4 Calcoli tipati e sistemi di tipi

In questa sezione introdurremo vari calcoli tipati, essi possono essere dati in due versioni: usando termini senza informazione di tipo, oppure usando termini con informazione di tipo (vi e' un certo grado di discrezionalita' quanta informazione di tipo esplicitare). Prima di considerare calcoli specifici, introduciamo un po' di terminologia e notazione per un generico linguaggio tipato L.

- $\tau \in T(L)$ espressioni di tipo
- $M \in \Lambda_T(L)$ termini con informazioni di tipo, $M \in \Lambda(L)$ termini senza informazione di tipo

  in generale un indice T indichera' la versione di L che usa termini con informazioni di tipo
- $|\_|: \Lambda_T(L) \to \Lambda(L)$ funzione di cancellazione dell'informazione di tipo
- $\Gamma \vdash M: \tau$ typing judgement, dove $\Gamma: Var \stackrel{fin}{\to} T(L)$ e' un typing context e $\mathrm{FV}(M) \subseteq dom(\Gamma)$. Nei calcoli piu' complessi si hanno anche regole per derivare judgements della forma $\Gamma \vdash$, $\Gamma \vdash A$ e $\Gamma \vdash A \leq B$.
- $V \in \mathsf{Val}(L) \subseteq \Lambda(L)$ valori aperti; $V \in \mathsf{Val}_0(L) = \mathsf{Val}(L) \cap \Lambda_0(L)$ valori chiusi
- $M \Downarrow_e R$ semantica operazionale CBV con err, dove $M \in \Lambda_0(L)$ e $R \in \mathsf{Val}_0(L) + \{err\}$.

  La semantica operazionale con err serve a descrivere in modo preciso la classe di errori che si vorrebbero identificare a tempo di compilazione (mediante il type-checking). In certi casi puo' essere conveniente usare una semantica operazionale con ambiente: $\rho \vdash M \Downarrow_e R$, dove $\rho: Var \stackrel{fin}{\to} \mathsf{Val}_0(L)$ ambiente e $\mathrm{FV}(M) \subseteq dom(\rho)$.
- Dato un sistema (di type-checking) per derivare typing judgements, diciamo che vale la **correttezza** del type-checking relativamente alla semantica operazionale con err: "se M ha un tipo, allora non genera err a tempo di esecuzione", cioe' $\emptyset \vdash M: \tau$ e $M \Downarrow R$ implicano $R \not\equiv err$.

Descriveremo anche la semantica semplice dei tipi, nella quale un tipo τ e' interpretato da un sottoinsieme $[\![\tau]\!] \subseteq \mathsf{Val}_0$ di valori. Tale semantica permette di definire la soddisfacibilita' di un typing judgement $\Gamma \models M: \tau$ in funzione della sola semantica operazione, ed analogamente permette una interpretazione ovvia della relazione di sub-typing $\tau_1 \subseteq \tau_2$, cioe' $[\![\tau_1]\!] \subseteq [\![\tau_2]\!]$, e quindi validare la soundness di regole di inferenza per il sub-typing.

4.1 I tipi semplici

I tipi semplici sono quelli che tipicamente si incontrano in linguaggio di programmazione come il Pascal o Algol, cioe' con parametri di tipo procedure ma senza *polimorfismo*.

- $\tau \in T ::= \overline{\tau}\ C \mid \tau_1 \to \tau_2$, dove C e' un costruttore di tipo, p.e. *bool* o τ *list*.
- $M \in \Lambda ::= x \mid c \mid \lambda x.M \mid M_1 M_2$, dove c sono delle costanti (higher-order), p.e. if, nil, $cons$, fix.

  $M \in \Lambda_T ::= x \mid c_{\overline{\tau}} \mid \lambda x{:}\,\tau.M \mid M_1 M_2$ e' l'insieme dei termini con informazioni di tipo, e la funzione $|\_|: \Lambda_T \to \Lambda$ di cancellazione della informazione di tipo e' data da: $|x| = x$, $|c_{\overline{\tau}}| = c$, $|\lambda x{:}\,\tau.M| = \lambda x.|M|$, $|M_1 M_2| = |M_1||M_2|$.
- Le regole di inferenza per derivare typing judgements con informazioni di tipo nei termini sono

  $$\text{(x)}\ \Gamma \vdash_T x{:}\,\Gamma(x) \qquad \text{(c)}\ \Gamma \vdash_T c{:}\,\tau_c \qquad (\to\text{I})\ \frac{\Gamma, x{:}\,\tau_1 \vdash_T M{:}\,\tau_2}{\Gamma \vdash_T \lambda x{:}\,\tau_1.M{:}\,\tau_1 \to \tau_2} \qquad (\to\text{E})\ \frac{\begin{array}{c}\Gamma \vdash_T M_1{:}\,\tau_1 \to \tau_2 \\ \Gamma \vdash_T M_2{:}\,\tau_1\end{array}}{\Gamma \vdash_T M_1 M_2{:}\,\tau_2}$$

  alcuni esempi di tipi per le costanti sono:

  - $T, F{:}\,bool$ e $if_\tau{:}\,bool \to \tau \to \tau \to \tau$
  - $nil_\tau{:}\,\tau\ list$, $cons_\tau{:}\,\tau \to \tau\ list \to \tau\ list$ e $case_{\tau,\sigma}{:}\,\tau\ list \to \sigma \to (\tau \to \tau\ list \to \sigma) \to \sigma$, dove *case* corrisponde al pattern matching per il tipo τ *list*
  - $fix_\tau{:}\,(\tau \to \tau) \to \tau$, che serve per modellare le definizioni ricorsive.

  La seguente regola di **sostitutivita'** risulta essere *ammissibile*, cioe' se la premessa e' derivabile anche la conclusione lo e', e quindi puo' essere aggiunta senza modificare l'insieme dei typing judgements derivabili

  $$\text{(sub)}\ \frac{\Gamma \vdash M{:}\,\tau \quad \Gamma, x{:}\,\tau \vdash M'{:}\,\tau'}{\Gamma \vdash [M/x]M'{:}\,\tau'}$$

  Le regole di inferenza per derivare typing judgements senza informazioni di tipo nei termini si ottengono dalle precedenti applicando la funzione di cancellazione $|\_|$ ai termini M che compaiono nella regola.

- La semantica operazionale CBV con *err* $M \Downarrow R$ estende quella CBV precedentemente introdotta con regole aggiuntive per trattare le situazioni di errore. La daremo per i termini senza informazione di tipo, poiche' la semantica operazione non fa' alcun uso di tale informazione. Per semplicita' consideriamo il linguaggio con le sole costanti per il tipo *bool*, quindi l'insieme dei valori e' dato da $V \in \mathsf{Val}::= x \mid T \mid F \mid \lambda x.M$.

$$\text{(V)}\ V \Downarrow V \qquad \text{(app)}\ \frac{M_1 \Downarrow \lambda x.M \quad M_2 \Downarrow V \quad [V/x]M \Downarrow R}{M_1 M_2 \Downarrow R} \qquad \text{(if.T)}\ \frac{M \Downarrow T \quad M_1 \Downarrow R}{if(M, M_1, M_2) \Downarrow R}$$

$$\text{(app1)}\ \frac{M_1 \Downarrow \lambda x.M \quad M_2 \Downarrow err}{M_1 M_2 \Downarrow err} \qquad \text{(app2)}\ \frac{M_1 \Downarrow R \not\equiv \lambda x.M}{M_1 M_2 \Downarrow err} \qquad \text{(if1)}\ \frac{M \Downarrow R \not\equiv T, F}{if(M, M_1, M_2) \Downarrow err}$$

- La semantica di una espressione di tipo τ e' data da un sottoinsieme $[\![\tau]\!] \subseteq \mathsf{Val}_0$ definito per induzione su τ. La definizione di $[\![\tau]\!]$ fa' uso della semantica operazione con *err*, ma e' indipendente dai typing judgments.

  Dato $X \subseteq \mathsf{Val}_0$ definiamo $\widetilde{X} \stackrel{\Delta}{=} \{M \in \Lambda_0 | \forall R. M \Downarrow R \supset R \in X\}$. Intuitivamente $\widetilde{X}$ per l'insieme dei programmi $M \in \Lambda_0$ che in caso di terminazione producono un valore in X (e quindi non possono terminare con *err*).

  Data una semantica delle espressioni di tipo, diciamo che $\Gamma \models M{:}\,\tau$ vale $\stackrel{\Delta}{\Longleftrightarrow} \forall \rho \in [\![\Gamma]\!].([\rho]M) \in \widetilde{[\![\tau]\!]}$, dove $[\![\Gamma]\!] \stackrel{\Delta}{=} \{\rho{:}\, Var \stackrel{fin}{\to} \mathsf{Val}_0 | \forall x \in \mathrm{DV}(\Gamma).\rho(x) \in [\![\Gamma(x)]\!]\}$ e $[\rho]M \in \Lambda_0$ e' l'istanza di M ottenuta sostituendo ogni $x \in \mathrm{FV}(M)$ con $\rho(x) \in \mathsf{Val}_0$.

  Diamo il corrispettivo semantico di *bool* e dei costruttori *list* e $\to$, che quindi agiscono su insiemi $X, Y \subseteq \mathsf{Val}_0$

  - $bool \stackrel{\Delta}{=} \{T, F\}$
  - $X \to Y \stackrel{\Delta}{=} \{\lambda x.M \in \mathsf{Val}_0 | \forall V \in X.[V/x]M \in \widetilde{Y}\}$
  - $X\ list \stackrel{\Delta}{=} \{[V_i|i{:}\,n] | \forall i{:}\,n.V_i \in X\}$, dove $[V_i|i{:}\,n]$ e' una abbreviazione per $cons(V_0, \ldots, cons(V_{n-1}, nil)\ldots)$

  La semantica dei tipi, oltre a spiegarne il significato, e' anche utile per stabilire la **soundness** di regole per derivare typing judgements (nello stesso modo in cui l'interpretazione della logica del primo ordine, permette di validare assioni e regole di inferenza). Una volta stabilita la soundness, cioe' $\Gamma \vdash M{:}\,\tau$ derivabile implica $\Gamma \models M{:}\,\tau$ vera, la correttezza discende come un facile corollario. Infatti, dalla definizione di $\Gamma \models M{:}\,\tau$ e' immediato dimostrare che $\emptyset \models M{:}\,\tau$ e $M \Downarrow R$ implicano $R \not\equiv err$.

Dimostriamo la correttezza delle regole di type-checking passando attraverso una dimostrazione di soundness. In seguito illustreremo una tecnica piu' diretta per dimostrare la correttezza (ma anche meno informativa).
Per semplicita' ci limiteremo a dimostrare la soundness delle regole ($\to$I) e ($\to$E).

- Per ($\to$I) dobbiamo dimostrare che $\Gamma, x{:}\,\tau_1 \models M{:}\,\tau_2$ implica $\Gamma \models \lambda x{:}\,\tau_1.M{:}\,\tau_1 \to \tau_2$. Per semplicita' supponiamo che Γ sia vuoto.

  Dalla definizione di $x{:}\,\tau_1 \models M{:}\,\tau_2$ abbiamo che $\forall V{:}\,[\![\tau_1]\!].[V/x]M \in \widetilde{[\![\tau_2]\!]}$, ma questa e' esattamente la definizione di $(\lambda x.M) \in [\![\tau_1]\!] \to [\![\tau_2]\!]$, che e' banalmente equivalente alla validita di $\lambda x{:}\,\tau_1.M{:}\,\tau_1 \to \tau_2 \models$.

- Per ($\to$E) dobbiamo dimostrare che $\Gamma \models M_1{:}\,\tau_1 \to \tau_2$ e $\Gamma \models M_2{:}\,\tau_1$ implicano $\Gamma \models M_1 M_2{:}\,\tau_2$. Anche in questo caso supponiamo che Γ sia vuoto.

  Espandendo la definizione di $\models M_1 M_2{:}\,\tau_2$, quello che dobbiamo dimostrare e': $M_1 M_2 \Downarrow R$ implica $R \in [\![\tau_2]\!]$. Facendo una analisi delle possibili regole per derivare $M_1 M_2 \Downarrow R$ si vede che l'unica regola possibile e' (app). Infatti se fosse (app2) si avrebbe $M_1 \Downarrow R_1 \not\equiv \lambda x.M$ che contraddice $\Gamma \models M_1{:}\,\tau_1 \to \tau_2$, e se fosse (app1) si avrebbe $M_2 \Downarrow err$ che contraddice $\Gamma \models M_2{:}\,\tau_1$. Quindi abbiamo $M_1 \Downarrow \lambda x.M$, $M_2 \Downarrow V$ e $[V/x]M \Downarrow R$. Ma per le assunzioni si ha che $\lambda x.M \in [\![\tau_1]\!] \to [\![\tau_2]\!]$ e $V \in [\![\tau_2]\!]$, e percio' $[V/x]M \in \widetilde{[\![\tau_2]\!]}$ e infine $R \in [\![\tau_2]\!]$.

4.2 Correttezza e Subject Reduction

Se si vuole solo dimostrare la correttezza, si puo' evitare di considerare la semantica dei tipi (e la dimostrazione di soundness). La proprieta' che si deve dimostrare va sotto il nome di **Subject Reduction** (SR): $\vdash M{:}\,\tau$ e $M \Downarrow R$ implicano $\vdash R{:}\,\tau$ (e quindi $R \not\equiv err$. Tuttavia la demostrazione richiede una ipotesi induttiva (SR)+ piu' forte.

- dato $\rho{:}\, Var \stackrel{fin}{\to} \mathsf{Val}_0$ e $\Gamma{:}\, Var \to T$ (per semplicita' supponiamo di non avere variabili di tipo) diciamo che "$\rho{:}\,\Gamma$ vale" sse $\vdash \rho(x){:}\,\Gamma(x)$ vale per ogni $x \in dom(\Gamma)$.
- Ipotesi induttiva (SR)+: se $\Gamma \vdash M{:}\,\tau$, $\rho{:}\,\Gamma$ e $([\rho]M) \Downarrow R$ allora $\vdash R{:}\,\tau$.

- (SR)+ si dimostra per induzione sulla derivazione di $([\rho]M)\Downarrow R$, case analysis sulla derivazione di $\Gamma \vdash M\colon\tau$, e sfruttando alcune regole ammissibili sui typing judgements, p.e. (sub).

Tale tecnica fornisce dei risultati meno forti, pero' rappresenta una alternativa che vale la pena considerare quando la semantica dei tipi richiede strumenti matematici piu' sofisticati, p.e. in presenza di ricorsione e/o side-effects, dove la semantica naive dei tipi come sottoinviemi non funziona.

4.3 Riepilogo type systems e semantica operazionale

Tipi semplici.

- sintassi $A \in T ::= K \mid A_1 \to A_2$, $\quad M \in \Lambda ::= c \mid x \mid \lambda x.M \mid M_1 M_2$, $\quad V \in \mathsf{Val} ::= c \mid x \mid \lambda x.M$
- typing (empty) $\emptyset \vdash$ $\quad$ (add-x) $\dfrac{\Gamma \vdash A}{\Gamma, x\colon A \vdash}\ x \notin \mathrm{DV}(\Gamma)$ $\quad$ (K) $\dfrac{\Gamma \vdash}{\Gamma \vdash K}$ $\quad$ $(\to)$ $\dfrac{\Gamma \vdash A_1, A_2}{\Gamma \vdash A_1 \to A_2}$

  (x) $\dfrac{\Gamma \vdash}{\Gamma \vdash x\colon A}\ A = \Gamma(x)$ $\quad$ (c) $\dfrac{\Gamma \vdash}{\Gamma \vdash c\colon A_c}$ $\quad$ $(\to\mathrm{I})$ $\dfrac{\Gamma, x\colon A_1 \vdash M\colon A_2}{\Gamma \vdash \lambda x.M\colon A_1 \to A_2}$ $\quad$ $(\to\mathrm{E})$ $\dfrac{\Gamma \vdash M_1\colon A_1 \to A_2 \quad \Gamma \vdash M_2\colon A_1}{\Gamma \vdash M_1 M_2\colon A_2}$
- semantica operazionale (V) $V \Downarrow V$ $\quad$ (app) $\dfrac{M_1 \Downarrow \lambda x.P \quad M_2 \Downarrow V \quad [V/x]P \Downarrow R}{M_1 M_2 \Downarrow R}$

Varibili di tipo (come estensione dei tipi semplici).

- sintassi $A \in T+ = X$
- typing (add-X) $\dfrac{\Gamma \vdash}{\Gamma, X \vdash}\ X \notin \mathrm{DV}(\Gamma)$ $\quad$ (X) $\dfrac{\Gamma \vdash}{\Gamma \vdash X}\ X \in \mathrm{DV}(\Gamma)$
- correttezza. Poiche' i tipi contengono variabili di tipo, la semantica di un tipo A diventa una funzione della interpretazione delle variabili (di tipo) in $\mathrm{FV}(A)$. Anche l'enunciato di correttezza del type system rispetto alla semantica operazionale richiede degli aggiustamenti.

Varibili di tipo e subtyping (come estensione dei tipi semplici).

- sintassi $A \in T+ = X \mid Top$
- typing (add-X) $\dfrac{\Gamma \vdash A}{\Gamma, X \le A \vdash}\ X \notin \mathrm{DV}(\Gamma)$ $\quad$ (X) $\dfrac{\Gamma \vdash}{\Gamma \vdash X}\ X \in \mathrm{DV}(\Gamma)$
- subtyping (refl) $\dfrac{\Gamma \vdash A}{\Gamma \vdash A \le A}$ $\quad$ (trans) $\dfrac{\Gamma \vdash A \le B \le C}{\Gamma \vdash A \le C}$ $\quad$ $(\le)$ $\dfrac{\Gamma \vdash}{\Gamma \vdash X \le A}\ X \le A\ \in \Gamma$

  $(Top. \le)$ $\dfrac{\Gamma \vdash A}{\Gamma \vdash A \le Top}$ $\quad$ $(\to . \le)$ $\dfrac{\Gamma \vdash B_1 \le A_1 \quad \Gamma \vdash A_2 \le B_2}{\Gamma \vdash (A_1 \to A_2) \le (B_1 \to B_2)}$ $\quad$ (subsum) $\dfrac{\Gamma \vdash M\colon A \le B}{\Gamma \vdash M\colon B}$

Tipi record (come estensione dei tipi semplici).

- sintassi $A \in T+ = \{l_i\colon A_i | i\colon m\}$,
  $M \in \Lambda+ = \{l_i = M_i | i\colon m\} \mid \text{with } \{l_i = x_i | i\colon m\} = M \text{ do } N$, $\quad V \in \mathsf{Val}+ = \{l_i = V_i | i\colon m\}$
- typing $(\{\})$ $\dfrac{\Gamma \vdash A_i \quad i\colon m}{\Gamma \vdash \{l_i\colon A_i | i\colon m\}}\ l_{i:m}$ distinte $\quad$ $(\{\}\mathrm{I})$ $\dfrac{\Gamma \vdash M_i\colon A_i \quad i\colon m}{\Gamma \vdash \{l_i = M_i | i\colon m\}\colon \{l_i\colon A_i | i\colon m\}}\ l_{i:m}$ distinte

  $(\{\}\mathrm{E})$ $\dfrac{\Gamma \vdash M\colon \{l_i\colon A_i | i\colon m\} \quad \Gamma, (x_i\colon A_i)_{i:m} \vdash N\colon A}{\Gamma \vdash \text{with } \{l_i = x_i | i\colon m\} = M \text{ do } N\colon A}\ x_i$ distinte
- subtyping $(\{\}. \le)$ $\dfrac{\Gamma \vdash A_i \le B_i \quad i\colon m}{\Gamma \vdash \{l_i\colon A_i | i\colon n\} \le \{l_i\colon B_i | i\colon m\}}\ m \le n$ e $l_{i:n}$ distinte

- semantica operazionale

$$\text{(record)}\ \frac{M_i \Downarrow V_i \quad i{:}\,m}{\{l_i = M_i | i{:}\,m\} \Downarrow \{l_i = V_i | i{:}\,m\}}\ l_i \text{ distinte} \qquad \text{(with)}\ \frac{M \Downarrow \{l_i = V_i | i{:}\,m\} \quad [\overline{V}/\overline{x}]N \Downarrow R}{\text{with } \{l_i = x_i | i{:}\,m\} = M \text{ do } N \Downarrow R}$$

- semantica dei tipi come sottoinsiemi $\{l_i{:}\,X_i | i{:}\,m\} \stackrel{\Delta}{=} \{\{l_i = V_i | i{:}\,m\} | \forall i{:}\,m.V_i \in X_i\}$ dove $X_i \subseteq \mathsf{Val}_0$. Nel caso di subsumption, la semantica dei tipi record va modificata $\{l_i{:}\,X_i | i{:}\,m\} \stackrel{\Delta}{=} \{\{l_i = V_i | i{:}\,n\} | m \leq n \wedge \forall i{:}\,m.V_i \in X_i\}$.

Tipi variant (come estensione dei tipi semplici).

- sintassi $A \in T+ = [l_i{:}\,A_i | i{:}\,m]$, $\quad M \in \Lambda+ = [l = M] \mid \text{case } M \text{ of } [l_i = x_i \text{ then } N_i \ | i{:}\,m]$, $\quad V \in \mathsf{Val}+ = [l = V]$

- typing $([])\ \dfrac{\Gamma \vdash A_i \quad i{:}\,m}{\Gamma \vdash [l_i{:}\,A_i | i{:}\,m]}\ l_{i:m}$ distinte $\qquad ([]\text{I})\ \dfrac{\begin{array}{c}\Gamma \vdash M_i{:}\,A_i \\ (\Gamma \vdash A_i \quad i{:}\,m)\end{array}}{\Gamma \vdash [l_i = M_i]{:}\,[l_i{:}\,A_i | i{:}\,m]}\ l_{i:m}$ distinte

$$([]\text{E})\ \frac{\begin{array}{c}\Gamma \vdash M{:}\,[l_i{:}\,A_i | i{:}\,m] \\ \Gamma, x_i{:}\,A_i \vdash N_i{:}\,A \quad i{:}\,m\end{array}}{\Gamma \vdash \text{case } M \text{ of } [l_i = x_i \text{ then } N_i \ | i{:}\,m]{:}\,A}$$

- subtyping $([].\leq)\ \dfrac{\Gamma \vdash A_i \leq B_i \quad i{:}\,m}{\Gamma \vdash [l_i{:}\,A_i | i{:}\,m] \leq [l_i{:}\,B_i | i{:}\,n]}\ m \leq n$ e $l_{i:n}$ distinte

- semantica operazionale $\text{(variant)}\ \dfrac{M \Downarrow V}{[l = M] \Downarrow [l = V]} \qquad \text{(case)}\ \dfrac{M \Downarrow [l_i = V] \quad [V/x_i]N \Downarrow R}{\text{case } M \text{ of } [l_i = x_i \text{ then } N_i \ | i{:}\,m] \Downarrow R}$

- semantica dei tipi come sottoinsiemi $[l_i{:}\,X_i | i{:}\,m] \stackrel{\Delta}{=} \{[l = V] | \exists i{:}\,m.l = l_i \wedge V \in X_i\}$ dove $X_i \subseteq \mathsf{Val}_0$. Nel caso di subsumption, non e' necessario modificare la semantica dei tipi variant.

Tipi reference (come estensione dei tipi semplici).

- sintassi $A \in T+ = Ref\ A$, $\quad M \in \Lambda+ = ref_A(M) \mid deref(M) \mid M{:}= N$

- typing $\text{(Ref)}\ \dfrac{\Gamma \vdash A}{\Gamma \vdash Ref\ A} \qquad \text{(Ref.I)}\ \dfrac{\Gamma \vdash M{:}\,A}{\Gamma \vdash ref_A(M){:}\,Ref\ A} \qquad \text{(Ref.E)}\ \dfrac{\Gamma \vdash M{:}\,Ref\ A}{\Gamma \vdash deref(M){:}\,A}$

$$(:=)\ \frac{\Gamma \vdash M{:}\,Ref\ A,\ N{:}\,A}{\Gamma \vdash M{:}= N{:}\,Unit}$$

- subtyping: la relazione di subtyping su $Ref\ A$ coincide con l'uguaglianza

- semantica operazionale (richiede varie modifiche)
  - $s \in Mem$ (stato della memoria) $\stackrel{\Delta}{\Longleftrightarrow}$ $s{:}\,Var \stackrel{fin}{\to} \mathsf{Val}$ e $\forall x \in dom(s).\mathrm{FV}(sx) \subseteq dom(s)$. Sarebbe opportuno associare ad ogni locazione di memoria x anche un tipo, che al contrario del valore non cambia.
  - $s, M \Downarrow V, s'$ (evaluation judgements) $\stackrel{\Delta}{\Longleftrightarrow}$
    * $s, s' \in Mem$ e $dom(s) \subseteq s'$
    * $M \in \Lambda$ e $\mathrm{FV}(M) \subseteq dom(s)$
    * $V \in \mathsf{Val}$ e $\mathrm{FV}(V) \subseteq dom(s')$

$$\text{(V)}\ s, V \Downarrow V, s \qquad \text{(app)}\ \frac{s_0, M_1 \Downarrow \lambda x.P, s_1 \quad s_1, M_2 \Downarrow V, s_2 \quad s_2, [V/x]P \Downarrow R, s_3}{s_0, M_1 M_2 \Downarrow R, s_3}$$

$$\text{(ref)}\ \frac{s_0, M \Downarrow V, s_1}{s_0, ref_A(M) \Downarrow x, s_1[x \mapsto (V, A)]}\ x \notin dom(s_1) \qquad \text{(deref)}\ \frac{s_0, M \Downarrow x, s_1}{s_0, deref(M) \Downarrow s_1(x), s_1}$$

$$(:=)\ \frac{s_0, M \Downarrow x, s_1 \quad s_1, NM \Downarrow V, s_2}{s_0, M{:}= N \Downarrow unit, s_2[x \mapsto V]}$$

- correttezza. La semantica dei tipi si complica notevolmente, mentre la correttezza del type system rispetto alla semantica operazionale richiede solo degli aggiustamenti, in particolare e' opportuno aggiungere ad s anche dell'informazione di tipo (per semplicita' supponiamo che non vi siano variabili di tipo).

- Se $\Gamma \vdash M\colon A$ e' derivabile
- se $s\colon Var \stackrel{fin}{\to} (\mathsf{Val} \times T)$ e' **type-correct**, cioe' $\Gamma_s \vdash V\colon A$ per ogni $(V\colon A) = s(x)$, dove $\Gamma_s \stackrel{\Delta}{\equiv} \{(x\colon Ref\ A) | x \in dom(s) \wedge (V\colon A) = s(x)\}$
- se $\rho\colon Var \stackrel{fin}{\to} \mathsf{Val}$ e' t.c. $\Gamma_s \vdash \rho(x)\colon \Gamma(x)$ per ogni $x \in \mathrm{DV}(\Gamma)$
- se $s, [\rho]M \Downarrow V, s'$
- allora s' e' type-correct e $\Gamma_{s'} \vdash V\colon A$.

La dimostrazione procede per induzione sulla derivazione di $s, [\rho]M \Downarrow V, s'$.

Tipi ricorsivi (come estensione dei tipi semplici con variabili di tipo).

- sintassi $A \in T+ = \mu X.A$, $\quad M \in \Lambda+ = fold(M) \mid unfold(M)$, $\quad V \in \mathsf{Val}+ = fold(V)$
- typing $(\mu)\ \dfrac{\Gamma, X \vdash A}{\Gamma \vdash \mu X.A}$ $\quad (\mu\mathrm{I})\ \dfrac{\Gamma \vdash M\colon [\mu X.A/X]A \quad (\Gamma, X \vdash A)}{\Gamma \vdash fold(M)\colon \mu X.A}$ $\quad (\mu\mathrm{E})\ \dfrac{\Gamma \vdash M\colon \mu X.A}{\Gamma \vdash unfold(M)\colon [\mu X.A/X]A}$
- subtyping $(\mu.\leq)\ \dfrac{\Gamma, X \leq Y \vdash A \leq B}{\Gamma \vdash \mu X.A \leq \mu Y.B}$
- semantica operazionale (fold) $\dfrac{M \Downarrow V}{fold(M) \Downarrow fold(V)}$ $\quad$ (unfold) $\dfrac{M \Downarrow fold(V)}{unfold(M) \Downarrow V}$
- correttezza. La semantica dei tipi si complica notevolmente, mentre la correttezza del type system rispetto alla semantica operazionale si dimostra nel modo usuale. In effetti la semantica dei tipi si complica gia' con in presenza di definizioni ricorsive a livello di termini $M \in \Lambda+ = \mu x.M$, anche rimanendo con i tipi semplici:
  $(\mu)\ \dfrac{\Gamma, x\colon A \vdash M\colon A}{\Gamma \vdash \mu x.M\colon A}$ $\quad (\mu)\ \dfrac{[\mu x.M/x]M \Downarrow V}{\mu x.M \Downarrow V}$
  in presenza di tipi ricorsivi, il termine $\mu x.M$ e' definibile.
- semantica dei tipi come sottoinsiemi. Data una funzione $F\colon \mathcal{P}(\mathsf{Val}_0) \to \mathcal{P}(\mathsf{Val}_0)$, che corrisponde all'interpretazione di un tipo con una variabile $X \vdash A$, l'interpretazione del tipo $\mu X.A$ e' data da un sottoinsieme $X \subseteq \mathsf{Val}_0$ che deve soddisfare la seguente proprieta' $X = \{fold(V) | V \in F(X)\}$. A priori non e' detto che un tale X esista, ne' che sia unico. Tuttavia, nell'ipotesi che F sia una funzione monotona esiste il piu' piccolo sottoinsieme $\mu(F)$ tra quelli che soddisfano la proprieta' richiesta per X.

Tipi universali (come estensione dei tipi semplici con variabili di tipo).

- sintassi $A \in T+ = \forall X.A$, non vi sono modifiche ai termini poiche' non contengono informazione di tipo.
- typing $(\forall)\ \dfrac{\Gamma, X \vdash A}{\Gamma \vdash \forall X.A}$ $\quad (\forall\mathrm{I})\ \dfrac{\Gamma, X \vdash M\colon A}{\Gamma \vdash M\colon \forall X.A}$ $\quad (\forall\mathrm{E})\ \dfrac{\Gamma \vdash M\colon \forall X.A \quad \Gamma \vdash B}{\Gamma \vdash M\colon [B/X]A}$
- subtyping $(\forall.\leq)\ \dfrac{\Gamma, X \vdash A \leq B}{\Gamma \vdash \forall X.A \leq \forall X.B}$ ma per termini senza tipi valgono delle regole piu' forti
  $(\forall.\leq)\ \dfrac{\Gamma \vdash B \quad \Gamma, X \vdash A}{\Gamma \vdash \forall X.A \leq [B/X]A}$ $\quad (\leq.\forall)\ \dfrac{\Gamma \vdash B \quad \Gamma, X \vdash B \leq A}{\Gamma \vdash B \leq \forall X.A}$
- semantica dei tipi come sottoinsiemi. Data una funzione $F\colon \mathcal{P}(\mathsf{Val}_0) \to \mathcal{P}(\mathsf{Val}_0)$, che corrisponde all'interpretazione di un tipo con una variabile $X \vdash A$, l'interpretazione del tipo $\forall X.A$ e' data da $\forall(F) \stackrel{\Delta}{\equiv} \{V | \forall X \subseteq \mathsf{Val}_0.V \in F(X)\}$, ovvero $\cap\{F(X) | X \subseteq \mathsf{Val}_0\}$.

Tipi esistenziali (come estensione dei tipi semplici con variabili di tipo).

- sintassi $A \in T+ = \exists X.A$, $\quad M \in \Lambda + = \text{open } M \text{ as } x \text{ in } N$
- typing $(\exists)\ \dfrac{\Gamma, X \vdash A}{\Gamma \vdash \exists X.A}$ $\quad (\exists\mathrm{I})\ \dfrac{\Gamma, X \vdash A \quad \Gamma \vdash B \quad \Gamma \vdash M{:}\,[B/X]A}{\Gamma \vdash M{:}\,\exists X.A}$ $\quad (\exists\mathrm{E})\ \dfrac{\Gamma \vdash B \quad \Gamma \vdash M{:}\,\exists X.A \quad \Gamma, X, x{:}\,A \vdash N{:}\,B}{\Gamma \vdash \text{open } M \text{ as } x \text{ in } N{:}\,B}$
- subtyping $(\exists.\leq)\ \dfrac{\Gamma, X \vdash A \leq B}{\Gamma \vdash \exists X.A \leq \exists X.B}$ ma per termini senza tipi valgono delle regole piu' forti

  $(\exists.\leq)\ \dfrac{\Gamma \vdash B \quad \Gamma, X \vdash A}{\Gamma \vdash [B/X]A \leq \exists X.A}$ $\quad (\leq.\exists)\ \dfrac{\Gamma \vdash B \quad \Gamma, X \vdash A \leq B}{\Gamma \vdash \exists X.A \leq B}$
- semantica operazionale: dal punto di vista della semantica operazionale il termine open M as x in N puo' essere identificato con $(\lambda x.N)M$.
- semantica dei tipi come sottoinsiemi. Data una funzione $F{:}\,\mathcal{P}(\mathsf{Val}_0) \to \mathcal{P}(\mathsf{Val}_0)$, che corrisponde all'interpretazione di un tipo con una variabile $X \vdash A$, l'interpretazione del tipo $\exists X.A$ e' data da $\exists(F) \stackrel{\Delta}{=} \{V | \exists X \subseteq \mathsf{Val}_0.V \in F(X)\}$, ovvero $\cup\{F(X) | X \subseteq \mathsf{Val}_0\}$.

Tipi universali bounded (come estensione dei tipi semplici con variabili di tipo e subtyping).

- sintassi $A \in T+ = \forall X \leq A_1.A_2$, non vi sono modifiche ai termini poiche' non contengono informazione di tipo.
- typing $(\forall_b)\ \dfrac{\Gamma, X \leq A_1 \vdash A_2}{\Gamma \vdash \forall X \leq A_1.A_2}$ $\quad (\forall_b\mathrm{I})\ \dfrac{\Gamma, X \leq A_1 \vdash M{:}\,A_2}{\Gamma \vdash M{:}\,\forall X \leq A_1.A_2}$ $\quad (\forall_b\mathrm{E})\ \dfrac{\Gamma \vdash M{:}\,\forall X \leq A_1.A_2 \quad \Gamma \vdash B \leq A_1}{\Gamma \vdash M{:}\,[B/X]A_2}$
- subtyping $(\forall_b.\leq)\ \dfrac{\Gamma \vdash B_1 \leq A_1 \quad \Gamma, X \leq B_1 \vdash A_2 \leq B_2}{\Gamma \vdash \forall X \leq A_1.A_2 \leq \forall X \leq B_1.B_2}$ ma per termini senza tipi valgono delle regole piu' forti

  $(\forall_b.\leq)\ \dfrac{\Gamma \vdash B \leq A_1 \quad \Gamma, X \leq A_1 \vdash A_2}{\Gamma \vdash \forall X \leq A_1.A_2 \leq [B/X]A_2}$ $\quad (\leq.\forall_b)\ \dfrac{\Gamma \vdash B \quad \Gamma, X \leq A_1 \vdash B \leq A_2}{\Gamma \vdash B \leq \forall X \leq A_1.A_2}$
- semantica dei tipi come sottoinsiemi. Dato $X \subseteq \mathsf{Val}_0$ ed una funzione $F{:}\,\mathcal{P}(\mathsf{Val}_0) \to \mathcal{P}(\mathsf{Val}_0)$, che corrispondono rispettivamente all'interpretazione di un tipo chiuso A ed uno con una variabile $X \vdash B$, l'interpretazione del tipo $\forall X \leq A.B$ e' data da $\forall(X, F) \stackrel{\Delta}{=} \{V | \forall Y \subseteq X.V \in F(Y)\}$, ovvero $\cap\{F(Y) | Y \subseteq X\}$.

I tipi universali bounded sono stati introdotti per riuscire a giustificare la correttezza delle regole di tipaggio dei linguaggi OO. Tuttavia un'analisi piu' accurata ha portato ad osservare che e' necessaria una forma di quantificazione ancora piu' complessa, detta f-bounded.