

# Linguaggi Formali e Compilatori

## Proff. Breveglieri, Crespi Reghizzi, Morzenti

### Prova scritta <sup>1</sup>: Domanda relativa alle esercitazioni

#### 08/09/2011

COGNOME: .....  
 NOME: ..... Matricola: .....  
 Iscritto a:  Laurea Specialistica  V. O.  Laurea Triennale  Altro: ...  
 Sezione:  Prof. Breveglieri  Prof. Crespi  Prof. Morzenti

Per la risoluzione della domanda relativa alle esercitazioni si deve utilizzare l'implementazione del compilatore **Acse** che viene fornita insieme al compito.

Si richiede di modificare la specifica dell'analizzatore lessicale da fornire a **flex**, quella dell'analizzatore sintattico da fornire a **bison** ed i file sorgenti per cui si ritengono necessarie delle modifiche in modo da estendere il compilatore **Acse** con la possibilità di gestire gli operatori di *shift-rotate*.

	Variabile		
Linea	a	b	c
1 a = 3;	1 0...11 <sub>2</sub>	undef	undef
2 b = a \$>>\$ 1;	2 0...11 <sub>2</sub>	10...01 <sub>2</sub>	undef
3 c = b \$<<\$ 1;	3 0...11 <sub>2</sub>	10...01 <sub>2</sub>	0...11 <sub>2</sub>

(a) Codice

(b) Contenuto delle variabili

Figura 1: Esempio di operatori di *shift-rotate*

Gli operatori di shift standard, << e >>, effettuano uno scorrimento a sinistra/destra di un valore. Sia *op* uno dei due operatori. Calcolare il valore dell'espressione *a op b* equivale a spostare a sinistra/destra i bit di *a* di *b* posizioni. I bit in eccesso sono scartati, mentre i bit lasciati liberi dallo spostamento sono impostati a 0.

Gli operatori di *shift-rotate*, \$<<\$ e \$>>\$, hanno un comportamento analogo: effettuano uno spostamento di un valore. A differenza degli operatori di shift standard essi non scartano alcun bit. I bit che normalmente sono eliminati perchè spostati oltre un confine, vengono reintrodotti dall'altro. Figura 1(a) contiene un esempio, mentre in Figura 1(b) è riportato il valore delle variabili dopo l'esecuzione dell'istruzione riportata in Figura 1(a) nella linea corrispondente.

Nell'implementazione, si ricordi che tutte le espressioni in *Acse* hanno tipo **int**.

<sup>1</sup>Tempo 45'. Libri e appunti personali possono essere consultati.  
 È consentito scrivere a matita. Scrivere il proprio nome sugli eventuali fogli aggiuntivi.

Si espliciti ogni eventuale ulteriore assunzione che sia ritenuta necessaria a completare la specifica data.

1. Definire i token (e le relative dichiarazioni in `Acse.lex` e `Acse.y`) necessari per ottenere la funzionalità richiesta. (3 punti)

La soluzione è riportata nella patch allegata.

2. Definire le regole sintattiche (o le modifiche a quelle esistenti) necessarie per ottenere la funzionalità richiesta. (4 punti)

La soluzione è riportata nella patch allegata.

3. Definire le azioni semantiche (o le modifiche a quelle esistenti) necessarie per ottenere la funzionalità richiesta. (18 punti)

La soluzione di questo tema d'esame dipende quindi dal tipo di dato con il quale si lavora. Per osservare l'effetto degli *shift-rotate* come riportato nell'esempio di Figura 1(b) è necessario avere a disposizione dei tipi di dato senza segno. Sfortunatamente la macchina *Acse* supporta come unico tipo di dato l'*intero con segno*.

Essendo costretto ad utilizzare un tipo di dato non corretto, il compilatore, l'assemblatore e l'interprete eseguono operazioni corrette su dati scorretti! Ne consegue, che l'output del programma non è quello atteso.

Posto di avere a disposizione il supporto ai tipi di dato senza segno, la soluzione proposta è corretta. Si controlli l'output del compilatore, `output.asm` per verificare che le istruzioni generate siano corrette.

Per ulteriori informazioni, si veda lo *standard ISO/EIC 9899:TC3 (C99)*, alla Sezione 6.2.6.2, *Integer types*.

La soluzione proposta modifica il framework delle espressioni di modo da introdurre due nuove istruzioni simulate. È ovviamente possibile non introdurre alcuna nuova istruzione ed utilizzare il framework delle espressioni senza modificarlo, agendo ad un livello più alto.

4. Dato il codice di Figura 2:

```
1 b || c | d && b
```

Figura 2: Operazioni logiche

Scrivere l'albero sintattico relativo partendo dalla grammatica Bison definita in `Acse.y` iniziando dal non-terminale `exp` a precedenza più bassa. (5 punti)

La soluzione è riportata in Figura 3.

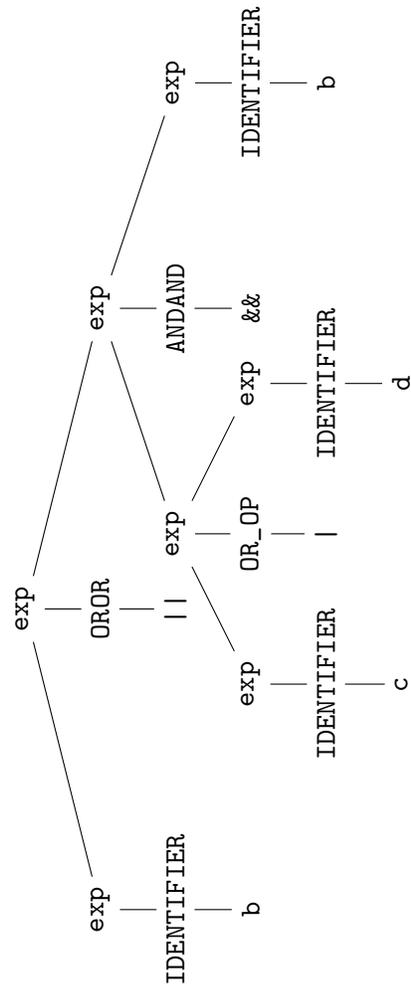


Figura 3: Albero sintattico del codice riportato in Figura 2

5. (Bonus) Si supponga di voler implementare una direttiva del compilatore che permette di selezionare il comportamento degli operatori di scorrimento `<<` e `>>`:

```
1 #pragma acse shift X
2
3 int a = 7;
4 a = a << 4;
```

Figura 4: Esempio d'uso della direttiva `shift`

Il token `X` può assumere due valori:

`classic` - gli operatori di scorrimento si comportano usualmente

`rotate` - gli operatori di scorrimento si comportano come degli *shift-rotate*

Come bisogna modificare il compilatore *Acse* per supportare questa caratteristica?

La modifica al compilatore è banale. Inizialmente bisogna introdurre le definizioni dei token e le regole grammaticali per riconoscere la nuova direttiva.

Successivamente, allo stato del compilatore è necessario aggiungere una variabile `shift_mode` che indica il comportamento degli operatori. Il suo valore di default equivale al comportamento `classic`.

L'azione semantica associata alla regola di riconoscimento della direttiva imposta opportunamente la variabile `shift_mode`.

Infine bisogna modificare l'azione semantica associata al riconoscimento delle espressioni `exp SHL_OP exp` e `exp SHR_OP exp` in modo che venga generato il codice corrispondente al comportamento `classic` piuttosto che a quello `rotate`.

## Applicare una patch

Sul sito del corso è disponibile una patch contenente la soluzione del tema d'esame per quanto riguarda la modifica della macchina **Acse**.

Per applicare la patch:

1. scaricare la macchina **Acse** versione 1.1.0
2. scaricare la patch `soluzione-08-09-11.diff`
3. scompattare l'archivio contenente la macchina **Acse**
4. usando il terminale, portarsi nella directory in cui è stata estratta la macchina **Acse**
5. copiare in tale cartella la patch
6. applicare la patch tramite il comando

```
patch -p1 < soluzione-08-09-11.diff
```

La patch è un normalissimo file di testo, contenente le differenze tra la versione di **Acse** con implementata la soluzione dell'esame e la versione 1.1.0.

Le righe che iniziano con il carattere `+` sono state aggiunte, mentre quelle con il carattere `-` sono state rimosse.