Forelesning 24
TILSTANDSMASKINER OG REGULÆRE SPRÅK

Innhold
- Formelle språk-repetisjon
- Regulære språk
- Regulære uttrykk
- Sammenheng: Regulære språk/uttrykk og tilstandsmaskiner

Formelle språk

A en endelig mengde, et alfabet.

Eksempel

$A = \{0,1\}$ eller $A = \{a,b,c,\ldots,z\}$.

En streng (eller et ord) av lengde n, er en n-tuppel av symboler fra A.

Husk: for eksempel er en 4-tuppel et element i $A \times A \times A \times A$, og vi skriver vanligvis et element som $\langle c,a,b,a \rangle$.

Men: her vil vi istedet skrive caba, og vi sier at caba har lengde 4.

Symbol for en tom streng: Λ (den greske bokstaven Lambda). Vi sier at Λ har lengde 0.

Språk-definisjon

Mengden av alle endelige strenger over et alfabet A, kaller vi A^*.

Eksempel

Alle endelige bistrenger er alle endelige strenger over alfabetet $A = \{0,1\}$.
Altså $A^* = \{\Lambda, 0, 1, 00, 01, 10, 11, 000, \ldots\}$.

Definisjon

Hvis A er et alfabet, og $S \subseteq A^*$, kalles S et (formelt) språk.

Eksempel

$\{0, 00, 10\}$ er et (endelig) språk over alfabetet $\{0,1\}$.

Eksempel

$\{01, 001, 0001, 00001, 000001, \ldots\} = \{0^n 1 \mid n \in \mathbb{N} \text{ and } n > 0\}$ er et (uendelig) språk.

NB! a^n betyr $\underbrace{aa\ldots a}_{n\ a\text{-er}}$, så $a^4 = aaaa$, og $a^0 = \Lambda$.

Virkelige eksempel

F.eks: Alle syntatisk gyldige programmer/kildekoder i Python utgjør et formelt språk.

Mengden av alle syntaktisk gyldige utsagnslogiske formler utgjør et formelt språk.

Mengden av alle endelige bitstrenger er et formelt språk.

Enhver mengde av endelige bistrenger utgjør et formelt språk.

Velg deg en norsk ordbok. Alle ordene som forekommer i den
utgjør et formelt språk over alfabetet $\{a, b, c, \dots, \}$.

Konkatenering av strenger og språk

Konkatenering (sammenslåing) av strenger. A alfabet. s, t strenger i A^*.
Da er st også en streng. Hvis s har lengde m og t har lengde n, så har st lengde $m+n$.

Eksempel

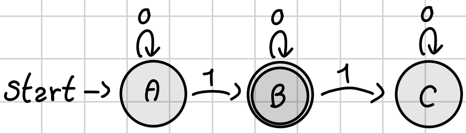
Konkateneringen av 1011 og 001 er 1011001.
Vi kan også konkatenere to språk over samme alfabet.

$S = \{00, 11, 000, 111\}$ og $T = \{01, 10\}$ konkateneres til
$ST = \{0001, 0010, 1101, 1110, 00001, 00010, 11101, 11110\}$

Altså vi setter sammen alle mulige ord i de to språkene, og får et nytt språk.
Generelt: S, T to språk, da er $ST = \{st \mid s \in S, t \in T\}$.
Og, mer generelt, kan vi sette sammen (konkatenere) en endelig mengde språk på samme måte.

Språk fra tilstandsmaskiner

Husk vår (endelige tilstands-) maskin som aksepterer bitstrenger som inneholder
nøyaktig en ener.



Mengden av strenger som gjenkjennes av denne maskinen, kan beskrives som en konkatinering av språk.

La N være språket som består av bitstrenger med bare nullere og den tomme strengen.
Altså $N = \{0^n \mid n \in \mathbb{N}\}$. (Husk at $0^0 = \Lambda$, per definisjon).

La E være språket som består av nøyaktig ett element, nemlig en 1-er. Altså $E = \{1\}$.

Da er språket som denne maskinen gjenkjenner konkateneringen NEN.

Flere operasjoner på språk

Union

Vi kan også ta unionen av to (eller flere) språk.
La S og T være to språk over samme alfabet A, dvs $S \subseteq A^*$ og $T \subseteq A^*$.
Da er også $(S \cup T) \subseteq A^*$.

Eksempel

La S være språket av bitstrenger som inneholder nøyaktig en 1-er, og la T være språket av bitstrenger som inneholder minst to 1-ere.

Da er $S \cup T$ språket av bitstrenger som inneholder minst en ener.

Tillukning

Tillukningen (kalles også Kleene-tillukningen) til et språk S er
$S^* = \bigcup_{i=0}^{\infty} S_i = S^0 \cup S \cup S^2 \cup S^3 \cup \dots$
(Symbolet $*$ kalles Kleene-stjerne i denne sammenhengen).

Eksempel

La $S = \{00, 1\}$ over alfabetet $A = \{0, 1\}$.
Da er $S^* = \{\Lambda, 00, 1, 0000, 001, 100, 11, 000000, 00001, \dots\}$

Regulære språk

Definisjon

Gitt et alfabet A, er mengden av regulære språk over A induktivt definert:
$\emptyset$, $\{\Lambda\}$ er regulære språk, og $\{a\}$ er et regulært språk for hvert symbol $a \in A$.
Hvis S og T er regulære språk, så er ST, $S \cup T$ og S^* regulære språk.

Eksempler

La $A = \{0, 1\}$.
Hele $A^* = \{0,1\}^*$ er et regulært språk. Siden både $\{0\}$ og $\{1\}$ er regulære språk, er $\{0\} \cup \{1\} = \{0,1\}$ et regulært språk. Dermed er også $\{0,1\}^*$ et regulært språk.

Mengden av alle bitstrenger som inneholder minst tre 1-ere på rad er et regulært språk:

$$\{0,1\}^*\{111\}\{0,1\}^*$$

Mengden av alle bitstrenger som totalt inneholder minst tre 1-ere er et regulært språk:

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

eller litt kortere:

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

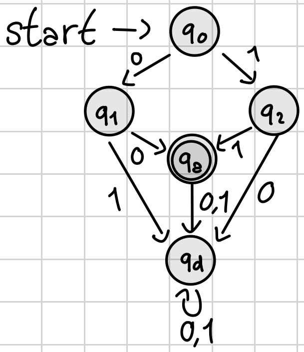
Endelige språk

Et endelig språk er regulært. Hvorfor?

- Språket som består av en bokstav er regulært.
- Språket som består av en streng er regulært (konkatenering).
- Språket som består av endelig mange strenger er regulært (union).

Oppgave

Finn en tilstandsmaskin som gjenkjenner det endelige språket $\{00, 11\}$.



Regulære uttrykk

Et hvert regulært språk kan uttrykkes ved et regulært uttrykk.

Husk: Et regulært språk er en **mengde**, mer presist en delmengde av A^* for et alfabet A.

Et regulært uttrykk er en måte å beskrive denne mengden på, ved hjelp av symboler.

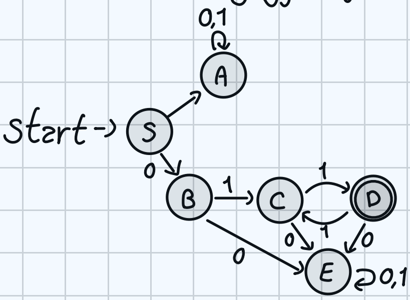
Vi beskriver mengden $A \cup B$ ved uttrykket $A|B$, og for et element eller en streng a, skriver vi a^* heller enn $\{a\}^*$.

Eksempler

- $(01)^*$ er et uttrykk for språket $\{\Lambda, 01, 0101, 010101, \ldots\}$.
- $(0|1)^*$ og $(1|0)^*$ er begge uttrykk for språket av alle bitstrenger altså $\{0,1\}^* = (\{0\} \cup \{1\})^* = (\{1\} \cup \{0\})^*$.
- $0|1^*$ er et uttrykk for språket $\{0, \Lambda, 1, 11, 111, 1111, \ldots\} = \{0\} \cup \{1\}^*$.
- 01^* er et uttrykk for språket $\{0, 01, 011, 0111, 01111, \ldots\}$.

Eksempel

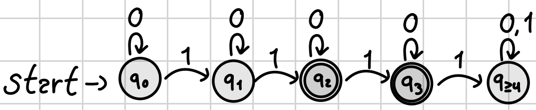
Språket $\{01\}\{11\}^*\{1\} \cup \{1\}\{0,1\}^*$ kan beskrives ved det regulære uttrykket $(01(11)^*1|(1(0|1)^*)$ og gjenkjennes av tilstandsmaskinen:



Oppgave

Finn et regulært uttrykk for det regulære språket som består av bitstrenger med nøyaktig to eller tre 1-ere.

Husk: denne maskinen gjenkjenner dette språket:



to 1-ere vilkårlig mengde 0-ere

Svar: $0^*10^*10^*(\Lambda|10^*)$ eller $(0^*10^*10^*)|(0^*10^*10^*10^*)$ ← tre 1-ere og vilkårlig mengde 0-ere

Sammenheng

For ethvert regulært språk S finnes en endelig tilstandsmaskin som gjenkjenner strengene i S.
For enhver endelig tilstandsmaskin M, finnes et regulært språk som beskriver strengene M gjenkjenner.

Husk: vi sier at en endelig tilstandsmaskin gjenkjenner språket S hvis den aksepterer alle strengene i S, og ingen andre strenger.

Husk: tilstandsmaskiner skal ha en starttilstand. Tenk over om den tomme strengen skal aksepteres eller ikke. Den aksepteres hvis og bare hvis starttilstanden er en aksepterende tilstand.

Vi har bare diskutert deterministiske endelige tilstandsmaskiner.
Det finnes også ikke-deterministiske (såvidt nevnt i RA). Man kan vise at disse også kun gjenkjenner regulære språk.