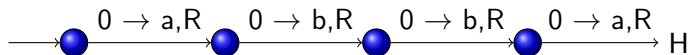


# Grunnleggende maskiner

February 12, 2013

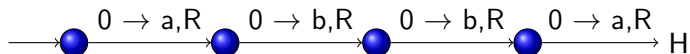
# Grunnleggende turing maskiner

## Basic 1: Skrive ord



# Grunnleggende turing maskiner

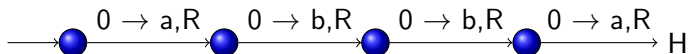
## Basic 1: Skrive ord



Alfabet:  $a, b, 0$  — 0 er blank

# Grunnleggende turing maskiner

## Basic 1: Skrive ord

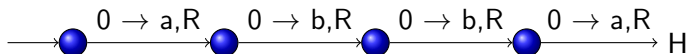


Alfabet:  $a, b, 0$  —  $0$  er blank

Spesifikasjon: Skriver *abba* og stopper

# Grunnleggende turing maskiner

## Basic 1: Skrive ord



Alfabet:  $a, b, 0$  —  $0$  er blank

Spesifikasjon: Skriver *abba* og stopper

# Grunnleggende turing maskiner

## Basic 1: Skrive ord



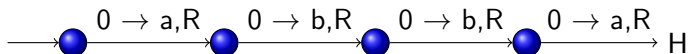
Alfabet:  $a, b, 0$  — 0 er blank

Spesifikasjon: Skriver *abba* og stopper

- ▶ Til ord av lengde  $n$  brukes  $n$  tilstander + stopp

# Grunnleggende turing maskiner

## Basic 1: Skrive ord



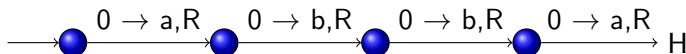
Alfabet:  $a, b, 0$  — 0 er blank

Spesifikasjon: Skriver *abba* og stopper

- ▶ Til ord av lengde  $n$  brukes  $n$  tilstander + stopp
- ▶ Et ord er random om vi ikke klarer oss med færre tilstander

# Grunnleggende turing maskiner

## Basic 1: Skrive ord



Alfabet:  $a, b, 0$  —  $0$  er blank

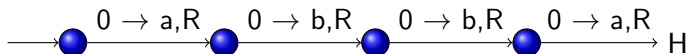
Spesifikasjon: Skriver *abba* og stopper

- ▶ Til ord av lengde  $n$  brukes  $n$  tilstander + stopp
- ▶ Et ord er random om vi ikke klarer oss med færre tilstander
- ▶ Kolmogorov kompleksitet — kompleksitet av et ord = minste antall tilstander som trengs for å produsere ordet



# Grunnleggende turing maskiner

## Basic 1: Skrive ord



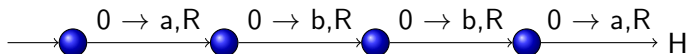
Alfabet:  $a, b, 0$  —  $0$  er blank

Spesifikasjon: Skriver *abba* og stopper

- ▶ Til ord av lengde  $n$  brukes  $n$  tilstander + stopp
- ▶ Et ord er random om vi ikke klarer oss med færre tilstander
- ▶ Kolmogorov kompleksitet — kompleksitet av et ord = minste antall tilstander som trengs for å produsere ordet
- ▶ Beveger seg bare i en retning — DFA

# Grunnleggende turing maskiner

## Basic 1: Skrive ord



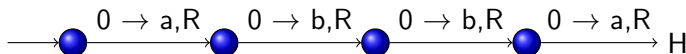
Alfabet:  $a, b, 0$  —  $0$  er blank

Spesifikasjon: Skriver *abba* og stopper

- ▶ Til ord av lengde  $n$  brukes  $n$  tilstander + stopp
- ▶ Et ord er random om vi ikke klarer oss med færre tilstander
- ▶ Kolmogorov kompleksitet — kompleksitet av et ord = minste antall tilstander som trengs for å produsere ordet
- ▶ Beveger seg bare i en retning — DFA
- ▶ Fargeleggingsmaskin

# Grunnleggende turing maskiner

## Basic 1: Skrive ord



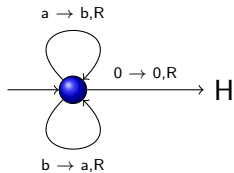
Alfabet:  $a, b, 0$  — 0 er blank

Spesifikasjon: Skriver *abba* og stopper

- ▶ Til ord av lengde  $n$  brukes  $n$  tilstander + stopp
- ▶ Et ord er random om vi ikke klarer oss med færre tilstander
- ▶ Kolmogorov kompleksitet — kompleksitet av et ord = minste antall tilstander som trengs for å produsere ordet
- ▶ Beveger seg bare i en retning — DFA
- ▶ Fargeleggingsmaskin
- ▶ Trenger ikke være blanke vi skriver over

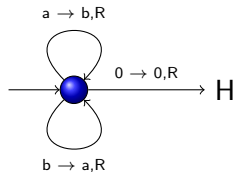
# Grunnleggende turing maskiner

## Basic 2: Bytte bokstaver



# Grunnleggende turing maskiner

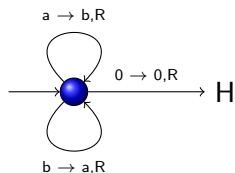
## Basic 2: Bytte bokstaver



Alfabet:  $a, b, 0$  — 0 er blank

# Grunnleggende turing maskiner

## Basic 2: Bytte bokstaver

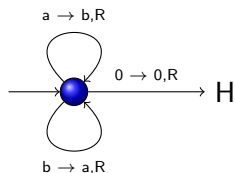


Alfabet:  $a, b, 0$  — 0 er blank

Spesifikasjon: Erstatter  $a$  med  $b$  og  $b$  med  $a$  inntil den treffer 0

# Grunnleggende turing maskiner

## Basic 2: Bytte bokstaver

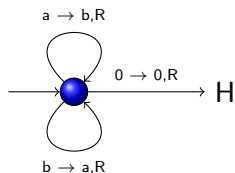


Alfabet:  $a, b, 0$  —  $0$  er blank

Spesifikasjon: Erstatte  $a$  med  $b$  og  $b$  med  $a$  inntil den treffer  $0$

# Grunnleggende turing maskiner

## Basic 2: Bytte bokstaver



**Alfabet:**  $a, b, 0$  —  $0$  er blank

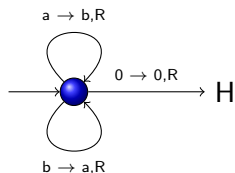
**Spesifikasjon:** Erstatter  $a$  med  $b$  og  $b$  med  $a$  inntil den treffer  $0$

- Renske opp etter beregning



# Grunnleggende turing maskiner

## Basic 2: Bytte bokstaver



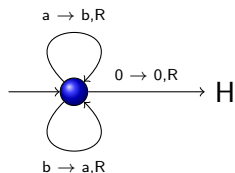
**Alfabet:**  $a, b, 0$  — 0 er blank

**Spesifikasjon:** Erstatter  $a$  med  $b$  og  $b$  med  $a$  inntil den treffer 0

- ▶ Renske opp etter beregning
- ▶ Beveger seg bare i en retning — DFA

# Grunnleggende turing maskiner

## Basic 2: Bytte bokstaver



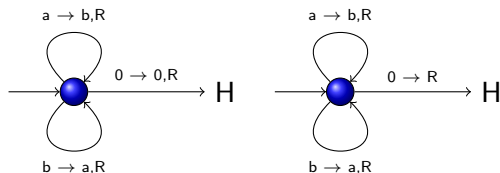
**Alfabet:**  $a, b, 0$  — 0 er blank

**Spesifikasjon:** Erstatter  $a$  med  $b$  og  $b$  med  $a$  inntil den treffer 0

- ▶ Renske opp etter beregning
- ▶ Beveger seg bare i en retning — DFA
- ▶ Fargeleggingsmaskin

# Grunnleggende turing maskiner

## Basic 2: Bytte bokstaver



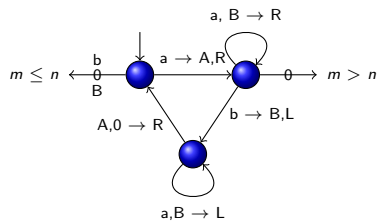
Alfabet:  $a, b, 0$  —  $0$  er blank

Spesifikasjon: Erstatter  $a$  med  $b$  og  $b$  med  $a$  inntil den treffer  $0$

- ▶ Renske opp etter beregning
- ▶ Beveger seg bare i en retning — DFA
- ▶ Fargeleggingsmaskin
- ▶ Forkortelser

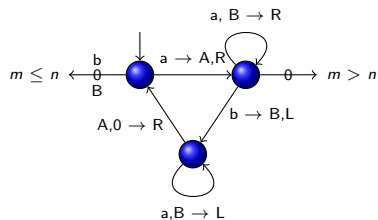
# Grunnleggende turing maskiner

## Basic 3: Sammenlikning



# Grunnleggende turing maskiner

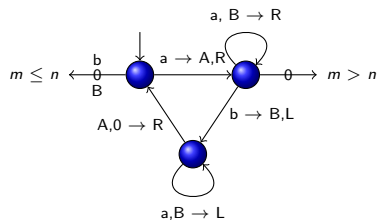
## Basic 3: Sammenlikning



Alfabet:  $a, b, A, B, 0$  —  $0$  er blank

# Grunnleggende turing maskiner

## Basic 3: Sammenlikning

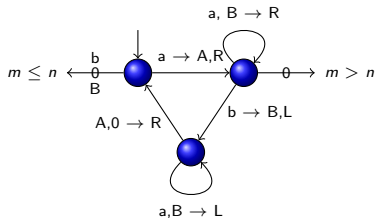


Alfabet:  $a, b, A, B, 0$  —  $0$  er blank

Starttape:  $0^* a^m b^n 0^*$  — start på venstre  $a$

# Grunnleggende turing maskiner

## Basic 3: Sammenlikning



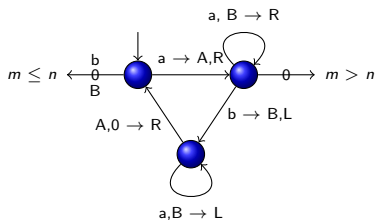
**Alfabet:**  $a, b, A, B, 0$  —  $0$  er blank

**Starttape:**  $0^* a^m b^n 0^*$  — start på venstre  $a$

**Spesifikasjon:** Ser om det er fler  $a$ 'er enn  $b$ 'er

# Grunnleggende turing maskiner

## Basic 3: Sammenlikning



**Alfabet:**  $a, b, A, B, 0$  — 0 er blank

**Starttape:**  $0^* a^m b^n 0^*$  — start på venstre  $a$

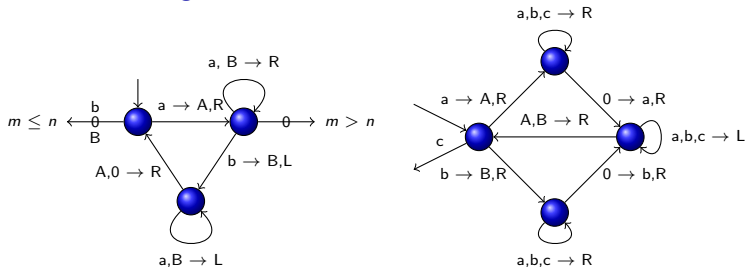
**Spesifikasjon:** Ser om det er fler  $a$ 'er enn  $b$ 'er

- Kan også sammenlikne ord i flere symboler



# Grunnleggende turing maskiner

## Basic 3: Sammenlikning



**Alfabet:**  $a, b, A, B, 0$  — 0 er blank

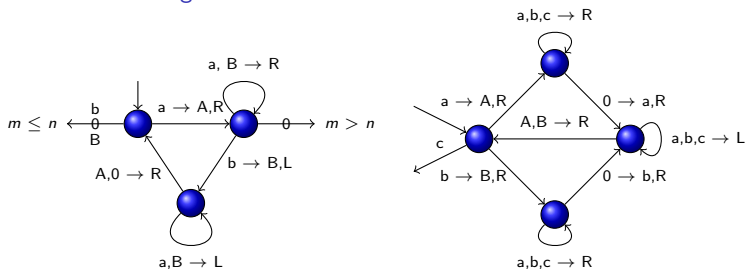
**Starttape:**  $0^* a^m b^n 0^*$  — start på venstre  $a$

**Spesifikasjon:** Ser om det er fler  $a$ 'er enn  $b$ 'er

- ▶ Kan også sammenlikne ord i flere symboler
- ▶ Kan kopiere ord —  $0^*(a \vee b)^* c 0^*$

# Grunnleggende turing maskiner

## Basic 3: Sammenlikning

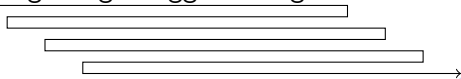


**Alfabet:**  $a, b, A, B, 0$  — 0 er blank

**Starttape:**  $0^* a^m b^n 0^*$  — start på venstre  $a$

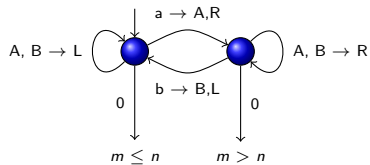
**Spesifikasjon:** Ser om det er fler  $a$ 'er enn  $b$ 'er

- ▶ Kan også sammenlikne ord i flere symboler
- ▶ Kan kopiere ord —  $0^*(a \vee b)^* c 0^*$
- ▶ Beveger seg i begge retninger



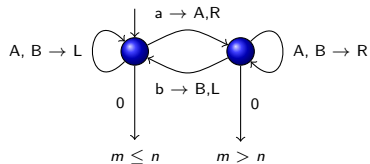
# Grunnleggende turing maskiner

## Basic 4: Omvendt sammenlikning



# Grunnleggende turing maskiner

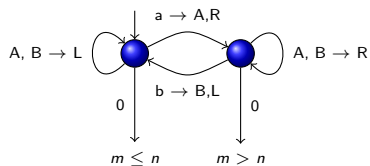
## Basic 4: Omvendt sammenlikning



Alfabet:  $a, b, A, B, 0$  —  $0$  er blank

# Grunnleggende turing maskiner

## Basic 4: Omvendt sammenlikning

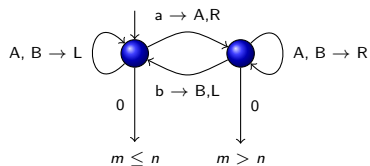


Alfabet:  $a, b, A, B, 0$  — 0 er blank

Starttape:  $0^* a^m b^n 0^*$  — start på venstre  $b$

# Grunnleggende turing maskiner

## Basic 4: Omvendt sammenlikning



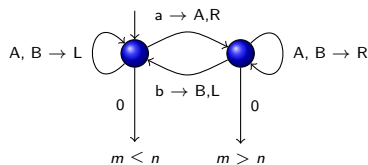
**Alfabet:**  $a, b, A, B, 0$  — 0 er blank

**Starttape:**  $0^* a^m b^n 0^*$  — start på venstre  $b$

**Spesifikasjon:** Ser om det er fler  $a$ 'er enn  $b$ 'er

# Grunnleggende turing maskiner

## Basic 4: Omvendt sammenlikning



**Alfabet:**  $a, b, A, B, 0$  — 0 er blank

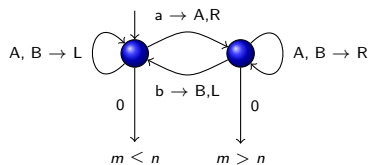
**Starttape:**  $0^* a^m b^n 0^*$  — start på venstre  $b$

**Spesifikasjon:** Ser om det er fler  $a$ 'er enn  $b$ 'er

- Kan også sammenlikne ord i flere symboler — ekstra problem

# Grunnleggende turing maskiner

## Basic 4: Omvendt sammenlikning



**Alfabet:**  $a, b, A, B, 0$  — 0 er blank

**Starttape:**  $0^* a^m b^n 0^*$  — start på venstre  $b$

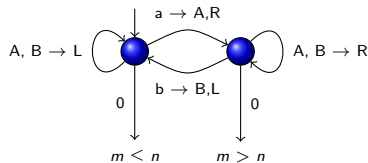
**Spesifikasjon:** Ser om det er fler  $a$ 'er enn  $b$ 'er

- ▶ Kan også sammenlikne ord i flere symboler — ekstra problem
- ▶ Kan kopiere ord



# Grunnleggende turing maskiner

## Basic 4: Omvendt sammenlikning



**Alfabet:**  $a, b, A, B, 0$  — 0 er blank

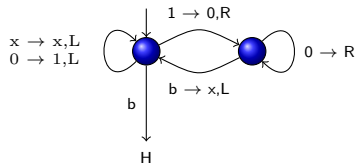
**Starttape:**  $0^* a^m b^n 0^*$  — start på venstre  $b$

**Spesifikasjon:** Ser om det er fler  $a$ 'er enn  $b$ 'er

- ▶ Kan også sammenlikne ord i flere symboler — ekstra problem
- ▶ Kan kopiere ord
- ▶ Raskere sammenlikning
- ▶ Beveger seg i begge retninger

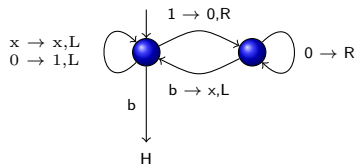
# Grunnleggende turing maskiner

## Basic 5: Konverter unær/binær



# Grunnleggende turing maskiner

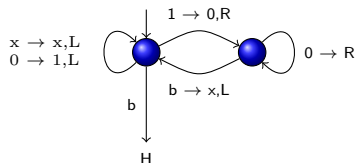
## Basic 5: Konverter unær/binær



Alfabet:  $0, 1, x, b$  —  $b$  er blank

# Grunnleggende turing maskiner

## Basic 5: Konverterer unær/binær

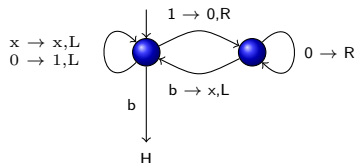


Alfabet:  $0, 1, x, b$  —  $b$  er blank

Starttape:  $b^*(0 \vee 1)^*b^*$  — start til høyre i tallet

# Grunnleggende turing maskiner

## Basic 5: Konverterer unær/binær



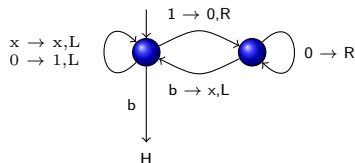
**Alfabet:**  $0, 1, x, b$  —  $b$  er blank

**Starttape:**  $b^*(0 \vee 1)^*b^*$  — start til høyre i tallet

**Spesifikasjon:** Konverterer det binære tallet  $(0 \vee 1)^*$  til det unære tallet  $x^*$

# Grunnleggende turing maskiner

## Basic 5: Konverterer unær/binær



**Alfabet:**  $0, 1, x, b$  —  $b$  er blank

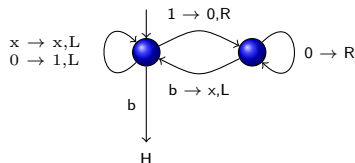
**Starttape:**  $b^*(0 \vee 1)^*b^*$  — start til høyre i tallet

**Spesifikasjon:** Konverterer det binære tallet  $(0 \vee 1)^*$  til det unære tallet  $x^*$

- ▶ Vi går i en løkke

# Grunnleggende turing maskiner

## Basic 5: Konverterer unær/binær



**Alfabet:**  $0, 1, x, b$  —  $b$  er blank

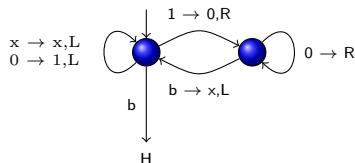
**Starttape:**  $b^*(0 \vee 1)^*b^*$  — start til høyre i tallet

**Spesifikasjon:** Konverterer det binære tallet  $(0 \vee 1)^*$  til det unære tallet  $x^*$

- ▶ Vi går i en løkke
- ▶ Trekker fra 1 binært og legger til  $x$  unært inntil vi treffer blank  $b$

# Grunnleggende turing maskiner

## Basic 5: Konverterer unær/binær



**Alfabet:**  $0, 1, x, b$  —  $b$  er blank

**Starttape:**  $b^*(0 \vee 1)^*b^*$  — start til høyre i tallet

**Spesifikasjon:** Konverterer det binære tallet  $(0 \vee 1)^*$  til det unære tallet  $x^*$

- ▶ Vi går i en løkke
- ▶ Trekker fra 1 binært og legger til  $x$  unært inntil vi treffer blank  $b$
- ▶ Går like bra i 10-tall systemet



# Grunnleggende turing maskiner

## Simulere turing mannen

- ▶ Turing mannen utfører enkle operasjoner — sammenlikne, slå opp i tabell, erstatte ord

# Grunnleggende turing maskiner

## Simulere turing mannen

- ▶ Turing mannen utfører enkle operasjoner — sammenlikne, slå opp i tabell, erstatte ord
- ▶ Med varianter av Basic 1-5 kan vi gjøre alt som turing mannen gjør

# Grunnleggende turing maskiner

## Simulere turing mannen

- ▶ Turing mannen utfører enkle operasjoner — sammenlikne, slå opp i tabell, erstatte ord
- ▶ Med varianter av Basic 1-5 kan vi gjøre alt som turing mannen gjør
- ▶ Dette er grunnlaget for Turings universelle maskin

# Grunnleggende turing maskiner

## Simulere turing mannen

- ▶ Turing mannen utfører enkle operasjoner — sammenlikne, slå opp i tabell, erstatte ord
- ▶ Med varianter av Basic 1-5 kan vi gjøre alt som turing mannen gjør
- ▶ Dette er grunnlaget for Turings universelle maskin
- ▶ Med den universelle maskinen får vi umulighets teoremer

# Grunnleggende turing maskiner

## Simulere turing mannen

- ▶ Turing mannen utfører enkle operasjoner — sammenlikne, slå opp i tabell, erstatte ord
- ▶ Med varianter av Basic 1-5 kan vi gjøre alt som turing mannen gjør
- ▶ Dette er grunnlaget for Turings universelle maskin
- ▶ Med den universelle maskinen får vi umulighets teoremer
- ▶ Maskiner som kode — intensjonale egenskaper

# Grunnleggende turing maskiner

## Simulere turing mannen

- ▶ Turing mannen utfører enkle operasjoner — sammenlikne, slå opp i tabell, erstatte ord
- ▶ Med varianter av Basic 1-5 kan vi gjøre alt som turing mannen gjør
- ▶ Dette er grunnlaget for Turings universelle maskin
- ▶ Med den universelle maskinen får vi umulighets teoremer
- ▶ Maskiner som kode — intensjonale egenskaper
- ▶ Maskiner som input/output — ekstensjonale egenskaper

# Grunnleggende turing maskiner

## Simulere turing mannen

- ▶ Turing mannen utfører enkle operasjoner — sammenlikne, slå opp i tabell, erstatte ord
- ▶ Med varianter av Basic 1-5 kan vi gjøre alt som turing mannen gjør
- ▶ Dette er grunnlaget for Turings universelle maskin
- ▶ Med den universelle maskinen får vi umulighets teoremer
- ▶ Maskiner som kode — intensjonale egenskaper
- ▶ Maskiner som input/output — ekstensjonale egenskaper
- ▶ Umulig å lage maskin som avgjør interessante ekstensjonal egenskaper