

1 Controllo dei tipi

Il controllo sui tipi viene garantito dal metodo `Node typeCheck()` che restituisce un nodo rappresentante il tipo del nodo (istanza di `Node`) su cui si è invocato, *null* in caso quel nodo non debba avere associato un tipo o un’eccezione in caso di errori durante il processo.

Il valore *null* viene ritornato da tutti i nodi che hanno come supertipo `TypeNode`, dagli argomenti delle funzioni, dalle dichiarazioni di variabili e di funzioni.

Le istruzioni, per scelta implementativa, ritornano il tipo `void` ma ci sono alcune eccezioni:

- `RetNode` che ritorna il tipo dell’espressione, `void` nel caso non ce ne sia alcuna;
- `CallNode` che ritorna il tipo di ritorno dichiarato dalla funzione;
- `IfNode` che ritorna il tipo comune ai due rami (che deve essere quindi uguale);
- `BlockNode` che ritorna `void` in assenza di istruzioni o quando non vi è né `return` né *if-then-else* altrimenti ritorna il tipo di `RetNode` o di `IfNode`.

1.1 Tipi

I tipi implementati nel linguaggio sono:

- `bool`: rappresenta un tipo booleano `true` o `false`;
- `int`: rappresenta un tipo intero;
- `pointer`: rappresenta un tipo puntatore. Questo può puntare ad un altro puntatore oppure ad un tipo primitivo quindi booleano o intero;
- `void`: rappresenta il tipo vuoto;
- `arrowType`: tipo utilizzato per le *funzioni*, contiene la lista dei tipi dei parametri e il tipo ritornato.