

Linguaggi Formali e Compilatori

Proff. Breveglieri, Crespi Reghizzi, Morzenti

Prova scritta ¹: Domanda relativa alle esercitazioni

04/03/2011

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 una forma semplificata di eccezioni.

```
1  try {  
2    ...  
3    throw;  
4    ...  
5  } catch {  
6    ...  
7  }
```

Figura 1: Esempio di eccezione

All'interno di un sorgente **Acse** deve essere possibile definire una regione di codice protetta dal costrutto **try catch**. Esso è analogo ai costrutti di gestione delle eccezioni dei linguaggi mainstream, con alcune semplificazioni:

- il blocco **try** è *obbligatoriamente* seguito da un blocco **catch**
- se nel blocco **try** viene eseguita l'istruzione **throw**, allora viene eseguito il blocco **catch**

Per semplicità, si considerino inoltre le seguenti limitazioni:

- l'istruzione **throw** non accetta alcun argomento
- il blocco **catch** cattura tutte le eccezioni lanciate nel blocco **try** associato

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

- per quanto riguarda la definizione delle azioni semantiche, si supponga che i costrutti `try catch` non si possano innestare

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 è riportata nella patch allegata.

4. Dato il codice di Figura 2:

```
1 ptr = ptr | (fast_locked << 1) | gc_locked;
```

Figura 2: Esempio di pointer-packing

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

La soluzione è riportata in Figura 3.

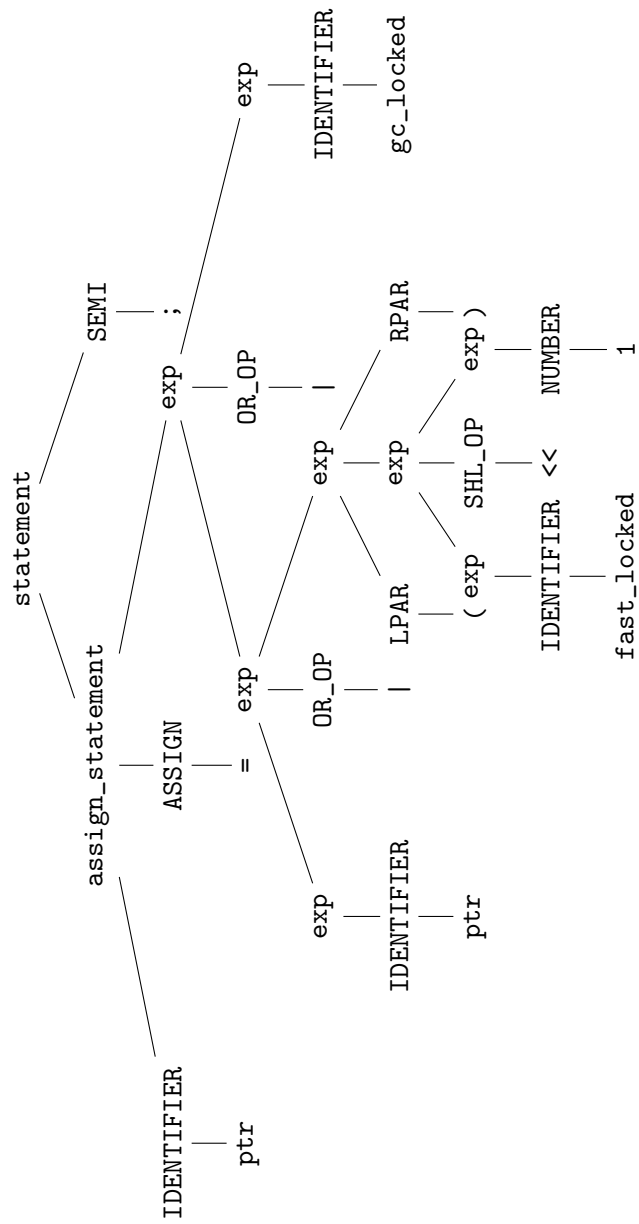


Figura 3: Albero sintattico del codice riportato in Figura 2

5. (Bonus) Si supponga di voler permettere la gestione di blocchi `try catch` annidati. Come deve essere modificata la soluzione proposta precedentemente?

La soluzione proposta si appoggia alla variabile `cur_try` per accedere al blocco `try catch` correntemente attivo. Nel caso in cui si annidino più costrutti l'uno all'interno dell'altro si può naturalmente estendere l'algoritmo in modo da tracciare il costrutto `try catch` attivo al livello corrente di annidamento. La struttura dati necessaria per tale compito è un semplice pila. Si veda la patch allegata per i dettagli implementativi.

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-04-03-10.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-04-03-10.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.