Valutazione "lazy" di $\wedge$

- In C, la valutazione della congiunzione *and* è chiamata *lazy* o pigra, dato che inizialmente si rimanda la valutazione della seconda espressione e poi la si fa solo se necessario.
- In particolare, prima si controlla se la prima espressione è falsa, nel qual caso si evita di valutare la seconda parte dell'espressione, dato che la valutazione complessiva sarà comunque falsa.
- Questo tipo di valutazione può essere reso con le regole della semantica operazionale seguenti:

$$\frac{\langle b_0, \sigma \rangle \to \textbf{false}}{\langle b_0 \wedge b_1, \sigma \rangle \to \textbf{false}} \qquad \frac{\langle b_0, \sigma \rangle \to \textbf{true} \quad \langle b_1, \sigma \rangle \to t}{\langle b_0 \wedge b_1, \sigma \rangle \to t}$$

Esecuzione dei comandi

L'esecuzione di un comando porta ad un nuovo stato: $\langle c, \sigma \rangle \to \sigma'$, ammesso che l'esecuzione termini!
Si suppone che lo *stato inziale* σ_0 abbia la proprietà che $\sigma_0(X) = 0$ per tutte le locazioni X.

- $\langle \textbf{skip}, \sigma \rangle \to \sigma$
- $\dfrac{\langle a, \sigma \rangle \to m}{\langle X = a, \sigma \rangle \to \sigma[m/X]}$ $\sigma[m/X]$ è lo stato dove m è associato a X^*
- $\dfrac{\langle c_0, \sigma \rangle \to \sigma'' \quad \langle c_1, \sigma'' \rangle \to \sigma'}{\langle c_0; c_1, \sigma \rangle \to \sigma'}$ prima va eseguito il primo comando
- $\dfrac{\langle b, \sigma \rangle \to \textbf{true}/\textbf{false} \quad \langle c_i, \sigma \rangle \to \sigma_i}{\langle \textbf{if } b \textbf{ then } c_0 \textbf{ else } c_1, \sigma \rangle \to \sigma_0/\sigma_1}$

$$^*\sigma[m/x](y) = \begin{cases} m & \text{if } y = x; \\ \sigma(y) & \text{if } y \neq x. \end{cases}$$

Valutazione dei comandi (cont.)

- $$\frac{\langle b, \sigma\rangle \rightarrow \textbf{false}}{\langle \textbf{while } b \textbf{ do } c, \sigma\rangle \rightarrow \sigma}$$
- $$\frac{\langle b, \sigma\rangle \rightarrow \textbf{true} \langle c; \textbf{while } b \textbf{ do } c, \sigma\rangle \rightarrow \sigma'}{\langle \textbf{while } b \textbf{ do } c, \sigma\rangle \rightarrow \sigma'}$$

Completezza e Correttezza

Vorremmo che la nostra semantica ci permettesse di dimostrare tutto ciò che è vero e niente di falso. La semantica è

- **completa** se può dimostrare ogni *asserzione* vera. Ad es., con la regola seguente, *non* potrei dimostrare che $\langle 5-3, \sigma\rangle \rightarrow 2$ (vero).

$$\frac{\langle a_0, \sigma\rangle \rightarrow n_0 \quad \langle a_1, \sigma\rangle \rightarrow 0}{\langle a_0 - a_1, \sigma\rangle \rightarrow n_0}$$

- **corretta** se ogni *asserzione* dimostrabile è vera. Ad es., con la regola seguente, *potrei* dimostrare che $\langle 5-3, \sigma\rangle \rightarrow 7$ (falso)

$$\frac{\langle a_0, \sigma\rangle \rightarrow n_0 \quad \langle a_1, \sigma\rangle \rightarrow n_1}{\langle a_0 - a_1, \sigma\rangle \rightarrow n_0+2}$$

Equivalenze

- Due espressioni aritmetiche a_0 e a_1 sono *equivalenti* $a_0 \sim a_1$ se e solo se
$$\forall n \in \mathbf{N}, \forall \sigma \in \Sigma. \langle a_0, \sigma \rangle \rightarrow n \Leftrightarrow \langle a_1, \sigma \rangle \rightarrow n$$
- Due espressioni booleane b_0 e b_1 sono *equivalenti* $b_0 \sim b_1$ se e solo se
$$\forall t, \forall \sigma \in \Sigma. \langle b_0, \sigma \rangle \rightarrow t \Leftrightarrow \langle b_1, \sigma \rangle \rightarrow t$$
- Due comandi c_0 e c_1 sono *equivalenti* $c_0 \sim c_1$ se e solo se
$$\forall \sigma, \sigma' \in \Sigma. \langle c_0, \sigma \rangle \rightarrow \sigma' \Leftrightarrow \langle c_1, \sigma \rangle \rightarrow \sigma'$$

Es. di dimostrazione sui comandi

Proposizione

Sia $w \equiv \textbf{while } b \textbf{ do } c$. Allora

$$w \sim \textbf{if } b \textbf{ then } c; w \textbf{ else skip}$$

Dimostrazione

Si deve dimostrare che per ogni scelta di σ, σ':

$$\langle w, \sigma \rangle \to \sigma' \text{ sse } \langle \textbf{if } b \textbf{ then } c; w \textbf{ else skip}, \sigma \rangle \to \sigma'$$

Lo si fa usando le derivazioni delle regole

Dimostrazione $\Leftarrow$

Siano $\sigma, \sigma' \in \Sigma$ e si supponga che $\langle \textbf{if } b \textbf{ then } c; w \textbf{ else skip}, \sigma\rangle \to \sigma'$.
Allora deve esistere una derivazione con una delle seguenti forme

- $$\frac{\dfrac{\cdots}{\langle b, \sigma\rangle \to \textbf{false}} \quad \dfrac{}{\langle \textbf{skip}, \sigma\rangle \to \sigma}}{\langle \textbf{if } b \textbf{ then } c; w \textbf{ else skip}, \sigma\rangle \to \sigma} \quad (1 \Leftarrow) \text{ (qui } \sigma' = \sigma)$$
- $$\frac{\dfrac{\cdots}{\langle b, \sigma\rangle \to \textbf{true}} \quad \dfrac{}{\langle c; w, \sigma\rangle \to \sigma'}}{\langle \textbf{if } b \textbf{ then } c; w \textbf{ else skip}, \sigma\rangle \to \sigma'} \quad (2 \Leftarrow)$$

Da ognuna si può partire per costruire la derivazione $\langle w, \sigma\rangle \to \sigma'$.

Dimostrazione caso (2 $\Leftarrow$)

La seconda derivazione deve contenere, per qualche stato σ'', qualcosa del tipo

$$\frac{\dfrac{\cdots}{\langle c, \sigma\rangle \to \sigma''} \quad \dfrac{\cdots}{\langle w, \sigma''\rangle \to \sigma'}}{\langle c; w, \sigma\rangle \to \sigma'}$$

Combinando con il resto:

$$\frac{\dfrac{\cdots}{\langle b, \sigma\rangle \to \mathbf{true}} \quad \dfrac{\cdots}{\langle c, \sigma\rangle \to \sigma''} \quad \dfrac{\cdots}{\langle w, \sigma''\rangle \to \sigma'}}{\langle w, \sigma\rangle \to \sigma'}$$

Analogamente per il caso (1). Si può quindi concludere che

$$\langle \mathbf{if}\ b\ \mathbf{then}\ c; w\ \mathbf{else\ skip}, \sigma\rangle \to \sigma' \Rightarrow \langle w, \sigma\rangle \to \sigma'$$