

COMP11212 Fundamentals of Computation
Part 1: Formal Languages

Andrea Schalk
A.Schalk@cs.man.ac.uk

February 11, 2012

COMP10042 Solutions, Sheet 1

Many of the exercises have more than one correct answer, only one of which is produced here. Some exercises marked with * aren't included.

Exercises from Chapter 2

Solution 1. The results are as follows.

- (a) $\{a, aaa, aaaaa, abb, aaabb, aaaaaabb, abbb, aaabbb, aaaaaabbb\}$
- (b) $\{aa, aaa, aaaa, aaaaa\}$
- (c) $\{a, a^2, a^6, ab^2, a^2b^2, a^6b^2, ab^3, a^2b^3, a^6b^3\}$
- (d) $\{0^m \mid m \in \mathbb{N}\} \cdot \{1^n \mid n \in \mathbb{N}\}$
- (e) $\{00, 001, 0001, 010, 0101, 01001, 0010, 00101, 001001\}$

Solution 2. (a) The results are as follows.

- $\{s \mid s \text{ is a word over } \{a, b\}\}$
- $\{s \mid s \text{ is a word consisting either entirely of } as \text{ or entirely of } bs.\}$
- $\{\epsilon\}$
- $\{s \mid s \text{ consists of a number of } as \text{ divisible by 2 but not by 4}\}$
- $\{s \mid s \text{ word over } \{a, b\} \text{ containing at least one } b\}$

(b) The result is

$$\{s \mid s \text{ word over } \Sigma \text{ such that } S \text{ consists of an even number of } 0\text{s}\}.$$

To see why this is the result call this set \mathcal{L} . Clearly \mathcal{L} is a subset of $\{0^{2n} \mid n \in \mathbb{N}\}^*$ since $\mathcal{L} = \mathcal{L}^1 \subseteq \mathcal{L}^*$. On the other hand, if we concatenate any two elements of \mathcal{L} , say 0^{2m} and 0^{2n} then we obtain $0^{2m+2n} = 0^{2(m+n)}$, which is another element of \mathcal{L} . Therefore concatenating any finite number of elements of \mathcal{L} will never take us out of \mathcal{L} .

(c) Σ^*

Solution 3. See the table.

Pattern	a	ab	b	aba	$abab$	aab	$aabb$	aa
$(ab)^*$		✓			✓			
$a*b^*$	✓	✓	✓			✓	✓	✓
$(a b)$	✓		✓					
$(a b)^*$	✓	✓	✓	✓	✓	✓	✓	✓
$ab b a^*$	✓	✓	✓					✓

Solution 4. We obtain the following.

- (a) $\{0, 1, 2\}$.
- (b) $\{002, 022, 102, 122\}$.
- (c) $\{01, 10\}$.
- (d) Words consisting of arbitrarily many 0s followed by one 1, or $\{0^n 1 \mid n \in \mathbb{N}\}$.

- (e) Words consisting of arbitrarily many occurrences of 01 followed by one 1, or $\{(01)^n 1 \mid n \in \mathbb{N}\}$.
- (f) Words consisting of arbitrarily many 0s followed by arbitrarily many 1s.
- (g) Words consisting of arbitrarily many copies of 010, so ϵ , 010, 010010, and so on.
- (h) Words consisting of arbitrarily many copies of 01 followed by arbitrarily many 0s.
- (i) Words consisting of arbitrarily many copies of 01.
- (j) The same as $\{0, 1\}^*$, that is, every word over the alphabet $\{0, 1\}$.
- (k) All words over the alphabet $\{0, 1, 2\}$ which mention at most one of the letters.
- (l) $\{0, 2\} \cup \{1^n \mid n \in \mathbb{N}\}$.

Solution 5. This is as easy as it looks. The only word that matches ϵ is ϵ and the only word that matches the pattern a (for $a \in \Sigma$) is the word a .

Solution 6. We explain the reasons.

- (a) The word aba matches the pattern $(ab)^*a$ as

ab matches $(ab)^*$	and	a matches a	as
ab matches ab	and	a matches a	as
a matches a and b matches b	and	a matches a .	

- (b) The word 10010 matches the pattern $(0|1)^*10$ as

100 matches $(0 1)^*$	and	10 matches 10	as
1, 0, 0 match $(0 1)$	and	1 matches 1 and 0 matches 0	as
1 matches 1 and 0 matches 0	and	1 matches 1 and 0 matches 0.	

- (c) The word 0010 matches the pattern $(0^*|1^*)10$ as

00 matches $(0^* 1^*)$	and	10 matches 10	as
00 matches 0^*	and	1 matches 1 and 0 matches 0	as
0 matches 0	and	1 matches 1 and 0 matches 0.	

- (d) The word a matches the pattern $(abc)^*a$ as

ϵ matches $(abc)^*$	and	a matches a .
------------------------------	-----	-------------------

Solution 7. The following applies to `grep` as it is installed on my machine. Different versions may vary slightly.

The ‘bracket expressions’ that appear in `grep` can all be generated using $(\cdot | \cdot | \dots | \cdot)$, listing all the elements in the corresponding bracket expression separated by the alternative operator `|`. Similarly for period—it requires taking the alternative of all available symbols.

The use of caret and dollar sign requires matching ϵ , and requires us to have a special symbol for carriage returns (or however linebreaks are marked

in the file in question). Similarly for $\backslash B$ and $\backslash b$, only involving the character for space \sqcup .

For $?$ we have to take the expression p this operator is applied to, and replace it by $(\epsilon|p)$.

For $*$ we have our $*$.

For $+$ applied to a pattern p take pp^* .

For $\{n\}$ applied to an expression p take p concatenated n times (this would be tedious to type, so one can see why a shortcut was developed here).

For $\{n, \}$ applied to an expression p , take p concatenated n times followed by p^* .

For $\{n, m\}$ take p concatenated n times, followed by $(\epsilon|p|\dots|p^{m-n})$.

Solution 8. We obtain the following.

$$\begin{aligned}
 \text{(a)} \quad \mathcal{L}((0|1)^*) &= (\mathcal{L}(0|1))^* \\
 &= (\mathcal{L}(0) \cup \mathcal{L}(1))^* \\
 &= (\{0\} \cup \{1\})^* \\
 &= \{0, 1\}^*
 \end{aligned}$$

This is the language of all words over the alphabet $\{0, 1\}$.

$$\begin{aligned}
 \text{(b)} \quad \mathcal{L}(0^*|1^*) &= \mathcal{L}(0^*) \cup \mathcal{L}(1^*) \\
 &= (\mathcal{L}(0))^* \cup (\mathcal{L}(1))^* \\
 &= \{0\}^* \cup \{1\}^*
 \end{aligned}$$

This is the language of all words over the alphabet $\{0, 1\}$ which consist either entirely of 0s or entirely of 1s.

$$\begin{aligned}
 \text{(c)} \quad \mathcal{L}((01)^*0) &= \mathcal{L}((01)^*) \cdot \mathcal{L}(0) \\
 &= (\mathcal{L}(01))^* \cdot \{0\} \\
 &= (\mathcal{L}(0) \cdot \mathcal{L}(1))^* \cdot \{0\} \\
 &= (\{0\} \cdot \{1\})^* \cdot \{0\} \\
 &= \{01\}^* \cdot \{0\}
 \end{aligned}$$

This is the language of all words that consist of an arbitrary number of 01s followed by 0.

$$\text{(d)} \quad \mathcal{L}((00)^*) = (\mathcal{L}(00))^* = (\mathcal{L}(0) \cdot \mathcal{L}(0))^* = (\{0\} \cdot \{0\})^* = \{00\}^*$$

This is the language of all words consisting of an even number of 0s.

$$\begin{aligned}
 \text{(e)} \quad \mathcal{L}(((0|1)(0|1))^*) &= (\mathcal{L}((0|1)(0|1)))^* \\
 &= (\mathcal{L}(0|1) \cdot \mathcal{L}(0|1))^* \\
 &= ((\mathcal{L}(0) \cup \mathcal{L}(1)) \cdot (\mathcal{L}(0) \cup \mathcal{L}(1)))^* \\
 &= ((\{0\} \cup \{1\}) \cdot (\{0\} \cup \{1\}))^* \\
 &= \{0, 1\} \cdot \{0, 1\}^* \\
 &= \{00, 01, 10, 11\}^*
 \end{aligned}$$

This is the language of all words over $\{0, 1\}$ which consist of an even number of symbols (that includes the empty word which consists of 0 symbols).

Solution 9. We give the required patterns.

- (a) $0(0|1)^*1$
- (b) $1^*01^*0(1|0)^*$
- (c) $(11^*0|00^*1)(0|1)^*$
- (d) $(0|1)^*0(0|1)$
- (e) $(0|1)^*11(0|1)^*$
- (f) $(0|1)(0|1)(0|1)(0|1)^*$
- (g) $(\epsilon|0|1)(\epsilon|0|1)(\epsilon|0|1)(\epsilon|0|1)^2$
- (h) $0(00|01|10|11)^*$
- (i) $((0|1)0)^*(0|1\epsilon)$
- (j) $(0|1)(0|1)^*$
- (k) $000^*|1000^*|0100^*|000^*10^*$
- (l) $((0|10|110|111(0|1))(0|1)^*)|\epsilon^3$
- (m) $((1^*01^*01^*)^*)|\epsilon$
- (n) $((1^*01^*01^*01^*)^*)|\epsilon$
- (o) $(0^*10)^*(\epsilon|0|1)$
- (p) 0^*1^*4
- (q) $(0^*11^*00)^*(0^*11^*(\epsilon|0)|\epsilon)$
- (r) Does anybody have a reasonable regular expression for this one?

Solution 10. Again we give the required regular expressions.

- (a) $(a|b)^*5$
- (b) $((b|c)^*ab)^*(b|c)^*$
- (c) A fairly short answer is $(aa^*c|b|c)^*(a^*)$. The following helps with the pattern required for the next part of this exercise:⁶

$$(((b|c)^*(aa^*c))^*((b|c)^*(\epsilon|aa^*)))$$

- (d) $((b|c)^*(aa^*(c|b(b|c))))^*((b|c)^*(\epsilon|aa^*(\epsilon|b)))$

²This is why we need a pattern that is matched by the empty word.

³Did you remember to make sure the empty word matches your pattern?

⁴As soon as we have seen 1 we may only see further 1s.

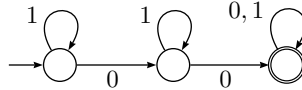
⁵The trick is to realize the pattern shouldn't contain c .

⁶Note that this is a little bit harder than part (n) from the previous exercise.

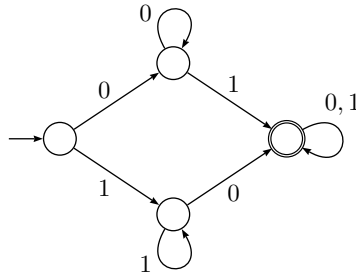
Exercises from Chapter 3

Solution 11. We give automata that work as required.

(b)



(c)



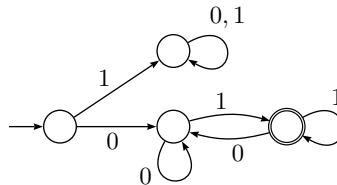
Solution 12. The first and third word end up in the right-most state, the second one in the only accepting state. The language of all words that finish there is

$$\{ab^n a \mid n \in \mathbb{N}\} \cup \{ba^n b \mid n \in \mathbb{N}\}.$$

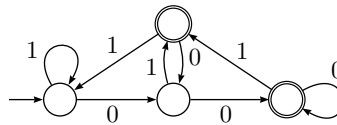
They are all the words that start with one letter (a or b), then have arbitrarily many copies of the other letter, and finish with the first letter again.

Solution 13. We give suitable automata.

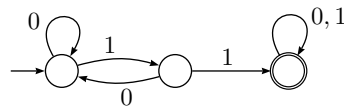
(a)



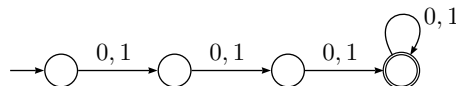
(d)



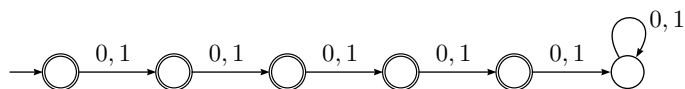
(e)



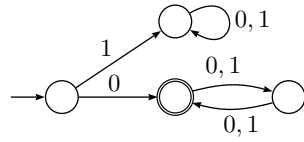
(f)



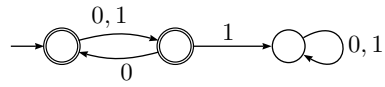
(g)



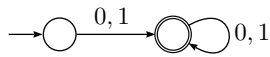
(h)



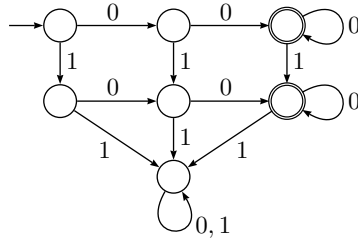
(i)



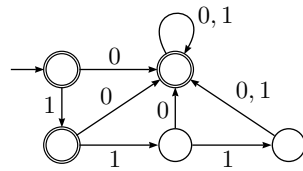
(j)



(k)

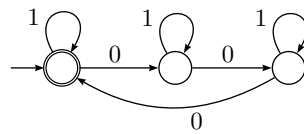


(l)

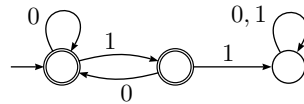


(m) This is the first example discussed in Chapter 3.

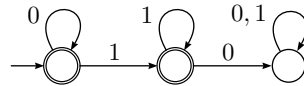
(n)



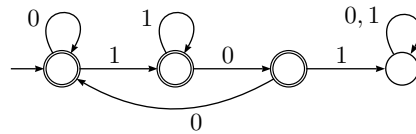
(o)



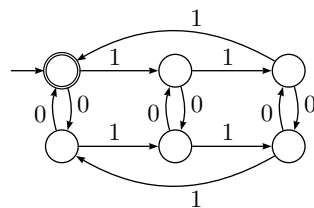
(p)



(q)

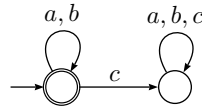


(r)

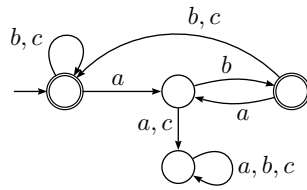


Solution 14. Again we give automata that do the job.

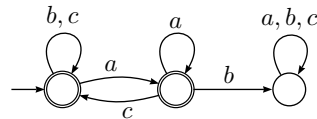
(a)



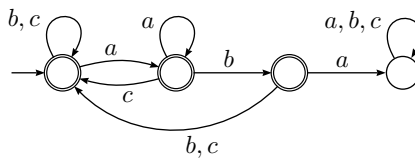
(b)



(c)



(d)



Exercises from Appendix A

Solution 48. We give the results.

(a) $\{aaa, aab, aa1, aba, abb, ab1, a1a, a1b, a11, baa, bab, ba1, bba, bbb, bb1, b1a, b1b, b11, 1aa, 1ab, 1a1, 1ba, 1bb, 1b1, 11a, 11b, 111\}$

(b) $\{\epsilon, 0, 00, 000, 0000, 00000, \dots\}$

(c) $\{\epsilon\}$.

Solution 49. We can define the length alternatively by setting

$$|x_1x_2 \cdots x_n| = n.$$

It is worth noting that this also works for the empty word since when $n = 0$ we have a word that consists of no letters at all.

To argue that this agrees with the original definition of the length function, we use this to derive the following:

$$\begin{aligned} |x_1x_2 \cdots x_n| &= |x_1x_2 \cdots x_{n-1}| + 1 \\ &= |x_1x_2 \cdots x_{n-2}| + 2 \\ &\dots \\ &= |\epsilon| + n \\ &= 0 + n \\ &= n \end{aligned}$$

Solution 50. You may find it most difficult to decide how to write down a suitable argument.

- (a) One way of presenting a coherent argument here is as follows. We have to show that for all words over some alphabet s, t and u we have that $(s \cdot t) \cdot u = s \cdot (t \cdot u)$. Let $s = x_1x_2 \cdots x_l$, $t = y_1y_2 \cdots y_m$ and $u = z_1z_2 \cdots z_n$. Then

$$\begin{aligned} (s \cdot t) \cdot u &= (x_1x_2 \cdots x_ly_1y_2 \cdots y_m) \cdot z_1z_2 \cdots z_n \\ &= x_1x_2 \cdots x_ly_1y_2 \cdots y_mz_1z_2 \cdots z_n \\ &= x_1x_2 \cdots x_l \cdot (y_1y_2 \cdots y_mz_1z_2 \cdots z_n) \\ &= s \cdot (t \cdot u). \end{aligned}$$

- (b) Showing that this operation is not commutative is very simple. For example, $0 \cdot 1 = 01 \neq 10 = 1 \cdot 0$.

Solution 51. The difficulty consists of deciding what to do about the two arguments. But since we don't have to look 'inside' the first argument it turns out that we can recurse just over the second. Let s and t be words over Σ .

$$s \cdot t = \begin{cases} s & t = \epsilon \\ (s \cdot t')x & t = t'x \end{cases}$$

Again note how we decompose t following the way a word is built in the recursive definition.

Solution 52. This is much easier if we follow the hint. If the original definition is used then the proof has to be by induction.

Assume that $s = x_1x_2 \cdots x_m$ and $t = y_1y_2 \cdots y_n$. Then

$$\begin{aligned} |s \cdot t| &= |(x_1x_2 \cdots x_m) \cdot (y_1y_2 \cdots y_n)| \\ &= |x_1x_2 \cdots x_my_1y_2 \cdots y_n| \\ &= |z_1z_2 \cdots z_{m+n}| \\ &= m + n \\ &= |x_1x_2 \cdots x_m| + |y_1y_2 \cdots y_n| \\ &= |s| + |t|, \end{aligned}$$

where

$$z_i = \begin{cases} x_i & 1 \leq i \leq m \\ y_{i-m} & m + 1 \leq i \leq n \end{cases}$$

Solution 53. We give the two recursive definitions.

- (a) We define the operation, say d , as follows:

- The result of applying d to ϵ is ϵ , that is, $d(\epsilon) = \epsilon$.
- The result of applying d to a word $s \cdot x$ is defined to be $d(s) \cdot xx$.

- (b) We define the operation, say r , as follows:

$$r(s) = \begin{cases} \epsilon & s = \epsilon \\ x \cdot r(s') & s = s'x \end{cases}$$

Solution 54. We get 00000, 000111, 010010, 0101010, ϵ .

Solution 55. These shouldn't be hard based on what you have learned in COMP10020. We calculate as follows.

$$\mathcal{L}_1 \cap \mathcal{L}_2 = \emptyset$$

$$\mathcal{L}_1 \cup \mathcal{L}_2 = \{s \mid \text{the first letter of } s \text{ is } 0 \text{ and the last letter of } s \text{ is } 2 \\ \text{or } s \text{ does not contain the letter } 2\}$$

$$\{s = x_1x_2 \cdots x_m \mid m \in \mathbb{N}, x_i \in \{0, 1, 2\} \text{ for } 1 \leq i \leq m \text{ and} \\ \exists i \in \{1, 2, \dots, m\} \text{ such that } x_i = 2\}$$

Solution 56. We get the following.

- (a) $\{aabbb, baabb, bbaab, bbbaa, ababb, abbab, abbba, babab, babba, bbaba\}$
- (b) $\{1^{2n} \mid n \in \mathbb{N}\}$ or $\{(11)^n \mid n \in \mathbb{N}\}$
- (c) $\{a^m b b^n \mid m, n \in \mathbb{N}\}$
- (d) $\{(ab)^n \mid n \in \mathbb{N}^+\}$
- (e) $\{(ab)^n a \mid n \in \mathbb{N}\}$
- (f) $\{0^n 1^n \mid n \in \mathbb{N}\}$
- (g) $\{x_1 x_2 \cdots x_n \mid n \in \mathbb{N}, x_i \in \Sigma \text{ for } 1 \leq i \leq n\}$

Solution 57. The results are given below.

- (a) We get $\{aa, ab, ba, bb\}$ and $\{abab, abba, baab, baba\}$.
- (b) $\{0, 1, 2\}^3$. Slightly longer, but describing the same set is

$$\{x_1 x_2 x_3 \mid x_1, x_2, x_3 \in \{0, 1, 2\}\}.$$

- (c) It is very difficult to find a concise notation for the set that arises here, which is one of the reasons why the notation we employ is so useful. The strings in question consists of an blocks of 0s and 1s of even lengths in an arbitrary order.
- (d) By definition, an element of Σ_1^* is a string consisting of a finite number of letters from Σ_1 . But every letter in Σ_1 is a letter in Σ_2 and so this is also a word over the alphabet Σ_2 and so an element of Σ_2^* .