Exercises

Exercises

Andrea Polini

Formal Languages and Compilers
Master in Computer Science
University of Camerino

ToC

Lexical Analysis

- All strings of lowercase letters that contain the five vowels in order
- All strings of lowercase letters in which the lettersare in ascending lexicographic order
- All strings of lowercase letters that begin and end in $'a'$
- All strings of digits that contain no leading zeroes
- All strings of a's and b's that contain no three consecutive b's
- All strings of a's and b's that do not contain the substring $'abb'$
- All strings of a's and b's with an even number of a's and an odd number of b's

Interesting exercises are those related to the demonstration of closure properties for regular languages (i.e. given the two regular languages $\mathcal{L}_1$ and $\mathcal{L}_2$ demonstrate that $\mathcal{L}_1 \cup \mathcal{L}_2$ is a regular language)

ToC

Si consideri il linguaggio $\mathcal{L} = \{a^n w a^n | \; n \geq 1 \; \wedge \; w \in \{a, b, c\}^* \; \wedge \; w = \overline{w}\} \cup \{b^n w b^n | \; n \geq 0 \; \wedge \; w \in \{a, b, c\}^* \; \wedge \; w = \overline{w}\}$ e se ne determini la classe di appartenenza in accordo alla classificazione di Chomsky.

- si derivi una grammatica capace di generare il linguaggio.

Si consideri il linguaggio $\mathcal{L} = \{a^{3n}b^{2n}c^{m} | m \geq 1, n \geq 0\}$ e si risolvano i seguenti punti:

1. Determinare la classe del linguaggio $\mathcal{L}$ in accordo alla classificazione di Chomsky definendo altresì le differenti componenti di un opportuno automa capace di accettare il linguaggio $\mathcal{L}$
2. Derivare una grammatica G, che non contenga ε-produzioni, tale che $L(G) = \mathcal{L}$

Si consideri la seguente grammatica G:

$$S \longrightarrow A \mid C \quad A \longrightarrow ccAa \mid a \quad C \longrightarrow cC \mid c$$

e si risolvano i seguenti punti:

1. Senza aver derivato gli insiemi FIRST e FOLLOW si decida se la grammatica G è LL(1) e perché?
2. Si derivino gli insiemi FIRST, FOLLOW e *nullable* per G indicando tutte le iterazioni necessarie.
3. Si costruisca l'automa LR(0) e le corrispondenti tabelle di parsing LR(0) ed SLR(1) decidendo per ogni tipo di parsing se è applicabile e perché.
4. Applicando una tra le due tipologie di parsing (a scelta LR(0) o SLR(1) se entrambe possibili) si mostrino le azioni del parser sulla stringa "*ccccaaa*"

Si consideri la grammatica definita nella slide precedente. Si derivi una nuova grammatica G’ rimuovendo i problemi identificati nel punto 1 dello stesso esercizio. Si discuta se esiste un $k \geq 1$ tale che la grammatica G’ risulti LL(k).

Suggerimento: può essere utile osservare che il linguaggio generato dalla grammatica G’ è descritto dalla seguente unione d’insiemi:

$$L(G') = \{c^n \mid n \geq 1\} \cup \{c^{2m}a^{m+1} \mid m \geq 0\}$$

Syntax analysis

LL Parsing

Let's consider the following language:

$$\mathcal{L} = \{a^n bc^k | n \geq 0, k > 0\} \cup \{b^n ac^k | n > 0, k \geq 0\}$$

- define a grammar that generates (all and only) the words of the language
- decide if the grammar is LL
- in case it is, derive the parsing table and apply it to the recognition of the word 'bba', otherwise try to modify the grammar so to derive an LL parsable grammar.

Syntax Analysis

Noteworthy languages

Consider the following languages and define grammars in order to parse them with LL and LR parsers:

- $\mathcal{L}_1 = \{a^n b^n | n \in \mathbb{N}\}$
- $\mathcal{L}_2 = \{a^n b^m | n \in \mathbb{N} \wedge n \geq m\}$
- $\mathcal{L}_3 = \{w \in \{a, b\}^* | w = \overline{w}\}$

Consider the following grammars and define grammars parsable with LL or LR parsing strategies:

- $\mathcal{L} = \{a^n a^n a | n \in \mathbb{N}\}$

Syntax Analysis

Noteworthy languages

Consider the following languages and define grammars in order to parse them with LL and LR parsers:

- $\mathcal{L}_1 = \{a^n b^n | n \in \mathbb{N}\}$
- $\mathcal{L}_2 = \{a^n b^m | n \in \mathbb{N} \wedge n \geq m\}$
- $\mathcal{L}_3 = \{w \in \{a, b\}^* | w = \overline{w}\}$

Consider the following grammars and define grammars parsable with LL or LR parsing strategies:

- $\mathcal{L} = \{a^n a^n a | n \in \mathbb{N}\}$

Syntax Analysis

LR Parsing

Consider the grammar:

$$Z \to S \;\; S \to AA \;\; A \to aA \;\; A \to b$$

- Build the table for the LR(1) parser
- decide if the LALR parser can be applied
- apply one of the two parser to the acceptance of string *abaab*

ToC

Let's $\mathcal{G}$ the grammar defined by the following productions, and that permits to represent nested lists of numbers:

$$S \longrightarrow Y \quad Y \longrightarrow Y; Z \mid Z \quad Z \longrightarrow (Y) \mid digit$$

The following are samples of correct sentences generated by the grammar: `((10;13);17)`, `((3);(7;8);15)`
Answer to the following requests:

1. define attributes and semantic rules (SDD) for the grammar, in order to permit:
   - the calculation of the sum of all numbers appearing in a sentence
   - the calculation of the total number of parenthesis opened in the sentence
   - the printing of the position in the list for each number in the sentence (consider that to implement this feature you could need accessory attributes).
2. show the evaluation tree for the sentence `((5);(8))`
3. modify the grammar in order to be parsable by an LL(1) parser modifying productions and rules to obtain a suitable L-attributed translation scheme.