

1. (2 p)

Lösning:

Nonsens “icke-accepterande slutsymboler” – slutsymboler är inte accepterande/icke-accepterande.

Felaktigt att göra lexikal analys ändrar inte huruvida det går att skriva en grammatik för ett språk.

Rättning:

1p för nonsens-delen av påståendet

1p för det felaktiga i påståendet

Vanliga **felaktiga** svar (som **inte ger poäng**):

- Grammatik går alltid att skriva för alla språk så nonsens att det inte gick.
- Kan inte göra lexikal analys utan att ha en grammatik först.

Lösning:

- (a) Reguljärt. Det finns ändligt många olika möjligheter för de första 10 bokstäverna, och för varje fix sådan möjlighet s är språket “2 eller fler upprepningar av s ” reguljärt, så vi kan skriva ett gigantiskt reguljärt uttryck i stil med “ $s_1s_1+|s_2s_2+|s_3s_3+|\dots$ ” där $s_1, s_2, \dots$ är en uppräkningslista av alla möjligheter för de 10 första bokstäverna.
- (b) Inte reguljärt. En automat för språket skulle t.ex. behöva kunna räkna godtyckligt många a :n för att se att de inte är för många, och automater har bara begränsat minne.
- (c) Reguljärt. För varje sträng i ordlistan finns bara ändligt många strängar inom redigeringsavstånd 3, så för en ändligt stor ordlista är L ändligt, och alla ändliga språk är reguljära.

Rättning:

- (a) 1p för korrekt klassificering (reguljärt/ej reguljärt).
2p för korrekt motivering.
- (b) 1p för korrekt klassificering (reguljärt/ej reguljärt).
2p för korrekt motivering. För full poäng på motiveringen krävs att man är tydlig med att *obegränsat* minne krävs. En motivering där detta inte framgår, t.ex. “automater har inget minne”, ger 1 poängs avdrag.
- (c) 1p för korrekt klassificering (reguljärt/ej reguljärt).
2p för korrekt motivering.

Vanliga **felaktiga** motiveringar (som **inte ger poängen** för motiveringen, bara för ja/nejsvaret):

- “Ja för det går att beskriva med ett reguljärt uttryck”, “Nej för det kan inte kännas igen av en automat”.

Lösning:

- (a) Möjlig lösning i BNF-notation:

```

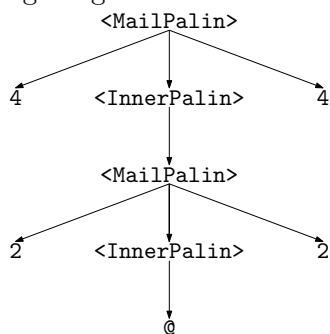
<MailPalin> ::= "a" <InnerPalin> "a" | "b" <InnerPalin> "b" | ...
<InnerPalin> ::= "@" | <MailPalin>

```

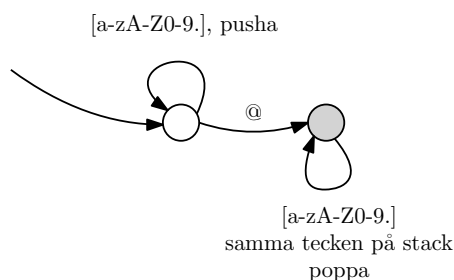
“...” ovan indikerar att det för varje tillåtet tecken (små bokstäver, stora bokstäver, siffror, punkter) finns motsvarande regel som visas för “a” och “b”.

Icke-slutsymbolen `InnerPalin` representerar ett palindrom med @ i mitten men där lokal del och domändel tillåts vara tomma.

- (b) Lösning för grammatiken ovan:



- (c) Ja, språket kan kännas igen av en stackautomat, så här:



Vänstra loopen är egentligen en pil för varje tecken i alfabetet (utom @), där tecknet läses och pushas på stacken. På samma sätt är högra loopen en pil för varje tecken i alfabetet där tecknet som lästes måste matcha det som ligger överst på stacken, och det som låg överst på stacken poppas.

Alternativ motivering:

Grammatiken i svaret på (a) kan parsas med rekursiv medåkning (detaljer skippas här i lösningsförslaget, ska beskrivas för full poäng). Rekursiv medåkning är mindre kraftfullt än stackautomater, så om språket kan parsas med rekursiv medåkning kan det också kännas igen av en stackautomat.

Rättning:

- (a)
- 3p för korrekt lösning
 - 0p om lösningen inte tvingar första och sista tecken att vara likadana, t.ex.

```

<MailPalin> ::= <Symbol> <MailPalin> <Symbol>
<Symbol> ::= "a" | "b" | "c" | ...

```
 - avdrag -1p om lösningen tillåter strängen “@” (e-post med tom lokal/domän-del), dvs om lösningen bara innehåller det som kallas `<InnerPalin>` ovan.
 - avdrag -1p för eventuella andra mindre slarvfel
- (b) Tre poäng om korrekt. 1-2 poängs avdrag för slarvfel. Poäng kan fås även om grammatiken från (a) inte är ett korrekt svar på (a), men det måste vara en korrekt definierad grammatik.

- (c) Ett poäng för korrekt klassificering (ja/nej).
Två poäng för motiveringen.