

CS150 Homework 4 Solution

Please read the text book [1] and the references [2].

Q1

(a)

S $\rightarrow$ AB
A $\rightarrow$ 0A1 | ϵ
B $\rightarrow$ 1B0 | ϵ

(b)

S $\rightarrow$ AB | CD
A $\rightarrow$ 00A1 | ϵ
B $\rightarrow$ 1B00 | ϵ
C $\rightarrow$ 00C1 | 01
D $\rightarrow$ 1D00 | 0

Q2

S $\rightarrow$ A1B $\rightarrow$ 0A1B $\rightarrow$ 01B $\rightarrow$ 010B $\rightarrow$ 0101B $\rightarrow$ 01010B
 $\rightarrow$ 010101B $\rightarrow$ 010101

Please see Figure 1. It is the parse tree.

Q3 (Ex. 5.4.5)

(a)

If there is only one leftmost derivation for any given string, then the grammar is unambiguous. So we only need to show that there is only one leftmost derivation for a string in the language.

When we start with leftmost derivations, we can only start with the first rule, because that is the only one we get. When we encounter symbol A, we can look ahead at the current symbol in the input string (we scan the input string from left to right as in the example of the lecture notes). If the current symbol is 0, then we can only apply the rule A $\rightarrow$ 0A. If the symbol is 1, then we have to apply the rule A $\rightarrow$ ϵ . So, there is no ambiguity here. When we encounter symbol B, similar reasoning can again be applied, but we observe the rule B $\rightarrow$ ϵ can only be applied at the end of the input. So there is only one leftmost derivation for any given input and the language is unambiguous.

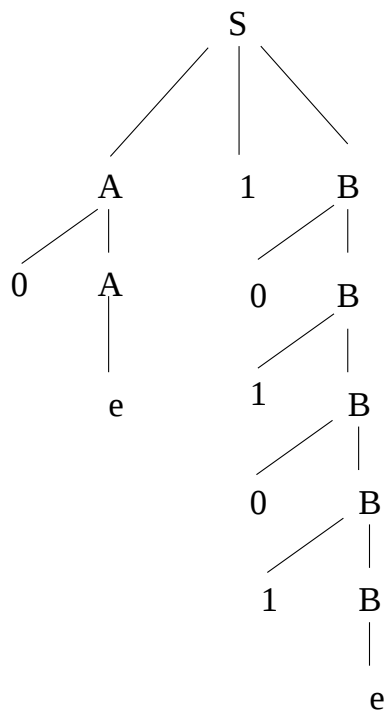


Figure 1: Parse tree for 010101

(b)

We can add some (redundant) B rules to make the grammar ambiguous: $B \rightarrow BB \mid 0 \mid 1 \mid \text{epsilon}$.

To show the new grammar is ambiguous, just give two parse trees for string 010.

Q4 (Ex. 6.1.1)

(b)

For string 0011:

ID 1: $(q, 0011, Z_0) \vdash (q, 011, XZ_0) \vdash (q, 11, XXZ_0) \vdash (q, 1, XXZ_0) \vdash (q, \text{epsilon}, XXZ_0) \vdash (p, \text{epsilon}, XZ_0) \vdash (p, \text{epsilon}, Z_0)$

ID 2: $(q, 0011, Z_0) \vdash (q, 011, XZ_0) \vdash (q, 11, XXZ_0) \vdash (q, 1, XXZ_0) \vdash (p, 1, XZ_0) \vdash (p, 1, Z_0) \vdash (p, \text{epsilon}, \text{epsilon})$

ID 3: $(q, 0011, Z_0) \vdash (q, 011, XZ_0) \vdash (q, 11, XXZ_0) \vdash (q, 1, XXZ_0) \vdash (p, 1, XZ_0) \vdash (p, \text{epsilon}, XXZ_0) \vdash (p, \text{epsilon}, XZ_0) \vdash (p, \text{epsilon}, Z_0)$

ID 4: $(q, 0011, Z_0) \vdash (q, 011, XZ_0) \vdash (p, 011, Z_0)$

ID 5: $(q, 0011, Z_0) \vdash (q, 011, XZ_0) \vdash (q, 11, XXZ_0) \vdash (p, 11, XZ_0) \vdash (p, 1, XXZ_0) \vdash (p, \text{epsilon}, XXXZ_0) \vdash (p, \text{epsilon}, XXZ_0) \vdash (p, \text{epsilon}, XZ_0) \vdash (p, \text{epsilon}, Z_0)$

ID 6: (q, 0011, Z₀) |- (q, 011, XZ₀) |- (q, 11, XXZ₀) |- (p, 11, XZ₀)
 |- (p, 1, XXZ₀) |- (p, 1, XZ₀) |- (p, 1, Z₀) |- (p, epsilon, epsilon)

ID 7: (q, 0011, Z₀) |- (q, 011, XZ₀) |- (q, 11, XXZ₀) |- (p, 11, XZ₀)
 |- (p, 1, XXZ₀) |- (p, 1, XZ₀) |- (p, epsilon, XXZ₀) |- (p, epsilon, XZ₀)
 |- (p, epsilon, Z₀)

ID 8: (q, 0011, Z₀) |- (q, 011, XZ₀) |- (q, 11, XXZ₀) |- (p, 11, XZ₀)
 |- (p, 11, Z₀) |- (p, 1, epsilon)

(c)

For string 010:

ID 1: (q, 010, Z₀) |- (q, 10, XZ₀) |- (q, 0, XZ₀) |- (q, epsilon, XXZ₀)
 |- (p, epsilon, XZ₀) |- (p, epsilon, Z₀)

ID 2: (q, 010, Z₀) |- (q, 10, XZ₀) |- (q, 0, XZ₀) |- (p, 0, Z₀)

ID 3: (q, 010, Z₀) |- (q, 10, XZ₀) |- (p, 10, Z₀) |- (p, 0, epsilon)

Q5

(a)

We use accept by final state: X is used to track the number of 1s, and Y is used to track the number of 0s, q is the start state and p is the accept state.

delta(q, 1, Z₀) = { (q, XZ₀) }
 delta(q, 0, Z₀) = { (q, YZ₀) }
 delta(q, 1, X) = { (q, XX) }
 delta(q, 1, Y) = { (q, epsilon) }
 delta(q, 0, X) = { (q, epsilon) }
 delta(q, 0, Y) = { (q, YY) }
 delta(q, epsilon, X) = { (p, X) }

Note that although our solution keys list a sequence of transition for each PDA, you are supposed to show a transition diagram as Figure 6.2 of the textbook.

(b)

We use accept by empty stack.

delta(q, 0, Z₀) = { (q, XZ₀) }
 delta(q, 0, X) = { (q, XX) }
 delta(q, 1, Z₀) = { (p, Y) }
 delta(q, 1, X) = { (p, e) }
 delta(p, 0, Y) = { (r, e) }
 delta(r, e, Z₀) = { (r, e) }

delta(q, e, Z₀) = { (q, e) }
 delta(q, 1, Y) = { (q, YY) }
 delta(p, 1, X) = { (p, e) }
 delta(r, 0, Y) = { (r, e) }

Here, e means epsilon

References

- [1] J. Hopcroft, R. Motwani, and J. Ullman, *Introduction to Automata Theory, Languages, and Computation*. Boston: Addison-Wesley, 2006.
- [2] M. Sipser, *Introduction to the Theory of Computation*. Boston: PWS Pub., 1997.

