Primeira Lista de Exercícios de Construção de Compiladores.
Primeiro Semestre de 2001. Departamento de Computação – UFSCar.
José de Oliveira Guimarães.

Muitas das questões desta lista admitem respostas subjetivas. As questões das provas serão mais exatas e com mais informações.
Considere que a linguagem do último compilador, o 10, se chame S2.
Quando se pede para fazer a ASA de uma expressão, queremos dizer fazer o desenho com objetos representados por círculos e referências por setas.

1. Cite as mudanças que devem ser feitas no seu analisador léxico para que ele faça uma estatística do uso das palavras chave em programas S2. A saída do seu programa, que usa o analisador léxico, poderia ser

```
and       : 2
begin     : 3
boolean   : 1
...
write     : 10
```

significando que no programa S2 analisado havia 2 `and`'s, 3 `begin`'s, etc.

2. Faça um programa completo em Java que leia um programa em C e produza como saída o mesmo programa, mas onde cada comentário foi substituído por um único espaço em branco. A leitura deve ser feita da entrada padrão. Admita que o programa em C não possua *strings* de caracteres. Isto dificultaria o trabalho pois poderia haver `/*` dentro de uma *string*.

3. Repita o exercício anterior, mas agora retirando os comentários de C++.

4. Como, a partir da gramática, são identificados os símbolos a serem encontrados pelo analisador léxico ?

5. Como o seu analisador deveria ser modificado para que ele considerasse letras maiúsculas e minúsculas como equivalentes ? Isto é, as *strings* `BEGIN`, `BEgin` e `begin` seriam consideradas iguais.

6. Dois símbolos terminais de um programa na linguagem LE, cuja gramática é mostrada a seguir, devem ser separados por caracteres brancos e números são formados por um único dígito. Faça, em Java, um analisador léxico e sintático para esta linguagem, sendo que o analisador sintático deve também construir a ASA do programa.

E ::= T E′
E′ ::= + T E′ | ϵ
T ::= F T′
T′ ::= $*$ F T′ | ϵ
F ::= Numero | (E)

7. Faça, em Java, um analisador léxico, sintático e semântico para a linguagem gerada pela gramática

S ::= Expr ExprList

ExprList ::= ϵ | Expr ExprList
Expr ::= "if" Expr "then" Expr "else" Expr "endif" | E > E | E == E | E
E ::= E + T | E "or" T | T
T ::= T * F | T "and" F | F
F ::= Numero | "(" Expr ")"

Qualquer número de caracteres brancos podem aparecer entre dois terminais em um programa e Numero representa números inteiros positivos com qualquer quantidade de dígitos. O tipo da expressão que se segue ao "if" deve ser booleana e os tipos das expressões que se seguem ao "then" e "else" devem ser iguais. As expressões aritméticas seguem as regras usuais.

A análise semântica deve ser feita sobre a ASA gerada na análise sintática. Crie as classes necessárias para a ASA.

8. Baseada na linguagem descrita no item anterior, faça a ASA da expressão

```
if 2 > 5 then 2*3 else 1 + (if 1 then 3 else 5 endif) endif
```

9. (Exercício muitíssimo importante — absolutamente fundamental) Faça uma expressão em Java que crie a ASA para a expressão do item anterior. Por exemplo, a expressão

```
if 2 > 1 then 0 else 3
```

poderia ser criada por

```
IfExpr ifExpr = new IfExpr(
    new CompositeStatement( new NumeroExpr(2), Symbol.GT, new NumeroExpr(1) ),
    new NumeroExpr(0),
    new NumeroExpr(3) );
```

Faça exercício equivalente com outras expressões de outras gramáticas.

10. Existem duas formas de descobrir, durante a análise léxica, se uma *string* corresponde a uma palavra chave ou um identificador e recuperar os dados do identificador:

- fazendo uma busca em um vetor de *strings* contendo apenas as palavras chaves. Se nada for encontrado, é feita uma busca na TS (Tabela de Símbolos);
- fazendo uma busca na TS, que contém tanto os identificadores quanto as palavras chaves.

Discuta as vantagens de 1 sobre 2 e vice-versa. Em 1, a busca no vetor de *strings* pode ser binária ou através de uma tabela *hash*.

11. Qual a função da análise sintática ?

12. A TS pode ser feita utilizando-se:

1. uma pilha implementada como uma lista encadeada de objetos alocados dinamicamente;
2. uma pilha implementada como um vetor estático. Cada elemento do vetor aponta para um objeto de `Symbol` ou suas subclasses. Admita que o tamanho do vetor seja suficiente para qualquer compilação. Assuma que todos os identificadores (variáveis, parâmetros, procedimentos, ...) herdem de Symbol;

3. uma tabela *hash* implementada como um vetor onde cada entrada aponta para uma pilha formada por uma lista encadeada de objetos alocados dinamicamente;

4. a mesma tabela *hash* de 3, mas com todos os objetos do mesmo nível léxico ligados através de uma outra lista encadeada. Isto é, haveria um ponteiro `level[0]` que apontaria para uma lista encadeada com todos os objetos correspondentes aos símbolos de nível léxico `0`. Esta lista seria independente das listas utilizadas pela tabela *hash*.

5. uma tabela *hash* como em 3 mas com um conjunto `level[0]` com um objeto da classe `IntSet`. Este objeto contém os índices na tabela *hash* (que é um vetor de listas) de todas as listas que contém símbolos do nível léxico `0`. Esta implementação foi utilizada na classe `SymbolTable` fornecida aos alunos.

Compare estas implementações quanto à eficiência:

1. na inserção de um elemento na tabela;

2. na busca por um elemento;

3. na eliminação de um nível léxico;

4. na quantidade de utilização de memória. Isto é, quais implementações utilizam mais memória.

Esta comparação não pode ser completamente exata em todos os casos acima pois não fornecemos todos os detalhes das implementações citadas.

13. Por que o compilador constrói uma ASA para o programa fonte sendo compilado se ele possui as mesmas informações, durante a compilação, na tabela de símbolos e no programa fonte ?

14. Faça a ASA do programa mostrado a seguir.

```
  var i, n : integer;
      escreveu : boolean;
begin
escreveu = false;
i = 1;
read(n);
if not ( n <= 1 )
then
  escreveu = true;
  while i <= n do
    begin
    write( i );
    i = i + 1;
    end;
endif;
end
```

15. Suponha que tenham sido introduzidos na linguagem S2 os seguites comandos :

1. `Records` como em Pascal ou `structs` de C:

```
var a : record
          x, y : integer;
        end
```

2. O comando

```
loop   C1; C2; ... Cn; end
```

   que é uma repetição infinita dos comandos `C1`, `C2`, ... `Cn`. O laço termina com uma instrução "`break`" em um dos comandos `Ci`. Observe que será necessário modificar a produção "Statement" da linguagem.

3. Vetores, como em Pascal:

```
var a : array[Inicio..Fim] of T;
```

4. Tipos enumerados, como em C/Pascal:

```
var cor : enum ( azul, vermelho, branco ) ;
```

   Não há um tipo associado às constantes, como é possível em C++.

5. Tipos, como em Pascal:

```
type NumElem = integer;
```

6. Um comando de repetição da forma:

```
loop C1; C2; ... Cn times E;
```

   onde `Ci` são comandos, `loop` e `times` são palavras chave e `E` é uma expressão. A seqüência de comandos `Ci` é repetida `N` vezes, onde `N` é o valor de `E`, calculado antes da execução de qualquer `Ci`. Se `N` for menor do que `0`, então `N` é zerado.

Para cada um destes comandos, especifique/defina:

- as regras da gramática;
- a classe da ASA, especificando as variáveis de instância e um construtor (não é necessário especificar outros métodos);
- os tokens que o Lex deve reconhecer para a regra (especifique mesmo se o seu compilador já reconhece estes tokens);
- as regras semânticas que devem ser obedecidas, em português;
- os métodos da classe `Analyzer` que analisam as regras para o comando. Escreva não só a interface como o código destes métodos. Admita que as palavras chave `record`, `loop`, `array`, `enum` e `type` já foram encontradas antes destes métodos serem chamados, como é usual.

16. Responda:

- Quando variáveis são retiradas da tabela de símbolos ?
- É possível que um objeto seja referenciado pela TS e pela ASA simultaneamente ?
- Existem objetos que nunca farão parte da ASA, mas que são referenciados pela TS ?
- Objetos representando os tipos `integer` e `boolean` e as constantes `true` e `false` são inseridos na TS do compilador apresentado em aula ? Eles são referenciados pela ASA ?