Kapittel 23

Formelle språk og grammatikker

A ER ALLTID ENDELIG (FOR OSS)

Definisjon 9.7 Alfabet, tegn, streng

Anta at A er en mengde av tegn. Mengden A kalles et **alfabet** (*eng: alphabet*). Mengden av **strenger** (*eng: strings*) over A er den minste mengden A^* slik at:

- $\Lambda \in A^*$, hvor symbolet Λ står for **den tomme strengen** (*eng: empty string*) og
- hvis $s \in A^*$ og $x \in A$, er $sx \in A^*$, hvor sx står for resultatet av å slå sammen s og x. Vi skriver s i stedet for Λs.

Et **språk** (*eng: language*) over A er en delmengde av A^*.

Eks: A – ALLE TEGN PÅ TASTATURET

$L \subseteq A^*$

- ALLE NORSKE ORD
- ALLE FRANSKE ORD

} ENDELIGE

- ALLE NORSKE SETNINGER
- ALLE STRENGER AV 0 OG 1 SOM REPRESENTERER BINÆRT TALL.

$L \subseteq A^*$

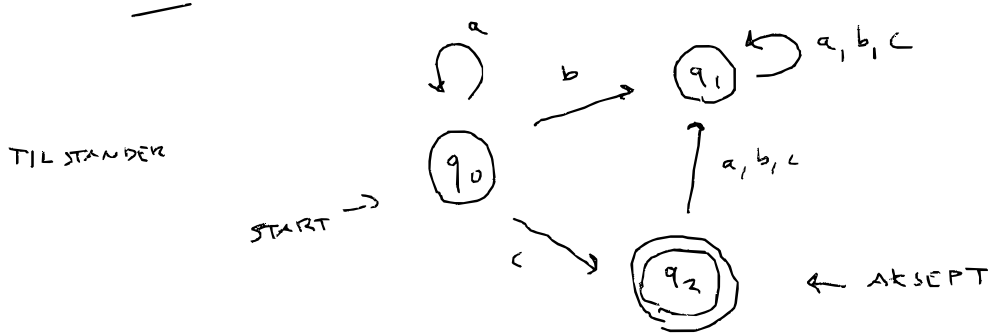
$L \in \mathcal{P}(A^*)$

- HVOR KOMPLISERT ER ET SPRÅK? HVOR KRAFTIG MÅ EN "MASKIN" (AUTOMATON) VÆRE SOM SKAL AVGJØRE OM EN STRENG ER MED I SPRÅKET ELLER IKKE?

DETERMINISTISKE ENDELIGE TILSTANDSMASKINER

- EN "MASKIN" SOM AVGJØR OM EN STRENG ER MED I ET GITT SPRÅK ELLER IKKE.

IDÉ: ALFABET A, F.EKS $A = \{a, b, c\}$



AKSEPTERER: STRENGER MED VILKÅRLIG MANGE a' ER FØRST OG EN c TIL SIST.

Definisjon 23.6 Deterministisk, endelig tilstandsmaskin

En **deterministisk, endelig tilstandsmaskin** (*eng: deterministic finite automaton, DFA*) består av følgende:

- En endelig mengde **tilstander** (*eng: states*).
- En endelig mengde **inputsymboler** (*eng: input symbols*), også kalt et **alfabet** (*eng: alphabet*).
- En **starttilstand** (*eng: start state*), som er en av tilstandene.
- En mengde **aksepterende tilstander** (*eng: final/accepting states*), som er en delmengde av tilstandene.
- En **overgangsfunksjon** (*eng: transition function*), som er en funksjon som tar to argumenter – en tilstand og et input-symbol – og som returnerer en tilstand.

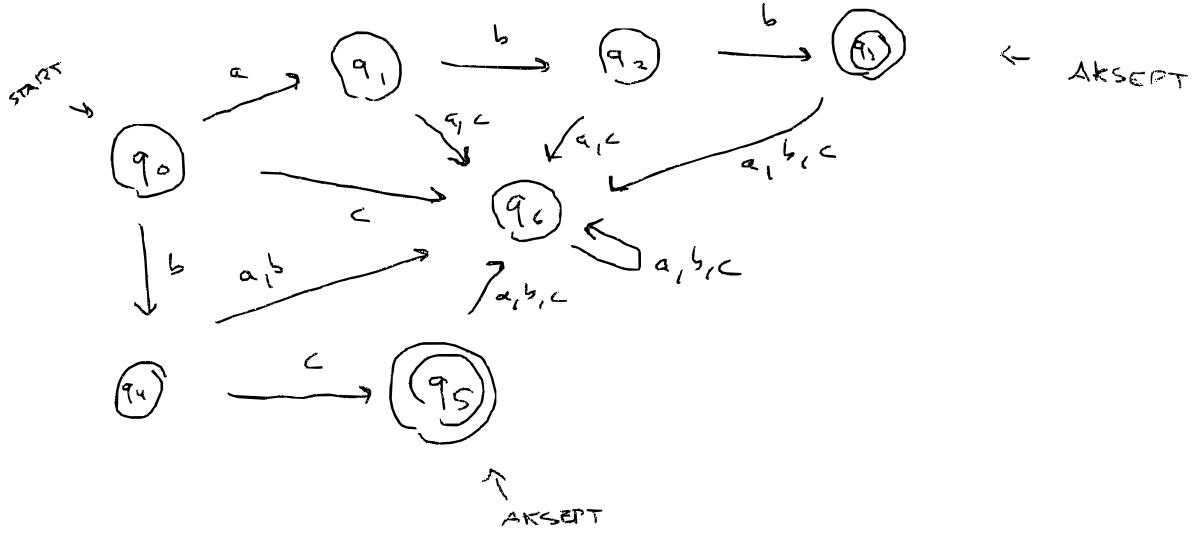
— VI SIER AT EN STRENG AKSEPTERES AV MASKINEN HVIS VI ENDER OPP I EN AKSEPTERENDE TILSTAND

— VI KAN SI AT EN TILSTANDSMASKIN DEFINERER SPRÅKET AV ALLE STRENGER SOM DEN AKSEPTERER.

EKSEMPLER

- $A = \{a, b, c\}$

LAG EN MASKIN SOM DEFINERER SPRÅKET $\{abb, bc\}$



SPØRSMÅL

GITT A, FOR HVILKE $L \in \mathcal{P}(A^*)$

FINS DET EN DETERMINISTISK ENDELIG TILSTANDS-MASKIN SOM DEFINERER L, ALTSÅ SOM AKSEPTERER NØYAKTIG DE STRENGENE SOM ER I L?

NYTT PERSPEKTIV: OPERASJONER PÅ SPRÅK

UNION:

$$\cup : \mathcal{P}(A^*) \times \mathcal{P}(A^*) \longrightarrow \mathcal{P}(A^*)$$

$$\langle L, M \rangle \longmapsto L \cup M$$

- OGSÅ SNITT, KOMPLEMENT, MENGDEDIFFERANSE

Definisjon 23.1 Konkatenering av to språk
Hvis L og M er språk, er **konkateneringen** (*eng: concatenation*) av L og M språket $LM = \{st \mid s \in L \text{ og } t \in M\}$.

$$\mathcal{P}(A^*) \times \mathcal{P}(A^*) \longrightarrow \mathcal{P}(A^*)$$

$$\langle L, M \rangle \longmapsto LM = \{ st \mid s \in L \text{ og } t \in M \}$$

Eks: $L = \{a\}$, $M = \{b\}$

$$LM = \{ab\}$$

$$ML = \{ba\}$$

$$(LM)P = L(MP)$$

- KONKATENERING ER ASSOSIATIVT, MEN IKKE KOMMUTATIVT.

- ENHETSELEMENT $\{\lambda\}$

$$L\{\lambda\} = L = \{\lambda\}L$$

$L \in \mathcal{P}(A^*)$

$$L^0 = \{\lambda\}$$

$$L^1 = L$$

$$L^2 = LL$$

$$\vdots$$

$$L^n = LL\ldots L \quad (n \text{ GANGER})$$

$$L^{n+1} = LL^n$$

$$L^0 \cup L \cup L^2 \cup \ldots \cup L^n$$

Definisjon 23.2 Tillukning av et språk

Hvis L er et språk, er **Kleene-tillukningen** (*eng: Kleene closure*) – eller bare **tillukningen** (*eng: closure*) – av L, språket $L^* = L^0 \cup L^1 \cup L^2 \cup L^3 \cup \cdots$, det vil si mengden av alle endelige strenger over L.

Eks $L = \{ ab \}$

$L^* = \{ \lambda, ab, abab, ababab, \ldots \}$

$M = \{ a, b \}$

$M^* = \{ \lambda, a, b, aa, ab, ba, aaa, aba, \ldots \}$

ALLE STRENGER OVER $\{a, b\}$.

Definisjon 23.3 Regulært språk

Mengden av **regulære språk** (*eng: regular languages*) over et alfabet A er den minste mengden slik at følgende holder:

- $\emptyset$ og $\{\Lambda\}$ er **regulære språk** over A, og $\{\mathtt{a}\}$ er et **regulært språk** over A for alle $\mathtt{a} \in A$.
- Hvis L og M er regulære språk, er $L \cup M$, LM og L^* også **regulære språk**.

— REGULÆRE SPRÅK ER LUKKET UNDER SNITT, KOMPLEMENT, DIFFERANSE

— ALLE ENDELIGE SPRÅK ER REGULÆRE

— DE REGULÆRE SPRÅKENE ER NØYAKTIG DE SOM KAN DEFINERES AV ENDELIGE DETERMINISTISKE TILSTANDSMASKINER!

$A = \{a, b, c\}$

Eksempel

- $L = \{abb, bc\}$

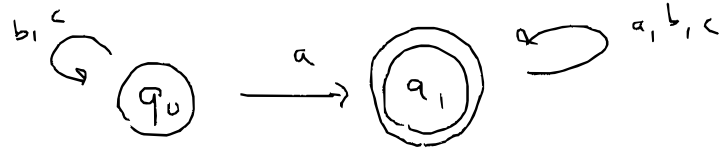
$\{a\}\{b\}\{b\} \cup \{b\}\{c\}$

- $L = \{c, ac, aac, \ldots\}$

$\{a\}^* \{c\}$

- L - ALLE STRENGER SOM HAR MINST ÉN a I SEC.

$(\{a\} \cup \{b\} \cup \{c\})^* \{a\} (\{a\} \cup \{b\} \cup \{c\})^*$



Definisjon 23.4 Regulært uttrykk

Mengden av **regulære uttrykk** (*eng: regular expressions*) over et alfabet A er induktivt definert som den minste mengden slik at:

- $\emptyset$ og Λ er **regulære uttrykk** over A, og a er et **regulært uttrykk** over A for alle $a \in A$.
- Hvis R og S er regulære uttrykk, er (R), $R|S$, RS og R^* også **regulære uttrykk**.

Eks $A = \{a, b, c\}$

- $abb \mid bc$ – $\{abb, bc\}$
- a^*c
- $(a|b|c)^* a (a|b|c)^*$

Definisjon 23.5 Tolkning av regulære uttrykk

La A være et alfabet. Vi definerer rekursivt funksjonen L fra mengden av regulære uttrykk over A til mengden av regulære språk over A på følgende måte:

- $L(\emptyset) = \emptyset$
- $L(\Lambda) = \{\Lambda\}$
- $L(a) = \{a\}$, for hver $a \in A$
- $L(R|S) = L(R) \cup L(S)$ (*union*)
- $L(RS) = L(R)L(S)$ (*konkatenering*)
- $L(R^*) = L(R)^*$ (*tillukning*)

I tillegg har parentesene ingen annen funksjon enn å gruppere: $L((R)) = L(R)$. For ethvert regulært uttrykk E får vi på denne måten et regulært språk $L(E)$.

$$L(abb \mid bc) = L(abb) \cup L(bc)$$
$$= L(a)L(b)L(b) \cup L(b)L(c)$$
$$= \{a\}\{b\}\{b\} \cup \{b\}\{c\}$$

PRESEDENSREGLER:

$*$ - STERKEST

$|$ - SVAKEST

SÅ $a \mid bc^*$

$= (a) \mid (b\,(c)^*)$

Eks (23.6)

UTTRYKKET DISSE REG. UTTRYKKENE SAMME SPRÅK ELLER IKKE

a) $(10)^*$ OG $(01)^*$

NEI

$\{\lambda, 10, 1010, ..\} \neq \{\lambda, 01, 0101, ..\}$

b) $(1 \mid 0)^*$ OG $(0 \mid 1)^*$

JA

$\{1,0\}^* = \{0,1\}^*$

EN NY MÅTE, OG MULIGHET TIL Å FANGE INN LANGT FLERE SPRÅK:

IDÉ: GI ET SPRÅK VED OMSKRIVINGS REGLER.

ALFABETET KALLES "TERMINAL SYMBOL"

NYE "IKKE-TERMINAL SYMBOLER" - INKLUDERT ET START SYMBOL,

SAMTEN MED SUBSTITUSJONS REGLER.

Definisjon 23.7 Formell grammatikk

En **formell grammatikk** (*eng: formal grammar*), eller en **grammatikk** (*eng: grammar*), består av følgende:

- En endelig mengde **terminalsymboler** (*eng: terminal symbols*).
- En endelig mengde **ikke-terminalsymboler** (*eng: nonterminal symbols*).
- Et **startsymbol** (*eng: start symbol*), ett av ikke-terminalsymbolene.
- En endelig mengde **produksjonsregler** (*eng: production rules*) på formen $X \to Y$, hvor X og Y er strenger av terminal- og ikke-terminalsymboler og X ikke kan være den tomme strengen.

Vi antar at mengdene med terminal- og ikke-terminalsymboler er disjunkte og at det finnes minst én produksjonsregel med kun startsymbolet på venstresiden.

FEKS S - START

REGLER: $S \Rightarrow \lambda$, $S \Rightarrow aS$ } $S \Rightarrow \lambda \mid aS$ (FORKORTELSE)

KAN UTLEDE: $S \Rightarrow \lambda$

$S \Rightarrow aS \Rightarrow a$

a^*

$S \Rightarrow aS \Rightarrow aaS \Rightarrow aa$

$S \Rightarrow aS \Rightarrow aaS \Rightarrow aaaS \Rightarrow aaa$

Definisjon 23.8 Utledninger

Anta at en grammatikk er gitt og at M er mengden av alle strenger over terminal- og ikke-terminalsymboler i grammatikken. La den binære relasjonen $\Rightarrow$ være definert på M slik at $AXB \Rightarrow AYB$ holder nøyaktig når $X \to Y$ er en produksjonsregel i grammatikken og A og B er strenger i M. En sekvens på formen $x_1 \Rightarrow x_2 \Rightarrow \cdots \Rightarrow x_n$ kalles en **utledning** (*eng: derivation*) av x_n fra x_1, og vi sier at x_n kan utledes fra x_1. **Språket** (*eng: language*) som defineres av grammatikken er mengden av strenger som kan utledes fra startsymbolet S og som kun inneholder terminalsymboler.

Eks

S => B | C

B => abb

C => bc

- {abb, bc}

$A = \{a, b, c\}$

Eks

- $S \Rightarrow a$

  $S \Rightarrow aS \mid bS \mid cS$

  SLUTTER MED EN a

- $S \Rightarrow bS \mid cS$

  $S \Rightarrow aB$

  $B \Rightarrow bB \mid cB \mid aB \mid \lambda$

  MINST ÉN a

  NØYAKTIG ÉN a

- $S \Rightarrow bS \mid cS$

  $S \Rightarrow aB$

  $B \Rightarrow bB \mid cB \mid \lambda$

Definisjon 23.9 Regulær grammatikk

En **regulær grammatikk** (*eng: regular grammar*) er en grammatikk hvor enhver produksjonsregel er på formen $A \to X$, hvor A er et ikke-terminalsymbol og X består av maksimalt ett ikke-terminalsymbol som i så fall forekommer lengst til høyre.

– DISSE DEFINERER NØYAKTIG DE REGULÆRE SPRÅKENE.

Definisjon 23.10 Kontekstfri grammatikk

En **kontekstfri grammatikk** (*eng: context-free grammar*) er en grammatikk hvor enhver produksjonsregel er på formen $A \to X$, hvor A er et ikke-terminalsymbol og X en streng av terminal- og ikke-terminalsymboler. Et språk som defineres av en kontekstfri grammatikk kalles et **kontekstfritt språk** (*eng: context-free language*).

Eks

$S \Rightarrow \lambda$

$S \Rightarrow aSb$

$L = \{\lambda, ab, aabb, \ldots, a^n b^n, \ldots\}$

– DETTE SPRÅKET ER IKKE REGULÆRT