

Structural operational semantics

structural regole di inferenza basate sulla struttura sintattica (una o più regole per ogni costruito nella grammatica);

operational relazione di transizione che descrive il comportamento dei programmi definendo passo-passo le operazioni da svolgere durante l'esecuzione.

Ci sono due approcci per la definizione di una semantica in stile SOS:

Small-step

$$\rightarrow_{ss} \subseteq Exp \times Exp$$

- ogni passo della relazione di transizione esegue una singola operazione;
- una computazione è una sequenza di passi;
- fornisce informazioni su *come* il programma viene eseguito, anche quando non termina (sequenza infinita di riduzioni).

Big-step

$$\rightarrow_{bs} \subseteq Exp \times Val$$

- la relazione descrive in un solo passo l'intera computazione;
- le singole operazioni sono descritte nell'albero di derivazione;
- più semplice da gestire quando non sono necessari dettagli su come esattamente sono calcolati i risultati.

Esempio

Semantica delle espressioni

Semantica *small-step*:

$$\begin{array}{c} n \rightarrow_{ss} n \quad \frac{E_1 \rightarrow_{ss} E'_1}{E_1 \text{ op } E_2 \rightarrow_{ss} E'_1 \text{ op } E_2} \\[1em] \frac{E_2 \rightarrow_{ss} E'_2}{n \text{ op } E_2 \rightarrow_{ss} n \text{ op } E'_2} \quad \frac{n_1 \text{ op } n_2 = n}{n_1 \text{ op } n_2 \rightarrow_{ss} n} \end{array}$$

Semantica *big-step*:

$$n \rightarrow_{bs} n \quad \frac{E_1 \rightarrow_{bs} n_1 \quad E_2 \rightarrow_{bs} n_2 \quad n_1 \text{ op } n_2 = n}{E_1 \text{ op } E_2 \rightarrow_{bs} n}$$

Implementazione

- pattern matching sui nodi dell'AST;
- ad ogni tipo di nodo corrisponde un costruito sintattico del linguaggio;
- per ogni pattern si identificano le regole della semantica corrispondenti;
- si verificano le precondizioni delle varie regole, chiamando ricorsivamente l'interprete se necessario;
- quando si trova una regola applicabile, si calcola il risultato della transizione.