Enhver NFA kan oversættes til en NFA-Λ

- Med den grafiske repræsentation er det trivielt

- Med de formelle definitioner:

  Givet en NFA $M=(Q, \Sigma, q_0, A, \boldsymbol{\delta}_M)$,

  definer en NFA-Λ $N=(Q, \Sigma, q_0, A, \boldsymbol{\delta}_N)$ hvor

  $\forall\, \boldsymbol{\delta}_N(q, a) = \boldsymbol{\delta}_M(q, a)$ for alle $q \in Q$ og $a \in \Sigma$

  $\forall\, \boldsymbol{\delta}_N(q, \Lambda) = \varnothing$ for alle $q \in Q$

  Bevis for at $L(N) = L(M)$: induktion...

Enhver NFA-Λ kan oversættes til en NFA (Λ-eliminering)

Givet en NFA-Λ $M=(Q, \Sigma, q_0, A, \delta)$,

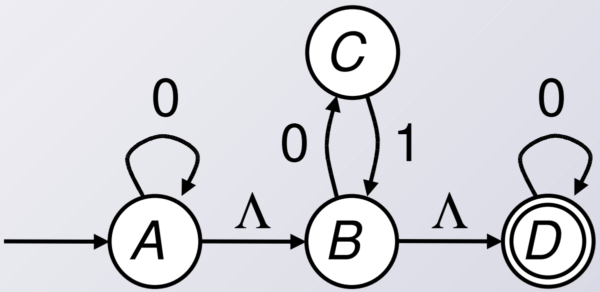
definer en NFA $M_1=(Q, \Sigma, q_0, A_1, \delta_1)$ ved

- $\delta_1(q, a) = \delta^*(q, a)$
- $A_1 = \begin{cases} A \cup \{q_0\} & \text{hvis } \Lambda(\{q_0\}) \cap A \neq \emptyset \\ A & \text{ellers} \end{cases}$

Der gælder nu: $L(M_1) = L(M)$

Eksempel

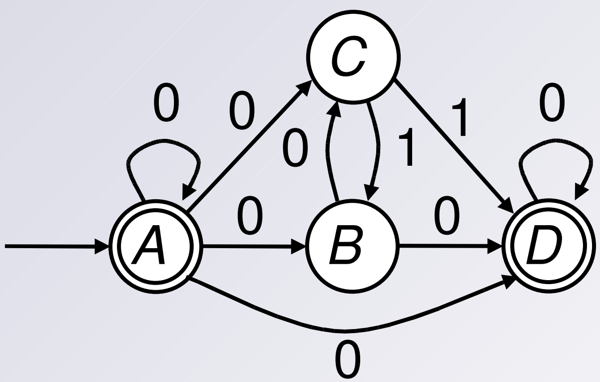
NFA-Λ



q	$\delta(q,\Lambda)$	$\delta(q,0)$	$\delta(q,1)$	$\delta^*(q,0)$	$\delta^*(q,1)$
A	$\{B\}$	$\{A\}$	$\varnothing$	$\{A,B,C,D\}$	$\varnothing$
B	$\{D\}$	$\{C\}$	$\varnothing$	$\{C,D\}$	$\varnothing$
C	$\varnothing$	$\varnothing$	$\{B\}$	$\varnothing$	$\{B,D\}$
D	$\varnothing$	$\{D\}$	$\varnothing$	$\{D\}$	$\varnothing$

- Find $\delta^*(q, a)$ for alle $q \in Q$ og $a \in \Sigma$
- Se om $\Lambda(\{q_0\}) \cap A \neq \varnothing$

NFA



Bevis for korrekthed af Λ-eliminering

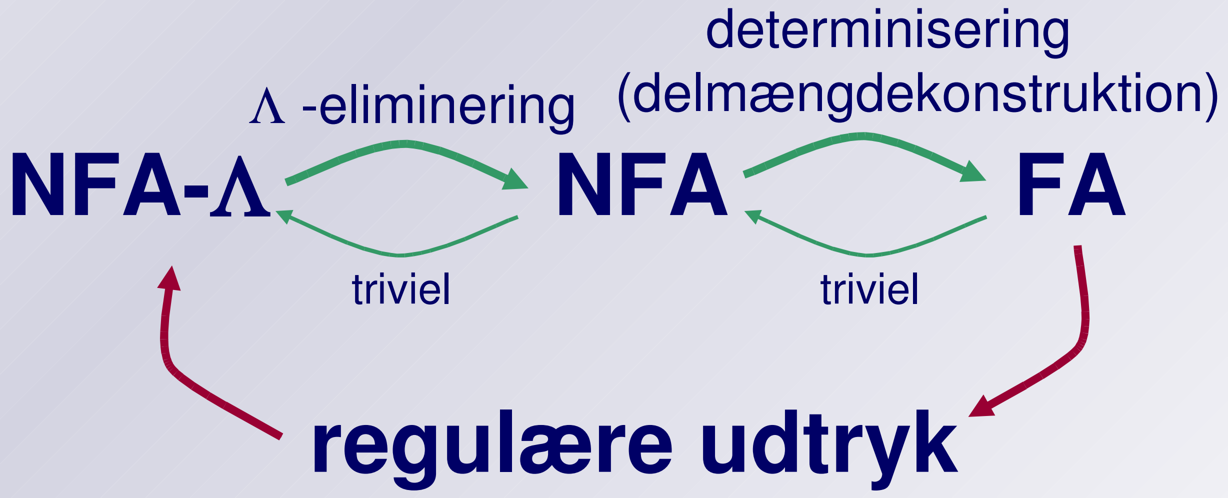
Vi skal vise: $\forall x \in \Sigma^*$: $x \in L(M_1) \Leftrightarrow x \in L(M)$

- $x = \Lambda$:
  - brug definition af A_1 og Λ-lukning...
- $x \neq \Lambda$:
  - Lemma: $\forall x \in \Sigma^*$, $x \neq \Lambda$: $\delta^*(q_0, x) = \delta_1^*(q_0, x)$
  - ...

– se bogen

Status

- Vi har defineret 4 formalismer
  - regulære udtryk
  - FA
  - NFA
  - NFA-Λ
- og er ved konstruktivt at bevise ækvivalens i udtrykskraft



Ethvert regulært udtryk kan oversættes til en NFA-Λ

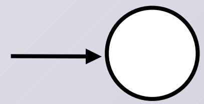
(Kleenes sætning, del 1)

Bevis:

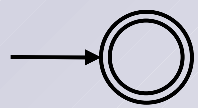
Induktion i strukturen af det regulære udtryk *r*

Basis

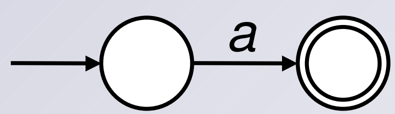
- $r = \emptyset$



- $r = \Lambda$



- $r = a$ hvor $a \in \Sigma$

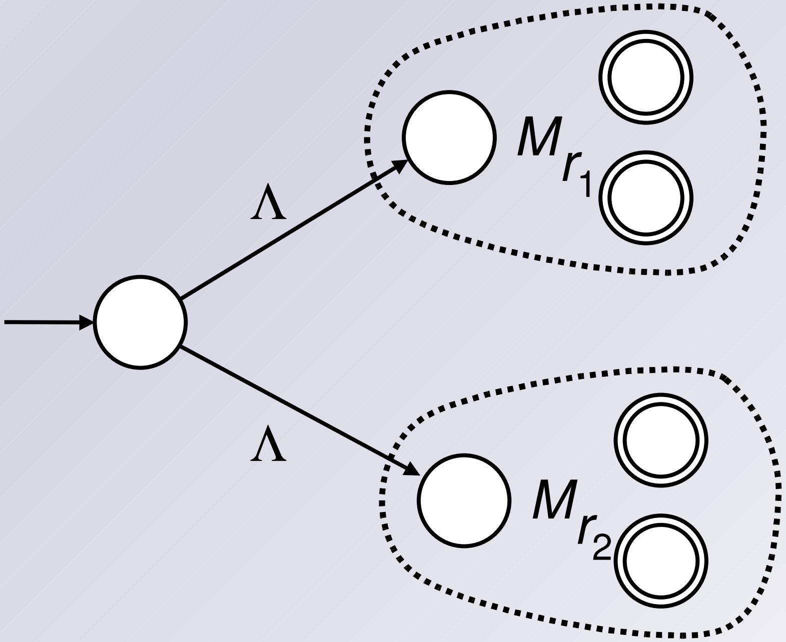


Induktionsskridt

For alle deludtryk *s* af *r* kan vi udnytte **induktionshypotesen**:

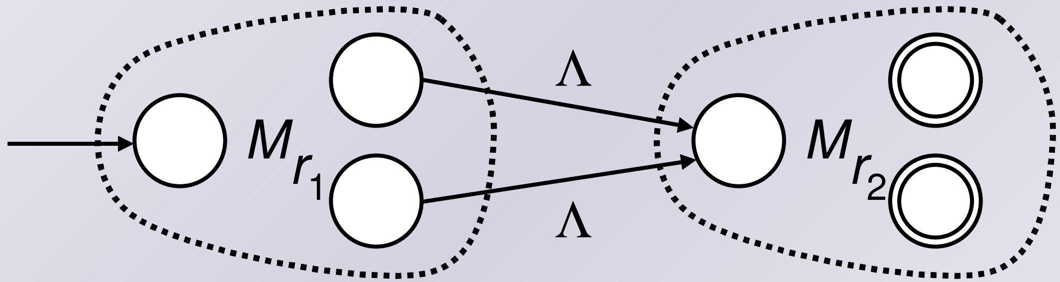
der eksisterer en NFA-Λ M_s hvor $L(M_s)=L(s)$

- $r = r_1 + r_2$

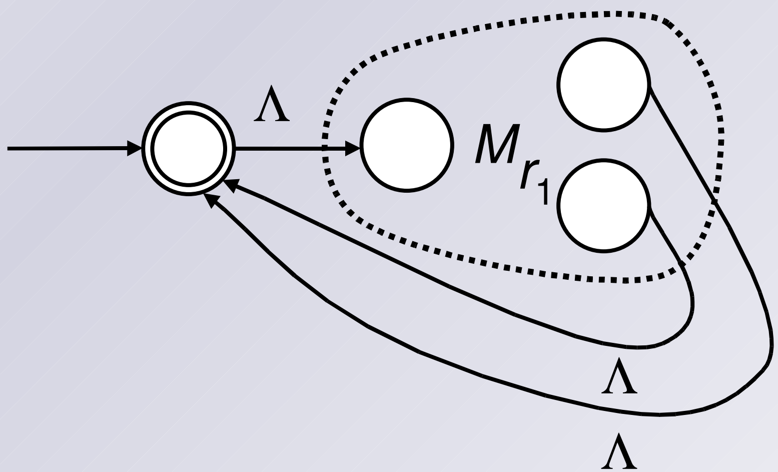


Induktionsskridt

- $r = r_1 r_2$



- $r = r_1^*$

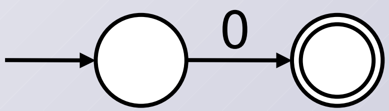


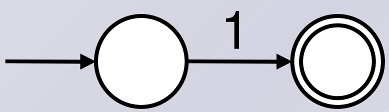
Formel beskrivelse og bevis for korrekthed

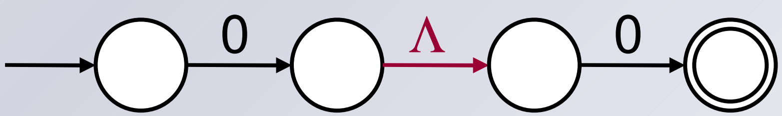
- Se bogen...

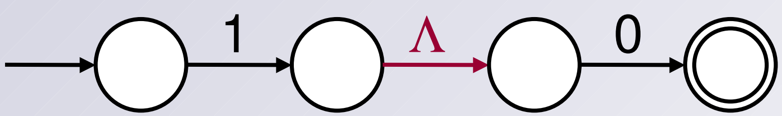
Eksempel

Konstruer en NFA-Λ for (00 + 1)*(10)*

- 0: 

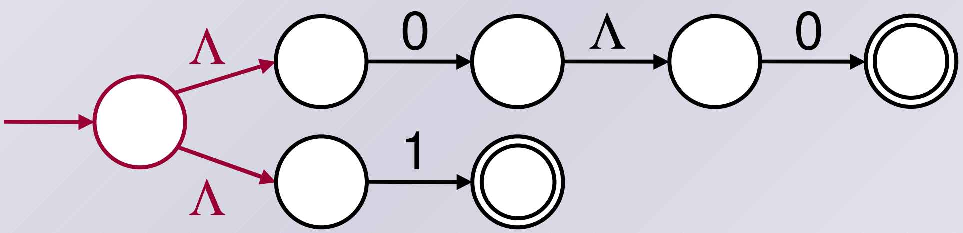
- 1: 

- 00: 

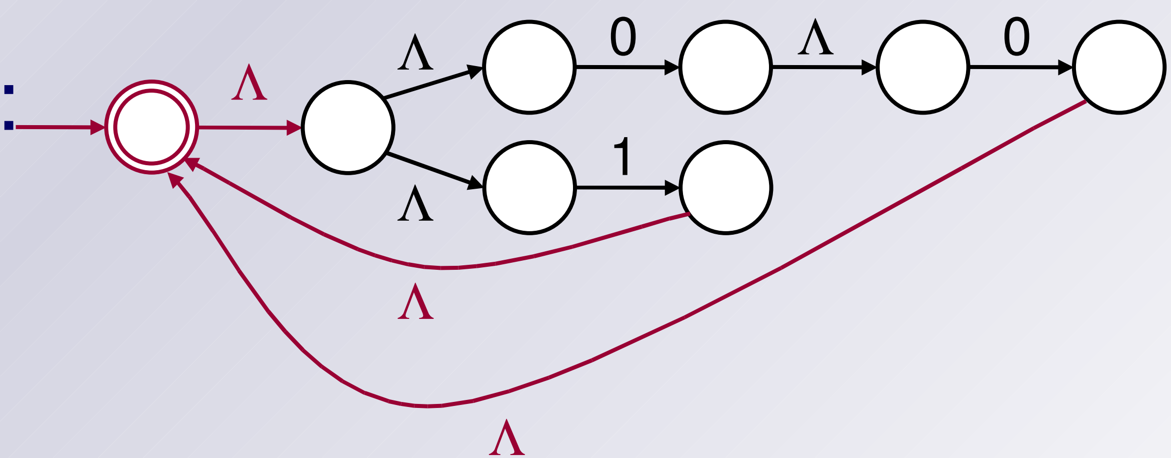
- 10: 

Eksempel, fortsat

- 00 + 1:

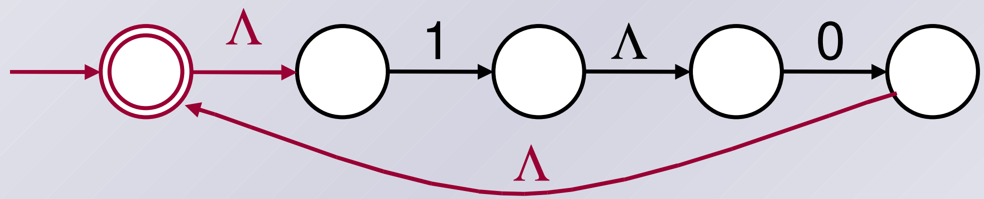


- (00 + 1)*:

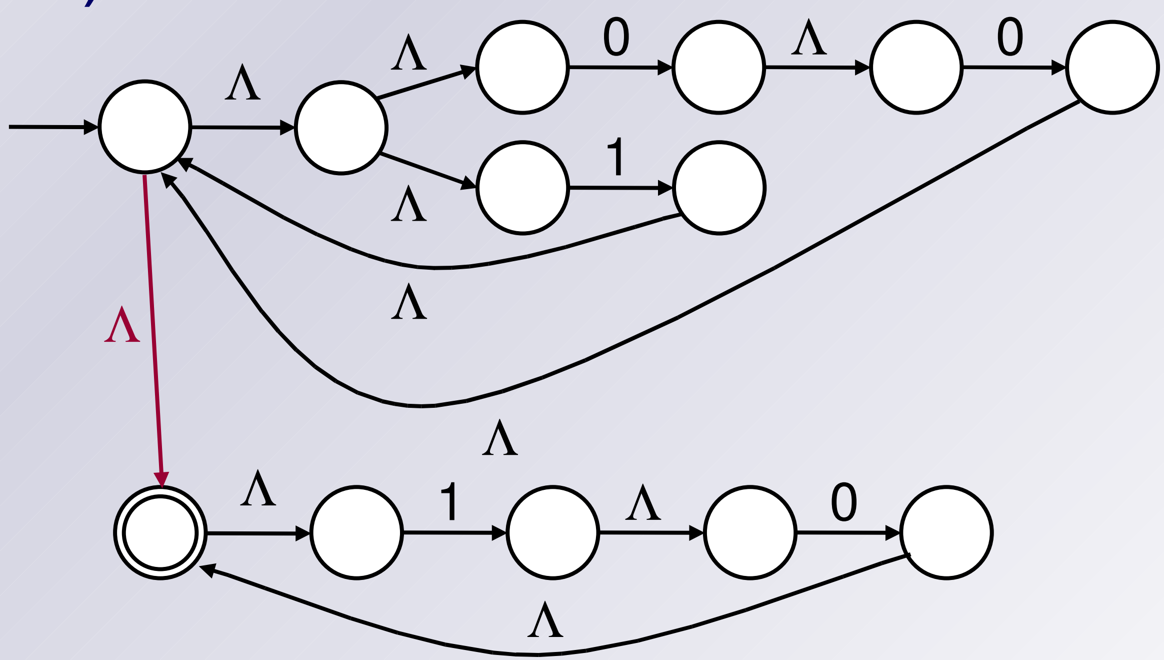


Eksempel, fortsat

- (10)*:



- (00 + 1)*(10)*:



Enhver FA kan oversættes til et regulært udtryk

(Kleenes sætning, del 2)

Bevis:

Induktion, naturligvis – men i hvad??

Fra FA til regulært udtryk

- For en FA $M=(Q, \Sigma, q_0, A, \boldsymbol{\delta})$ er $L(M)$ defineret som
  $$L(M) = \{ x \in \Sigma^* \mid \delta^*(q_0, x) \in A \}$$

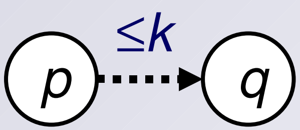
- Da A er endelig kan $L(M)$ udtrykkes som en endelig forening af sprog på form
  $$L(p, q) = \{ x \in \Sigma^* \mid \delta^*(p, x) = q \}$$

- Vi vil vise at hvert af disse sprog kan oversættes til et regulært udtryk, $r(p, q)$, og derefter kombinere disse med "+"

Induktion i tilstandsnumre

- Antag tilstandene i M er nummereret 1, ..., $|Q|$

- Definer $L(p, q, k)$ hvor $p,q\in Q$ og $k\in\mathbb{N}$
som mængden af strenge, der fører fra p til q og kun går gennem tilstande med nummer $\leq k$ (fraregnet endepunkterne)



- dvs. $L(p, q) = L(p, q, |Q|)$

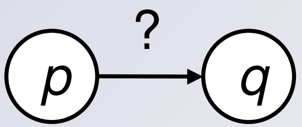
- Vi vil vise ved induktion i k at $L(p, q, k)$ svarer til et regulært udtryk, $r(p, q, k)$
- dvs. vælg $r(p, q) = r(p, q, |Q|)$

Basis

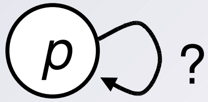
$\underline{k = 0}$

- $L(p, q, 0)$ er mængden af strenge, der fører fra p til q uden at gå gennem nogen tilstande (fraregnet endepunkterne)

- hvis $p \neq q$:
  $L(p, q, 0) = \{ a \in \Sigma \mid \delta(p, a) = q \}$



- hvis $p = q$:
  $L(p, q, 0) = \{ a \in \Sigma \mid \delta(p, a) = p \} \cup \{\Lambda\}$



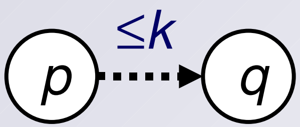
- dvs. vi kan altid finde et regulært udtryk $r(p, q, 0)$ for $L(p, q, 0)$

Induktionsskridt

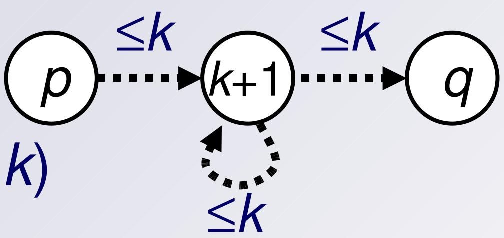
$\underline{k + 1}$

- $L(p, q, k + 1)$ er mængden af strenge, der fører fra p til q og kun går gennem tilstande med nummer $\leq k + 1$

- To tilfælde:



  - strenge der **ikke** går gennem tilstand $k + 1$:
    $L(p, q, k)$

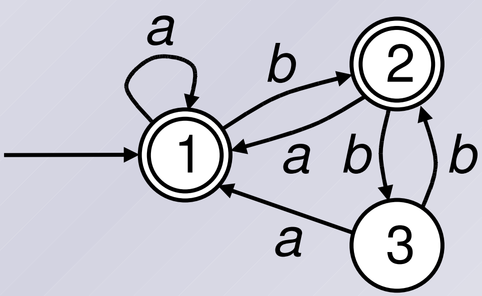


  - strenge der **går** gennem tilstand $k + 1$:
    $L(p, k + 1, k)\ L(k + 1, k + 1, k)^*\ L(k + 1, q, k)$

- dvs. $L(p, q, k + 1) = L(p, q, k) \cup$
  $L(p, k + 1, k)\ L(k + 1, k + 1, k)^*\ L(k + 1, q, k)$
  som vha. induktionshypotesen svarer til et regulært udtryk
  $r(p, q, k + 1) = r(p, q, k) +$
  $r(p, k + 1, k)\ r(k + 1, k + 1, k)^*\ r(k + 1, q, k)$

Eksempel

Oversæt denne FA til et regulært udtryk:



- $r = r(1,1,3) + r(1,2,3)$
- $r(1,1,3) = r(1,1,2) + r(1,3,2)r(3,3,2)^*r(3,1,2)$
- $r(1,1,2) = r(1,1,1) + r(1,2,1)r(2,2,1)^*r(2,1,1)$
- $r(1,1,1) = r(1,1,0) + r(1,1,0)r(1,1,0)^*r(1,1,0)$
- $r(1,1,0) = a + \Lambda$
- ...

– vi kan heldigvis skrive et Java-program, der laver oversættelsen for os ☺

Eksempel, fortsat

r = ((Ø+(((((Ø+a)+Λ)+(Ø(((Ø+Λ)*)(Ø+a))))+((((Ø+a)+Λ)+(Ø(((Ø+Λ)
)(Ø+a))))(((((Ø+a)+Λ)+(Ø(((Ø+Λ))(Ø+a))))*)(((Ø+a)+Λ)+(Ø(((Ø+Λ)
)(Ø+a)))))))+((((Ø+b)+(Ø(((Ø+Λ))(Ø+b))))+((((Ø+a)+Λ)+(Ø(((Ø+Λ)
)(Ø+a))))(((((Ø+a)+Λ)+(Ø(((Ø+Λ))(Ø+a))))*)((Ø+b)+(Ø(((Ø+Λ)
)(Ø+b)))))))(((((Ø+Λ)+((Ø+b)(((Ø+Λ))(Ø+b))))+(((Ø+a)+((Ø+b)(((Ø+
Λ)*)(Ø+a))))(((((Ø+a)+Λ)+(Ø(((Ø+Λ)*)(Ø+a))))*)((Ø+b)+(Ø(((Ø+Λ)
)(Ø+b))))))))(((Ø+a)+((Ø+b)(((Ø+Λ)*)(Ø+a))))+(((Ø+a)+((Ø+b)(((Ø+
Λ)*)(Ø+a))))(((((Ø+a)+Λ)+(Ø(((Ø+Λ)*)(Ø+a))))*)(((Ø+a)+Λ)+(Ø(((Ø+Λ)
)(Ø+a))))))))))+((((Ø+b)+(Ø(((Ø+Λ))(Ø+b))))+((((Ø+a)+Λ)+(Ø(((Ø+
Λ)*)(Ø+a))))(((((Ø+a)+Λ)+(Ø(((Ø+Λ)*)(Ø+a))))*)((Ø+b)+(Ø(((Ø+Λ)
)(Ø+b)))))))+((((Ø+b)+(Ø(((Ø+Λ))(Ø+b))))+((((Ø+a)+Λ)+(Ø(((Ø+Λ)
)(Ø+a))))(((((Ø+a)+Λ)+(Ø(((Ø+Λ))(Ø+a))))*)((Ø+b)+(Ø(((Ø+Λ)
)(Ø+b)))))))(((((Ø+Λ)+((Ø+b)(((Ø+Λ))(Ø+b))))+(((Ø+a)+((Ø+b)(((Ø+
Λ)*)(Ø+a))))(((((Ø+a)+Λ)+(Ø(((Ø+Λ)*)(Ø+a))))*)((Ø+b)+(Ø(((Ø+Λ)
)(Ø+b))))))))(((Ø+Λ)+((Ø+b)(((Ø+Λ)*)(Ø+b))))+(((Ø+a)+((Ø+b)(((Ø+
Λ)*)(Ø+a))))(((((Ø+a)+Λ)+(Ø(((Ø+Λ)*)(Ø+a))))*)((Ø+b)+(Ø(((Ø+Λ)
*)(Ø+b))))))))))))

(hvis programmet ikke simplificerer undervejs...)

dRegAut Java-pakken

Udleverede programdele:

- `NFA.java` og `NFALambda.java`:
  repræsentation af NFA'er og NFA-Λ'er
- `RegExp.java`:
  repræsentation af regulære udtryk

- parser for regulære udtryk
- de trivielle oversættelser: FA $\rightarrow$ NFA, NFA $\rightarrow$ NFA-Λ

`NFA.java`

- Repræsentation som `FA.java`, med én undtagelse:

  `transitions` er en funktion fra `StateSymbolPair` til en **mængde** af `State` objekter

`NFALambda.java`

- Repræsentation som `NFA.java`, med én undtagelse:

  Λ repræsenteres som `\uFFFF` (= `NFALambda.LAMBDA`)

`RegExp.java`

- `RegExp(String, Alphabet)`
  – parser et regulært udtryk

- `toString()`
  – til udskrift af et parsed regulært udtryk

- `toNFALambda()`
  – konstruktionen fra Kleene's sætning del 1

- `simplify()`
  – simplificerer et parsed regulært udtryk,
  nyttig efter `FA.toRegExp()` (Kleene's sætning del 2)

Resume

- Regulære udtryk, FA'er, NFA'er og NFA-Λ'er svarer alle til klassen af regulære sprog

- Algoritmer fra de konstruktive beviser:
  - determinisering (delmængdekonstruktionen)
  - $\forall\,\Lambda$-eliminering
  - regulært udtryk $\rightarrow$ NFA-Λ
  - FA $\rightarrow$ regulære udtryk (primært et teoretisk resultat)