo su coppie di stati di istanziazione.
- Che corrisponde all'unificazione di due variabili.
- Nel linguaggio abbiamo il caso generico di unificazione fra una variabile ed un termine.
- Per via della forma normale possiamo limitarci ai temini flattered.
- Serve unificazione astratta che rappresenti questo caso nel dominio astratto degli stati di istanziazione.

Funtori di stati di istanziazione

Un *funtore di stati di istanziazione* è l'applicazione di un simbolo di funzione a degli stati di instanziazione

$f(i_1, \ldots, i_n)$ dove $f/n \in \Sigma \wedge \forall j \in \{1, \ldots, n\}.\ i_j \in Inst(\Sigma)$

Unificazione astratta fra uno stato di istanziazione i_1 ed un funtore $f(i_1, \ldots, i_n)$.

$$
\begin{aligned}
&abs\_unify\_inst\_func(i, f, \{i_1, \ldots, i_n\}, i', \{i'_1, \ldots, i'_n\}) \iff \\
&\qquad abs\_unify\_inst(i, bound(\{f(i_1, \ldots, i_n)\}), i') \land \\
&\qquad (\ i' = bound(\{f(i'_1, \ldots, i'_n)\}) \lor \\
&\qquad\qquad (i' = not\_reached \land \forall j \in \{1, \ldots, n\}.\ i'_j = not\_reached)\ )
\end{aligned}
$$

Correttezza di una procedura

La procedura $\langle p(v_1, \ldots, v_n) \leftarrow B\ ,\ \langle F, F' \rangle \rangle$ **è *mode corretta* solo se**

per ogni sostituzione di variabili $\theta : Var \rightarrow Term$ tale che
$\forall v \in dom(F).\ \theta(v) \in \gamma(F(v))$,

la sostituzione finale θ' risultante dall'esecuzione del corpo della procedura è tale che
$\forall v \in dom(F).\ \theta'(v) \in \gamma(F'(v))$.

Mode judgement

La *mode correttezza* **di una procedura** R rispetto ad un ambiente Γ viene espressa attraverso il *mode judgement* $\Gamma \Vdash R$.

L'ambiene Γ è un insieme di procedure.

$$\Gamma \subseteq Proc(\Sigma, Var, \Pi)$$

Mode judgement

La *mode correttezza* **di una associazione formula-*mode*** $B : M$ rispetto ad un ambiente esteso $\langle \Gamma, V \rangle$ viene espressa attraverso il *mode judgement* $\langle \Gamma, V \rangle \vdash B : M$.

$V \subseteq Var$ è l'insieme delle variabili non quantificate in B.

Regola per le procedure

PROC

$$\frac{\begin{array}{c} R \in \Gamma \\ R = \langle p(v_1, \ldots, v_n) \leftarrow B, M \rangle \\ \langle \Gamma, V \rangle \vdash B : M' \\ dom(M) = dom(M') = \{v_1, \ldots, v_n\} = V = uq(B) \\ \forall v \in \{v_1, \ldots, v_n\}.\ M_{init}(v) \sqsubseteq M'_{init} \\ \forall v \in \{v_1, \ldots, v_n\}.\ M'_{fin}(v) \sqsubseteq M_{fin} \end{array}}{\Gamma \Vdash R}$$

Regole per le formule

CONJ

$$\dfrac{\begin{array}{c}\langle \Gamma, uq(B_1)\rangle \vdash B_1 : M_1 \\ \vdots \\ \langle \Gamma, uq(B_n)\rangle \vdash B_n : M_n \\ M = M_1 \triangleright \cdots \triangleright M_n\end{array}}{\langle \Gamma, V\rangle \vdash \wedge\ B_1, \ldots, B_n : M}$$

Regole per le formule

DISJ

$$\frac{\langle \Gamma, uq(B_1) \rangle \vdash B_1 : M_1 \quad \vdots \quad \langle \Gamma, uq(B_n) \rangle \vdash B_n : M_n \quad M = M_1 \bowtie \ldots \bowtie M_n}{\langle \Gamma, V \rangle \vdash \vee\ B_1, \ldots, B_n : M}$$

Regole per le formule

SOME

$$\frac{\langle \Gamma, V \backslash \{v_1, \ldots, v_n\} \rangle \vdash B : M' \qquad M = M' \ominus \{v_1, \ldots, v_n\}}{\langle \Gamma, V \rangle \vdash \exists v_1, \ldots, v_n.\ B : M}$$

NOT

$$\frac{\langle \Gamma, V \rangle \vdash B : M \qquad nobind(M, V)}{\langle \Gamma, V \rangle \vdash \neg B : M}$$

dove

$$nobind(M, V) \iff \forall v \in V.\ M_{init}(v) = M_{fin}(v)$$

Regole per le formule

ITE

$$\frac{\begin{array}{c}\langle \Gamma, uq(B_1) \rangle \vdash B_1 : M_1 \\ \langle \Gamma, uq(B_2) \rangle \vdash B_2 : M_2 \\ \langle \Gamma, uq(B_3) \rangle \vdash B_3 : M_3 \\ nobind(M_1, V) \\ M = (M_1 \triangleright M_2) \bowtie M_3\end{array}}{\langle \Gamma, V \rangle \vdash if\ B_1\ then\ B_2\ else\ B_3 : M}$$

Regole per le formule

CALL

$$\frac{\begin{array}{c}\langle p(v'_1,\ldots,v'_n) \leftarrow B,\ M' \rangle \in \Gamma \\ \forall j \in \{1,\ldots,n\}.\ M_{init}(v_j) \sqsubseteq M'_{init}(v'_j) \\ M_{fin} = M_{init} \oplus \{v_j \mapsto i_j \mid j \in \{1,\ldots,n\} \wedge \\ \qquad\qquad i_j = M'_{fin}(v'_j) \curlywedge M_{init}(v_j)\}\end{array}}{\langle \Gamma, V \rangle \vdash p(v_1,\ldots,v_n) : M}$$

Regole per le formule

UNIFY-VV

$$\frac{\begin{array}{c} abs\_unify\_inst(M_{init}(v), M_{init}(v'), i) \\ M_{fin} = M_{init} \oplus \{v \mapsto i, v' \mapsto i\} \end{array}}{\langle \Gamma, V \rangle \vdash v = v' : M}$$

UNIFY-VF

$$\frac{\begin{array}{c} v \notin \{v_1, \ldots, v_n\} \\ i = M_{init}(v) \\ \bar{i} = \langle M_{init}(v_1), \ldots, M_{init}(v_n) \rangle \\ abs\_unify\_inst\_func(i, f, \bar{i}, i', \langle i'_1, \ldots, i'_n \rangle) \\ M_{fin} = M_{init} \oplus \{v \mapsto i', v_1 \mapsto i'_1, \ldots, v_n \mapsto i'_n\} \end{array}}{\langle \Gamma, V \rangle \vdash v = f(v_1, \ldots, v_n) : M}$$

Esempio

$$
\left.
\begin{array}{l}
\texttt{append(Xs, Ys, Zs)} \leftarrow \\
\qquad \vee(\ \wedge(\texttt{Xs=[]}, \\
\qquad\qquad\quad \texttt{Ys=Zs} \\
\qquad\qquad\ ), \\
\qquad\qquad \exists \texttt{Xs0,Zs0,X.}(\\
\qquad\qquad\quad \wedge(\texttt{Xs = [X | Xs0]}, \\
\qquad\qquad\qquad \texttt{append(Xs0, Ys, Zs0)}, \\
\qquad\qquad\qquad \texttt{Zs = [X | Zs0]} \\
\qquad\qquad\quad) \\
\qquad\qquad) \\
\qquad)
\end{array}
\right\} \qquad (B)
$$

Esempio

Assumiamo

$$
\begin{aligned}
&\Gamma = \{R\} \\
&R = \langle\ \texttt{append(Xs, Ys, Zs)} \leftarrow B,\ M\rangle \\
&M = \langle\{\texttt{Xs} \mapsto ground, \texttt{Ys} \mapsto ground, \texttt{Zs} \mapsto free\}, \\
&\qquad\quad \{\texttt{Xs} \mapsto ground, \texttt{Ys} \mapsto ground, \texttt{Zs} \mapsto ground\}\rangle
\end{aligned}
$$

Vogliamo provare che

$$\Gamma \Vdash R$$

Esempio

$$\Gamma \Vdash R$$

Usiamo **PROC**

$$\frac{\begin{array}{c} R \in \Gamma \\ R = \langle p(v_1, \ldots, v_n) \leftarrow B, M \rangle \\ \langle \Gamma, V \rangle \vdash B : M' \\ dom(M) = dom(M') = \{v_1, \ldots, v_n\} = V = uq(B) \\ \forall v \in \{v_1, \ldots, v_n\}.\ M_{init}(v) \sqsubseteq M'_{init} \\ \forall v \in \{v_1, \ldots, v_n\}.\ M'_{fin}(v) \sqsubseteq M_{fin} \end{array}}{\Gamma \Vdash R}$$

Esempio

$$\Gamma \Vdash R$$

Usiamo **PROC**

$$\frac{\begin{array}{c} R \in \Gamma \\ R = \langle p(v_1, \ldots, v_n) \leftarrow B, M \rangle \\ \langle \Gamma, V \rangle \vdash B : M' \\ dom(M) = dom(M') = \{v_1, \ldots, v_n\} = V = uq(B) \\ \forall v \in \{v_1, \ldots, v_n\}.\ M_{init}(v) \sqsubseteq M'_{init} \\ \forall v \in \{v_1, \ldots, v_n\}.\ M'_{fin}(v) \sqsubseteq M_{fin} \end{array}}{\Gamma \Vdash R}$$

Esempio

$$\Gamma \Vdash R$$

Usiamo **PROC**

$$\frac{\begin{array}{c} R \in \Gamma \\ R = \langle append(Xs, Ys, Zs) \leftarrow B, M \rangle \\ \langle \Gamma, V \rangle \vdash B : M' \\ dom(M) = dom(M') = \{Xs, Ys, Zs\} = V = uq(B) \\ \forall v \in \{Xs, Ys, Zs\}.\ M_{init}(v) \sqsubseteq M'_{init} \\ \forall v \in \{Xs, Ys, Zs\}.\ M'_{fin}(v) \sqsubseteq M_{fin} \end{array}}{\Gamma \Vdash R}$$

Esempio

$$\Gamma \Vdash R$$

Usiamo **PROC**

$$\frac{\begin{array}{c} R \in \Gamma \\ R = \langle append(Xs, Ys, Zs) \leftarrow B, M \rangle \\ \langle \Gamma, \{Xs, Ys, Zs\} \rangle \vdash B : M' \\ dom(M) = dom(M') = \{Xs, Ys, Zs\} = V = uq(B) \\ \forall v \in \{Xs, Ys, Zs\}.\ M_{init}(v) \sqsubseteq M'_{init} \\ \forall v \in \{Xs, Ys, Zs\}.\ M'_{fin}(v) \sqsubseteq M_{fin} \end{array}}{\Gamma \Vdash R}$$

Esempio

$$\Gamma \Vdash R$$

Usiamo **PROC**

$$\frac{\begin{array}{c} R \in \Gamma \\ R = \langle append(Xs, Ys, Zs) \leftarrow B, M \rangle \\ \langle \Gamma, \{Xs, Ys, Zs\} \rangle \vdash B : M' \\ dom(M) = dom(M') = \{Xs, Ys, Zs\} = V = uq(B) \\ \forall v \in \{Xs, Ys, Zs\}.\ M_{init}(v) \sqsubseteq M'_{init} \\ \forall v \in \{Xs, Ys, Zs\}.\ M'_{fin}(v) \sqsubseteq M_{fin} \end{array}}{\Gamma \Vdash R}$$

Esempio

Assumiamo che

$$M' = \langle \{ \texttt{Xs} \mapsto \ldots, \texttt{Ys} \mapsto \ldots, \texttt{Zs} \mapsto \ldots \}, \\ \{ \texttt{Xs} \mapsto \ldots, \texttt{Ys} \mapsto \ldots, \texttt{Zs} \mapsto \ldots \} \rangle$$

Passiamo a

$$\langle \Gamma, \{ Xs, Ys, Zs \} \rangle \vdash B : M'$$

Esempio

$$
\begin{array}{l}
\texttt{append(Xs, Ys, Zs)} \leftarrow \\
\quad \vee(\left.\begin{array}{l} \wedge(\texttt{Xs=[],} \\ \quad \texttt{Ys=Zs} \\ \quad), \end{array}\right\} \quad (B_1) \\
\quad\quad \left.\begin{array}{l} \exists \texttt{Xs0,Zs0,X.}(\\ \quad \wedge(\texttt{Xs = [X | Xs0],} \\ \quad\quad \texttt{append(Xs0, Ys, Zs0),} \\ \quad\quad \texttt{Zs = [X | Zs0]} \\ \quad\quad) \\ \quad) \\ ) \end{array}\right\} \quad (B_2)
\end{array}
$$

Esempio

$$\langle \Gamma, \{Xs, Ys, Zs\} \rangle \vdash \vee\ B_1, B_2 : M'$$

Usiamo **DISJ**

$$\frac{\begin{array}{c}\langle \Gamma, uq(B_1) \rangle \vdash B_1 : M_1 \\ \vdots \\ \langle \Gamma, uq(B_n) \rangle \vdash B_n : M_n \\ M = M_1 \bowtie \ldots \bowtie M_n\end{array}}{\langle \Gamma, V \rangle \vdash \vee\ B_1, \ldots, B_n : M}$$

Esempio

$$\langle \Gamma, \{Xs, Ys, Zs\} \rangle \vdash \vee\ B_1, B_2 : M'$$

Usiamo **DISJ**

$$\frac{\begin{array}{c}\langle \Gamma, uq(B_1) \rangle \vdash B_1 : M_1 \\ \langle \Gamma, uq(B_2) \rangle \vdash B_2 : M_2 \\ M' = M_1 \bowtie M_2\end{array}}{\langle \Gamma, \{Xs, Ys, Zs\} \rangle \vdash \vee\ B_1, B_2 : M'}$$

Esempio

$$\langle \Gamma, \{Xs, Ys, Zs\} \rangle \vdash \vee\ B_1, B_2 : M'$$

Usiamo **DISJ**

$$\frac{\begin{array}{c}\langle \Gamma, \{Xs, Ys, Zs\} \rangle \vdash B_1 : M_1 \\ \langle \Gamma, \{Xs, Ys, Zs\} \rangle \vdash B_2 : M_2 \\ M' = M_1 \bowtie M_2\end{array}}{\langle \Gamma, \{Xs, Ys, Zs\} \rangle \vdash \vee\ B_1, B_2 : M'}$$

Esempio

Notiamo che per definizione di $\bowtie$ abbiamo che

$$dom(M_1) = dom(M_2) = dom(M')$$

Quindi anche M_1 ed M_2 sono della forma

$$\begin{aligned}&\langle\{\mathtt{Xs} \mapsto \ldots, \mathtt{Ys} \mapsto \ldots, \mathtt{Zs} \mapsto \ldots\},\\ &\ \{\mathtt{Xs} \mapsto \ldots, \mathtt{Ys} \mapsto \ldots, \mathtt{Zs} \mapsto \ldots\}\rangle\end{aligned}$$

Passiamo a

$$\langle\Gamma, \{Xs, Ys, Zs\}\rangle \vdash B_1 : M_1$$

Esempio

$$
\begin{array}{ll}
\texttt{append(Xs, Ys, Zs)} \leftarrow & \\
\quad \vee \big(\wedge(\texttt{Xs=[]}, & (B_{11}) \\
\qquad\quad \texttt{Ys=Zs} & (B_{12}) \\
\qquad\quad), & \\
\qquad\quad \exists\texttt{Xs0,Zs0,X.}(& \\
\qquad\qquad \wedge\,(\texttt{Xs = [X | Xs0]}, & \\
\qquad\qquad\quad \texttt{append(Xs0, Ys, Zs0)}, & \\
\qquad\qquad\quad \texttt{Zs = [X | Zs0]} & \\
\qquad\qquad\quad) & \\
\qquad\quad) & \\
\quad \big) &
\end{array}
$$

Esempio

$$\langle \Gamma, \{Xs, Ys, Zs\} \rangle \vdash \wedge\ B_{11}, B_{12} : M_1$$

Usiamo **CONJ**

$$\frac{\begin{array}{c} \langle \Gamma, uq(B_1) \rangle \vdash B_1 : M_1 \\ \vdots \\ \langle \Gamma, uq(B_n) \rangle \vdash B_n : M_n \\ M = M_1 \triangleright \cdots \triangleright M_n \end{array}}{\langle \Gamma, V \rangle \vdash \wedge\ B_1, \ldots, B_n : M}$$

Esempio

$$\langle \Gamma, \{Xs, Ys, Zs\}\rangle \vdash \wedge\ B_{11}, B_{12} : M_1$$

Usiamo **CONJ**

$$\frac{\begin{array}{c}\langle \Gamma, \{Xs\}\rangle \vdash B_{11} : M_{11} \\ \langle \Gamma, \{Ys, Zs\}\rangle \vdash B_{12} : M_{12} \\ M_1 = M_{11} \triangleright M_{12}\end{array}}{\langle \Gamma, \{Xs, Ys, Zs\}\rangle \vdash \wedge\ B_{11}, B_{12} : M_1}$$

Esempio

Notiamo che per definizione di $\triangleright$ abbiamo che

$$dom(M_{11}) = dom(M_{12}) = dom(M_1)$$

Quindi anche M_{11} ed M_{12} sono della forma

$$\begin{aligned}\langle\{\mathtt{Xs} \mapsto \dots, \mathtt{Ys} \mapsto \dots, \mathtt{Zs} \mapsto \dots\},\\ \{\mathtt{Xs} \mapsto \dots, \mathtt{Ys} \mapsto \dots, \mathtt{Zs} \mapsto \dots\}\rangle\end{aligned}$$

Passiamo a

$$\langle\Gamma, \{Xs\}\rangle \vdash B_{11} : M_{11}$$

Esempio

$$
\begin{array}{l}
\texttt{append(Xs, Ys, Zs)} \leftarrow \\
\qquad \vee\, (\ \wedge(\texttt{Xs=[],} \hfill (B_{11}) \\
\qquad\qquad \texttt{Ys=Zs} \hfill (B_{12}) \\
\qquad\qquad), \\
\qquad\qquad \exists \texttt{Xs0,Zs0,X.}(\\
\qquad\qquad\quad \wedge\,(\texttt{Xs = [X | Xs0],} \\
\qquad\qquad\qquad \texttt{append(Xs0, Ys, Zs0),} \\
\qquad\qquad\qquad \texttt{Zs = [X | Zs0]} \\
\qquad\qquad\qquad) \\
\qquad\qquad) \\
\qquad)
\end{array}
$$

Esempio

$$\langle \Gamma, \{Xs\} \rangle \vdash \texttt{Xs=[]} : M_{11}$$

Usiamo **UNIFY-VF**

$$\frac{\begin{array}{c} v \notin \{v_1, \ldots, v_n\} \\ i = M_{init}(v) \\ \bar{i} = \langle M_{init}(v_1), \ldots, M_{init}(v_n) \rangle \\ abs\_unify\_inst\_func(i, f, \bar{i}, i', \langle i'_1, \ldots, i'_n \rangle) \\ M_{fin} = M_{init} \oplus \{v \mapsto i', v_1 \mapsto i'_1, \ldots, v_n \mapsto i'_n\} \end{array}}{\langle \Gamma, V \rangle \vdash v = f(v_1, \ldots, v_n) : M}$$

Esempio

$$\langle \Gamma, \{Xs\} \rangle \vdash \texttt{Xs=[]} : M_{11}$$

Usiamo **UNIFY-VF**

$$\frac{\begin{array}{c} Xs \notin \emptyset \\ i = M_{11_{init}}(Xs) \\ \overline{i} = \langle\rangle \\ abs\_unify\_inst\_func(i, [], \overline{i}, i', \langle\rangle) \\ M_{11_{fin}} = M_{11_{init}} \oplus \{Xs \mapsto i'\} \end{array}}{\langle \Gamma, \{Xs\} \rangle \vdash \texttt{Xs=[]} : M_{11}}$$

Esempio

$$\langle \Gamma, \{Xs\} \rangle \vdash \texttt{Xs=[]} : M_{11}$$

Usiamo **UNIFY-VF**

$$\frac{\begin{array}{c} Xs \notin \emptyset \\ i = M_{11_{init}}(Xs) \\ \overline{i} = \langle\rangle \\ abs\_unify\_inst\_func(i, [], \overline{i}, i', \langle\rangle) \\ M_{11_{fin}} = M_{11_{init}} \oplus \{Xs \mapsto i'\} \end{array}}{\langle \Gamma, \{Xs\} \rangle \vdash \texttt{Xs=[]} : M_{11}}$$

Esempio

Assumiamo

$$i = M_{11_{init}}(Xs) = bound(\{\ [],\ \dots\ \})$$

Allora vale che

$$abs\_unify\_inst\_func(i, [], \bar{i}, i', \langle\rangle)$$

se

$$i' = bound(\{\ []\ \})$$

$$\langle \Gamma, \{Xs\} \rangle \vdash \texttt{Xs=[]} : M_{11}$$

Usiamo **UNIFY-VF**

$$\frac{\begin{array}{c} Xs \notin \emptyset \\ i = bound(\{\ [],\ \dots\ \}) = M_{11_{init}}(Xs) \\ \bar{i} = \langle\rangle \\ abs\_unify\_inst\_func(i, [], \bar{i}, bound(\{\ []\ \}), \langle\rangle) \\ M_{11_{fin}} = M_{11_{init}} \oplus \{Xs \mapsto bound(\{\ []\ \})\} \end{array}}{\langle \Gamma, \{Xs\} \rangle \vdash \texttt{Xs=[]} : M_{11}}$$

Esempio

Abbiamo quindi che M_{11} è della forma

$$\begin{aligned}
&\langle\{\texttt{Xs} \mapsto bound(\{\ [],\ \dots\ \}), \texttt{Ys} \mapsto I_{Ys_{11}}, \texttt{Zs} \mapsto I_{Zs_{11}}\},\\
&\ \{\texttt{Xs} \mapsto bound(\{\ []\ \}), \qquad\ \ \texttt{Ys} \mapsto I_{Ys_{11}}, \texttt{Zs} \mapsto I_{Zs_{11}}\}\rangle
\end{aligned}$$

Esempio

Torniamo a

$$\langle \Gamma, \{Xs, Ys, Zs\} \rangle \vdash \wedge\ B_{11}, B_{12} : M_1$$

Usiamo **CONJ**

$$\frac{\begin{array}{c}\langle \Gamma, \{Xs\} \rangle \vdash B_{11} : M_{11} \\ \langle \Gamma, \{Ys, Zs\} \rangle \vdash B_{12} : M_{12} \\ M_1 = M_{11} \triangleright M_{12}\end{array}}{\langle \Gamma, \{Xs, Ys, Zs\} \rangle \vdash \wedge\ B_{11}, B_{12} : M_1}$$

Esempio

$$\langle \Gamma, \{Xs, Ys, Zs\}\rangle \vdash \wedge\ B_{11}, B_{12} : M_1$$

Usiamo **CONJ**

$$\frac{\begin{array}{c}\langle \Gamma, \{Xs\}\rangle \vdash B_{11} : M_{11} \\ \langle \Gamma, \{Ys, Zs\}\rangle \vdash B_{12} : M_{12} \\ M_1 = M_{11} \triangleright M_{12}\end{array}}{\langle \Gamma, \{Xs, Ys, Zs\}\rangle \vdash \wedge\ B_{11}, B_{12} : M_1}$$

Esempio

Passiamo a

$$\langle \Gamma, \{Ys, Zs\} \rangle \vdash B_{12} : M_{12}$$

Ovvero a

$$\langle \Gamma, \{Ys, Zs\} \rangle \vdash Ys = Zs : M_{12}$$

Esempio

$$\langle \Gamma, \{Ys, Zs\} \rangle \vdash Ys = Zs : M_{12}$$

Usiamo **UNIFY-VV**

$$\frac{\begin{array}{c} abs\_unify\_inst(M_{init}(v), M_{init}(v'), i) \\ M_{fin} = M_{init} \oplus \{v \mapsto i, v' \mapsto i\} \end{array}}{\langle \Gamma, V \rangle \vdash v = v' : M}$$

Esempio

$$\langle \Gamma, \{Ys, Zs\} \rangle \vdash Ys = Zs : M_{12}$$

Usiamo **UNIFY-VV**

$$\frac{\begin{array}{c} abs\_unify\_inst(M_{12_{init}}(Ys), M_{12_{init}}(Zs), i) \\ M_{12_{fin}} = M_{12_{init}} \oplus \{Ys \mapsto i, Zs \mapsto i\} \end{array}}{\langle \Gamma, \{Ys, Zs\} \rangle \vdash Ys = Zs : M_{12}}$$

Esempio

Assumiamo

$$M_{12_{init}}(Ys) = bound(\{\ \dots\ \})$$
$$M_{12_{init}}(Zs) = free$$

Allora vale che

$$abs\_unify\_inst(M_{12_{init}}(Ys), M_{12_{init}}(Zs), i)$$

se

$$i = M_{12_{init}}(Ys)$$

Esempio

$$\langle \Gamma, \{Ys, Zs\} \rangle \vdash Ys = Zs : M_{12}$$

Usiamo **UNIFY-VV**

$$\frac{\begin{array}{c} abs\_unify\_inst(M_{12_{init}}(Ys), M_{12_{init}}(Zs), i) \\ M_{12_{fin}} = M_{12_{init}} \oplus \{Ys \mapsto i, Zs \mapsto i\} \end{array}}{\langle \Gamma, \{Ys, Zs\} \rangle \vdash Ys = Zs : M_{12}}$$

Esempio

Abbiamo quindi che M_{12} è della forma

$$\langle \{ \mathtt{Xs} \mapsto I_{Xs_{12}}, \mathtt{Ys} \mapsto bound(\{\ I_{A_{12}}\ \}), \mathtt{Zs} \mapsto free \},$$
$$\{ \mathtt{Xs} \mapsto I_{Xs_{12}}, \mathtt{Ys} \mapsto bound(\{\ I_{A_{12}}\ \}), \mathtt{Zs} \mapsto bound(\{\ I_{A_{12}}\ \}) \} \rangle$$

Esempio

Torniamo a

$$\langle \Gamma, \{Xs, Ys, Zs\} \rangle \vdash \wedge\ B_{11}, B_{12} : M_1$$

Usiamo **CONJ**

$$\frac{\begin{array}{c} \langle \Gamma, \{Xs\} \rangle \vdash B_{11} : M_{11} \\ \langle \Gamma, \{Ys, Zs\} \rangle \vdash B_{12} : M_{12} \\ M_1 = M_{11} \triangleright M_{12} \end{array}}{\langle \Gamma, \{Xs, Ys, Zs\} \rangle \vdash \wedge\ B_{11}, B_{12} : M_1}$$

Esempio

$$M_1 = M_{11} \triangleright M_{12}$$

Richiede che

$$M_{11_{fin}} = M_{12_{init}}$$

e

$$M_1 = \langle M_{11_{init}},\ M_{12_{fin}} \rangle$$

Esempio

Quindi

$$\begin{aligned} M_{11} = & \langle \{ \mathtt{Xs} \mapsto bound(\{\ [],\ \ldots\ \}), \mathtt{Ys} \mapsto I_{Ys_{11}}, \mathtt{Zs} \mapsto I_{Zs_{11}} \}, \\ & \{ \mathtt{Xs} \mapsto bound(\{\ []\ \}), \qquad \mathtt{Ys} \mapsto I_{Ys_{11}}, \mathtt{Zs} \mapsto I_{Zs_{11}} \} \rangle \end{aligned}$$

$$\begin{aligned} M_{12} = & \langle \{ \mathtt{Xs} \mapsto I_{Xs_{12}}, \mathtt{Ys} \mapsto bound(\{\ I_{A_{12}}\ \}), \mathtt{Zs} \mapsto free \}, \\ & \{ \mathtt{Xs} \mapsto I_{Xs_{12}}, \mathtt{Ys} \mapsto bound(\{\ I_{A_{12}}\ \}), \mathtt{Zs} \mapsto bound(\{\ I_{A_{12}}\ \}) \} \rangle \end{aligned}$$

Esempio

Quindi

$$
\begin{aligned}
M_{11} =& \langle \{\mathtt{Xs} \mapsto \mathit{bound}(\{\ [],\ \ldots\ \}), \mathtt{Ys} \mapsto I_{Ys_{11}}, \mathtt{Zs} \mapsto I_{Zs_{11}}\},\\
& \{\mathtt{Xs} \mapsto \mathit{bound}(\{\ []\ \}), \qquad \mathtt{Ys} \mapsto I_{Ys_{11}}, \mathtt{Zs} \mapsto I_{Zs_{11}}\}\rangle
\end{aligned}
$$

$$
\begin{aligned}
M_{12} =& \langle \{\mathtt{Xs} \mapsto I_{Xs_{12}}, \mathtt{Ys} \mapsto \mathit{bound}(\{\ I_{A_{12}}\ \}), \mathtt{Zs} \mapsto \mathit{free}\},\\
& \{\mathtt{Xs} \mapsto I_{Xs_{12}}, \mathtt{Ys} \mapsto \mathit{bound}(\{\ I_{A_{12}}\ \}), \mathtt{Zs} \mapsto \mathit{bound}(\{\ I_{A_{12}}\ \})\}\rangle
\end{aligned}
$$

sono

$$
\begin{aligned}
M_{11} =& \langle \{\mathtt{Xs} \mapsto \mathit{bound}(\{\ [],\ \ldots\ \}), \mathtt{Ys} \mapsto \mathit{bound}(\{\ I_{A_{12}}\ \}), \mathtt{Zs} \mapsto \mathit{free}\},\\
& \{\mathtt{Xs} \mapsto \mathit{bound}(\{\ []\ \}), \qquad \mathtt{Ys} \mapsto \mathit{bound}(\{\ I_{A_{12}}\ \}), \mathtt{Zs} \mapsto \mathit{free}\}\rangle
\end{aligned}
$$

$$
\begin{aligned}
M_{12} =& \langle \{\mathtt{Xs} \mapsto \mathit{bound}(\{\ []\ \}), \mathtt{Ys} \mapsto \mathit{bound}(\{\ I_{A_{12}}\ \}), \mathtt{Zs} \mapsto \mathit{free}\},\\
& \{\mathtt{Xs} \mapsto \mathit{bound}(\{\ []\ \}), \mathtt{Ys} \mapsto \mathit{bound}(\{\ I_{A_{12}}\ \}), \mathtt{Zs} \mapsto \mathit{bound}(\{\ I_{A_{12}}\ \})\}\rangle
\end{aligned}
$$

Esempio

$$\begin{aligned} M_1 = &\langle \{ \mathtt{Xs} \mapsto bound(\{\ [],\ \dots\ \}), \mathtt{Ys} \mapsto bound(\{\ I_{A_{12}}\ \}), \mathtt{Zs} \mapsto free \}, \\ &\{ \mathtt{Xs} \mapsto bound(\{\ []\ \}), \mathtt{Ys} \mapsto bound(\{\ I_{A_{12}}\ \}), \mathtt{Zs} \mapsto bound(\{\ I_{A_{12}}\ \}) \} \rangle \end{aligned}$$

Esempio

Torniamo a

$$\langle \Gamma, \{Xs, Ys, Zs\} \rangle \vdash \wedge\; B_{11}, B_{12} : M_1$$

Usiamo **CONJ**

$$\frac{\begin{array}{c} \langle \Gamma, \{Xs\} \rangle \vdash B_{11} : M_{11} \\ \langle \Gamma, \{Ys, Zs\} \rangle \vdash B_{12} : M_{12} \\ M_1 = M_{11} \triangleright M_{12} \end{array}}{\langle \Gamma, \{Xs, Ys, Zs\} \rangle \vdash \wedge\; B_{11}, B_{12} : M_1}$$

Esempio

Torniamo a

$$\langle \Gamma, \{Xs, Ys, Zs\} \rangle \vdash \vee\ B_1, B_2 : M'$$

Usiamo **DISJ**

$$\frac{\begin{array}{c} \langle \Gamma, \{Xs, Ys, Zs\} \rangle \vdash B_1 : M_1 \\ \langle \Gamma, \{Xs, Ys, Zs\} \rangle \vdash B_2 : M_2 \\ M' = M_1 \bowtie M_2 \end{array}}{\langle \Gamma, \{Xs, Ys, Zs\} \rangle \vdash \vee\ B_1, B_2 : M'}$$

Esempio

Dalla definizione di $\bowtie$ sappiamo che $M_{1_{init}} = M_{2_{init}}$.

Quindi M_2 sarà della forma

$$\langle \{\texttt{Xs} \mapsto bound(\{\ [],\ \ldots\ \}), \texttt{Ys} \mapsto bound(\{\ I_{A_{12}}\ \}), \texttt{Zs} \mapsto free\},$$
$$\{\texttt{Xs} \mapsto \ldots,\ \texttt{Ys} \mapsto \ldots,\ \texttt{Zs} \mapsto \ldots\}\rangle$$

Passiamo a

$$\langle \Gamma, \{Xs, Ys, Zs\} \rangle \vdash B_2 : M_2$$

Esempio

```
append(Xs, Ys, Zs) ←
        ∨ ( ∧(Xs=[],
               Ys=Zs
             ),
             ∃Xs0,Zs0,X.(
                 ∧(Xs = [X | Xs0],
                    append(Xs0, Ys, Zs0),
                    Zs = [X | Zs0]
                  )
             )
          )
```

(B_{21})

Esempio

$$\langle \Gamma, \{Xs, Ys, Zs\} \rangle \vdash \exists \texttt{Xs0,Zs0,X.}\ B_{21} : M_2$$

Usiamo **SOME**

$$\frac{\begin{array}{c}\langle \Gamma, V \backslash \{v_1, \ldots, v_n\} \rangle \vdash B : M' \\ M = M' \ominus \{v_1, \ldots, v_n\}\end{array}}{\langle \Gamma, V \rangle \vdash \exists v_1, \ldots, v_n.\ B : M}$$

Esempio

$$\langle \Gamma, \{Xs, Ys, Zs\} \rangle \vdash \exists \texttt{Xs0,Zs0,X.}\ B_{21} : M_2$$

Usiamo **SOME**

$$\frac{\begin{array}{c}\langle \Gamma, \{Xs, Ys, Zs\} \backslash \{Xs0, Zs0, X\} \rangle \vdash B_{21} : M_{21} \\ M_2 = M_{21} \ominus \{Xs0, Zs0, X\}\end{array}}{\langle \Gamma, \{Xs, Ys, Zs\} \rangle \vdash \exists \texttt{Xs0,Zs0,X.}\ B_{21} : M_2}$$

Esempio

$$\langle \Gamma, \{Xs, Ys, Zs\} \rangle \vdash \exists \texttt{Xs0,Zs0,X.}\ B_{21} : M_2$$

Usiamo **SOME**

$$\frac{\begin{array}{c}\langle \Gamma, \{Xs, Ys, Zs\} \rangle \vdash B_{21} : M_{21} \\ M_2 = M_{21} \ominus \{Xs0, Zs0, X\}\end{array}}{\langle \Gamma, \{Xs, Ys, Zs\} \rangle \vdash \exists \texttt{Xs0,Zs0,X.}\ B_{21} : M_2}$$

Esempio

M_{21} è della forma

$$\langle \{ \texttt{Xs} \mapsto bound(\{ \ [], \ \dots \ \}), \texttt{Ys} \mapsto bound(\{ \ I_{A_{12}} \ \}), \texttt{Zs} \mapsto free,$$
$$Xs0 \mapsto \dots, Zs0 \mapsto \dots, X \mapsto \dots \},$$
$$\{ \texttt{Xs} \mapsto \dots, \ \texttt{Ys} \mapsto \dots, \ \texttt{Zs} \mapsto \dots, Xs0 \mapsto \dots, Zs0 \mapsto \dots, X \mapsto \dots \} \rangle$$

Passiamo a

$$\langle \Gamma, \{Xs, Ys, Zs\} \rangle \vdash B_{21} : M_{21}$$

Esempio

$$
\begin{array}{ll}
\texttt{append(Xs, Ys, Zs)} \leftarrow & \\
\quad \vee\,(\ \wedge(\texttt{Xs=[],} & \\
\qquad \texttt{Ys=Zs} & \\
\qquad), & \\
\qquad \exists\texttt{Xs0,Zs0,X.}(& \\
\qquad\quad \wedge\,(\texttt{Xs = [X | Xs0],} & (B_{211}) \\
\qquad\qquad \texttt{append(Xs0, Ys, Zs0),} & (B_{212}) \\
\qquad\qquad \texttt{Zs = [X | Zs0]} & (B_{213}) \\
\qquad\quad) & \\
\qquad) & \\
\quad) &
\end{array}
$$

Esempio

$$\langle \Gamma, \{Xs, Ys, Zs\}\rangle \vdash \wedge\ B_{211}, B_{212}, B_{213} : M_{21}$$

Usiamo **CONJ**

$$\frac{\begin{array}{c}\langle \Gamma, uq(B_1)\rangle \vdash B_1 : M_1 \\ \vdots \\ \langle \Gamma, uq(B_n)\rangle \vdash B_n : M_n \\ M = M_1 \triangleright \cdots \triangleright M_n\end{array}}{\langle \Gamma, V\rangle \vdash \wedge\ B_1, \ldots, B_n : M}$$

Esempio

$$\langle \Gamma, \{Xs, Ys, Zs\} \rangle \vdash \wedge\ B_{211}, B_{212}, B_{213} : M_{21}$$

Usiamo **CONJ**

$$\frac{\begin{array}{c} \langle \Gamma, uq(B_{211}) \rangle \vdash B_{211} : M_{211} \\ \langle \Gamma, uq(B_{212}) \rangle \vdash B_{212} : M_{212} \\ \langle \Gamma, uq(B_{213}) \rangle \vdash B_{213} : M_{213} \\ M_{21} = M_{211} \triangleright M_{212} \triangleright M_{213} \end{array}}{\langle \Gamma, \{Xs, Ys, Zs\} \rangle \vdash \wedge\ B_{211}, B_{212}, B_{213} : M_{21}}$$

Esempio

$$\langle \Gamma, \{Xs, Ys, Zs\} \rangle \vdash \wedge\ B_{211}, B_{212}, B_{213} : M_{21}$$

Usiamo **CONJ**

$$\frac{\begin{array}{c}\langle \Gamma, \{X, Xs, Xs0\} \rangle \vdash B_{211} : M_{211} \\ \langle \Gamma, \{Ys, Zs0, Xs0\} \rangle \vdash B_{212} : M_{212} \\ \langle \Gamma, \{X, Zs, Zs0\} \rangle \vdash B_{213} : M_{213} \\ M_{21} = M_{211} \triangleright M_{212} \triangleright M_{213}\end{array}}{\langle \Gamma, \{Xs, Ys, Zs\} \rangle \vdash \wedge\ B_{211}, B_{212}, B_{213} : M_{21}}$$

Esempio

Per $\triangleright$ abbiamo che $M_{211_{init}} = M_{21_{init}}$

Abbiamo quindi che M_{211} è della forma

$$\begin{array}{l}\langle\{\mathtt{Xs} \mapsto bound(\{\ [],\ \ldots\ \}), \mathtt{Ys} \mapsto bound(\{\ I_{A_{12}}\ \}), \mathtt{Zs} \mapsto free, \\ \qquad Xs0 \mapsto \ldots, Zs0 \mapsto \ldots, X \mapsto \ldots\}, \\ \{\mathtt{Xs} \mapsto \ldots,\ \mathtt{Ys} \mapsto \ldots,\ \mathtt{Zs} \mapsto \ldots, Xs0 \mapsto \ldots, Zs0 \mapsto \ldots, X \mapsto \ldots\}\rangle\end{array}$$

Passiamo a

$$\langle\Gamma, \{X, Xs, Xs0\}\rangle \vdash B_{211} : M_{211}$$

Esempio

```
append(Xs, Ys, Zs) ←
        ∨ ( ∧(Xs=[],
                Ys=Zs
              ),
              ∃Xs0,Zs0,X.(
                  ∧ (Xs = [X | Xs0],                  (B211)
                     append(Xs0, Ys, Zs0),            (B212)
                     Zs = [X | Zs0]                   (B213)
                    )
              )
          )
```

Esempio

$$\langle \Gamma, \{X, Xs, Xs0\}\rangle \vdash Xs = [X|Xs0] : M_{211}$$

Usiamo **UNIFY-VF**

$$\frac{\begin{array}{c} v \notin \{v_1, \ldots, v_n\} \\ i = M_{init}(v) \\ \bar{i} = \langle M_{init}(v_1), \ldots, M_{init}(v_n)\rangle \\ abs\_unify\_inst\_func(i, f, \bar{i}, i', \langle i'_1, \ldots, i'_n\rangle) \\ M_{fin} = M_{init} \oplus \{v \mapsto i', v_1 \mapsto i'_1, \ldots, v_n \mapsto i'_n\} \end{array}}{\langle \Gamma, V\rangle \vdash v = f(v_1, \ldots, v_n) : M}$$

Esempio

$$\langle \Gamma, \{X, Xs, Xs0\} \rangle \vdash Xs = [X|Xs0] : M_{211}$$

Usiamo **UNIFY-VF**

$$\frac{\begin{array}{c} Xs \notin \{X, Xs0\} \\ i = M_{211_{init}}(Xs) \\ \overline{i} = \langle M_{211_{init}}(X), M_{211_{init}}(Xs0) \rangle \\ abs\_unify\_inst\_func(i, [||], \overline{i}, i', \langle i'_1, i'_2 \rangle) \\ M_{211_{fin}} = M_{211_{init}} \oplus \{Xs \mapsto i', X \mapsto i'_1, Xs0 \mapsto i'_2\} \end{array}}{\langle \Gamma, \{X, Xs, Xs0\} \rangle \vdash Xs = [X|Xs0] : M_{211}}$$

Esempio

Assumiamo

$$M_{211_{init}}(Xs) = bound(\{[], [ground|ground]\})$$
$$M_{211_{init}}(X) = M_{211_{init}}(Xs0) = free$$

Abbiamo quindi

$$abs\_unify\_inst\_func(i, [|], \bar{i}, i', \langle i'_1, i'_2 \rangle)$$

se

$$i' = bound(\{[ground|ground]\})$$
$$i'_1 = i'_2 = ground$$

Esempio

Abbiamo quindi che M_{211} è della forma

$$
\begin{aligned}
\langle\{&\texttt{Xs} \mapsto bound(\{[],\ [ground|ground]\}),\\
&\qquad \texttt{Ys} \mapsto bound(\{\ I_{A_{211}}\ \}), \texttt{Zs} \mapsto free,\\
&\qquad Xs0 \mapsto free,\ Zs0 \mapsto I_{Zs0_{211}},\ X \mapsto free\},\\
\{&\texttt{Xs} \mapsto bound(\{[ground|ground]\}),\\
&\qquad \texttt{Ys} \mapsto bound(\{\ I_{A_{211}}\ \}), \texttt{Zs} \mapsto free,\\
&\qquad Xs0 \mapsto ground,\ Zs0 \mapsto I_{Zs0_{211}},\ X \mapsto ground\}\rangle
\end{aligned}
$$

$$\langle \Gamma, \{X, Xs, Xs0\} \rangle \vdash Xs = [X|Xs0] : M_{211}$$

Usiamo **UNIFY-VF**

$$\frac{\begin{array}{c} Xs \notin \{X, Xs0\} \\ i = M_{211_{init}}(Xs) \\ \bar{i} = \langle M_{211_{init}}(X), M_{211_{init}}(Xs0) \rangle \\ abs\_unify\_inst\_func(i, [||], \bar{i}, i', \langle i'_1, i'_2 \rangle) \\ M_{211_{fin}} = M_{211_{init}} \oplus \{Xs \mapsto i', X \mapsto i'_1, Xs0 \mapsto i'_2\} \end{array}}{\langle \Gamma, \{X, Xs, Xs0\} \rangle \vdash Xs = [X|Xs0] : M_{211}}$$

Esempio

Torniamo a

$$\langle \Gamma, \{Xs, Ys, Zs\} \rangle \vdash \wedge\ B_{211}, B_{212}, B_{213} : M_{21}$$

Usiamo **CONJ**

$$\frac{\begin{array}{c} \langle \Gamma, \{X, Xs, Xs0\} \rangle \vdash B_{211} : M_{211} \\ \langle \Gamma, \{Ys, Zs0, Xs0\} \rangle \vdash B_{212} : M_{212} \\ \langle \Gamma, \{X, Zs, Zs0\} \rangle \vdash B_{213} : M_{213} \\ M_{21} = M_{211} \triangleright M_{212} \triangleright M_{213} \end{array}}{\langle \Gamma, \{Xs, Ys, Zs\} \rangle \vdash \wedge\ B_{211}, B_{212}, B_{213} : M_{21}}$$

Esempio

Per $\triangleright$ abbiamo che $M_{212_{init}} = M_{211_{fin}}$

Abbiamo quindi che M_{212} è della forma

$$\begin{aligned}
\langle\{\texttt{Xs} &\mapsto bound(\{[ground|ground]\}),\\
&\texttt{Ys} \mapsto bound(\{\ I_{A_{211}}\ \}), \texttt{Zs} \mapsto free,\\
&Xs0 \mapsto ground,\ Zs0 \mapsto I_{Zs0_{211}},\ X \mapsto ground\},\\
\{\texttt{Xs} &\mapsto bound(\{[ground|ground]\}),\\
&\texttt{Ys} \mapsto bound(\{\ \ldots\}), \texttt{Zs} \mapsto free,\\
&Xs0 \mapsto ground,\ Zs0 \mapsto \ldots,\ X \mapsto ground\}\rangle
\end{aligned}$$

Passiamo a

$$\langle\Gamma, \{Ys, Zs0, Xs0\}\rangle \vdash B_{212} : M_{212}$$

Esempio

$$
\begin{array}{ll}
\texttt{append(Xs, Ys, Zs)} \leftarrow & \\
\quad \vee\,(\ \wedge(\texttt{Xs=[]}, & \\
\qquad \texttt{Ys=Zs} & \\
\qquad), & \\
\qquad \exists\texttt{Xs0,Zs0,X.}(& \\
\qquad\quad \wedge\,(\texttt{Xs = [X | Xs0]}, & (B_{211}) \\
\qquad\qquad \texttt{append(Xs0, Ys, Zs0)}, & (B_{212}) \\
\qquad\qquad \texttt{Zs = [X | Zs0]} & (B_{213}) \\
\qquad\quad) & \\
\qquad) & \\
\quad) &
\end{array}
$$

$$\langle \Gamma, \{Ys, Zs0, Xs0\}\rangle \vdash \texttt{append(Xs0, Ys, Zs0)} : M_{212}$$

Usiamo **CALL**

$$\frac{\begin{array}{c}\langle p(v'_1, \ldots, v'_n) \leftarrow B,\ M'\rangle \in \Gamma \\ \forall j \in \{1, \ldots, n\}.\ M_{init}(v_j) \sqsubseteq M'_{init}(v'_j) \\ M_{fin} = M_{init} \oplus \{v_j \mapsto i_j \mid j \in \{1, \ldots, n\} \wedge \\ \qquad\qquad i_j = M'_{fin}(v'_j) \curlywedge M_{init}(v_j)\}\end{array}}{\langle \Gamma, V\rangle \vdash p(v_1, \ldots, v_n) : M}$$

Esempio

$$\langle \Gamma, \{Ys, Zs0, Xs0\}\rangle \vdash \texttt{append(Xs0, Ys, Zs0)} : M_{212}$$

Usiamo **CALL**

$$\frac{\begin{array}{c}\langle \texttt{append(Xs, Ys, Zs)} \leftarrow B,\ M\rangle \in \Gamma \\ M_{212_{init}}(Xs0) \sqsubseteq M_{init}(Xs) \wedge M_{212_{init}}(Ys) \sqsubseteq M_{init}(Ys) \\ \wedge\ M_{212_{init}}(Zs0) \sqsubseteq M_{init}(Zs) \\ \begin{aligned} M_{212_{fin}} = M_{212_{init}} \oplus \{ & Xs0 \mapsto M_{fin}(Xs) \curlywedge M_{212_{init}}(Xs0), \\ & Ys \mapsto M_{fin}(Ys) \curlywedge M_{212_{init}}(Ys), \\ & Zs0 \mapsto M_{fin}(Zs) \curlywedge M_{212_{init}}(Zs)\} \end{aligned}\end{array}}{\langle \Gamma, \{Ys, Zs0, Xs0\}\rangle \vdash \texttt{append(Xs0, Ys, Zs0)} : M_{212}}$$

Esempio

$$\langle \Gamma, \{Ys, Zs0, Xs0\}\rangle \vdash \texttt{append(Xs0, Ys, Zs0)} : M_{212}$$

Usiamo **CALL**

$$\frac{\begin{array}{c}\langle \texttt{append(Xs, Ys, Zs)} \leftarrow B,\ M\rangle \in \Gamma \\ M_{212_{init}}(Xs0) \sqsubseteq M_{init}(Xs) \wedge M_{212_{init}}(Ys) \sqsubseteq M_{init}(Ys) \\ \wedge\ M_{212_{init}}(Zs0) \sqsubseteq M_{init}(Zs) \\ \begin{aligned} M_{212_{fin}} = M_{212_{init}} \oplus \{ & Xs0 \mapsto M_{fin}(Xs) \curlywedge M_{212_{init}}(Xs0), \\ & Ys \mapsto M_{fin}(Ys) \curlywedge M_{212_{init}}(Ys), \\ & Zs0 \mapsto M_{fin}(Zs) \curlywedge M_{212_{init}}(Zs)\} \end{aligned}\end{array}}{\langle \Gamma, \{Ys, Zs0, Xs0\}\rangle \vdash \texttt{append(Xs0, Ys, Zs0)} : M_{212}}$$

Esempio

Ricordiamo che

$$M = \langle \{\texttt{Xs} \mapsto ground, \texttt{Ys} \mapsto ground, \texttt{Zs} \mapsto free\},$$
$$\{\texttt{Xs} \mapsto ground, \texttt{Ys} \mapsto ground, \texttt{Zs} \mapsto ground\}\rangle$$

Esempio

$$\langle \Gamma, \{Ys, Zs0, Xs0\} \rangle \vdash \texttt{append(Xs0, Ys, Zs0)} : M_{212}$$

Usiamo **CALL**

$$\frac{\begin{array}{c} \langle \texttt{append(Xs, Ys, Zs)} \leftarrow B,\ M \rangle \in \Gamma \\ ground \sqsubseteq ground \wedge bound(I_{A_{211}}) \sqsubseteq ground \\ \wedge\ I_{Zs0_{211}} \sqsubseteq free \\ M212_{fin} = M212_{init} \oplus \{Xs0 \mapsto ground \curlywedge ground, \\ Ys \mapsto ground \curlywedge bound(I_{A_{211}}), \\ Zs0 \mapsto ground \curlywedge free\} \end{array}}{\langle \Gamma, \{Ys, Zs0, Xs0\} \rangle \vdash \texttt{append(Xs0, Ys, Zs0)} : M_{212}}$$

Esempio

$$bound(I_{A_{211}}) \sqsubseteq ground$$
$$I_{Zs0_{211}} \sqsubseteq free$$

Assumiamo

$$I_{Zs0_{211}} = free$$
$$bound(I_{A_{211}}) = ground$$

Esempio

$$\langle \Gamma, \{Ys, Zs0, Xs0\} \rangle \vdash \texttt{append(Xs0, Ys, Zs0)} : M_{212}$$

Usiamo **CALL**

$$\frac{\begin{array}{c}\langle \texttt{append(Xs, Ys, Zs)} \leftarrow B,\ M \rangle \in \Gamma \\ ground \sqsubseteq ground \wedge ground \sqsubseteq ground \\ \wedge\ free \sqsubseteq free \\ M_{212_{fin}} = M_{212_{init}} \oplus \{Xs0 \mapsto ground \curlywedge ground, \\ Ys \mapsto ground \curlywedge ground, \\ Zs0 \mapsto ground \curlywedge free\}\end{array}}{\langle \Gamma, \{Ys, Zs0, Xs0\} \rangle \vdash \texttt{append(Xs0, Ys, Zs0)} : M_{212}}$$

Esempio

$$\langle \Gamma, \{Ys, Zs0, Xs0\} \rangle \vdash \texttt{append(Xs0, Ys, Zs0)} : M_{212}$$

Usiamo **CALL**

$$\frac{\begin{array}{c}\langle \texttt{append(Xs, Ys, Zs)} \leftarrow B,\ M \rangle \in \Gamma \\ ground \sqsubseteq ground \wedge ground \sqsubseteq ground \\ \wedge\ free \sqsubseteq free \\ M212_{fin} = M212_{init} \oplus \{Xs0 \mapsto ground \curlywedge ground, \\ Ys \mapsto ground \curlywedge ground, \\ Zs0 \mapsto ground \curlywedge free\}\end{array}}{\langle \Gamma, \{Ys, Zs0, Xs0\} \rangle \vdash \texttt{append(Xs0, Ys, Zs0)} : M_{212}}$$

Esempio

$$
\begin{aligned}
M_{212} =\langle \{ & \texttt{Xs} \mapsto bound(\{[ground|ground]\}), \\
& \texttt{Ys} \mapsto ground, \texttt{Zs} \mapsto free, \\
& Xs0 \mapsto ground,\ Zs0 \mapsto free,\ X \mapsto ground\}, \\
\{ & \texttt{Xs} \mapsto bound(\{[ground|ground]\}), \\
& \texttt{Ys} \mapsto ground, \texttt{Zs} \mapsto free, \\
& Xs0 \mapsto ground,\ Zs0 \mapsto ground,\ X \mapsto ground\}\rangle
\end{aligned}
$$

Esempio

Torniamo a

$$\langle \Gamma, \{Xs, Ys, Zs\} \rangle \vdash \wedge\ B_{211}, B_{212}, B_{213} : M_{21}$$

Usiamo **CONJ**

$$\frac{\begin{array}{c} \langle \Gamma, \{X, Xs, Xs0\} \rangle \vdash B_{211} : M_{211} \\ \langle \Gamma, \{Ys, Zs0, Xs0\} \rangle \vdash B_{212} : M_{212} \\ \langle \Gamma, \{X, Zs, Zs0\} \rangle \vdash B_{213} : M_{213} \\ M_{21} = M_{211} \triangleright M_{212} \triangleright M_{213} \end{array}}{\langle \Gamma, \{Xs, Ys, Zs\} \rangle \vdash \wedge\ B_{211}, B_{212}, B_{213} : M_{21}}$$

Esempio

Per $\triangleright$ abbiamo che $M_{213_{init}} = M_{212_{fin}}$

Abbiamo quindi che M_{213} è della forma

$$\begin{aligned}
\langle\{\texttt{Xs} \mapsto\ & bound(\{[ground|ground]\}), \\
& \texttt{Ys} \mapsto ground, \texttt{Zs} \mapsto free, \\
& Xs0 \mapsto ground,\ Zs0 \mapsto ground,\ X \mapsto ground\}, \\
\{\texttt{Xs} \mapsto\ & \dots, \\
& \texttt{Ys} \mapsto \dots, \texttt{Zs} \mapsto \dots, \\
& Xs0 \mapsto \dots,\ Zs0 \mapsto \dots,\ X \mapsto \dots\}\rangle
\end{aligned}$$

Passiamo a

$$\langle\Gamma, \{X, Zs, Zs0\}\rangle \vdash B_{213} : M_{213}$$

Esempio

$$
\begin{array}{ll}
\texttt{append(Xs, Ys, Zs)} \leftarrow & \\
\quad \vee\, (\ \wedge(\texttt{Xs=[]}, & \\
\qquad \texttt{Ys=Zs} & \\
\qquad), & \\
\qquad \exists \texttt{Xs0,Zs0,X.}(& \\
\qquad\quad \wedge\, (\texttt{Xs = [X | Xs0]}, & (B_{211}) \\
\qquad\qquad \texttt{append(Xs0, Ys, Zs0)}, & (B_{212}) \\
\qquad\qquad \texttt{Zs = [X | Zs0]} & (B_{213}) \\
\qquad\quad) & \\
\qquad) & \\
\quad) &
\end{array}
$$

Esempio

$$\langle \Gamma, \{X, Zs, Zs0\}\rangle \vdash \texttt{Zs = [X | Zs0]} : M_{213}$$

Usiamo **UNIFY-VF**

$$\frac{\begin{array}{c} v \notin \{v_1, \ldots, v_n\} \\ i = M_{init}(v) \\ \bar{i} = \langle M_{init}(v_1), \ldots, M_{init}(v_n)\rangle \\ abs\_unify\_inst\_func(i, f, \bar{i}, i', \langle i'_1, \ldots, i'_n\rangle) \\ M_{fin} = M_{init} \oplus \{v \mapsto i', v_1 \mapsto i'_1, \ldots, v_n \mapsto i'_n\} \end{array}}{\langle \Gamma, V\rangle \vdash v = f(v_1, \ldots, v_n) : M}$$

Esempio

$$\langle \Gamma, \{X, Zs, Zs0\} \rangle \vdash \texttt{Zs = [X | Zs0]} : M_{213}$$

Usiamo **UNIFY-VF**

$$\frac{\begin{array}{c} Zs \notin \{X, Zs0\} \\ i = M_{213_{init}}(Zs) \\ \overline{i} = \langle M_{213_{init}}(X), M_{213_{init}}(Zs0) \rangle \\ abs\_unify\_inst\_func(i, [||], \overline{i}, i', \langle i'_1, i'_2 \rangle) \\ M_{213_{fin}} = M_{213_{init}} \oplus \{Zs \mapsto i', X \mapsto i'_1, Zs0 \mapsto i'_2\} \end{array}}{\langle \Gamma, \{X, Zs, Zs0\} \rangle \vdash \texttt{Zs = [X | Zs0]} : M_{213}}$$

Esempio

$$\langle \Gamma, \{X, Zs, Zs0\} \rangle \vdash \texttt{Zs = [X | Zs0]} : M_{213}$$

Usiamo **UNIFY-VF**

$$\frac{\begin{array}{c} Zs \notin \{X, Zs0\} \\ i = free \\ \overline{i} = \langle ground, ground \rangle \\ abs\_unify\_inst\_func(i, [||], \overline{i}, i', \langle i'_1, i'_2 \rangle) \\ M213_{fin} = M213_{init} \oplus \{Zs \mapsto i', X \mapsto i'_1, Zs0 \mapsto i'_2\} \end{array}}{\langle \Gamma, \{X, Zs, Zs0\} \rangle \vdash \texttt{Zs = [X | Zs0]} : M_{213}}$$

Esempio

$$\langle \Gamma, \{X, Zs, Zs0\} \rangle \vdash \texttt{Zs = [X | Zs0]} : M_{213}$$

Usiamo **UNIFY-VF**

$$Zs \notin \{X, Zs0\}$$

$$i = free$$

$$\bar{i} = \langle ground, ground \rangle$$

$$abs\_unify\_inst\_func(i, [||], \bar{i}, bound(\{[ground|ground]\}), \langle ground, ground \rangle)$$

$$M_{213_{fin}} = M_{213_{init}} \oplus \{Zs \mapsto bound(\{[ground|ground]\}), X \mapsto ground, Zs0 \mapsto ground\}$$

$$\langle \Gamma, \{X, Zs, Zs0\} \rangle \vdash \texttt{Zs = [X | Zs0]} : M_{213}$$

Esempio

$$
\begin{aligned}
M_{213} = \langle \{ & \mathtt{Xs} \mapsto bound(\{[ground|ground]\}), \\
& \mathtt{Ys} \mapsto ground, \mathtt{Zs} \mapsto free, \\
& Xs0 \mapsto ground,\ Zs0 \mapsto ground,\ X \mapsto ground\}, \\
\{ & \mathtt{Xs} \mapsto bound(\{[ground|ground]\}), \\
& \mathtt{Ys} \mapsto ground, \mathtt{Zs} \mapsto bound(\{[ground|ground]\}), \\
& Xs0 \mapsto ground,\ Zs0 \mapsto ground,\ X \mapsto ground\} \rangle
\end{aligned}
$$

Esempio

Torniamo a

$$\langle \Gamma, \{Xs, Ys, Zs\} \rangle \vdash \wedge\ B_{211}, B_{212}, B_{213} : M_{21}$$

Usiamo **CONJ**

$$\frac{\begin{array}{c} \langle \Gamma, \{X, Xs, Xs0\} \rangle \vdash B_{211} : M_{211} \\ \langle \Gamma, \{Ys, Zs0, Xs0\} \rangle \vdash B_{212} : M_{212} \\ \langle \Gamma, \{X, Zs, Zs0\} \rangle \vdash B_{213} : M_{213} \\ M_{21} = M_{211} \triangleright M_{212} \triangleright M_{213} \end{array}}{\langle \Gamma, \{Xs, Ys, Zs\} \rangle \vdash \wedge\ B_{211}, B_{212}, B_{213} : M_{21}}$$

Esempio

Torniamo a

$$\langle \Gamma, \{Xs, Ys, Zs\}\rangle \vdash \wedge\ B_{211}, B_{212}, B_{213} : M_{21}$$

Usiamo **CONJ**

$$\frac{\begin{array}{c}\langle \Gamma, \{X, Xs, Xs0\}\rangle \vdash B_{211} : M_{211} \\ \langle \Gamma, \{Ys, Zs0, Xs0\}\rangle \vdash B_{212} : M_{212} \\ \langle \Gamma, \{X, Zs, Zs0\}\rangle \vdash B_{213} : M_{213} \\ M_{21} = M_{211} \triangleright M_{212} \triangleright M_{213}\end{array}}{\langle \Gamma, \{Xs, Ys, Zs\}\rangle \vdash \wedge\ B_{211}, B_{212}, B_{213} : M_{21}}$$

Esempio

$$\begin{aligned} M_{21} = \langle \{ & \texttt{Xs} \mapsto bound(\{\ [],\ [ground|ground]\}), \\ & \texttt{Ys} \mapsto ground, \texttt{Zs} \mapsto free, \\ & Xs0 \mapsto free,\ Zs0 \mapsto free,\ X \mapsto free\}, \\ \{ & \texttt{Xs} \mapsto bound(\{[ground|ground]\}), \\ & \texttt{Ys} \mapsto ground, \texttt{Zs} \mapsto bound(\{[ground|ground]\}), \\ & Xs0 \mapsto ground,\ Zs0 \mapsto ground,\ X \mapsto ground\}\rangle \end{aligned}$$

Esempio

Torniamo a

$$\langle \Gamma, \{Xs, Ys, Zs\}\rangle \vdash \exists \texttt{Xs0,Zs0,X.}\ B_{21} : M_2$$

Usiamo **SOME**

$$\frac{\langle \Gamma, \{Xs, Ys, Zs\}\rangle \vdash B_{21} : M_{21} \qquad M_2 = M_{21} \ominus \{Xs0, Zs0, X\}}{\langle \Gamma, \{Xs, Ys, Zs\}\rangle \vdash \exists \texttt{Xs0,Zs0,X.}\ B_{21} : M_2}$$

Esempio

Torniamo a

$$\langle \Gamma, \{Xs, Ys, Zs\} \rangle \vdash \exists \texttt{Xs0,Zs0,X.}\ B_{21} : M_2$$

Usiamo **SOME**

$$\frac{\begin{array}{c}\langle \Gamma, \{Xs, Ys, Zs\} \rangle \vdash B_{21} : M_{21} \\ M_2 = M_{21} \ominus \{Xs0, Zs0, X\}\end{array}}{\langle \Gamma, \{Xs, Ys, Zs\} \rangle \vdash \exists \texttt{Xs0,Zs0,X.}\ B_{21} : M_2}$$

Esempio

$$
\begin{aligned}
M_2 = \langle \{ & \mathtt{Xs} \mapsto bound(\{\ [],\ [ground|ground]\}), \\
& \qquad \mathtt{Ys} \mapsto ground, \mathtt{Zs} \mapsto free\}, \\
\{ & \mathtt{Xs} \mapsto bound(\{[ground|ground]\}), \\
& \qquad \mathtt{Ys} \mapsto ground, \mathtt{Zs} \mapsto bound(\{[ground|ground]\})\}\rangle
\end{aligned}
$$

Esempio

Torniamo a

$$\langle \Gamma, \{Xs, Ys, Zs\}\rangle \vdash \vee\ B_1, B_2 : M'$$

Usiamo **DISJ**

$$\frac{\begin{array}{c}\langle \Gamma, \{Xs, Ys, Zs\}\rangle \vdash B_1 : M_1 \\ \langle \Gamma, \{Xs, Ys, Zs\}\rangle \vdash B_2 : M_2 \\ M' = M_1 \bowtie M_2\end{array}}{\langle \Gamma, \{Xs, Ys, Zs\}\rangle \vdash \vee\ B_1, B_2 : M'}$$

Esempio

$$
\begin{aligned}
M_1 = & \langle \{ \texttt{Xs} \mapsto bound(\{\ [],\ [ground|ground]\}), \\
& \qquad \texttt{Ys} \mapsto ground, \texttt{Zs} \mapsto free\}, \\
& \{ \texttt{Xs} \mapsto bound(\{\ []\ \}), \texttt{Ys} \mapsto ground, \texttt{Zs} \mapsto ground\} \rangle
\end{aligned}
$$

$$
\begin{aligned}
M_2 = & \langle \{ \texttt{Xs} \mapsto bound(\{\ [],\ [ground|ground]\}), \\
& \qquad \texttt{Ys} \mapsto ground, \texttt{Zs} \mapsto free\}, \\
& \{ \texttt{Xs} \mapsto bound(\{\ [ground|ground]\}), \\
& \qquad \texttt{Ys} \mapsto ground, \texttt{Zs} \mapsto bound(\{[ground|ground]\})\} \rangle
\end{aligned}
$$

Esempio

$$
\begin{aligned}
M' = &\langle \{\texttt{Xs} \mapsto bound(\{\ [],\ [ground|ground]\}),\\
&\qquad\qquad \texttt{Ys} \mapsto ground, \texttt{Zs} \mapsto free\},\\
&\quad \{\texttt{Xs} \mapsto bound(\{\ []\ \}) \sqcup bound(\{[ground|ground]\}),\\
&\quad\ \ \texttt{Ys} \mapsto ground,\\
&\quad\ \ \texttt{Zs} \mapsto ground \sqcup bound(\{[ground|ground]\})\}\rangle
\end{aligned}
$$

Esempio

$$
\begin{aligned}
M' =& \langle \{ \texttt{Xs} \mapsto bound(\{\ [],\ [ground|ground] \}), \\
&\qquad\qquad \texttt{Ys} \mapsto ground, \texttt{Zs} \mapsto free \}, \\
&\{ \texttt{Xs} \mapsto bound(\{\ [],\ [ground|ground] \}), \\
&\ \ \texttt{Ys} \mapsto ground, \\
&\ \ \texttt{Zs} \mapsto ground \} \rangle
\end{aligned}
$$

Esempio

Torniamo a

$$\langle \Gamma, \{Xs, Ys, Zs\} \rangle \vdash \vee\ B_1, B_2 : M'$$

Usiamo **DISJ**

$$\frac{\begin{array}{c}\langle \Gamma, \{Xs, Ys, Zs\} \rangle \vdash B_1 : M_1 \\ \langle \Gamma, \{Xs, Ys, Zs\} \rangle \vdash B_2 : M_2 \\ M' = M_1 \bowtie M_2\end{array}}{\langle \Gamma, \{Xs, Ys, Zs\} \rangle \vdash \vee\ B_1, B_2 : M'}$$

Esempio

Finalmente torniamo a

$$\Gamma \Vdash R$$

Usiamo **PROC**

$$\frac{\begin{array}{c} R \in \Gamma \\ R = \langle append(Xs, Ys, Zs) \leftarrow B, M \rangle \\ \langle \Gamma, \{Xs, Ys, Zs\} \rangle \vdash B : M' \\ dom(M) = dom(M') = \{Xs, Ys, Zs\} = V = uq(B) \\ \forall v \in \{Xs, Ys, Zs\}.\ M_{init}(v) \sqsubseteq M'_{init}(v) \\ \forall v \in \{Xs, Ys, Zs\}.\ M'_{fin}(v) \sqsubseteq M_{fin}(v) \end{array}}{\Gamma \Vdash R}$$

Esempio

Finalmente torniamo a

$$\Gamma \Vdash R$$

Usiamo **PROC**

$$\frac{\begin{array}{c} R \in \Gamma \\ R = \langle append(Xs, Ys, Zs) \leftarrow B, M \rangle \\ \langle \Gamma, \{Xs, Ys, Zs\} \rangle \vdash B : M' \\ dom(M) = dom(M') = \{Xs, Ys, Zs\} = V = uq(B) \\ \forall v \in \{Xs, Ys, Zs\}.\ M_{init}(v) \sqsubseteq M'_{init}(v) \\ \forall v \in \{Xs, Ys, Zs\}.\ M'_{fin}(v) \sqsubseteq M_{fin}(v) \end{array}}{\Gamma \Vdash R}$$

Esempio

$$\forall v \in \{Xs, Ys, Zs\}.\ M_{init}(v) \sqsubseteq M'_{init}(v)$$
$$\forall v \in \{Xs, Ys, Zs\}.\ M'_{fin}(v) \sqsubseteq M_{fin}(v)$$

$$M_{init}(Xs) \sqsubseteq M'_{init}(Xs)$$
$$M_{init}(Ys) \sqsubseteq M'_{init}(Ys)$$
$$M_{init}(Zs) \sqsubseteq M'_{init}(Zs)$$

$$M'_{fin}(Xs) \sqsubseteq M_{fin}(Xs)$$
$$M'_{fin}(Ys) \sqsubseteq M_{fin}(Ys)$$
$$M'_{fin}(Zs) \sqsubseteq M_{fin}(Zs)$$

Esempio

$$\forall v \in \{Xs, Ys, Zs\}.\ M_{init}(v) \sqsubseteq M'_{init}(v)$$
$$\forall v \in \{Xs, Ys, Zs\}.\ M'_{fin}(v) \sqsubseteq M_{fin}(v)$$

$$ground \sqsubseteq bound(\{\ [],\ [ground|ground]\})$$
$$ground \sqsubseteq ground$$
$$free \sqsubseteq free$$

$$bound(\{\ [],\ [ground|ground]\}) \sqsubseteq ground$$
$$ground \sqsubseteq ground$$
$$ground \sqsubseteq ground$$

Esempio

$$\forall v \in \{Xs, Ys, Zs\}.\ M_{init}(v) \sqsubseteq M'_{init}(v)$$
$$\forall v \in \{Xs, Ys, Zs\}.\ M'_{fin}(v) \sqsubseteq M_{fin}(v)$$

$$ground \sqsubseteq bound(\{\ [],\ [ground|ground]\})$$
$$ground \sqsubseteq ground$$
$$free \sqsubseteq free$$

$$bound(\{\ [],\ [ground|ground]\}) \sqsubseteq ground$$
$$ground \sqsubseteq ground$$
$$ground \sqsubseteq ground$$

Esempio

$$ground \sqsubseteq bound(\{\ [],\ [ground|ground]\})$$

Fortunatamente il *type_system* ci dice che per un termine di tipo lista

$$ground = bound(\{\ [],\ [ground|ground]\})$$

Esempio

$$\forall v \in \{Xs, Ys, Zs\}.\ M_{init}(v) \sqsubseteq M'_{init}(v)$$
$$\forall v \in \{Xs, Ys, Zs\}.\ M'_{fin}(v) \sqsubseteq M_{fin}(v)$$

$$ground \sqsubseteq bound(\{\ [],\ [ground|ground]\})$$
$$ground \sqsubseteq ground$$
$$free \sqsubseteq free$$

$$bound(\{\ [],\ [ground|ground]\}) \sqsubseteq ground$$
$$ground \sqsubseteq ground$$
$$ground \sqsubseteq ground$$

Esempio

$$\Gamma \Vdash R$$

Usiamo **PROC**

$$\frac{\begin{array}{c} R \in \Gamma \\ R = \langle append(Xs, Ys, Zs) \leftarrow B, M \rangle \\ \langle \Gamma, \{Xs, Ys, Zs\} \rangle \vdash B : M' \\ dom(M) = dom(M') = \{Xs, Ys, Zs\} = V = uq(B) \\ \forall v \in \{Xs, Ys, Zs\}.\ M_{init}(v) \sqsubseteq M'_{init}(v) \\ \forall v \in \{Xs, Ys, Zs\}.\ M'_{fin}(v) \sqsubseteq M_{fin}(v) \end{array}}{\Gamma \Vdash R}$$

Esempio

$$\Gamma \Vdash R$$

Usiamo **PROC**

$$\frac{\begin{array}{c} R \in \Gamma \\ R = \langle append(Xs, Ys, Zs) \leftarrow B, M \rangle \\ \langle \Gamma, \{Xs, Ys, Zs\} \rangle \vdash B : M' \\ dom(M) = dom(M') = \{Xs, Ys, Zs\} = V = uq(B) \\ \forall v \in \{Xs, Ys, Zs\}.\ M_{init}(v) \sqsubseteq M'_{init}(v) \\ \forall v \in \{Xs, Ys, Zs\}.\ M'_{fin}(v) \sqsubseteq M_{fin}(v) \end{array}}{\Gamma \Vdash R}$$

Evviva!

Bibliografia

Bibliografia

- **D.Overton. *Precise and Expressive Mode Systems for Typed Logic Programming Languages*. 2003.**
- L.Sterling, E.Shapiro. *The Art of Prolog: Advanced Programmin Techniques, Second Edition*. MIT Press. 1994.
- S.Russell, P.Norvig. *Artificial Intelligence: A Modern Approach, Third edition*. Pearson. 2009.
- P.Cousot, R.Cousot. *Abstact interpretation and application to logic programs*. journal of logic programming, 1992.

Bibliografia

- Z.Somogyi. *A system of precise modes for logic programs*.
- Z.Somogyi, F.Henderson and T.Conway. *the execution algorithm of Mercury, an efficient purely declarative logic programming language*. journal of logic programming, 1996.
- D.Overton, Z.Somogyi, P.J.Stuckey. *Contraint-Based Mode Analysis of Mercury*. 2002.
- F.Henderson, T.Conway, Z.Somogyi, D.Jeffery, P.Schachte, S.Taylor, C.Speirs, T.Dowd, R.Becket, M.Brown, P.Wang. The Mercury Language Reference Manual. 2014.