Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Networked Systems Group (NSG)

HS 2023

Prof. L. Vanbever / R. Schmid
based on Prof. R. Wattenhofer's material

Discrete Event Systems

Solution to Exercise Sheet 5

1 Revisiting Context-Free Grammars

a) Recall the solution from last week with start symbol X:

$$X \to XAX \mid A,$$
$$A \to 0 \mid 1$$

We begin by transforming the grammer's productions to CNF:

(i) Ensure that start is not on the right by introducing a new start symbol S:

$$S \to X,$$
$$X \to XAX \mid A,$$
$$A \to 0 \mid 1$$

(ii) Remove ε productions (except from the start symbol): already none ✓

(iii) Remove ($A \to B$)-type productions:

$$S \to XAX \mid 0 \mid 1,$$
$$X \to XAX \mid 0 \mid 1,$$
$$A \to 0 \mid 1$$

(iv) Replace longer variable productions by *dyadic*, i.e., ($A \to BC$)-type productions by introducing additional symbols:

$$S \to XY \mid 0 \mid 1,$$
$$X \to XY \mid 0 \mid 1,$$
$$Y \to AX,$$
$$A \to 0 \mid 1$$

This grammar is in CNF. Note that the language is regular. Hence, there exist both a right-linear and a left-linear grammar for it.

Right-linear:

$$X \to 0Y \mid 1Y \mid 0 \mid 1,$$
$$Y \to 0X \mid 1X$$

Left-linear:

$$X \to Y0 \mid Y1 \mid 0 \mid 1,$$
$$Y \to X0 \mid X1$$

L_1 can also be described using a single non-terminal symbol:

$$S \to 0 \mid 1 \mid 0SS \mid 1SS$$

b) Recall the solution from last week with start symbol X:

$$\begin{aligned} S &\rightarrow\ A1A, \\ A &\rightarrow\ A1 \mid 1A \mid A01 \mid 0A1 \mid 01A \mid A10 \mid 1A0 \mid 10A \mid \varepsilon \end{aligned}$$

We begin by transforming the grammer's productions to CNF:

(i) Ensure that start is not on the right: already ok ✓

(ii) Remove ε productions (except from the start symbol):

$$\begin{aligned} S &\rightarrow\ A1A \mid A1 \mid 1A \mid 1, \\ A &\rightarrow\ A1 \mid 1A \mid A01 \mid 0A1 \mid 01A \mid A10 \mid 1A0 \mid 10A \mid 1 \mid 01 \mid 10 \end{aligned}$$

(iii) Remove $(A \rightarrow B)$-type productions: already none ✓

(iv) Replace longer productions by *dyadic* variable productions, i.e., $(A \rightarrow BC)$-type productions by introducing additional symbols:

$$\begin{aligned} S &\rightarrow\ BA \mid AX \mid XA \mid 1, \\ A &\rightarrow\ AX \mid XA \mid CX \mid ZB \mid ZD \mid BZ \mid XC \mid XE \mid 1 \mid ZX \mid XZ, \\ B &\rightarrow\ AX, \\ C &\rightarrow\ AZ, \\ D &\rightarrow\ XA, \\ E &\rightarrow\ ZA, \\ X &\rightarrow\ 1, \\ Z &\rightarrow\ 0 \end{aligned}$$

This grammar is in CNF. Note that the language is **not** regular. Hence, there is no right-/left-linear grammar for it. We've seen a grammar using the minimum number of non-terminal symbols generating it last week (or above).

Finally, we consider last week's alternative (also minimal) solution:

$$\begin{aligned} S &\rightarrow\ A1A, \\ A &\rightarrow\ AA \mid 1A0 \mid 0A1 \mid 1 \mid \varepsilon \end{aligned}$$

It can also be transformed to CNF:

(i) Ensure that start is not on the right: already ok ✓

(ii) Remove ε productions (except from the start symbol):

$$\begin{aligned} S &\rightarrow\ A1A \mid A1 \mid 1A \mid 1, \\ A &\rightarrow\ AA \mid \underbrace{A}_{\text{not producing anything}} \mid 1A0 \mid 10 \mid 0A1 \mid 01 \mid 1 \end{aligned}$$

(iii) Remove $(A \rightarrow B)$-type productions: only remove $A \rightarrow A$ ✓

(iv) Replace longer variable productions by *dyadic*, i.e., $(A \rightarrow BC)$-type productions by introducing additional symbols:

$$\begin{aligned} S &\rightarrow\ A1A \mid A1 \mid 1A \mid 1, \\ A &\rightarrow\ AA \mid XC \mid XZ \mid ZB \mid ZX \mid 1, \\ B &\rightarrow\ AX, \\ C &\rightarrow\ AZ, \\ X &\rightarrow\ 1, \\ Z &\rightarrow\ 0 \end{aligned}$$

2 Regular, Context-Free or Not?

a) We begin by proving that L is not regular using the pumping lemma recipe:

1. Assume for contradiction that L was regular.
2. There must exist some p, s.t. any word $w \in L$ with $|w| \geq p$ is pumpable.
3. Choose the string $w = 1^k$ for some $\underbrace{k \gg p}_{k \text{ much greater than } p}$ and p prime, $w \in L$ with length $|w| > p$.
4. Consider all ways to split $w = xyz$ s.t. $|xy| \leq p$ and $|y| \geq 1$.
   $\rightarrow$ Hence, $y \in 1^+$ and $|z| > 1$ (since $k \gg p$).
5. Observe that $xy^iz \notin L$ for $i = |xz|$, since
   $$|w| = |xy^iz| = |xz| + i \cdot |y| = \underbrace{|xz|}_{>1} \cdot \underbrace{(1 + |y|)}_{>1}$$
   is not prime.
6. This constitutes a contradiction to p being a valid pumping length.
7. Consequently, L cannot be regular.

Similarly, one can proof that L is not context-free using the tandem-pumping lemma:

1. Assume for contradiction that L was **context-free**.
2. There must exist some p, s.t. any word $w \in L$ with $|w| \geq p$ is **tandem-**pumpable.
3. Choose the string $w = 1^k$ for some $\underbrace{k \gg p}_{k \text{ much greater than } p}$ and p prime, $w \in L$ with length $|w| > p$.
4. Consider all ways to split $w = \mathbf{uv}xyz$ s.t. $|\mathbf{v}xy| \leq p$ and $|\mathbf{v}y| \geq 1$.
   $\rightarrow$ Hence, $\mathbf{v}y \in 1^+$ and $|\mathbf{u}z| > 1$ (since $k \gg p$).
5. Observe that $\mathbf{uv^i}xy^iz \notin L$ for $i = |\mathbf{u}xz|$, since
   $$|w| = |\mathbf{uv^i}xy^iz| = |\mathbf{u}xz| + i \cdot |\mathbf{v}y| = \underbrace{|\mathbf{u}xz|}_{>1} \cdot \underbrace{(1 + |\mathbf{v}y|)}_{>1}$$
   is not prime.
6. This constitutes a contradiction to p being a valid **tandem-**pumping length.
7. Consequently, L cannot be **context-free**.

It would have been enough to show that L is not context-free to prove that L is not regular.

b) First, it can be shown using the pumping lemma that L is not regular:

1. Assume for contradiction that L was regular.
2. There must exist some p, s.t. any word $w \in L$ with $|w| \geq p$ is pumpable.
3. Choose the string $w = a^p\#a^p\#a^p\#a^p$; hence, $w \in L$ with length $|w| > p$.
4. Consider all ways to split $w = xyz$ s.t. $|xy| \leq p$ and $|y| \geq 1$.
   $\rightarrow$ Hence, $y \in a^+$.
5. Observe that $xy^0z \notin L$ – a contradiction to p being a valid pumping length.
6. Consequently, L cannot be regular.

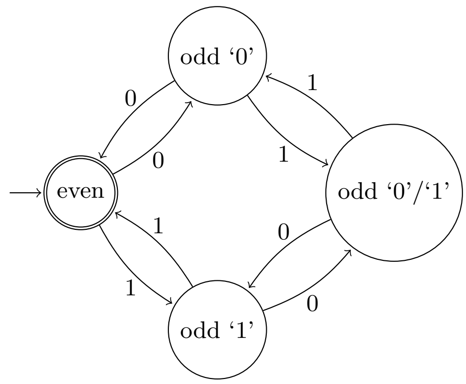
Next, we show that L is context-free by providing a CFG that produces the language L. First, we create an equal number of symbols for w and z using rule (2), and then an equal number of symbols for x and y using rule (3).

$$S \rightarrow A \quad (1)$$
$$A \rightarrow YAY \mid \#B\# \quad (2)$$
$$B \rightarrow YBY \mid \# \quad (3)$$
$$Y \rightarrow a \mid b \quad (4)$$

c) If $|w| = |y|$ and $|x| = |z|$, the resulting language is not context free, thus a CFG does not exist. This can be seen using the tandem pumping lemma as follows.

Let the word considered be $s = a^p\#a^p\#a^p\#a^p \in L$ with $|s| = 4p+3 \geq p$. For any division $s = defgh$ with $|eg| \geq 1$ and $|efg| \leq p$, the pumpable regions e and g can never consist of boths as from w and y or both x and z because of the condition $|efg| \leq p$. Hence, any pumping would inevitably only modify the number of as in one part thereby creating a word $s' \notin L$. Therefore, L cannot be context free.

d) L is regular. Consider the following DFA on the alphabet $\Sigma = \{0, 1\}$ recognizing it:



3 Tandem-Pumping Lemma [Exam HS21]

a) $w = 1^p\#0\#1^p0$ is tandem-pumpable for the split $w = uvxyz$ where $u = 1^{p-1}$, $v = 1$, $x = \#0\#$, $y = 1$, and $z = 1^{p-1}0$:

- $w \in L$, because "1^p0" $= 2 \cdot$ "1^p" and $\#_0(b) = 1 = \#_0(c)$.
- $uv^0xy^0z = 1^{p-1}\#0\#1^{p-1}0$, which is in L.
  (i.e. removing v and y from w does not break any of the language's rules)
- v and y are part of a's and c's leading 1s, respectively. As $v = y = 1$, both numbers are modified identically, while c's trailing 0 ensures that $c = 2a$ remains true.
- v and y do not contain any 0s, so $\#_0(b) = \#_0(c)$ is preserved.

b) We prove that L is not context-free using the tandem-pumping lemma.

1. Assume for contradiction that L was context-free.
2. There must exist some p, s.t. any word $w \in L$ with $|w| \geq p$ is tandem-pumpable.
3. Choose the string $w = 10^{p-1}\#0^p\#10^p \in L$ with length $|w| > p$.
4. Consider all ways to split $w = uvxyz$ s.t. $|vxy| \leq p$ and $|vy| \geq 1$.
   - First, we observe that if the vxy part was completely part of a, b, or c (for $w = a\#b\#c$), then $uv^0xy^0z \notin L$.
   - Next, as $|\#b\#| > p$, the vxy part cannot span parts from both a and c.
   - Hence, while pumping w, we cannot change the (arithmetic) value of a or c as we could only change one of these values.
   - As both a and c do not contain leading 0s, we cannot change either of them.
   - Moreover, note that we can neither add nor remove a 0 to/from b as c is fixed.
   - Finally, observe that the number of # signs in w is fixed.
5. In conclusion, there is no split $w = uvxyz$ that satisfies all criteria of the tandem-pumping lemma – a contradiction to p being a valid tandem-pumping length.
6. Consequently, L cannot be context-free. □

c) If we chose any string $w = a\#b\#c$ with $1 \in b$, i.e. $b = b_1 1 b_2$, it would be tandem-pumpable. To see this, let $b = b_1 1 b_2$. Then, observe that $w = a\#b\#c$ is tandem-pumpable for the split $w = uvxyz$ where $u = a\#b_1$, $v = 1$, $x = \varepsilon$, $y = \varepsilon$, and $z = b_2\#c$.

4 Java is not regular! [Bonus question]

This question is just for fun. Please excuse if part of this sample solution is not fully formal.

Note that if `java` is regular, then

$$L = \texttt{java} \cap L\Big(\underbrace{(\{ \cup \})^*}_{\textit{REX for arbitrary curly-brace expressions}}\Big)$$

would have to be regular as well. However, L can also be written as:

$$L = \Big\{ w \mid w \in \{\{,\}\}^*, \#_\{(w) = \#_\}(w) \Big\},$$

which can be shown to be irregular using the pumping lemma.