Plan

- Hvad er Regularitet og Automater
- Praktiske oplysninger om kurset
- Regulære udtryk
- **Induktionsbevis**
- Frokost
- Endelige automater
- Skelnelighed, Produktkonstruktion
- Præsentation af Java projekt

Reverse-operatoren

- Givet en streng $x \in \Sigma^*$, definer *reverse*(x) i strukturen af x:
  - *reverse*(Λ) = Λ
  - *reverse*(ya) = a(*reverse*(y))
    hvor $y \in \Sigma^*$, $a \in \Sigma$

- Eksempel: *reverse*(123) = ... = 321

Reverse på sprog

- Givet et sprog $L \subseteq \Sigma^*$, define
  $$Reverse(L) = \{\ reverse(x) \mid x \in L\ \}$$

- Eksempel:
  Hvis $L=\{\Lambda,123,abc\}$ så er
  $Reverse(L)=\{\Lambda,321,cba\}$

Rekursive definitioner og induktionsbeviser

- **Rekursive definitioner** giver ofte anledning til **induktionsbeviser**
- Hvis vi skal bevise noget på form *"for alle X gælder P(X)"*, hvor mængden af *X'*er er defineret rekursivt, så kan vi prøve bevisteknikken *"induktion i strukturen af X"*

Et induktionsbevis

- Påstand: Hvis S er et regulært sprog, så er $Reverse(S)$ også regulært
  (dvs. de regulære sprog er lukkede under *Reverse*)
- Bevis:
  - S er regulært, så der eksisterer et regulært udtryk r så $L(r)=S$
  - Vi vil vise ved **induktion** i strukturen af r, at der eksisterer et regulært udtryk r' hvor $L(r')=Reverse(L(r))$, hvilket medfører, at $Reverse(S)$ er regulært

Basis

- $\underline{r = \emptyset:}$

  vælg $r' = \emptyset$

  $L(r') = \emptyset = Reverse(\emptyset) = Reverse(L(r))$

- $\underline{r = \Lambda:}$

  vælg $r' = \Lambda$ ...

- $\underline{r = a}$ hvor $a \in \Sigma$:

  vælg $r' = a$ ...

Induktionsskridt

For alle deludtryk *s* af *r* kan vi udnytte **induktionshypotesen**: der eksisterer et regulært udtryk *s'* hvor $L(s')=Reverse(L(s))$

- $\underline{r = r_1+r_2}$ hvor $r_1,r_2 \in R$:
  vælg $r' = r_1'+r_2'$ hvor r_1' og r_2' er givet af i.h. ...
- $\underline{r = r_1 r_2}$ hvor $r_1,r_2 \in R$:
  vælg $r' = r_2' r_1'$ ...
- $\underline{r = r_1^*}$ hvor $r_1 \in R$:
  vælg $r' = (r_1')^*$ ...

<u>Lemma 1:</u>
$\forall x,y \in \Sigma^*$: $reverse(xy) = reverse(y)\, reverse(x)$
bevis: induktion i strukturen (eller længden) af *y*

<u>Lemma 2:</u>
$\forall i \geq 0, E \subseteq \Sigma^*$:
$Reverse(E^i) = (Reverse(E))^i$
bevis: induktion i *i*

Konstruktive beviser

- Bemærk at dette induktionsbevis indeholder en **algoritme** til – givet et regulært udtryk for *S* – at konstruere et regulært udtryk for *Reverse*(*S*)

- Sådanne beviser kaldes **konstruktive**

- Husk altid både *konstruktionen* og *beviset for dens korrekthed*

Algoritme

Input: et regulært udtryk *r*

Definer en rekursiv funktion *REV* ved:

- $REV(\emptyset) = \emptyset$
- $REV(\Lambda) = \Lambda$
- $REV(a) = a$, hvor $a \in \Sigma$
- $REV(r_1+r_2) = REV(r_1) + REV(r_2)$
- $REV(r_1 r_2) = REV(r_2)\ REV(r_1)$
- $REV(r_1^*) = \left(REV(r_1)\right)^*$

Output: det regulære udtryk $REV(r)$

dRegAut.RegExp

- Java-repræsentation af regulære udtryk
- Speciel syntax:
  - # betyder Ø
  - % betyder Λ
- Alfabetet angives som en mængde af Unicode tegn

Resume

- Alfabeter, strenge, sprog
- Regulære udtryk og regulære sprog
- Rekursive definitioner, induktionsbeviser, konstruktive beviser
- Java: `dRegAut.RegExp` klassen

Øvelse

- Lad r være det regulære udtryk $((a+\Lambda)cbc)^*$ over alfabetet {a,b,c}. Bevis at enhver streng i sproget sproget L(r) har et lige antal c'er. Argumentér kort og præcist for hvert trin i beviset.

- Hint: Brug definitionen af sprog for regulære udtryk (Definition 3.1 i [Martin]), definitionen af '*' på sprog (s. 31 øverst i [Martin]), og lav induktion.

Plan

- Hvad er Regularitet og Automater
- Praktiske oplysninger om kurset
- Regulære udtryk
- Induktionsbevis
- **Frokost**
- Endelige automater
- Skelnelighed, Produktkonstruktion
- Præsentation af Java projekt