Logikk og beregninger

En repetisjon

hrj – høst 2009

Beregning

Data

- Syntaktiske objekter – endelige
  - Mengde $\{ \quad \}$ $\cap \cup \subseteq \in \notin \varnothing$
  - Multimengde $[\quad]$
  - Liste $< \quad >$
  - Symbol
  - String = Liste av symboler

- Vi kan alltid finne ut om to syntaktiske objekter er like eller ulike
- Produkt $A \times B$
- Funksjon $A \rightarrow B$

Telling

- $|A| = $ antallet elementer i A
- $|A \times B| = |A| \cdot |B|$
- $|A \rightarrow B| = |B|^{|A|}$
- $|BOOL| = 2$
- Antall delmengder av $A = |A \rightarrow BOOL| = 2^{|A|}$

- Antall partikler i universet omtrent som 2^{256}

Binære relasjoner

- Refleksiv : $\forall x\ (Rxx)$
- Irrefleksiv : $\forall x\ \neg(Rxx)$
- Symmetrisk : $\forall x \forall y\ (Rxy \rightarrow Ryx)$
- Antisymmetrisk : $\forall x \forall y\ (Rxy \wedge Ryx \rightarrow x=y)$
- Transitiv : $\forall x \forall y \forall z\ (Rxy \wedge Ryz \rightarrow Rxz)$
- Total : $\forall x \forall y\ (Rxy \vee x=y \vee Ryx)$
- Ekvivalensrelasjon $\sim$: Refleksiv, symmetrisk, transitiv – gir en partisjon av universet
- Partiell ordning $\leq$: Refleksiv, antisymmetrisk, transitiv
- Total ordning $\leq$: Partiell ordning som er total

Utsagnslogikk – språk

- Utsagn bygd opp av
  - Atomære utsagn
  - Ved bruk av konnektivene: $\neg$ $\wedge$ $\vee$ $\rightarrow$
- Kan sees som funksjoner av type
  - BOOL
  - BOOL $\rightarrow$ BOOL
  - BOOL $\times$ BOOL $\rightarrow$ BOOL
  - BOOL $\times$ BOOL $\times$ BOOL $\rightarrow$ BOOL
  - ……

Utsagn

BOOL^n Maskin BOOL

Utsagnslogikk

- Finne ut om et utsagn er
  - Gyldig – alltid sant
  - Tilfredsstillbar – noen ganger sann
  - Falsifiserbar – noen ganger usann
  - Kontradiktorisk – alltid usann
- Gjøres ved sannhetstabeller

Sannhetstabell

A	B	$\neg$ A	A $\wedge$ B	A $\vee$ B	A $\rightarrow$ B	A $\leftarrow$ B	A $\leftrightarrow$ B
1	1	0	1	1	1	1	1
1	0	0	0	1	0	1	0
0	1	1	0	1	1	0	0
0	0	1	0	0	1	1	1

Første ordens logikk – språk

- Har univers U
- Termer – navn på elementer i U
  - Konstanter og variable
  - Funksjoner
- Utsagn – bygd opp ved
  - Atomære utsagn – relasjoner og termer
  - Konnektiver : $\neg \wedge \vee \rightarrow$
  - Kvantorer : $\forall \; \exists$
- Frie og bundne variable
  - Setning – utsagn uten frie variable

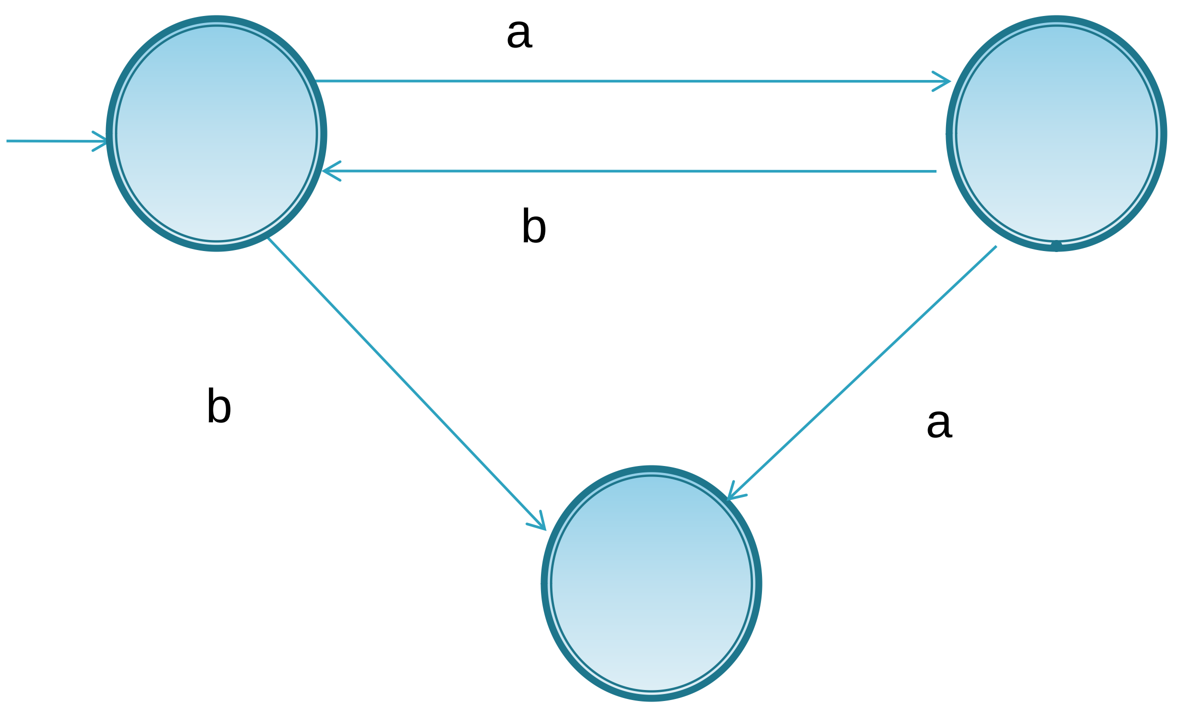
Typer

- Term : U
- Unær funksjon : $U \to U$
- Unær relasjon : $U \to BOOL$
- Binært konnektiv : $BOOL \times BOOL \to BOOL$
- Kvantor : $(U \to BOOL) \to BOOL$

- Tilsvarende med flere argumenter

Automat

DFA – deterministisk



a b a b b

Endelige tilstandsautomater

- DFA – deterministisk
- NFA – ikke deterministisk
  - Flere mulige transisjoner
  - Tomme transisjoner – bruk Λ
- REG – regulære uttrykk
  - For hvert symbol ,for Λ og for $\emptyset$
  - Serie : A B
  - Parallell : $A \vee B$
  - Tilbake : A^*

Overganger

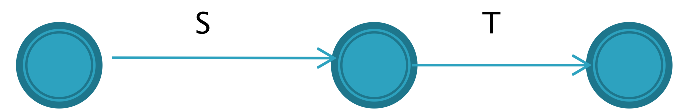
- Fra NFA til DFA
  - Bruk delmengdekonstruksjon
- Fra REG til NFA
  - Bruk transisjoner med regulære uttrykk
  - Skriv om transisjonene til vi kommer til symboler
- Fra NFA til REG
  - Start med transisjoner med symboler
  - Innfør transisjoner med regulære uttrykk

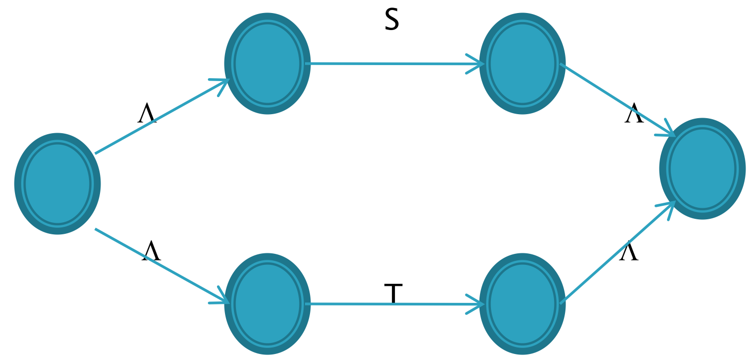
Fra NFA til DFA

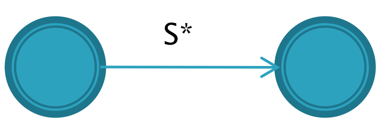
- Gitt NFA med
  - Alfabet A
  - Tilstander Q – start $s \in Q$, final $F \subseteq Q$
  - Transisjoner

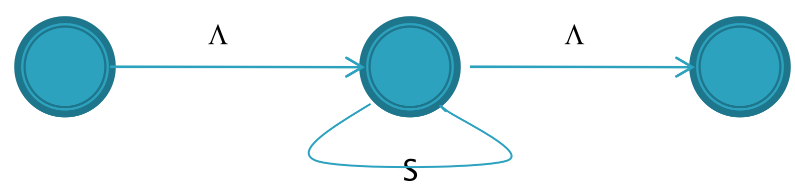
- Konstruer DFA som gjør det samme
  - Alfabet A – som over
  - Tilstander – $\wp Q$ – delmengder av Q
  - Start $\{s\}$ og tilstander fått fra s ved tom transisjon
  - Final – alle delmengder som inneholder F
  - Transisjoner – direkte konstruksjon
- Eksponensiell vekst

Fra REG til NFA



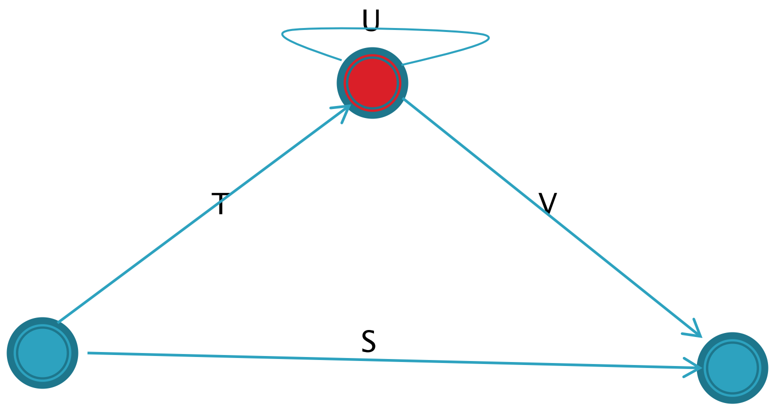




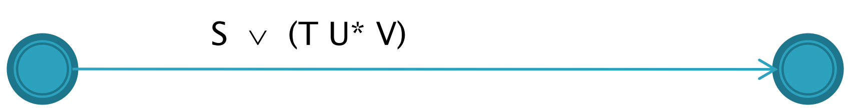


Fra NFA til REG

- Rød tilstand og alle blå par



- Erstattes med



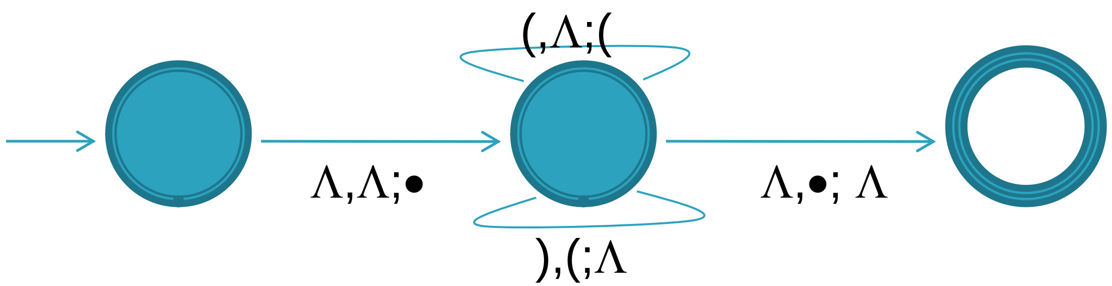
Pumpelemma

- Inputstringen gir en sti gjennom automaten
- Om inputstringen er lengre enn antall tilstander, må stien gå i en ring

- Brukes til å vise at parentesspråket er ikke regulært – kan ikke avgjøres med en endelig tilstandsautomat

Parentesspråket

- PDA



- CFG
  - S → Λ
  - S → S S
  - S → (S)

Simulering av NFA

- Et symbol – en unær funksjon
- En tilstand – en unær relasjon
- Transisjon P til Q ved lesing av symbol a
  - $\forall x\ (\ P(\alpha x) \rightarrow Q(x)\ )$
  - Konjunksjon av disse gir TRANSISJON
- Startstring t gir utsagn START(t)
- Finale tilstander gir utsagn FINAL
- Får simuleringsutsagn
  - START(t) $\wedge$ TRANSISJON $\rightarrow$ FINAL

Simuleringsutsagn

- Om t er akseptert, så kan simuleringsutsagnet bevises
  - Fra beregningen av t lages beviset i sekventkalkyle
- Om t er ikke akseptert, så kan simuleringsutsagnet falsifiseres
  - Lager modell
    - Univers – alle termer
    - Atomære utsagn – sann om den kan beregnes, usann ellers
  - Om t er ikke akseptert, så får vi falsifikasjon

Sekventer

- $\Gamma \vdash \Delta$ multimengder av utsagn

- De to tolkningene av sekventer
  - Gyldighet : Om alle Γ er sanne, så er minst en av Δ
  - Falsifikasjon : Alle Γ sanne og alle Δ usanne

- De to tolkningene av sekventkalkyle
  - Syntese : Om alle premisser gyldige, så også konklusjon
  - Analyse : En falsifikasjon av konklusjonen gir en falsifikasjon for et av premissene

Sekventkalkyle – utsagnslogikk

- Aksiom : $\Gamma, F \vdash \Delta, F$
  - Alltid gyldig, kan ikke falsifiseres
- Regler

	Antesedent	Suksedent
$\neg$	$\dfrac{\Gamma \vdash \Delta, F}{\Gamma, \neg F \vdash \Delta}$	$\dfrac{\Gamma, F \vdash \Delta}{\Gamma \vdash \Delta, \neg F}$
$\wedge$	$\dfrac{\Gamma, F, G \vdash \Delta}{\Gamma, F \wedge G \vdash \Delta}$	$\dfrac{\Gamma \vdash \Delta, F \qquad \Gamma \vdash \Delta, G}{\Gamma \vdash \Delta, F \wedge G}$
$\vee$	$\dfrac{\Gamma, F \vdash \Delta \qquad \Gamma, G \vdash \Delta}{\Gamma, F \vee G \vdash \Delta}$	$\dfrac{\Gamma \vdash \Delta, F, G}{\Gamma \vdash \Delta, F \vee G}$
$\rightarrow$	$\dfrac{\Gamma \vdash \Delta, F \qquad \Gamma, G \vdash \Delta}{\Gamma, F \rightarrow G \vdash \Delta}$	$\dfrac{\Gamma, F \vdash \Delta, G}{\Gamma \vdash \Delta, F \rightarrow G}$

Sekventkalkyle – første orden

- Regler

	Antesedent	Suksedent
$\forall$	$\dfrac{\Gamma, \forall x\, Fx, Ft \vdash \Delta}{\Gamma, \forall x\, Fx \vdash \Delta}$	$\dfrac{\Gamma \vdash \Delta, Fa}{\Gamma, F \vdash \Delta, \forall x\, Fx}$
$\exists$	$\dfrac{\Gamma, Fa \vdash \Delta}{\Gamma, \exists x\, Fx \vdash \Delta}$	$\dfrac{\Gamma \vdash \Delta, \exists x\, Fx, Ft}{\Gamma \vdash \Delta, \exists F}$

- Betingelser
  - $\forall$–antesedent, $\exists$–suksedent : t en term
  - $\forall$–suksedent, $\exists$–antesedent : a en fersk term

Sekventkalkyle – likhet

	Antesedens	Suksedens
=	$\dfrac{\Gamma, \Gamma^*, a=b \vdash \Delta, \Delta^*}{\Gamma \vdash \Delta}$	$\Gamma \vdash \Delta, a=a$

I antesedenten har vi lagt til nye sekventer fått ved å erstatte a med b eller b med a i Γ og Δ

I suksedenten har vi lagt til et nytt aksiom

Sekventkalkyle – flere sorter

- Vi må bare holde orden på hvilken sort de ulike termene har, og hvilke sorter de ulike argumentene i relasjoner og funksjoner skal ha.

- Akkurat like enkel som vanlig første ordens logikk, men kan være viktig i anvendelser.

Terminering

- Utsagnslogikk
  - Hver regel tar vekk et konnektiv
  - Analyseprosessen vil stoppe opp
- Første ordens logikk
  - Analyseprosessen vil ikke stoppe opp om
    - Analyserer $\forall$–antesedent eller $\exists$–suksedent og har funksjonssymboler
    - Analyserer $\forall$ utenfor $\exists$ i antesedent, eller $\exists$ utenfor $\forall$ i suksedent
- Passer på at analyseprosessen blir fair – alt som kan analyseres blir før eller senere analysert (om nødvendig)

Turings analyse

- Analyserer en menneskelig beregner som utfører en beregning etter bestemt forskrift
  - Eksternt regnemedium – ubegrenset tape

Λ	Λ	Λ	a	a	b	a	b	b	b	a	b	b	b	a	b	Λ	Λ	Λ	Λ

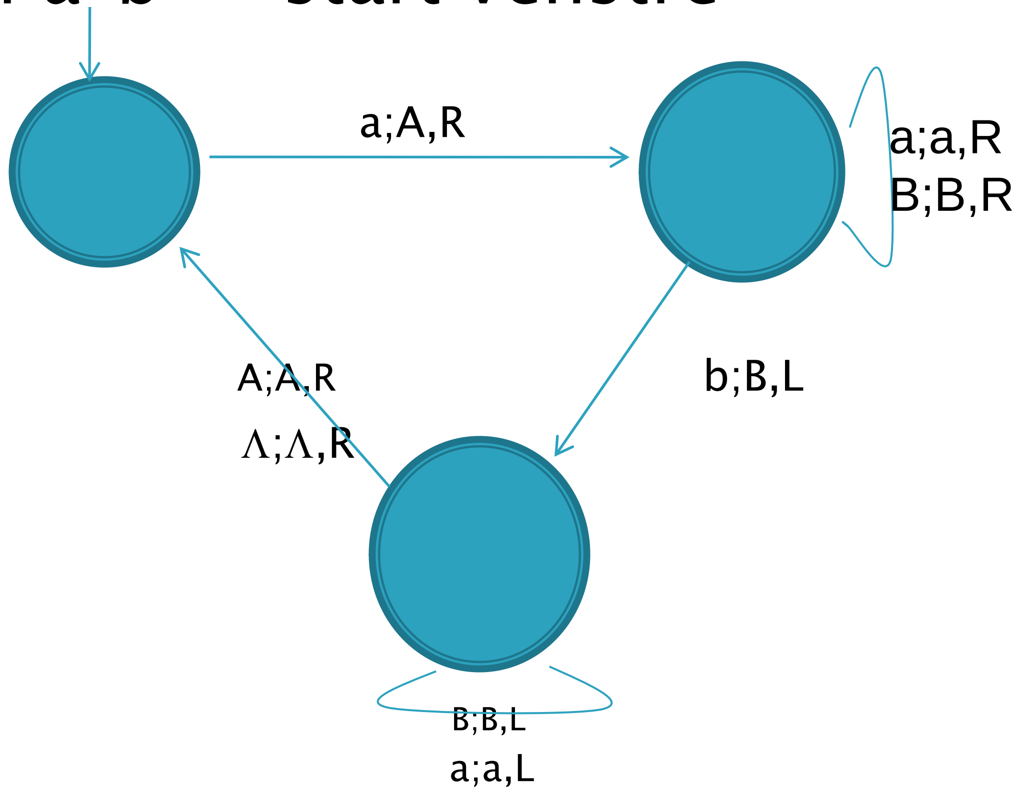
  - Endelig alfabet med symboler – inklusive Λ
  - Beregner – med endelig antall tilstander og som kan
    - Lese en rute
    - Skrive i en rute
    - Bevege seg til neste rute

Maskin 1 – endelig automat

- En endelig tilstandsautomat kan sees som en turingmaskin som bare beveger seg i en retning
- Den vil ikke kunne bruke det eksterne regnemediet som ekstra hukommelse
- Mange beregningsoppgaver kan løses med endelige tilstandsautomater

Maskin 2 – Sammenlikning

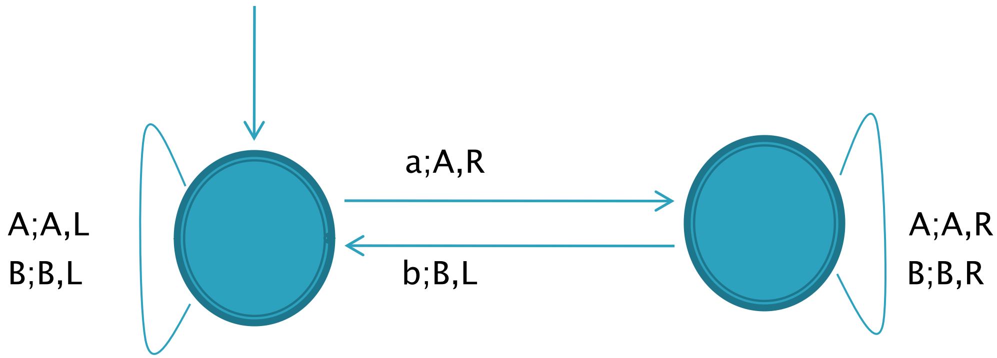
- Tape: a*b* – start venstre



- Stopp ved venstre tilstand : Antall a $\leq$ antall b
- Stopp ved høyre tilstand : Antall a $>$ antall b

Maskin 3 – invers sammenlikning

Starttape: a*b* – start høyre a



Utganger

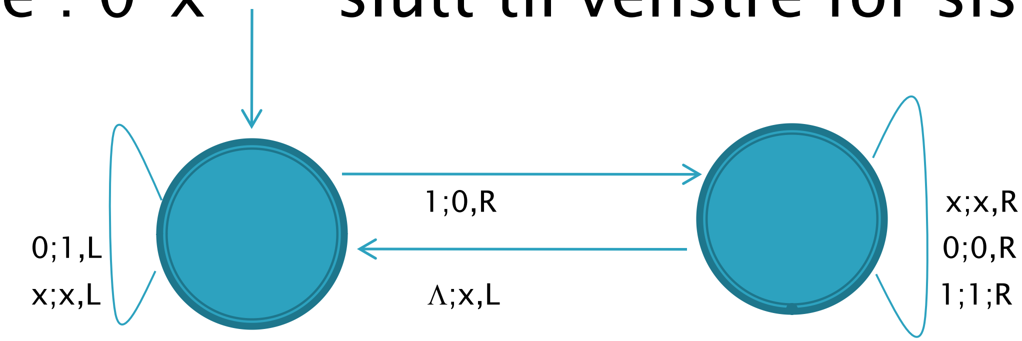
Venstre : Antall a $\leq$ antall b

Høyre : Antall a > antall b

Maskin 4 – Binær / unær

Starttape: $(0 \vee 1)^*$ – start til høyre

Slutttape : 0^*x^* – slutt til venstre for siste 0



Til venstre på tapen er et binært tall. Den trekker fra1binært og så legger til en x til høyre på tapen. Dette fortsettes med inntil det er bare 0'er i det binære tallet. Da har en konvertert det binære tallet til et unært tall med x'er.

Simulering av maskiner med logikk

- Ser på maskiner med stringer inn og BOOL ut
- Har et unært funksjonssymbol for hvert symbol i stringalfabetet
- Har relasjonssymboler for hver tilstand – like mange argumenter som stringer vi må passe på
  - DFA og NFA – unære relasjoner
  - Turingmaskin – binære relasjoner
- Simulering ved
  - START $\wedge$ TRANSISJON $\rightarrow$ FINAL

Avgjørbarhet

- En beregning

String Maskin BOOL

- er avgjørbar, om den alltid terminerer

Ikke avgjørbar

- Vi kan ikke avgjøre om et program stopper.
- Vi kan ikke avgjøre om et utsagn i første ordens logikk er gyldig

LYKKE TIL 14. DESEMBER