

$$e ::= \begin{array}{l} n \\ | x \\ | (e_1 \circ e_2) \\ | !(e) \\ | g(e_1, \dots, e_k) \end{array} \quad (1)$$

$$c ::= \begin{array}{l} x = e; \\ | \text{if } (e) \ c_1 \ [\text{else } c_2] \\ | \text{while } (e) \ c \\ | \text{read}(x); \\ | \text{write}(e); \\ | \text{wsp}(e); \\ | \text{wnl}(e); \\ | \text{return } e; \\ | \{c_1 \dots c_n\} \\ | ; \end{array} \quad (2)$$

$$d ::= \text{int } x_0, \dots, x_k; \quad (3)$$

$$f ::= \text{int } g(\text{int } x_1, \dots, \text{int } x_k) \ (; \ | \ \{d_1 \dots d_m \ c_1 \dots c_n\}) \quad (4)$$

$$p ::= d_1 \dots d_m \ /* \ */ \ f_1 \dots f_n \quad (5)$$

Figura 1: La sintassi di CCINO

```

/* Tipo per la memorizzazione di una variabile
 * Ogni variabile e' rappresentata tramite un intero che indica la sua
 * posizione nella sequenza delle definizioni di variabili locali o globali
 * per il quale riserviamo un campo di 7 bit (pari a 128 variabili)
 * Il bit isglobal indica se si tratta di variabile globale (se pari a 1)
 * o locale (se pari a 0)
 */
struct variabile {
    unsigned isglobal : 1;
    unsigned ofs : 7;
};

```

### 1.1.2 Le espressioni

Per quanto riguarda le espressioni, in base alla regola (1) in Figura 1, possiamo individuare i seguenti casi:

1. l'espressione è un intero  $n$ ;
2. l'espressione è una variabile  $x$ . Nel qual caso occorre distinguere se
  - (a)  $x$  è locale alla funzione che si sta analizzando, in particolare,  $x$  è un parametro della funzione oppure è stata definita con una dichiarazione locale;
  - (b)  $x$  è stata definita con una dichiarazione globale.
3. l'espressione è ottenuta componendo le espressioni  $e_1$  ed  $e_2$  mediante un operatore binario;
4. l'espressione è ottenuta applicando l'operatore unario  $!$  all'espressione  $e$ ;
5. l'espressione è ottenuta applicando una funzione  $g$  definita dall'utente a  $k$  espressioni  $e_1, \dots, e_k$ .

In base alle precedenti considerazioni, nel file `ccindefs.h`, sono definite le seguenti strutture dati:

```

/* Tipi di espressione */
enum exp_tags { EXP_NUM, /* numero */

```