

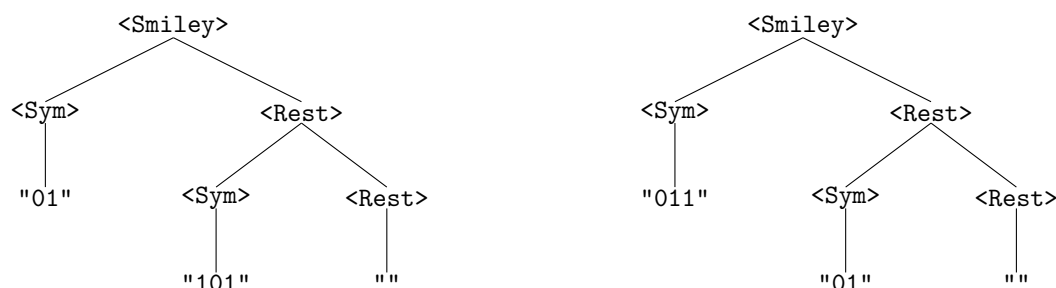
Lösningsförslag för KS i Programmeringsparadigm 2017, del 3: Formella Språk och Syntaxanalys
2017-11-24 08.15-09.45

Formella språk och syntaxanalys

1

(14p)

- (a) Binärt kodade smileys beskrivs t.ex. av det reguljära uttrycket $(01|101|011)^+$ och är således ett reguljärt språk.
- (b) En möjlig sträng är: 01101
Denna har följande två syntaxträd enligt grammatiken.



- (c) Vännen blandar ihop koncepten “språk” och “grammatik” som är två olika saker. Ett språk kan ha flera olika grammatiker. Även om en specifik grammatik är tvetydig kan det fortfarande finnas andra grammatiker för språket som inte är tvetydiga. Det faktum att språket är reguljärt implicerar att det finns någon grammatik för språket som kan parsas med rekursiv medåkning.

(d) **Möjlig lösning 1:**

Låt L_1 vara språket med balanserade smileys och L_2 vara språket med alla strängar över vänster- och höger-parenteser. L_2 är reguljärt, eftersom det kan beskrivas av det reguljära uttrycket $[(\)]^*$ (dvs, 0 eller flera förekomster av vänster- och högerparenteser). Så om L_1 är reguljärt (vilket vi försöker bevisa att det inte är) så är också $L_1 \cap L_2$ reguljärt. Men $L_1 \cap L_2$ är språket med alla (icke-tomma) balanserade parentesuttryck, och vi vet att detta inte är reguljärt. Alltså är inte heller L_1 reguljärt.

Vanliga fel längs detta spår inkluderade olika kombinationer av följande:

- Definiera L_2 som alla strängar som består av bara kolon, och påstå att $L_1 - L_2$ är balanserade parentesuttryck. Detta är fel.
- Låt L_1 vara balanserade parentesuttryck och definiera något språk L_2 sådant att balanserade smileys är $L_1 \cup L_2$, påstå att unionen av ett reguljärt och ett icke-reguljärt språk är icke-reguljärt. Detta är jättefel.

Möjlig lösning 2:

Antag för motsägelse att balanserade smileys är reguljärt. Då finns en DFA för språket. Tag denna DFA, och för varje övergång med ett “:”, ändra denna övergång så att den går till fail-tillståndet istället till vart den nu kan ha gått. Denna nya modifierade DFA kommer nu vara en DFA för balanserade parentesuttryck, men vi vet att en sådan DFA inte existerar, så vi för den önskade motsägelsen och vårt ursprungliga antagande att balanserade smileys är ett reguljärt språk måste ha varit felaktigt.

Vanliga fel längs detta spår inkluderade:

- i. Gå åt andra hållet: börja med en hypotetisk DFA för balanserade parentesuttryck och förklara hur den kan modifieras till en DFA för balanserade smileys. Detta är väsentligen samma logiska groda som fel 2 för lösning 1.

(e) Möjlig lösning:

```
<Smiley> ::= ":" <SmileyOrEmpty> | "(" <SmileyOrEmpty> ")" <SmileyOrEmpty>  
<SmileyOrEmpty> ::= <Smiley> | ""
```

Denna lösning tar hänsyn till villkoret att smileys ska vara icke-tomma. Inga avdrag gjordes på (d) och (e) för lösningar som inte tog hänsyn till detta.

2

(6p)

- (a) I detta lösningsförslag beskriver vi stacken som en sträng, där första tecknet i strängen är nedersta tecknet i stacken, och sista tecknet i strängen är översta tecknet i stacken.

Start: tillstånd **a**

1. Läs 0, pusha z, gå till **b**. Stacken: **z**
2. Läs 0, pusha z, gå till **c**. Stacken: **zz**
3. Läs 1, poppa z, pusha c, gå till **e**. Stacken: **zc**
4. Läs 1, poppa c, gå till **d**. Stacken: **z**
5. Läs 0, pusha z, gå till **a**. Stacken: **zz**
6. Läs 1, poppa z, pusha a, gå till **e**. Stacken: **za**
7. Läs 1, poppa a, gå till **b**. Stacken: **z**

Automaten slutar i **b** som inte är ett accepterande tillstånd, så accepterar ej strängen.

- (b) Ingen av de två strängarna accepteras av automaten.