Compiler Construction

Exercises

1 Review of some Topics in Formal Languages

1.

(a) Prove that two words x, y commute (i.e., satisfy $xy = yx$) if and only if there exists a word w such that $x = w^m, y = w^n$ for some non-negative integers m, n.

(b) Characterize all pairs of words x, y, z satisfying the equality $x^2y^2 = z^2$.

(c) Let $k \geq 2$ be an arbitrary fixed integer. Characterize all pairs of words x, y, z satisfying the equality $x^ky^k = z^k$.

(d) Characterize all triples of words x, y, z satisfying the equality $xyz = yzx$.

(e) Characterize all triples of words x, y, z satisfying the equality $xyz = zyx$.

2. Let $\Sigma = \{a, b, \ldots, z, 0, 1, \ldots, 9, \_\}$. Let L be the language consisting of all non-empty words over Σ which (i) do not start with a digit, (ii) do not contain two consecutive occurrences of '_', and (iii) do not end with '_'.

(a) Construct a regular expression r such that $L(r) = L$.

(b) Construct a DFA accepting L.

(c) How many words of each length n does L include?

3. Show that the following languages over Σ are regular:

(a) The collection of all words whose length is congruent to k modulo m (where $0 \le k \le m-1$).

(b) The collection of all words having the property that, for some fixed positive integer k and all pairs of letters σ_1, σ_2, the difference between the number of occurrences of σ_1 and of σ_2 in every prefix does not exceed k.

(c) The collection of all words containing exactly m_1 occurrences of the word w_1, exactly m_2 occurrences of the word w_2, ..., exactly m_k occurrences of the word w_k.

4. How many words of length n do the languages, corresponding to the following regular expressions, contain?

(a) $\sigma_1^* \sigma_2^* \dots \sigma_k^*$ (where the σ_i's are all distinct).

(b) $(0 \cup 11 \cup 22 \cup 3333 \cup 4444 \cup 5555 \cup 6666)^*$.

5. Show that, if L is a regular language, then so are the following languages:

(a) The language obtained by replacing, in each word of L, each occurrence of aa by b. (The replacement is done consecutively; thus, the block a^{2k} is replaced by b^k and the block a^{2k+1} by $b^k a$.)

(b) The language obtained from L by deleting the second last letter in every word of length 2 or more:

(c) The language consisting of all words in L in which the number of occurrences of the letter σ is r modulo d.

(d) The language obtained from L by omitting from each word of L any number of occurrences of the letter σ. (For example, if $\sigma_1\sigma^4\sigma_2\sigma^7\sigma_3 \in L$, then both words $\sigma_1\sigma^2\sigma_2\sigma^7\sigma_3$ and $\sigma_1\sigma^3\sigma_2\sigma\sigma_3$ belong to the language we construct.)

6. Let L_1, L_2 be two languages over Σ. Show that the "equation"

$$L_1 L \cup L_2 = L$$

has a solution. Moreover, if L_1, L_2 are both regular (or both context-free), then there exists a solution L with the same property.

7. Let $\Sigma = \{a, b, c\}$. Construct DFA's accepting the following languages:

(a) All words containing neither aaa nor aca as a subword.

(b) All words containing either $ababa$ or $abcba$ as a subword.

(c) All words containing both a^2 and b^2, but not c^2, as subwords.

8. Construct NFAs accepting the languages corresponding to the following regular expressions:

(a) $bab(bba \cup abb)^* bab$.

(b) $ab(ab \cup bba)^* \cup a(ba \cup \phi^*)bba$.

9. Present an algorithm which, given a DFA, returns all words of minimal length accepted by it (or an error if it accepts the empty language).

10. Present an algorithm that, given a DFA, returns a DFA accepting a language strictly containing the language accepted by the original DFA and strictly contained in Σ^* (or an error if no such language exists).

11.

(a) Show that an infinite regular language may be written as an infinite disjoint union of infinite regular languages.

(b) Show that an infinite context-free language may be written as an infinite disjoint union of infinite context-free languages.

(c) Does an infinite context-free language necessarily contain an infinite regular language?

(d) Does an infinite language, accepted by a Turing machine, necessarily contain an infinite context-free language?

12. Show that the following languages are not regular:

(a) $\{a^m b^n c^{m+n} : m, n \geq 0\}$.

(b) $\{a^k b^l c^m d^n : k, l, m, n \geq 0, |\{k, l, m, n\}| \geq 2\}$.

(c) $\{0^{m^3+n^3} : m, n \geq 0\}$.

(c) $\{0^{l^2+m^3+n^7} : l, m, n \geq 0\}$.

13. Given a set of non-negative integers, the set of their expansions in base 10 forms a partial language of $\{0, 1, \ldots, 9\}^*$. For each of the following sets show that the corresponding language is regular or not (as indicated):

(a) All powers of 1000 (regular).

(b) All powers of 7 (not regular).

(c) All perfect cubes (not regular).

(d) $\{n! : n \geq 0\}$ (not regular).

(e) All numbers whose distance from some number of the form $777\ldots7$ is at most 7 (regular).

14. Find the languages accepted by the grammars:

(a)

$$\begin{aligned} S &\to AB \mid BA, \\ A &\to aAb \mid \varepsilon, \\ B &\to bBa \mid \varepsilon. \end{aligned}$$

(b)

$$\begin{aligned} S &\to \varepsilon, \\ S &\to \alpha_i S \alpha_j, \qquad 1 \leq i, j \leq m \end{aligned}$$

(where $\Sigma = \{\alpha_1, \alpha_2, \ldots, \alpha_m\}$).

15. Let $a_1, a_2, \ldots, a_r$ be positive integers and $b_1, b_2, \ldots, b_r$ non-negative integers. Consider the language $\{\sigma_1^{a_1 n + b_1} \sigma_2^{a_2 n + b_2} \ldots \sigma_r^{a_r n + b_r} : n \geq 0\}$, where $\sigma_1, \sigma_2, \ldots, \sigma_r$ are any letters, not necessarily distinct. Specify the conditions under which this language is context-free.

16. Construct DFA's accepting the same languages as the grammars:

(a)

$$\begin{aligned} S &\to abS \mid baS \mid bcA, \\ A &\to babB, \\ B &\to bcbaC, \\ C &\to abaC \mid bC \mid \varepsilon. \end{aligned}$$

(b)

$$\begin{aligned} S &\to acbS \mid bcbS \mid bA \mid aC \mid aca, \\ A &\to bS \mid caS \mid baB \mid babC \mid a^2, \\ B &\to bbA \mid a^2B \mid a^2 \mid b^2 \mid c, \\ C &\to bcS \mid aA \mid bB \mid cC \mid cba \mid \varepsilon. \end{aligned}$$

17. Construct pushdown automata accepting the same language as the grammars:

(a)

$$S \to \varepsilon \mid SbS \mid AbS \mid SaB,$$
$$A \to Bb \mid a^2,$$
$$B \to b^2S \mid baA \mid b^2.$$

(b)

$$S \to SaBS \mid bAS \mid ba^2,$$
$$A \to BaAa \mid aSB \mid ab \mid \varepsilon,$$
$$B \to b^3B \mid ASA \mid baba.$$

18. Consider the grammar G defined by:

$$S \to S + S \mid S * S \mid a \mid b \mid c.$$

(a) Is the language $L(G)$ regular? If yes – find a DFA accepting the same language as G and a regular grammar G' equivalent to G. If not – prove it.

(b) How many words of each length does the language $L(G)$ include?

(c) How many derivation trees yield each word in $L(G)$? (Hint: Find a recurrence for the sequence which expresses the required number as a function of the word's length.)

(d) How do your answers to the first three parts change if we add the productions $S \to S - S$ and $S \to S/S$?

2 Lexical Analysis

19.

(a) Write a regular expression r such that $L(r)$ consists of all identifiers (i.e., all strings of letters and digits, starting with a letter), with the exception of the three strings "if", "int" and "integer".

(b) Construct a DFA which recognizes the string "if" as a token of type IF, the strings "int" and "integer" as tokens of type NUM, and all other identifiers as tokens of type ID.

20.

(a) Given an arbitrary fixed integer $d \geq 2$, construct a regular expression r such that $L(r)$ consists of all base d expansions of even (positive) integers. (Thus, the alphabet consists of all digits in base d.)

(b) Construct a DFA which recognizes every non-negative number, expanded in base d, as EVEN or as ODD.

21.

(a) Given a regular expression r, define a regular expression $\vec{r}$ for vectors (of any non-negative length) of elements of type r. The entries of a vector are separated by commas and grouped by parentheses.

(b) Given a DFA recognizing elements of type r, construct a DFA which recognizes vectors of such elements.

(c) Can you design your DFA so as to recognize vectors whose length is (i) 3 modulo 10? (ii) a prime?

(d) The regular expression $\vec{\vec{r}}$ represents vectors of vectors of elements of type r. Can you construct a regular expression $[r]$ for matrices (i.e., vectors whose entries are vectors of the same length) of elements of type r?

22.

(a) For any $n \geq 0$, write a regular definition for the language of balanced parentheses of nesting level up to n.

(b) Write a computer program which, for given n, will output a DFA for this language. The DFA should inform of the nesting level. (Represent the DFA in any way you like – by a transition table, a graph, etc.)

23. Write a regular definition for the language of all strings over $\{a, b, \ldots, z\}$, not containing "if" as a substring.

24. In Java, the command

```
a = b+++--c;
```

passes compilation, whereas the command

```
a = b+++++c;
```

does not. Why?

25. In a certain computer language, identifiers are strings of letters and digits, starting with a letter, with the additional constraint that a character should not appear more than once in the name. (A lower-case letter and the corresponding upper-case letter are considered as distinct.) Construct a DFA, with a minimal possible number of states, recognizing identifiers. How many states does this DFA consist of?

3 Syntactic Analysis

26. Consider the grammar G_1, given by:

$E \rightarrow E + P \mid P,$

$P \rightarrow P * V \mid V,$

$V \rightarrow a \mid b \mid c.$

(a) Show that $L(G_1) = L(G)$, where G is the grammar defined in Question 18.

(b) Show that G_1 is unambiguous.

27. Consider the grammar G_1 given by:

$S \rightarrow iSeS \mid iS \mid \varepsilon,$

and the grammar G_2 given by:

$S \rightarrow M \mid U,$

$M \rightarrow iMeM \mid \varepsilon,$

$U \rightarrow iMeU \mid iS.$

(Intuitively, you should think of these grammars as the two grammars presented in class for conditional statements. Here we deal only with occurrences of the words *if* and *else*, represented by i and e, respectively. M and U stand for *matched* and *unmatched*, respectively.)

(a) Prove that $L(G_1) = L(G_2)$.

(b) Which words are obtained by a unique derivation tree in G_1?

(c) Write a program that, given a word in $\{i, e\}^*$, finds the number of derivation trees (if any) over G_1 producing this word. Prove that your algorithm works in polynomial time in the length of the input.

(d) Prove that G_2 is unambiguous.

(e) Write a program that, given a word in $\{i, e\}^*$, finds the unique derivation sequence over G_2 producing this word (and gives an error message if the word does not belong to $L(G_2)$).

28. Consider the following grammar, designed to solve the ambiguity problem of the **if-then-else** grammar presented in class:

stmt $\rightarrow$ **if** *cond* **then** *stmt* | *matched*,

matched $\rightarrow$ **if** *cond* **then** *matched* **else** *stmt* | *unconditionalStmt*.

Show that the grammar is still ambiguous.

29. Consider the grammar G given by:

$S \rightarrow SS\sigma_1 \mid SS\sigma_2 \mid \ \ldots \ \mid SS\sigma_r \mid \sigma_{r+1}$,

where $\sigma_1, \sigma_2, \ldots, \sigma_{r+1}$ are distinct terminals. Is the grammar unambiguous? If yes – prove your claim, if not – produce a word in $L(G)$ with two distinct derivation trees.

30. Consider the grammar G, given by:

$S \rightarrow FS \mid \varepsilon$,

$F \rightarrow aB \mid bA$,

$A \rightarrow a \mid bAA$,

$B \rightarrow b \mid aBB$.

(a) Show that $L(G)$ consists of all words over $\{a, b\}$ with an equal number of occurrences of a and of b.

(b) Show that G is unambiguous.

31. Construct an unambiguous grammar G, such that $L(G)$ consists of all words over $\{a, b\}$ in which the number of occurrences of a is not less than that of b.

32. For each of the following grammars, find the minimal k for which it is $LL(k)$. In case no such k exists, determine whether the grammar is unambiguous.

(a)

$$E \rightarrow M + E \mid M - E \mid M * E \mid M/E \mid M,$$
$$M \rightarrow V \mid V{+}{+} \mid V{-}{-},$$
$$V \rightarrow a \mid b \mid c.$$

(b)

$$E \to E + M \mid E - M \mid E * M \mid E/M \mid M,$$
$$M \to V \mid V{+}{+} \mid V{-}{-},$$
$$V \to a \mid b \mid c.$$

(c)

$$E \to M + E \mid M - E \mid M * E \mid M/E \mid M,$$
$$M \to V \mid V{+}{+} \mid V{-}{-} \mid {+}{+}V \mid {-}{-}V,$$
$$V \to a \mid b \mid c.$$

33. Two of the algorithms discussed in class enable us deciding which letters $X \in N \cup T$ have the property that there exists a word $w \in L$ such that $X \overset{*}{\Longrightarrow} w$, where L is any one of the two languages T^* and $\{\varepsilon\}$. Show that you can do the same for any regular language $L \subseteq T^*$. (You may use algorithms studied in the Automata course without detailing them.)

34. Find the FIRST sets of all non-terminals and right-hand sides of all rules for the following grammars:

(a)

$$S \to ABc,$$
$$A \to a \mid \varepsilon,$$
$$B \to b \mid \varepsilon.$$

(b)

$$S \to aS \mid AS \mid BAb,$$
$$A \to Aab \mid AB \mid \varepsilon,$$
$$B \to Aa \mid BbB \mid \varepsilon.$$

(c)

$$S \to aSe \mid A,$$
$$A \to bAe \mid B,$$
$$B \to cBe \mid d.$$

(d)

$$S \to ABCS \mid SS \mid aba,$$
$$A \to ACB \mid cb \mid \varepsilon,$$
$$B \to BCB \mid A \mid bc,$$
$$C \to AS \mid c.$$

(e)

$$S \to SS \mid AB \mid c,$$
$$A \to Aa \mid Aab \mid a \mid \varepsilon,$$
$$B \to bC \mid bB \mid Sb \mid b,$$
$$C \to cA \mid SC \mid c.$$

35. Find the FOLLOW sets of all non-terminals for the grammars in Question 34.

36. Determine whether each of the following grammars is $LR(k)$ for some k. If yes – find the minimal such k. (Hint: In some cases it may be helpful to find first the minimal k for which the grammar is $LL(k)$, and then use the fact that the grammar is $LR(k)$ for this k.)

(a) $S \to a^2Sb^3 \mid a^3b^4$.

(b) $S \to aSa \mid \varepsilon$.

(c) $S \to aSba \mid aba$.

(d) The grammar defined in Question 18.

(e) The grammar defined in Question 26.

(f) The grammar defined in Question 29.

37.

(a) Find the left contexts of all non-terminals and the $LR(0)$ contexts of all rules for the grammars in the preceding question.

(b) Same for the two grammars in Question 27.