

1. (8 p)

Lösning:

- (a) Möjliga lösningar (samma lösning med olika nivå av syntaktiskt socker):
[0-9] [0-9] [0-9] [0-9] - [0-9] [0-9] - [0-9] [0-9]
[0-9]{4} - [0-9]{2} - [0-9]{2}
- (b) Möjlig lösning:
[0-9]{4} - (0[1-9] | 1[0-2]) - (0[1-9] | [1-2][0-9] | 3[0-1])
- (c) Ja, det är alltid reguljärt. Det finns bara ändligt många strängar (högst 10000 möjliga år, 100 möjliga månader, 100 möjliga dagar), och ändliga språk är alltid reguljära eftersom man kan skriva det reguljära uttrycket " $s_1 | s_2 | \dots | s_n$ " där $s_1, \dots, s_n$ är strängarna i språket.

Rättning:

- (a) (1p)
- 1 poäng för ett korrekt uttryck, 0 poäng annars.
- (b) (4p)
- Ett poängs avdrag för vart och ett av följande test som blir fel (högst 4 poängs avdrag, poängen kan inte bli negativ):
- Strängar som inte matchar uppgift (a) ska inte accepteras.
 - Strängen "2015-**13**-14" ska **inte** accepteras
 - Strängen "2015-09-**32**" ska **inte** accepteras
 - Strängen "2015-**20**-10" ska **inte** accepteras
 - Strängen "2015-09-**00**" ska **inte** accepteras
 - Strängen "2015-**00**-10" ska **inte** accepteras
 - Strängen "2015-12-31" ska accepteras
 - Strängen "4711-02-20" ska accepteras
 - Strängen "9999-10-15" ska accepteras
 - Strängen "0001-01-01" ska accepteras
- (c) (3p)
- 1 poäng för rätt svar (ja/nej).
 - 2 poäng för korrekt motivering (räcker att säga att språket är ändligt och därför reguljärt, behöver inte motivera varför ändliga språk är reguljära).

2. (9 p)

Lösning:

- (a) Indata har preprocessats. De tecken-sekvenser som bildar de enkla datatyperna strängar, tal, och Booleans har redan identifierats och ersatts av de nya slutsymbolerna **string**, **number** och **bool**. Vid parsning kan man nu tänka på dessa som symboler i indata istället för enskilda tecken.

(b)

$$\text{Value} \rightarrow \text{string} \mid \text{number} \mid \text{bool} \mid \text{List} \mid \text{Object}$$

(c) Möjlig lösning:

$$\begin{aligned}\text{List} &\rightarrow [] \mid [\text{ListElems}] \\ \text{ListElems} &\rightarrow \text{Value} \mid \text{Value}, \text{ListElems}\end{aligned}$$

(d) Möjlig lösning:

$$\begin{aligned}\text{Object} &\rightarrow \{ \} \mid \{ \text{ObjectElems} \} \\ \text{ObjectElems} &\rightarrow \text{KeyValue} \mid \text{KeyValue}, \text{ObjectElems} \\ \text{KeyValue} &\rightarrow \text{string} : \text{Value}\end{aligned}$$

(e) Object

Rättning:

- (a) Två poäng för en korrekt och tydlig beskrivning.
Ett poäng om det verkar rätt men förklaringen är otydlig och svårförstådd.
- (b) Ett poäng för korrekt svar.
- (c) Två poäng för en korrekt lösning.
Ett poängs avdrag för upp till två mindre fel, t.ex. om tom lista inte tillåts eller om “[Value,]” tillåts.
- (d) Tre poäng för en korrekt lösning.
Ett poängs avdrag för upp till två mindre fel, t.ex.:
- om tom lista inte tillåts
 - om “{KeyValue,}” tillåts.
 - om godtyckliga värden (inte bara strängar) tillåts som nycklar
- Två poängs avdrag för upp till fyra mindre fel.
- (e) Ett poäng för korrekt svar.

3. (3 p)

Lösning:

- (a) **Sant.** De språk som kan beskrivas av en stackautomat kallas deterministiska kontextfria språk och är en strikt delmängd till de kontextfria språken. Ett exempel på ett språk som är kontextfritt men inte deterministiskt kontextfritt är palindrom.
- (b) **Sant.** LL(1)-grammatiker är aldrig tvetydiga – det finns aldrig mer än en härledning eftersom man kan använda en rekursiv medåknings-parser som i varje steg kan identifiera exakt vilken produktionsregel som ska användas baserat på nästa tecken i indata.
- (c) **Falskt.** Till exempel är språket som består av alla strängar ett reguljärt språk (reguljärt uttryck “.”). En delmängd av detta är språket som består av alla balanserade parentesuttryck (eller för all del alla syntaktiskt korrekta Java-program, eller vilket språk som helst), men detta är inte reguljärt.

Rättning:

- −1 för felaktiga sant/falskt-svar
- +1 för korrekta sant/falskt-svar