**Relazione progetto Compilatori**

**Specifiche del linguaggio**

* Una sola routine main(). Non possono essere definite altre funzioni
* Solo variabili intere
* Una variabile deve essere definita prima di poter essere usata. È possibile inizializzare una variabile al momento della dichiarazione ed è possibile dichiarare una sola variabile per volta
* Il nome di una variabile deve essere formato da un carattere alfabetico seguito da uno o più caratteri alfabetici o da numeri
* È disponibile un’istruzione di stampa nel formato: print(id) dove id è una variabile precedentemente dichiarata
* Gestisce espressioni aritmetiche ed espressioni logiche
* È possibile assegnare a una variabile il valore di un’espressione aritmetica o di un’espressione logica
* Sono presenti inoltre istruzioni if, if-else e while. La condizione dell’if e del while deve essere il risultato di un’espressione logica e deve essere contenuta tra parentesi tonde. I corpi dell’if, dell’else e del while devono essere racchiusi in parentesi graffe (blocchi), inoltre questi devono presentare sempre almeno un’istruzione
* Tutte le istruzioni devono terminare con il ‘;’. Fanno eccezione if, else e while.
* È possibile definire nuovi variabili all’interno dei blocchi delimitati da parentesi graffe. È possibile anche usare nomi di variabili già definite. Queste variabili hanno una visibilità locale, quindi le variabili e i loro valori saranno eliminati all’uscita dal blocco. Variabili definite in blocchi più esterni sono invece visibili in tutti i blocchi innestati
* È possibile aggiungere dei commenti, questi possono essere presenti su una sola riga alla volta e devono essere preceduti da ‘//’ e la loro fine è identificata dal carattere speciale ‘\n’ (new line).

**Scelte progettuali**

**Analizzatore lessicale (Flex)**

L’analizzatore lessicale si occupa di individuare i token del linguaggio. In base al token rilevato vengono compiute le opportune azioni. Per esempio, nel caso di “int”, “if”, “main”, ecc… viene restituito il relativo token, nel caso in cui si rilevi un “id” (cioè il nome di una variabile) il lessema viene duplicato e salvato in un campo di yylval, che è la variabile globale che permette a Flex di passare il valore semantico associato al token all’analizzatore sintattico (Bison), e viene poi anche restituito il relativo token. Discorso analogo anche per il token “num”: in questo caso il lessema viene convertito in un intero, salvato in un altro campo di yylval e viene ritornato il relativo token.

**Analizzatore sintattico (Bison)**

In Bison è stato ridefinito il YYSTYPE tramite la direttiva ‘%union’. Sono presenti 3 campi: number (intero), identifier (stringa) e label (una struttura utilizzata esclusivamente per le istruzioni di controllo di flusso). Sono stati definiti i vari token, la loro precedenza e l’associatività degli operatori aritmetici e logici.

È stata definita una grammatica context-free che rispecchia le specifiche del linguaggio precedentemente illustrate. L’analizzatore sintattico si occupa di tradurre il codice sorgente in Three Address Code, stampando quindi in uscita le istruzioni equivalenti in 3AC.

Non viene fatto uso di attributi ereditari per la gestione delle istruzioni di controllo del flusso (if e while), ma vengono utilizzati solo attributi sintetizzati. L’implementazione di queste istruzioni avviene tramite l’uso di marker inseriti nella grammatica.

*IF:*

L’istruzione if viene gestita tramite la creazione di due nuove etichette. La prima viene salvata in una variabile temporanea mentre la seconda viene memorizzata nel campo ‘label1’ di $$ (nonterminale ifstmt). Vengono a questo punto stampate le istruzioni relative al salto condizionato (prima etichetta) e al salto incondizionato (seconda etichetta). Viene poi stampata l’etichetta relativa al salto condizionato. L’azione semantica associata alla riduzione della regola permette di stampare l’etichetta era stata memorizzata in $$ in quanto l’informazione è stata passata tramite un attributo sintetizzato (‘label1’) dal nodo figlio al nodo padre nell’albero di parsing.

È stata inoltre generata un’altra etichetta salvata nel campo ‘label2’ il cui uso è però relativo all’istruzione if-else.

*IF-ELSE:*

L’if viene gestito esattamente come nel caso precedente. Nelle produzioni che identificano l’if-else sono presenti due marker: ‘er1’ e ‘er2’. Il primo effettua la stampa del salto incondizionato che bisogna eseguire nel caso in cui il programma sia entrato nel corpo dell’if (in modo da evitare di eseguire il corpo dell’else). Il secondo invece stampa l’etichetta relativa al corpo dell’else. Terminato il corpo dell’else vengono stampate l’istruzione di salto incondizionato (uscita dall’else) e in seguito la relativa etichetta.

*WHILE:*

Anche in questo caso la gestione è molto simile al caso IF. Viene solamente aggiunto un marker nella produzione di whilestmt che permette di stampare l’etichetta relativa al salto per ritornare all’inizio del while, in modo da testare nuovamente la condizione.

**Nota sui file consegnati**

La cartella consegnata contiene due versioni del compilatore: la prima produce in output il 3AC relativo al codice in input e gli esempi di input/output sono in Versione 1/Esempi.

La seconda versione invece ha un file Bison leggermente modificato. La grammatica è la stessa ma questa volta anziché stampare semplicemente in output le istruzioni in 3AC viene creato un file eseguibile C (se ovviamente non vengono riscontrati errori durante la compilazione). L’analizzatore sintattico crea inoltre una libreria “temp.h” che contiene le librerie “uthash.h” (usata per la symbol table) e “list.h” (usata per lo stack di symbol table) e dichiara la variabile “list” inizializzandola a NULL e tutte le variabili temporanee necessarie per l’esecuzione del file C di output. In questo caso gli output sono ognuno all’interno di una cartella separata (in Versione 2) insieme alle librerie necessarie per compilare e mandare in esecuzione il file C.

Le due versioni hanno in comune il file Flex, le librerie “list.h” e “uthash.h” e i file di input.

**Lavoro di:**

**DIEGO SAPORITO MATRICOLA: 1197676**

**BENVENUTO BORDIGNON MATRICOLA:1210175**