Theoretical Computer Science (M21276)

Part A/8: Application of context-free languages

(Oct 16-20, 2023)

Question 1. Consider the following grammar G with non-terminal start symbol S and terminal symbols 0, 1:

$$S \to 0S1 \mid SS \mid 10$$

Show a parse tree produced by G for each of the following strings:

(a) 010110

(b) 00101101

Question 2. Consider the fragment of English grammar given in the lecture. Use it to construct a parse tree (using top down parsing) for the following sentences:
(a) "The boy sees a flower."
(b) "A girl likes the boy with the flower."

Answer: (a) <Sentence> $\Longrightarrow$ <Noun Phrase> <Verb Phrase>
$\Longrightarrow$ <Cmplx-Noun> <Verb Phrase>
$\Longrightarrow$ <Article> <Noun> <Verb Phrase>
$\Longrightarrow$ The <Noun> <Verb Phrase>
$\Longrightarrow$ The boy <Cmplx-verb>
$\Longrightarrow$ The boy <Verb> <Noun Phrase>
$\Longrightarrow$ The boy sees <Cmplx-Noun>
$\Longrightarrow$ The boy sees <Article> <Noun>
$\Longrightarrow$ The boy sees a <Noun>
$\Longrightarrow$ The boy sees a flower.

Question 3. Show that the following grammar is ambiguous: S the start non-terminal, A, B two non-terminals and a, b terminals

$$S \to AB \mid aaB$$
$$A \to a \mid Aa$$
$$B \to b$$

Answer: Consider for example the string aab.

Question 4. Show that the following grammar is ambiguous: S the start non-terminal, a, b terminals

$$S \to aSbS \mid bSaS \mid \Lambda$$

Answer: Consider for example the string $abab$.

Question 5. Consider the following grammar G with the non-terminal start symbol S, two non-terminals B, C and terminal symbols a, c, d, e, f, g, x, y, z:

$$S \to xyz \mid aBC$$
$$B \to c \mid cd$$
$$C \to eg \mid df$$

Use two different methods of parsing (top-down and bottom-up) to derive the strings
(a) $acddf$,
(b) $acdg$.

Answer: Bottom-up parsing:
(a) Steps

- Λ can't be reduced
- a can't be reduced
- ac can be reduced, as follows:
- reduce ac to aB
  - aB can't be reduced
  - aBd can't be reduced
  - $aBdd$ can't be reduced
  - $aBddf$ can't be reduced
  - End of string. Stack is $aBddf$, not S. Failure! Must backtrack.
- ac can't be reduced
- acd can be reduced, as follows:
- reduce acd to aB
  - aB can't be reduced
  - aBd can't be reduced
  - $aBdf$ can be reduced, as follows:
  - reduce $aBdf$ to aBC
    - aBC can be reduced, as follows:
    - reduce aBC to S

End of string. Stack is S. Success!

(b) If all combinations fail, then the string cannot be parsed, string is $acdg$.
Steps:

- Λ can't be reduced
- a can't be reduced

- ac can be reduced, as follows:
- reduce ac to aB
  - aB can't be reduced
  - aBd can't be reduced
  - $aBdg$ can't be reduced
  - End of string. Stack is $aBdg$, not S. Failure! Must backtrack.
- ac can't be reduced
- acd can be reduced, as follows:
- reduce acd to aB
  - aB can't be reduced
  - aBg can't be reduced
  - End of string. stack is aBg, not S. Failure! Must backtrack.
- acd can't be reduced
- $acdg$ can't be reduced
- End of string. Stack is is $acdg$, not S. No backtracking is possible. Failure!

Top-down parsing.
(a) String is $acddf$.
Steps

- Assertion 1: $acddf$ matches S
  - Assertion 2: $acddf$ matches xyz:
  - Assertion is false. Try another.
  - Assertion 2: $acddf$ matches aBC i.e. $cddf$ matches BC:
    - Assertion 3: $cddf$ matches cC i.e. ddf matches C:
      - Assertion 4: ddf matches eg:
      - False.
      - Assertion 4: ddf matches df:
      - False.
    - Assertion 3 is false. Try another.
    - Assertion 3: $cddf$ matches cdC i.e. df matches C:
      - Assertion 4: df matches eg:
      - False.
      - Assertion 4: df matches df:
      - Assertion 4 is true.

    * Assertion 3 is true.

  - Assertion 2 is true.

- Assertion 1 is true. Success!

Question 6. Give an example of a string (of length at least 5) from the language described by the grammar $S \to aSc \mid b$ with the initial non-terminal S. Show that you can find unique derivations generating the string from left looking only at the current symbol. ($LL(1)$ grammar).

Question 7. Give an example of a string (of length at least 5) from the language described by the grammar $S \to AB$, $A \to aA \mid a$, $B \to bB \mid c$. Show that you can always find derivations used for generation of your string (from left) looking only at most two symbols ahead. ($LL(2)$ grammar).
Can you rewrite this grammar as an $LL(1)$ grammar?

Answer: $S \to aAB$, $A \to aA \mid \Lambda$, $B \to bB \mid c$

Question 8. Explain why the following grammar is $LR(1)$ and not $LL(1)$: $S \to a \mid ab$

Question 9. Explain why the following grammar is unambiguous and not $LR(1)$: $S \to Uab \mid Vac$, $U \to d$, $V \to d$. *Answer:* The grammar is unambiguous because it contains exactly two strings and each string has a unique derivation. $S \implies Uab \implies dab$, $S \implies Vac \implies dac$.
The grammar is not $LR(1)$ because of the following reason. Consider an input that begins with da. After seeing d, the parser must decide and use either U or V. However, based solely on the lookahead character of a, we cannot decide which of U or V is correct. Note that the grammar is $LR(2)$.

Question 10. Find a language which is described by the grammar: $S \to Sa \mid b$. Show that the grammar is not $LL(1)$.
Can you find the $LL(1)$ grammar for the same language?

Answer: $S \to bB$, $B \to aB \mid \Lambda$

Question 11. Show that the following grammar is $LR(1)$, but not an $LR(0)$ grammar. $S \to AB$, $A \to aAb$, $A \to \Lambda$, $B \to Bb$, $B \to b$. Describe the language which is generated by this grammar. Also, find the derivation tree for a^2b^4.

Answer: $L = \{a^m b^n \mid n > m \geq 0\}$.
To get the derivation tree for a^2b^4, we scan a^2b^4 from left to right. After scanning a, we look ahead. If the next symbol is a, we continue to scan. If the next symbol is b, we decide that $A \to \Lambda$ is the required handle production. Thus the last step of the right-most derivation of a^2b^4 is

$$a^2Ab^4 \underset{R}{\implies} a^2\Lambda b^4.$$

To get the last step of a^2Ab^4, we scan a^2Ab^4 from L to R. aAb is a possible handle. We are able to decide that this is the right handle without looking ahead and so we get

$$aAbb^2 \underset{R}{\Longrightarrow} a^2Ab^4$$

Once again using the handle aAb, we obtain

$$Ab^2 \underset{R}{\Longrightarrow} aAbb^2.$$

To get the last step of the rightmost derivation of Ab^2, we scan Ab^2. A possible handle production is $B \to b$. We also note that this handle production can be applied to the first b we encounter, but not to the last b. So, we get

$$ABb \underset{R}{\Longrightarrow} Ab^2.$$

For ABb, a possible a-handle is Bb. Hence, we get $AB \underset{R}{\Longrightarrow} ABb$. Finally we obtain

$$S \underset{R}{\Longrightarrow} AB.$$

Question 12. Find an $LL(k)$ grammar for the language $\{aa^n \mid n \in \mathbb{N}\} \cup \{aab^n \mid n \in \mathbb{N}\}$. What is k for your grammar?

Answer: An $LL(2)$ grammar: $S \to aC$, $C \to A \mid aB$, $A \to aA \mid \Lambda$, $B \to bB \mid \Lambda$.

Question 13. Find the minimum k such that the following grammar is $LL(k)$ grammar: $S \to SS \mid aSb \mid ab$.

Answer: The grammar is not an $LL(k)$ grammar for any k.

Question 14. Find the minimum k such that the following grammar is $LR(k)$ grammar: $S \to ADC \mid aaaddd$, $A \to aaa$, $D \to ddd$, $C \to Cc \mid c$.

Answer: $k = 4$

Question 15. Find the minimum values k_1, k_2 such that the following grammar is $LL(k_1)$, $LR(k_2)$ grammar: $S \to A \mid B$, $A \to aAb \mid 0$, $B \to aBbb \mid 1$.

Answer: No k_1 exists, $k_2 = 0$.

Question 16. Is it possible for a regular grammar to be ambiguous?

Answer: Yes, e.g. $S \to aS \mid aA \mid aB$, $A \to bA \mid b$, $B \to bB \mid b$, e.g. for ab