Formal Reasoning 2025
Solutions Test Block 3: Languages
(24/11/25)

Languages

1. Which of the following four languages is different from the other three?

   (a) $(\{a, b\})^*$

   (b) $(\{a, b, a\})^*$

   (c) is correct (c) $(\{a\}^* \cup \{b\}^*)$

   (d) $(\{a\} \cup \{b\})^*$

   Answer (c) is correct.

   The languages $(\{a, b\})^*$, $(\{a, b, a\})^*$, and $(\{a\} \cup \{b\})^*$ basically all state: choose either a or b and do that as often as you want. This gives all words over the alphabet $\{a, b\}$. The language $(\{a\}^* \cup \{b\}^*)$, however, states: take any number of a's or any number of b's. In particular, this last language doesn't contain words like ab that contain a's and b's.

2. Let

$$\begin{array}{lcl} \Sigma & := & \{a, b\}, \\ L_2 & := & \{w \in \Sigma^* \mid w \text{ starts with the symbol } a\}, \text{ and} \\ L_2' & := & \{w \in \Sigma^* \mid w \text{ starts with the symbol } b\}. \end{array}$$

   In which of the alternatives are all statements correct?

   (a) $L_2 \cap L_2' \neq \varnothing$, $L_2^* = \Sigma^*$, $L_2 \cup L_2' \neq \Sigma^*$, and $L_2^* \cup L_2'^* = \Sigma^*$.

   (b) $L_2 \cap L_2' = \varnothing$, $L_2^* \neq \Sigma^*$, $L_2 \cup L_2' = \Sigma^*$, and $L_2^* \cup L_2'^* \neq \Sigma^*$.

   (c) is correct (c) $L_2 \cap L_2' = \varnothing$, $L_2^* \neq \Sigma^*$, $L_2 \cup L_2' \neq \Sigma^*$, and $L_2^* \cup L_2'^* = \Sigma^*$.

   (d) $L_2 \cap L_2' \neq \varnothing$, $L_2^* = \Sigma^*$, $L_2 \cup L_2' = \Sigma^*$, and $L_2^* \cup L_2'^* \neq \Sigma^*$.

   Answer (c) is correct.

   - $L_2 \cap L_2' = \emptyset$. Because words in L_2 always start with an a and words in L_2' always start with a b, the intersection $L_2 \cap L_2'$ must be empty, because there are no words that start with an a and start with a b at the same time.
   - $L_2^* \neq \Sigma^*$. Note that $L_2^* = L_2 \cup \{\lambda\}$. So if $w \in L_2^*$ then it holds that $w = \lambda$ or w starts with an a. But Σ^* also contains words that start with a b and therefore Σ is strictly larger than L_2^*.
   - $L_2 \cup L_2' \neq \Sigma^*$. We know that $\lambda \in \Sigma^*$. However, λ does not start with an a and it does not start with a b. Therefore λ is not in L_2 and also not in L_2'. Hence also not in $L_2 \cup L_2'$.
   - $L_2^* \cup L_2'^* = \Sigma^*$. Because Σ^* contains all words over the alphabet $\{a, b\}$, it is clear that $L_2^* \cup L_2'^* \subseteq \Sigma^*$. In addition, we have already seen that $L_2^* = L_2 \cup \{\lambda\}$ and $L_2'^* = L_2' \cup \{\lambda\}$. So $L_2^* \cup L_2'^* = L_2 \cup L_2' \cup \{\lambda\}$. But if $w \in \Sigma^*$ then it holds that $w = \lambda$, w starts with an a, or w starts with a b. And hence it follows that $w \in L_2 \cup L_2' \cup \{\lambda\} = L_2^* \cup L_2'^*$.

3. Consider the language

$$L_3 := \{w \in \{a,b,c\}^* \mid w \text{ does not contain } ab\}$$

So *aacbbacbca* and *bbccaaccbb* are elements of L_3, but *bbcaabcacb* is not. Here are two claims about this language:

1. $L_3 = \mathcal{L}((aa^*c \cup b \cup c)^*a^*)$
2. $L_3 = \mathcal{L}((b^*a^*c)^*b^*a^*)$

Now what is the case?

(a) Both claims are incorrect.

(b) Claim 1 is correct, but claim 2 is incorrect.

(c) Claim 1 is incorrect, but claim 2 is correct.

(d) is correct — (d) Both claims are correct.

Hint: Try to recognize the structure of the expression(s) in the examples.

Answer (d) is correct.

Both claims are indeed correct. This is how we can explain that.

1. If we construct a word that doesn't contain ab from left to right, then if we write an a, it will be followed by a possibly empty series of a's, followed by a c, unless we are at the end of the word (which is the aa^*c part). And if we write a b, there are no restrictions for the next symbol (which is the b part). And if we write a c, there are no restrictions for the next symbol (which is the c part). These three parts can be repeated zero or more times (which is the $(aa^*c \cup b \cup c)^*$ part). The only thing that is not covered yet is the situation that the word ends on a series of a's (which is the a^* part). Note that words ending on a b or a c are already covered in the $(aa^*c \cup b \cup c)^*$ part. So indeed $L_3 = \mathcal{L}((aa^*c \cup b \cup c)^*a^*)$.
2. We can split a word that doesn't contain ab after the c's as in $ac|c|c|bbac|bbbc|baaa$. From this example, it is hopefully clear that each block starts with a possibly empty series of b's, followed by a possibly empty series of a's, followed by exactly one c (which is the b^*a^*c part). And there can be zero or more of these blocks (which is the $(b^*a^*c)^*$ part).The only thing missing is that after the last c, there can still be a possibly empty series of b's, followed by a possibly empty series of a's (which is the b^*a^* part). So indeed $L_3 = \mathcal{L}((b^*a^*c)^*b^*a^*)$.

4. Consider the language

$$L_4 = \mathcal{L}\left(\big(a(a \cup b) \cup b\big)\big((a \cup b)(a \cup b)\big)^*\right)$$

Which words are elements of L_4?

(a) *aabbaa*

(b) *baaaaab*

(c) *ababbaba*

(d) is correct — (d) All of the above.

Answer (d) is correct.

The total expression is a concatenation of two parts. The first part $(a(a \cup b) \cup b)$ means: either take an a followed by another symbol, or take a single b. And the second part $((a \cup b)(a \cup b))^*$ means: take zero or more copies of exactly two symbols. This leads to the observation:

$$L_4 = \left\{ w \in \{a, b\}^* \;\middle|\; \begin{array}{l} w \text{ starts with the symbol } a \text{ and has an even} \\ \text{length or} \\ w \text{ starts with the symbol } b \text{ and has an odd} \\ \text{length} \end{array} \right\}$$

Now note that $aabbaa$ starts with an a and has length six, $baaaaab$ starts with a b and has length seven, and $ababbaba$ starts with an a and has length eight. So all words are in L_4.

5. Consider the grammar

$$G_5 = \langle \{a, b\}, \{S, B\}, \{S \to B, S \to aB, B \to bS, B \to \lambda\} \rangle$$

What is the minimum number of steps in a production for the word $abab$?

(a) 4 or less

(b) 5

(c) is correct (c) 6

(d) 7 or more

Answer (c) is correct.

There is only one production for this word:

$$S \to aB \to abS \to abaB \to ababS \to ababB \to abab$$

It takes six steps.

6. Consider the grammar G_6 given by the following rules:

$$\begin{array}{lcl} S & \to & A \mid aA \\ A & \to & bS \mid BS \mid \lambda \\ B & \to & bB \end{array}$$

We want to prove that $baaab \notin \mathcal{L}(G_6)$. Someone claims that this can be done with the invariant

$$P_6(w) := [\, w \text{ does not contain } aS \text{ and } w \text{ does not contain } aa \,]$$

Will this work? Explain your answer.

It won't work as $P_6(w)$ is not an invariant. Let $v = aAa$. Then $P_6(v)$ holds because aAa does not contain aS and it does not contain aa. However, because of the rule $A \to \lambda$, we have that $v \to v'$ where $v' = aa$ and $P_6(v')$ clearly doesn't hold, since it contains aa.

7. Consider the grammar G_7 given by the following rules:

$$\begin{array}{lcl} S & \to & aA \mid aB \\ A & \to & aS \\ B & \to & BB \mid b \mid \lambda \end{array}$$

Now what is the case?

(b) is correct

(a) Grammar G_7 is right linear and $\mathcal{L}(G_7)$ is a regular language.
(b) Grammar G_7 is not right linear and $\mathcal{L}(G_7)$ is a regular language.
(c) Grammar G_7 is right linear and $\mathcal{L}(G_7)$ is not a regular language.
(d) Grammar G_7 is not right linear and $\mathcal{L}(G_7)$ is not a regular language.

Answer (b) is correct.

The rule $B \to BB$ is not a right linear rule, making the grammar not right linear. However, this rule simply produces a series of zero or more b's. Hence the rule can be replaced by the right linear rule $B \to bB$. In fact, adding this rule makes the rule $B \to b$ superfluous. So if G'_7 is the grammar

$$\begin{array}{rcl} S & \to & aA \mid aB \\ A & \to & aS \\ B & \to & bB \mid \lambda \end{array}$$

then $\mathcal{L}(G_7) = \mathcal{L}(G'_7)$ and hence $\mathcal{L}(G_7)$ is a regular language. In fact, $\mathcal{L}(G_7) = \mathcal{L}((aa)^*ab^*)$.

8. Consider the grammar G_8 given by the following rules:

$$\begin{array}{rcl} S & \to & AB \mid BCS \\ A & \to & aaA \mid C \\ B & \to & bB \mid bbC \\ C & \to & Cc \mid \lambda \end{array}$$

First, give the nullable nonterminals in G_8 and then give a grammar G'_8 without any λ-rules such that $\mathcal{L}(G_8) = \mathcal{L}(G'_8)$.

The nullable nonterminals in G_8 are A and C as $C \to \lambda$ and $A \to C \to \lambda$.

Knowing that A and C are nullable, we have to add the following rules: $S \to B$, $S \to BS$, $A \to aa$, $A \to \lambda$, $B \to bb$, and $C \to c$. This leads to

$$\begin{array}{rcl} S & \to & AB \mid B \mid BCS \mid BS \\ A & \to & aaA \mid aa \mid C \mid \lambda \\ B & \to & bB \mid bbC \mid bb \\ C & \to & Cc \mid c \mid \lambda \end{array}$$

We can now safely remove the λ-rules and we get G'_8:

$$\begin{array}{rcl} S & \to & AB \mid B \mid BCS \mid BS \\ A & \to & aaA \mid aa \mid C \\ B & \to & bB \mid bbC \mid bb \\ C & \to & Cc \mid c \end{array}$$