Problem Set for Tutorial 7 — TDDD14/TDDD85

1 The Pumping Lemma for Context-Free Languages

Exercise 1. Show that the following languages are not context-free:

1. $L_1 = \{ a^j b^k a^l \mid 0 < j < k < l \}$,
2. $L_2 = \{ w \in \{a, b, c\}^* \mid w \text{ has an equal number of } a\text{'s, } b\text{'s and } c\text{'s} \}$,
3. $L_3 = \{ ww \mid w \in \{a, b\}^* \}$.

Solutions

Solution to Exercise 1.

1. Assume that L_1 is context-free. Then the pumping lemma holds. According to the lemma, there exists a number n such that if a string z, not shorter than n, is in L_1 (i.e. $|z| \geq n$, $z \in L_1$) then z can be split into five strings u, v, w, x, y:

   $$z = uvwxy$$

   such that

   $$|vx| \geq 1, \qquad |vwx| \leq n, \qquad uv^iwx^iy \in L_1 \text{ for all } i \geq 0$$

   We show that this leads to a contradiction. Take

   $$z = a^n b^{n+1} a^{n+2} \in L_1.$$

   Then there exist strings u, v, w, x, y satisfying the conditions above. We have two possibilities:

   (a) vwx does not overlap with the initial a^n. In other words, $u = a^n u'$ (for some u'). Take $i = 0$. Then $uv^0wx^0y = uwy = a^n b^k a^l$, for some k and l. The string $a^n b^k a^l$ is shorter than $a^n b^{n+1} a^{n+2}$ (as $|vx| \geq 1$), hence $k < n+1$ or $l < n+2$ (or both). In both cases $n < k < l$ is impossibe, so $uwy \notin L_1$ and we have a contradiction.

   (b) Otherwise vwx overlaps with the initial a^n. So it does not overlap with the final a^{n+2}, as $|vwx| \leq n$. In other words $y = y' a^{n+2}$, for some y'. Take $i = 2$. If uv^2wx^2y is not of the form $a^j b^k a^l$ then $uv^2wx^2y \notin L_1$, contradiction. If $uv^2wx^2y = a^j b^k a^l$ then $l = n+2$ but $j > n$ or $k > n+1$ (as $|vx| \geq 1$). Thus $j < k < l$ does not hold and $uv^2wx^2y \notin L_1$. Contradiction.

   This completes the proof. As usually in proofs with pumping lemmas, choosing an appropriate string z was crucial. For instance, if $z = ab^n a^{2n}$ then we may take $v = w = \epsilon$, $x = a$, $u = ab^n$ and and we do not obtain contradiction.

2. We know (from the lecture) that the language $M = \{\, a^i b^i c^i \mid i \geq 0 \,\}$ is not context-free. Notice that $L_2 \cap a^* b^* c^* = M$. Remember that the intersection of a context-free language with a regular language is context-free. So if L_2 were context-free then M would also be context-free. The latter is not true, so L_2 is not context-free.

   (A proof using the pumping lemma is also possible; take for instance $z = a^n b^n c^n$).

3. To show that L_3 is not a context-free language, we show that the pumping lemma does not hold for L_3. To do this, for every number n we have to find a string $z \in L_3$, $|z| \geq n$ such that for every splitting of z into five pieces

   $$z = uvwxy, \quad \text{where } |vx| \geq 1 \text{ and } |vwx| \leq n$$

   some of the strings uv^iwx^iy $(i = 0, 1, \ldots)$ are not in L_3.

Let us try with

$$z = a^n b^n a^n b^n \in L.$$

Consider an arbitrary splitting as above. If $|vx|$ is odd then uv^0wx^0y has an odd length and thus is not in L_3. So it remains to consider the case of $|vx|$ being even. Notice that $3n \leq |uv^0wx^0y| \leq 4n - 1$. We have three possibilities:

(a) vwx is contained in the first half of z (so $y = y'a^nb^n$, for some y'). Then the last symbol of the first half of uv^0wx^0y is a (we removed some symbols from the first half of z and "the middle moved to the right"). Thus uv^0wx^0y is not in L_3 (as the last symbol of its second half is b).

(b) vwx is contained in both halves of z. So it begins with a b and ends with an a. Thus v contains (at least one) b or x contains (at least one) a. Thus the first half of uv^0wx^0y has fewer b's than the second one[a] or the second half of uv^0wx^0y has fewer a's than the first one. Hence uv^0wx^0y is not in L_3.

(c) vwx is contained in the second half of z (so $u = a^nb^nu'$, for some u'). This case is symmetric to case 3a. The first symbol of the second half of uv^0wx^0y is b and $uv^0wx^0y \notin L_3$.

[a]As the halves of uv^0wx^0y are not shorter than $1.5n$, the first half begins with a^n and the second half ends with b^n.

2 LR-Parsing

Exercise 2. Consider the following CFG G (S is the start symbol):

$S \to A\$$
$A \to AB \mid B$
$B \to (A) \mid ()$

1. Construct an NFA which shows the valid LR(0) items for each viable prefix. (You may choose to skip this step and go directly to b).
2. Construct an equivalent DFA (exclude the error state and all transitions to it).
3. Is G an LR(0) grammar?

Exercise 3. Is the following CFG an LR(0) grammar? (S' is the start symbol.)

$S' \to S$
$S \to bA \mid aB$
$A \to a \mid aS \mid bAA$
$B \to b \mid bS \mid aBB$

Exercise 4. Show how the following strings are parsed by the LR(0) parser whose finite control is given by the DFA in Exercise 2(2). For each step show the stack, the remaining input and whether the operation is "shift" or "reduce". For reduce operations show which productions are involved.

1. $(()())\$$
2. $()((())())\$$

Exercise 5. Is the following CFG an LR(0) grammar? (S is the start symbol.)

$S \to A$
$A \to aAa \mid bAb \mid c$

Exercise 6. Consider the following CFG G (S is the start symbol):

$S \to A$
$A \to AB \mid \epsilon$
$B \to b \mid aB$

1. Construct an NFA showing the valid LR(1) items for each viable prefix. (You may choose to skip this step and construct a DFA instead).
2. Is G an LR(1) grammar?

Exercise 7. Consider the following CFG G (S is the start symbol):

$S \to E$
$E \to E + T \mid T$
$T \to a \mid (E)$

1. Construct a DFA showing the valid LR(1) items for a viable prefix.
2. Show how the string $a + (a + a)\$$ is parsed by the LR(1) parser corresponding to the DFA in a) (\$ is used to denote 'end-of-input').

   For each step show the stack, the remaining input, the kind of action ("shift" or "reduce") and the production used in "reduce".

Solutions

Solution to Exercise 2.

1. The NFA is shown in Figure 1.
2. The DFA is shown in Figure 2.
3. Yes, the grammar is LR(0).

Solution to Exercise 3. If we construct a DFA which determines the set of valid LR(0) items for each viable prefix (see Figure 3), then we will find that both $A \to a\bullet$ and $S \to \bullet bA$ are valid for the viable prefix ab, for instance. Since $A \to a\bullet$ is a complete item, and another item is valid for the same viable prefix, the grammar is *not* LR(0).

Solution to Exercise 4.

Stack	**Remaining input**	**Comment**
0	(()())\$	Start
0(5	()())\$	Shift
0(5(5	)())\$	Shift
0(5(5)7	())\$	Shift
$0(5B4$	())\$	Reduce by $B \to ()$
$0(5A6$	())\$	Reduce by $A \to B$
$0(5A6(5$	))\$	Shift
$0(5A6(5)7$	)\$	Shift
$0(5A6B3$	)\$	Reduce by $B \to ()$
$0(5A6$	)\$	Reduce by $A \to AB$
$0(5A6)8$	\$	Shift
$0B4$	\$	Reduce by $B \to (A)$
$0A1$	\$	Reduce by $A \to B$
$0A1\$2$	-	Shift
$0S$	-	Reduce by $S \to A\$$ and accept

Solution to Exercise 5. If a DFA is constructed as in Exercise 2 (see Figure 4), we will find that the grammar is LR(0).

Solution to Exercise 6.

1. See the NFA in Figure 5 (\$ denotes 'end-of-input').

2. If a DFA equivalent to the NFA in the first subtask is constructed (see Figure 6), we will find that the grammar is LR(1).

Solution to Exercise 7.

1. See Figure 7.

2.

Stack	**Remaining input**	**Comment**
A	$a + (a + a)\$$	Start
AaE	$+(a + a)\$$	Shift
ATF	$+(a + a)\$$	Reduce by $T \to a$
AEB	$+(a + a)\$$	Reduce by $E \to T$
AEB+C	$(a + a)\$$	Shift
AEB+C(G	$a + a)\$$	Shift
AEB+C(GaK	$+a)\$$	Shift
AEB+C(GTJ	$+a)\$$	Reduce by $T \to a$
AEB+C(GEH	$+a)\$$	Reduce by $E \to T$
AEB+C(GEH+M	$a)\$$	Shift
AEB+C(GEH+MaK	$)\$$	Shift
AEB+C(GEH+MTN	$)\$$	Reduce by $T \to a$
AEB+C(GEH	$)\$$	Reduce by $E \to E + T$
AEB+C(GEH)I	$\$$	Shift
AEB+CTD	$\$$	Reduce by $T \to (E)$
AEB	$\$$	Reduce by $E \to E + T$
AS	$\$$	Reduce by $S \to E$ and accept

An LR(1) parser can be described by a decision table. For an input symbol and a DFA state from the stack, the table gives the parser's action. The decision table corresponding to the DFA above is shown here.

Stack top	Next input symbol				
	a	$+$	(	)	\$
A	Shift		Shift		
B		Shift			Reduce by $S \to E$ and accept
C	Shift		Shift		
D		Reduce by $E \to E + T$			Reduce by $E \to E + T$
E		Reduce by $T \to a$			Reduce by $T \to a$
F		Reduce by $E \to T$			Reduce by $E \to T$
G	Shift		Shift		
H		Shift		Shift	
I		Reduce by $T \to (E)$			Reduce by $T \to (E)$
J		Reduce by $E \to T$		Reduce by $E \to T$	
K		Reduce by $T \to a$		Reduce by $T \to a$	
L	Shift		Shift		
M	Shift		Shift		
N		Reduce by $E \to E + T$		Reduce by $E \to E + T$	
O		Shift		Shift	
P		Reduce by $T \to (E)$		Reduce by $T \to (E)$	

Figures

This section contains the figures for the solutions to Exercise 2 – Exercise 7

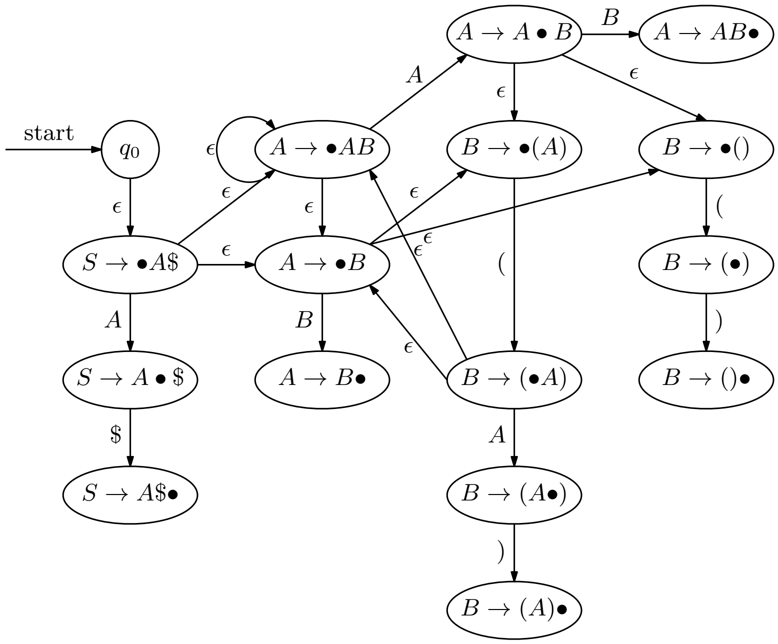

Figure 1: The NFA for Exercise 2.

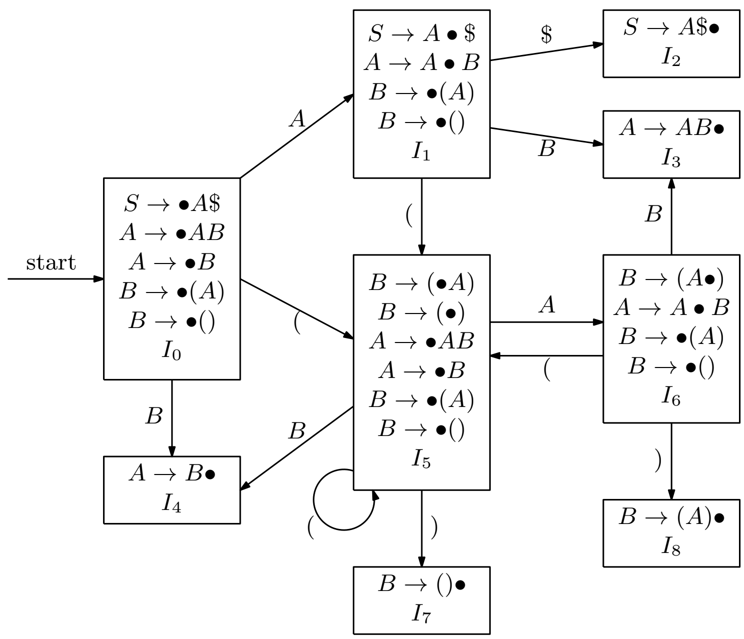

Figure 2: The DFA for Exercise 2.

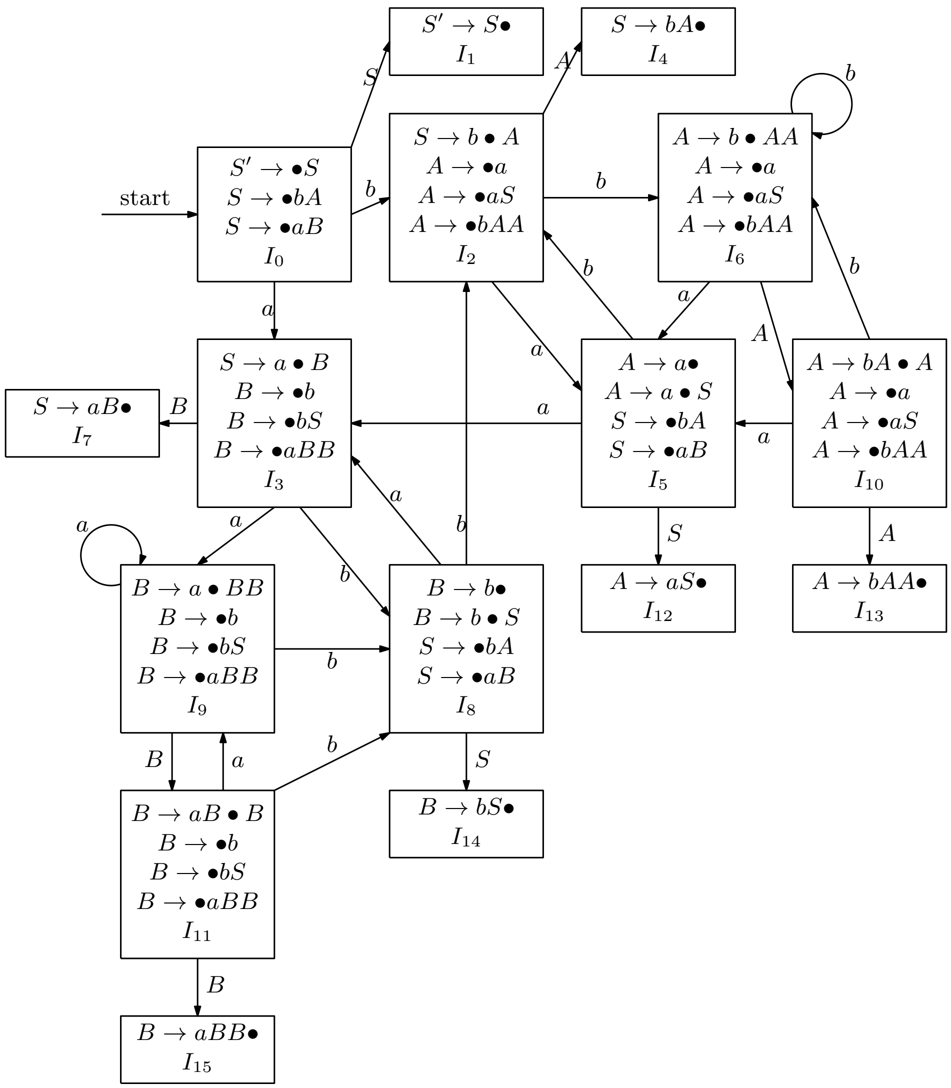

Figure 3: The DFA for Exercise 3.

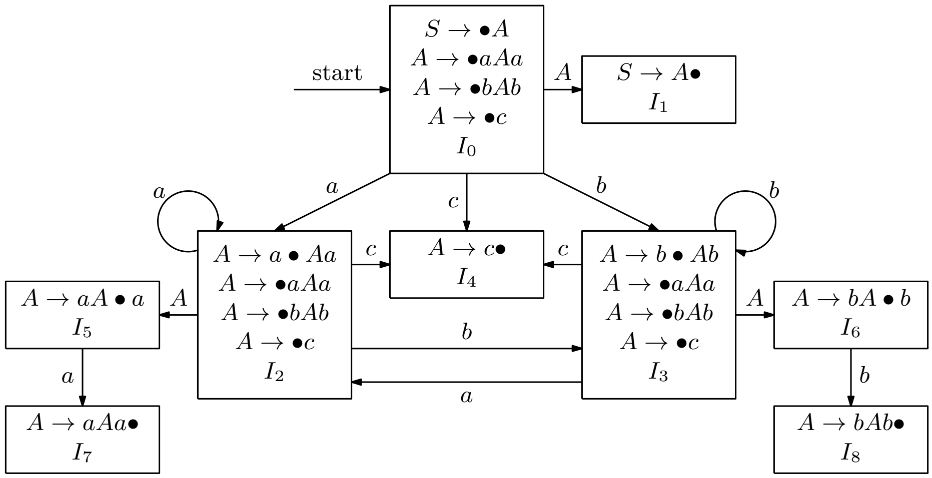

Figure 4: The DFA for Exercise 5.

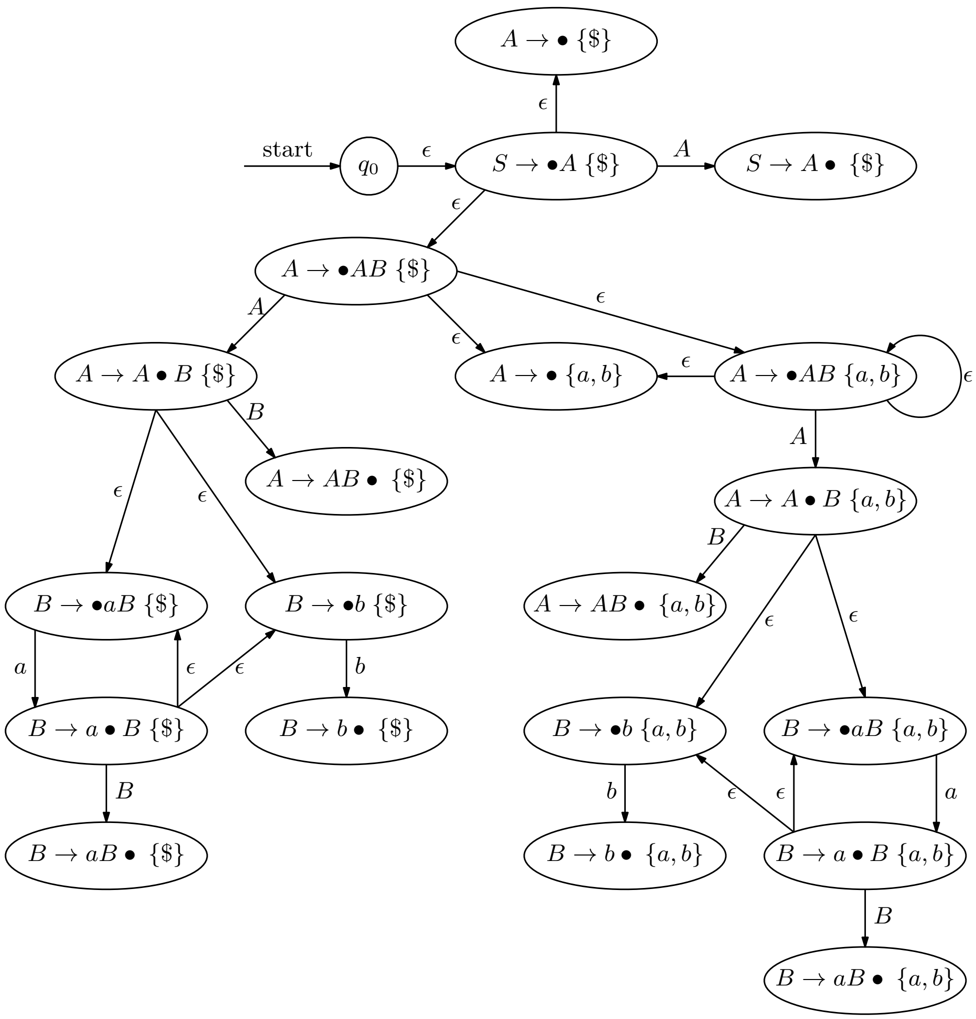

Figure 5: The NFA for Exercise 6.

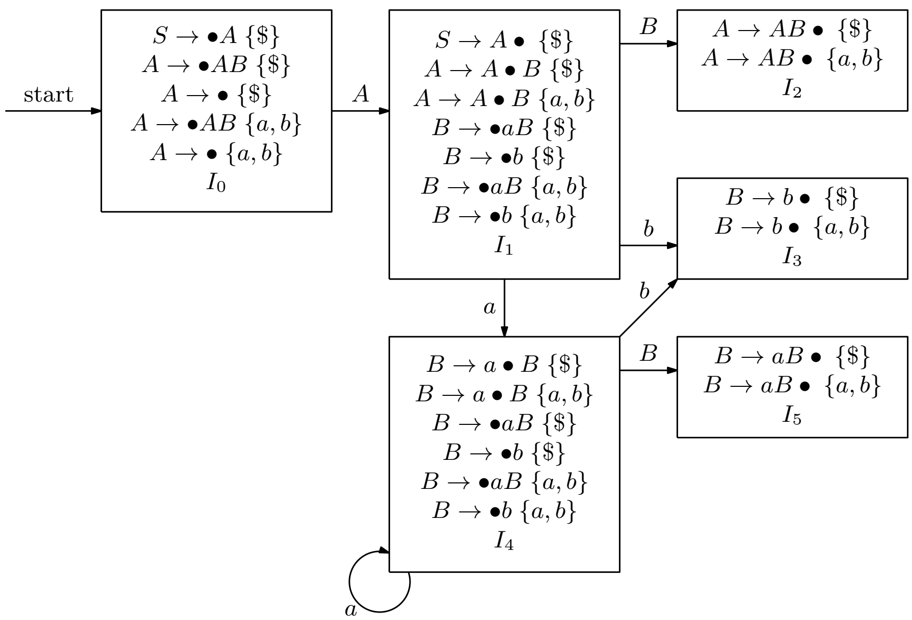

Figure 6: The DFA for Exercise 6.

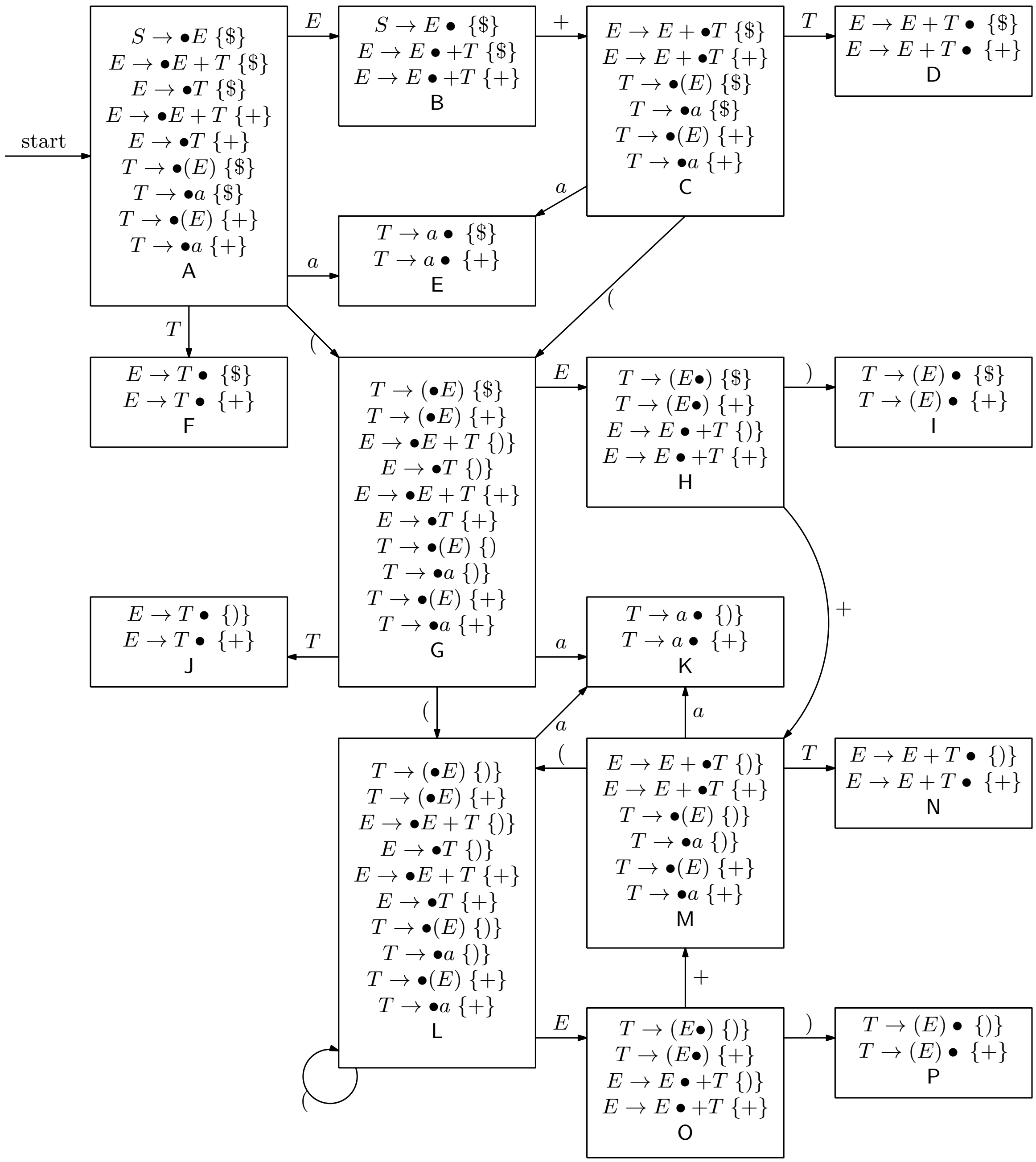

Figure 7: The DFA for Exercise 7.

3 Advanced and Exam Like Exercises

Exercise 8. Prove that the language $L = \{a^i b^j c^{ij} \mid 0 < i < j\}$ is not context-free by using the pumping lemma for context-free languages.

Exercise 9. Consider the set of O of strings over $\{0, 1, \ldots, 9\}$ containing an odd number of digits. For example, $5,000,083 \in O$.

Prove that O is context-free by constructing a context-free grammar for O. Your grammar should LR(0) or LR(1). To simplify the last task you may restrict yourself to a sufficiently rich subset of the grammar (e.g., instead of using 9 distinct digits, you could use a single symbol).

Solutions

Solution to Exercise 8. We provide a proof sketch which illustrates the most important case distinctions but does not go into every case in detail.

To prove that the language $L = \{a^i b^j c^{ij} \mid 0 < i < j\}$ is not context-free, we use the (inverted) pumping lemma for context-free languages. Let $p \geq 1$ be arbitrary and consider the string $s = a^p b^{p+1} c^{(p)(p+1)} \in L$. Let $s = uvwxy$ be a partitioning where $|vwx| \leq p$ and $|vx| \geq 1$. We observe that since $|vwx| \leq p$, vwx can contain at most two different types of symbols (either as and bs, or bs and cs).

We then consider three cases based on vwx:

1. vwx consists of a single symbol): in this case uv^2wx^2y is not included in the language.

2. vwx consists of only a's and b's: then pumping v and x changes the number of a's or b's, but does not change the number of c's accordingly. Thus, uv^2wx^2y would have either more a's or b's without the corresponding number of c's to match the condition c^{ij}, where i and j are the counts of a and b respectively.

3. vwx consists of only b's and c's: then pumping v and x changes the number of b's or c's without the corresponding adjustment to a's. If $p = 1$ then we choose $i = 0$ since then uv^0wx^0y is not included in the language. For $p > 1$ one can prove that we can always find $i \geq 2$ such that uv^iwx^iy is not in the language.

Solution to Exercise 9. A context-free grammar for the language can be given by:

$$D \to 0 \mid 1 \mid 2 \mid 3 \mid 4 \mid 5 \mid 6 \mid 7 \mid 8 \mid 9$$
$$O \to D$$
$$O \to DDO$$

This grammar can be proven to not be LR(0) by constructing the DFA with LR(0) items. However, it can be proven to be LR(1).