

[20241219] INFOB3TC - Talen en compilers - 2 - USP

Cursus: BETA-INFOB3TC Talen en compilers (INFOB3TC)

Tijdsduur: 2 uur
Aantal vragen: 17
Gegenereerd op: 19 dec 2024

Inhoud:

Pagina's:

- A. Voorpagina **1**
- B. Vragen **14**
- C. Antwoordformulier **2**
- D. Uitwerking van het antwoord **4**

[20241219] INFOB3TC - Talen en compilers - 2 - USP

Cursus: Talen en compilers (INFOB3TC)

Aantal vragen: 17

In totaal zijn 51 punten voor deze toets te behalen, 30,85 punten zijn nodig om voor de toets te slagen.

1

2 pt.

Which abstract syntax is the best fit for the following grammar:

$V \rightarrow Va \mid bV \mid V[V] = V$

- a. data $V \ a \ b = \text{SingleA } a \mid \text{SingleB } b \mid \text{Assign } (V \ a \ b) \ (V \ a \ b) \ (V \ ab)$
- b. data $V = V \ ((\text{Maybe } V), (\text{Maybe } V), [V, V, V])$
- c. data $V = Av \ V \mid Bv \ V \mid \text{Assign } V \ V \ V$
- d. data $T = \text{SingleA Char} \mid \text{SingleB Char} \mid \text{Assign String String String}$

2 Given is the following Haskell datatype:

```
data RoseTree a = Rose a (List (RoseTree a))
data List b = Nil | Cons b (List b)
```

4 pt. a. We define a combined algebra for *RoseTree* and *List* as follows

```
type RoseTreeAlgebra a r l = (... , ..., ...)
```

Here, we choose *a* to correspond to *a*, *r* to correspond to *RoseTree a*, and *l* to correspond to *List (RoseTree a)*.

What is the type of the component corresponding to the constructor *Cons*?

3 pt. b. The fold function for *RoseTree* has the following template:

```
foldRoseTree (rose, nil, cons) = f where
```

```
  f (Rose a rs) = ...
```

```
  g Nil = ...
```

```
  g (Cons r rs) = ...
```

What code should be written on the dots for the *Rose* case?

3 pt. c. We want to use the function *foldRoseTree* to compute the height (a.k.a. maximum depth) of a *RoseTree*.

For example, the height of

```
`RoseTree a []`
```

is 1, and the height of

```
`RoseTree a [ RoseTree b [ RoseTree c [] ], RoseTree d [] ]`
```

is 3.

The following template is given:

```
height r = foldRoseTree ((+1), 0, ...)
```

What should be written on the dots?

Give your answer as a single line of simple code.

3

2 pt.

A grammar has the following productions:

$S \rightarrow aSb \mid Sa \mid \varepsilon$

We need one more production to be able to derive the sentence *aabbaabb*. Which of the following productions should we add?

- a. $S \rightarrow bbSaa$
- b. $S \rightarrow aSbb$
- c. All the other answers are correct.
- d. $S \rightarrow SS$

4

1 pt.

Is the following statement **false** or **true**?

If $A \rightarrow bCX$ is the only production with A as a left hand side in a context-free grammar G , then it can never be the case that $A \Rightarrow^* bC$

- a. False
- b. True

5

1 pt.

Is the following statement **false** or **true**?

A context-free grammar that employs a finite alphabet can describe a language with infinitely long sentences.

- a. False
- b. True

6

1 pt.

Is the following statement **false** or **true**?

A context-free grammar that employs a finite alphabet can describe a language with infinitely many sentences.

- a. False
- b. True

7 Is the following statement **false** or **true**?

1 pt.

If $A \rightarrow bC$ is a production of a context-free grammar G , then $AXY \Rightarrow^* bCXY$

a. False

b. True

8 Assume that the function *abs* computes the absolute value of an integer.

3 pt.

Which of the following grammars generates the language

$$\{a^k b^n c^m \mid k = \text{abs}(n - m), k, n, m \geq 0\}$$

a. $S \rightarrow T \mid U$

$T \rightarrow AB$

$A \rightarrow aAb \mid \varepsilon$

$B \rightarrow bBc \mid \varepsilon$

$U \rightarrow aUc \mid V$

$V \rightarrow bVc \mid \varepsilon$

b. $S \rightarrow aSc \mid T$

$T \rightarrow bTc \mid \varepsilon$

c. $S \rightarrow U \mid T$

$U \rightarrow aUb \mid bUc \mid \varepsilon$

$T \rightarrow bTc \mid aTc \mid \varepsilon$

d. $S \rightarrow TU$

$T \rightarrow A \mid B$

$A \rightarrow aAb \mid \varepsilon$

$B \rightarrow bBc \mid \varepsilon$

$U \rightarrow V \mid W$

$W \rightarrow aWc \mid \varepsilon$

$V \rightarrow bVc \mid \varepsilon$

9

3 pt.

What is the language generated by the following grammar:

$S \rightarrow AB$

$A \rightarrow aAb \mid ab$

$B \rightarrow bbBa \mid bba$

- a. $\{a^n b^{3n} a^n \mid n \geq 1\}$
- b. $\{a^n b^n b^{2m} a^m \mid n, m \geq 1\}$
- c. $\{vwx \mid v, w \in a^n, x \in b^{3n}, n \geq 1\}$
- d. All other answers are wrong.

10

3 pt.

Given is the following grammar:

$$X \rightarrow X T T X \mid S$$
$$S \rightarrow ab \mid cd$$
$$T \rightarrow e$$

X is the start symbol. Which of the following grammars is not left recursive, but generates the same language?

a.

$$X \rightarrow X (T T S)^*$$
$$S \rightarrow ab \mid cd$$
$$T \rightarrow e$$

b.

$$X \rightarrow S \mid S Z$$
$$Z \rightarrow T T X \mid T T X Z$$
$$S \rightarrow ab \mid cd$$
$$T \rightarrow e$$

c.

$$X \rightarrow Z T T X \mid S$$
$$Z \rightarrow X \mid \varepsilon$$
$$S \rightarrow ab \mid cd$$
$$T \rightarrow e$$

d.

$$X \rightarrow S T T Z$$
$$Z \rightarrow S \mid X$$
$$S \rightarrow ab \mid cd$$
$$T \rightarrow e$$

11

3 pt.

Consider the following parser:

$pZ :: \text{Parser Token Int}$

$pZ = f \<\$> \text{plnt} \<*> \text{many plnt}$

The parser plnt parses a single token to an integer. It has the type Parser Token Int .

What is the type of f ?

- a. $\text{Parser Token } (\text{Int} \rightarrow [\text{Int}] \rightarrow \text{Int})$
- b. $\text{Parser Token } [\text{Int}]$
- c. $\text{Int} \rightarrow \text{Int}$
- d. $\text{Int} \rightarrow [\text{Int}] \rightarrow \text{Int}$

12

2 pt.

Consider the following parser:

$pX :: \text{Parser Token Int}$

$pX = pY \<*> \text{plnt}$

The parser plnt parses a single token to an integer. It has the type Parser Token Int .

Consider the parser pY used by pX above. What would be the type of the result obtained by applying pY to a list of tokens? In other words, what is the type of:

$\text{runParser } pY \text{ input}$

where input is a list of tokens?

- a. $\text{Int} \rightarrow \text{Int}$
- b. $(\text{Int} \rightarrow \text{Int}, [[\text{Token}]])$
- c. $[(\text{Int}, [\text{Token}])]$
- d. $[(\text{Int} \rightarrow \text{Int}, [\text{Token}])]$

13

3 pt.

What follows are several properties involving parser combinators.

We say that $p \cong q$ if $\text{runParser } p \text{ input} == \text{runParser } q \text{ input}$ for all possible inputs.

Which of these properties is **NOT** true?

- a. $\text{many } (\text{some } p) \cong \text{some } (\text{many } p)$
- b. $(:) <\$> p <*> \text{many } p \cong \text{some } p$
- c. $\text{satisfy } (== c) \cong \text{symbol } c$
- d. $\text{greedy1 } (\text{greedy1 } p) \cong (\backslash x \rightarrow [x]) <\$> \text{greedy1 } p$

14

3 pt.

For any two regexps P and Q, we say that

$$P \cong Q$$

iff P matches exactly the same language as Q.

Which of the following statements are true?

- a. For every regexp P, Q
 $(PQ)^+ \cong P(QP)^+$
- b. For every regexp P
 $P^* | P \cong P^+$
- c. For every regexp P
 $(PP^*)^+ \cong P^+$
- d. For every regexp P, Q
 $(P|Q)^+ \cong P^+ | Q^+$
- e. For every regexp P, Q, R, S
 $P(Q|R)S \cong PQS | PRS$

15

4 pt. Consider these two lists of words:

List P:

gabbed
gaberloonie
garbed
gibblegable
giblet
gobleted
gombeen
gooseberry
griegen
guberla
gubernation
orthobenzoquinone
overbeetling
overblew
overexuberant

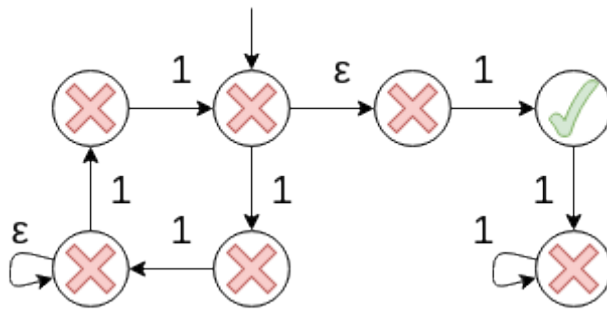
List N:

obvolute
guntub
gigantoblast
obdormition
octosyllabic
specklebelly
soberest
reputabley
beachmen
fiberize
matchable

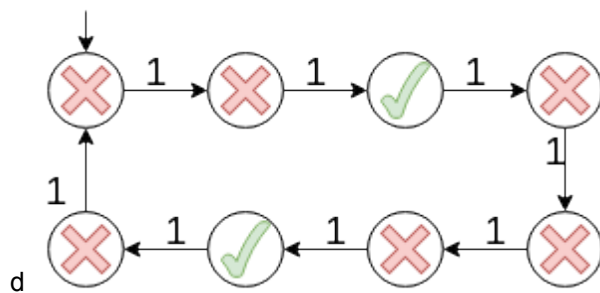
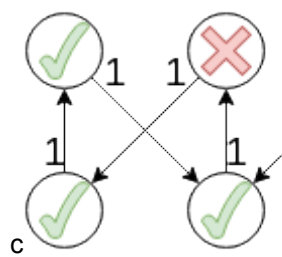
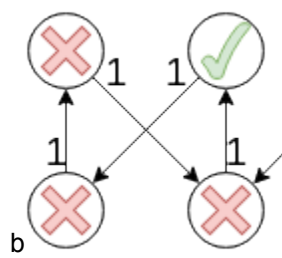
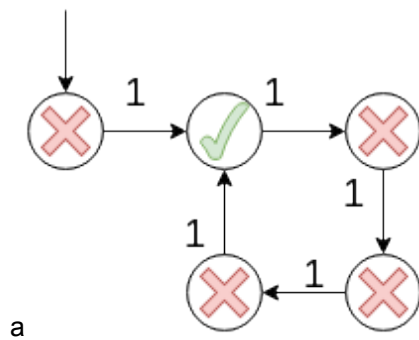
Which of the following regexps both match all of the words in List P, and match none of the words in List N?

- a. `.*g(a|o|r|(i|u)b)|o(r|v).*`
- b. `(g|o).*bl?e.*`
- c. `(g|o)b*l?e.*`
- d. `(g|o).*b*l?.*`
- e. `gabbed|gaberloonie|garbed|gibblegable|giblet|gobleted|gombeen|gooseberry|griegen|guberla|gubernation|orthobenzoquinone|overbeetling|overblew|overexuberant`
- f. `g(a|o|r|(i|u)b)|o(r|v).+`

16 This question is about one NFA ϵ :



and several DFAs:



All of these automata work with the alphabet $\{1\}$.

3 pt. **a.** Which of these DFAs represent the same language as the NFA ϵ ? Check all that apply.

a. a

b. b

c. c

d. d

2 pt. **b.** Which of these DFAs is the result of the subset construction on the NFA?

a. a

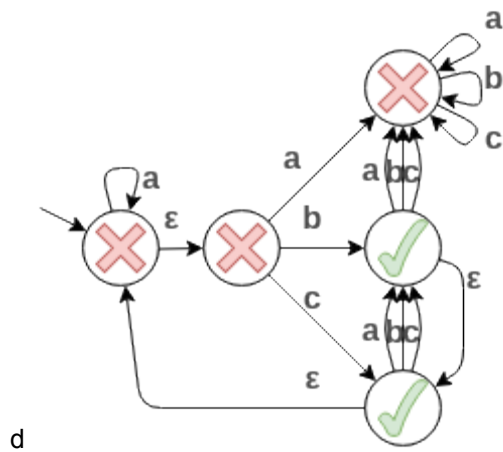
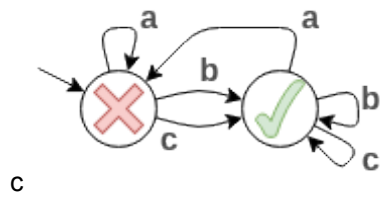
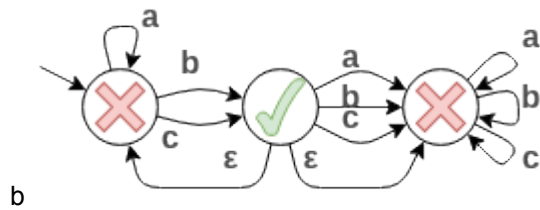
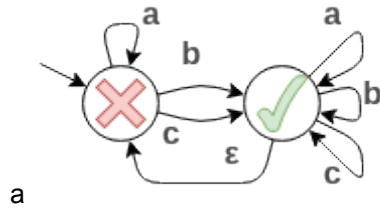
b. b

c. c

d. d

17 This question is about the following Regular Expression over the alphabet {a,b,c}:
 4 pt. $(a^*(b|c))^+$

And the following four diagrams:



Which of these are valid NFA_εs that represent the regular expression? Check all that apply.

- a.** a
- b.** b
- c.** c
- d.** d

Naam:

Handtekening:

Datum:

Geboortedatum:

Cursus: BETA-INFOB3TC Talen en compilers (INFOB3TC) - Vragen: [20241219] INFOB3TC - Talen en compilers - 2 - USP

1 A B C D
2 pt. ☐ ☐ ☐ ☐

2
10 pt. **a.** Antwoord:

2

b. Antwoord:

2

c. Antwoord:

2

3 A B C D
2 pt. ☐ ☐ ☐ ☐

4 A B
1 pt. ☐ ☐

5 A B
1 pt. ☐ ☐

6 A B
1 pt. ☐ ☐

7 A B
1 pt. ☐ ☐

8 A B C D
3 pt. ☐ ☐ ☐ ☐

9 A B C D
3 pt. ☐ ☐ ☐ ☐

10 A B C D
3 pt. ☐ ☐ ☐ ☐

11 A B C D
3 pt. ☐ ☐ ☐ ☐

12 A B C D
2 pt. ☐ ☐ ☐ ☐

13 A B C D
3 pt. ☐ ☐ ☐ ☐

14
3 pt.

A B C D E
☐ ☐ ☐ ☐ ☐

Er zijn meerdere antwoorden mogelijk

15
4 pt.

A B C D E F
☐ ☐ ☐ ☐ ☐ ☐

Er zijn meerdere antwoorden mogelijk

16
5 pt.

a. A B C D
☐ ☐ ☐ ☐

Er zijn meerdere antwoorden mogelijk

b. A B C D
☐ ☐ ☐ ☐

17
4 pt.

A B C D
☐ ☐ ☐ ☐

Er zijn meerdere antwoorden mogelijk

Uitwerking van het antwoord

1. C

2 pt.

2.

10 pt.

a.

Correction criterion	Points
function type with arity 2	0 to 1 points
arguments "r" and "l"	0 to 1 points
result type "l"	0 to 1 points
Precisely written: <code>r -> l -> l</code>	0 to 1 points
<i>Total points:</i>	<i>4 points</i>

b.

Correction criterion	Points
calls g on rs	0 to 1 points
calls rose	0 to 1 points
Precisely written: <code>"rose a (g rs)"</code>	0 to 1 points
<i>Total points:</i>	<i>3 points</i>

c.

Correction criterion	Points
Precise, elegant code: <code>\x y -> if x > y then x else y,</code> or alternatively <code>max</code> Look at things like correct capitalization etc.	0 to 1 points
<code>Int -> Int -> Int</code>	0 to 1 points
compare two numbers	0 to 1 points
<i>Total points:</i>	<i>3 points</i>

3. C

2 pt.

4. A

1 pt.

5. A

1 pt.

6. B

1 pt.

7. B

1 pt.

8. A

3 pt.

9. B
3 pt.

10. B
3 pt.

11. D
3 pt.

12. D
2 pt.

13. A
3 pt.

14. 1 pt. A
3 pt. B
1 pt. C
D
1 pt. E
Bonus: 1 pt.

15. A
4 pt. 1 pt. B
C
D
1 pt. E
1 pt. F
Bonus: 2 pt.

16. a. 1 pt. A
5 pt. 1 pt. B
C
D
Bonus: 1 pt.
b. 2 pt. B

17. A
4 pt. 1 pt. B
1 pt. C
D
Bonus: 2 pt.

Cesuur

Toegepaste raadscore: 10,697 pt

Behaalde punten	Cijfer
51	10
50	9,78
49	9,55
48	9,33
47	9,11
46	8,88
45	8,66
44	8,44
43	8,21
42	7,99
41	7,77
40	7,54
39	7,32
38	7,10
37	6,87
36	6,65
35	6,43
34	6,20
33	5,98
32	5,76
31	5,53
30	5,31
29	5,09
28	4,86
27	4,64
26	4,42
25	4,19
24	3,97
23	3,75

22	3,52
21	3,30
20	3,08
19	2,85
18	2,63
17	2,41
16	2,18
15	1,96
14	1,74
13	1,51
12	1,29
11	1,07
10	1,00
9	1,00
8	1,00
7	1,00
6	1,00
5	1,00
4	1,00
3	1,00
2	1,00
1	1,00
0	1,00

Vraag-identificatiecodes

Deze identifiërs kunnen worden gebruikt om de precieze vraag in de vragenbanken te identificeren. Gebruik deze code in combinatie met de documentcode wanneer u feedback doorgeeft, zodat precies duidelijk is op welke vraag en -versie uw feedback van toepassing is.

Documentidentificatiecode: 66952-112500

Vraagnummer	Vraag-identificatiecode	Versie-identificatiecode
1	332565	d16dea0f-42b8-4a17-931e-ea5417d81378
2	334005	bb155a3c-9036-401b-a560-a8182b2ec8ce
3	329032	18e92082-205b-4f46-9a03-dae96cbc0571
4	328804	bde965bb-61ee-4bb5-8f63-780f59669cf7
5	328808	054952e1-1433-4071-9b0c-1131ac0049aa
6	328816	701bfa18-8c06-4ece-b0ca-59279bb4a50e
7	328764	305288ac-ecf5-4b37-9c76-be4a95e2a6a0
8	334590	d6075810-ccb4-49af-9103-815427e10154
9	332860	0b6357c6-9106-4716-a345-9516a0a284bb
10	328820	a2c12c63-5f1b-4088-8786-8faa8f2197f1
11	585017	d5074e06-5b74-469a-83a8-4470c924b0e6
12	585007	a123de08-569f-4080-af6e-f71f8cda8cf2
13	334500	d28aa504-20cb-4fce-87c5-489e2737a70d
14	328704	a6eefe16-9556-4e63-9a48-5c890309e88d
15	627970	b8c71cf6-5995-4b92-8063-e87f15f7a723
16	627976	e3336ed3-123c-4e17-bb1a-8bf7fc1f5267
17	627980	2def8887-641c-47a1-924f-a7a51fcfbe56