

- Le condizioni C sono espressioni dal valore booleano: Odd è un predicato unario che verifica se l'espressione è 0, poi ci sono vari operatori di confronto di espressioni E.
- Le espressioni E sono identificatori, valori naturali, operazioni tra espressioni e espressioni tra parentesi.
- Qui sintetizziamo con op la metavariable dell'insieme $\{+, -, *, /\}$. La stessa cosa potevamo farla per gli operatori di confronto, e mettere in C la produzione in E cop E con cop metavariable in $\{>, >=, =, \#, <=, <\}$.

Per la grammatica delle espressioni dobbiamo fare attenzione all'ambiguità: questa è ambigua non considerando le parentesi. Se invece mettiamo tutte le parentesi, allora non abbiamo più l'ambiguità ma diventa un linguaggio pesante da usare, pronò ad errori (tante parentesi da scrivere) e difficile da leggere. Dobbiamo quindi cercare un compromesso che permetta di tener conto le precedenze matematiche in assenza di parentesi, senza obbligare ad usare le parentesi e comunque eliminando l'ambiguità.

<Expr> $E \rightarrow [+,-] T [A T]$

<Add op> $A \rightarrow + \mid -$

<Terms> $T \rightarrow F [M F]$

$M \rightarrow * \mid /$

<Factor> $F \rightarrow I \mid N \mid (E)$

<Numbers> $N \rightarrow [+,-] d\{d\}$

<Digit> $d \rightarrow 0 \mid \dots \mid 9$

<Ident> $I \rightarrow l\{l, d\}$

<Letter> $l \rightarrow A \mid \dots \mid Z$

Questa grammatica delle espressioni rispetta tutte le convenzioni e non è ambigua. Questa è la definizione che serve ai compilatori, mentre per parlare di semantica possiamo anche semplicemente usare la grammatica ambigua, più facile da usare.