Dagens föreläsning

- Inledning till beräkningsbarhet.
- Turingmaskiner
- Church-Turing tesen.

Turingmaskiner

- Den sista beräkningsmodell vi tar upp på kursen är Turing maskinen.
- Utvecklades av Alan Turing 1936, som även har gett namn åt Turingtestet.
- I princip en finit automat som har tillgång till ett obegränsat minne som tillåter både läsning och skrivning (chmod 666):
- minnet i en TM är ett oändligt långt band, och på detta finns ett skrivhuvud som kan flyttas fram och tillbaka på bandet och skriva och läsa. Hur detta skall flyttas bestäms av en finit automat. *tavlan*
- Liknar mer en vanlig dator än vad dom andra modellerna gör, men är faktiskt kraftfullare.

Mer (informellt) om Turingmaskiner

- Det som skiljer Turing maskiner från finita automater är
  - En TM kan både skriva och läsa bandet.
  - Skrivhuvudet kan flyttas både framåt och bakåt.
  - Bandet är oändligt långt.
  - De speciella tillstånden för acceptans eller refusering gäller direkt maskinen nått dem.
- Vi tittar på en TM M_1 som avgör om en sträng tillhör språket $B = \{w\#w \mid w \in \{0,1\}^*\}$. *tavlan*

Definition av Turingmaskin

En *Turingmaskin* är en tupel $(Q, \Sigma, \Gamma, \delta, q_0, q_{\text{ja}}, q_{\text{nej}})$, där Q, Σ, och Γ är finita mängder och

- Q är maskinens tillståndsmängd,
- Σ är inputalfabetet som inte får innehålla *blanksymbolen* $\sqcup$,
- Γ är band alfabetet, där $\sqcup \in \Gamma$ och $\Sigma \subset \gamma$,
- $\delta : Q \times \Gamma \mapsto Q \times \Gamma \times \{L, R\}$ är transitionsfunktionen,
- $q_0 \in Q$ är starttillståndet,
- $q_{\text{ja}} \in Q$ är det accepterande tillståndet, och
- $q_{\text{nej}} \in Q$ är det refuserande tillståndet.

Hur en Turingmaskin exekverar - informellt

- Maskinen får sin indata på i början på bandet,
- skrivhuvudet står initialt längst till vänster, och
- innan beräkning börjar är maskinen in tillstånd q_0.
- När maskinen startat rör den sig enligt δ, som är en funktion, så därför är maskinen deterministisk.
- Om maskinen försöker röra skrivhuvudet till vänster om början på bandet så blir det kvar på samma ställe.
- Beräkningen forsätter tills maskinen når tillståndet q_{ja} eller q_{nej}.
- Om maskinen inte når något av dessa tillstånd fortsätter beräkning för evigt.

Konfigurationer

- En konfiguration av en TM berättar hur bandet ser ut, i vilket tillstånd maskinen är, och var på bandet skrivhuvudet befinner sig.
- Om en maskin är i tillstånd q, strängen framför skrivhuvudet är u, symbolen under skrivhuvudet är a, och strängen efter skrivhuvudet är v så skriver vi den konfigurationen som $uqav$.
- Vi tar inte med alla omgivande blanksteg i konfigurationen, dom är ju oändligt många så det går för mycket tid & blyerts.
- En TM kan gå från konfigurationen uaq_1bv till uq_jacv om $\delta(q_1, b) = (q_j, c, L)$, och
- från konfigurationen uaq_1bv till $uacq_jv$ om $\delta(q_1, b) = (q_j, c, R)$.
- Startkonfigurationen är q_0w, en konfiguration är accepterande om maskinen är i tillstånd q_{ja} och refuserande om den är i tillstånd q_{nej}. De senare två typerna kallas även haltkonfigurationer.

Hur en Turingmaskin exekverar - formellt

En Turingmaskin M accepterar en inputsträng w om det finns en sekvens konfigurationer $C_1, C_2, \ldots, C_k$, så att

- C_1 är startkonfigurationen givet inputsträngen w,
- maskinen kan gå från C_i till C_{i+1} enligt δ, och
- C_k är en accepterande konfiguration.

Exempel: Addition av binära tal.

Exempel: Distinkthetsproblemet

Distinkthetsproblemet är att godkänna att en lista av #-separerade strängar över $\{0, 1\}$ är fri från upprepningar. Med andra ord, att känna igen språket

$$E = \{\#x_1\#x_2\# \dots \#x_n \mid x_i \neq x_j \text{ om } i \neq j\} \ .$$

Beskriv en Turingmaskin som löser det här problemet.

Distinkthetsproblemet, forts.

1. Placera en markör på den vänstraste symbolen på bandet. Om den symbolen är ett blanksteg accepterar vi strängen. Om symbolen var # forsätter vi med nästa steg. För alla andra symboler refuserar vi.
2. Placera en markör även på nästa #. Om det inte finns en till #, då var x_1 den enda strängen, så vi accepterar.
3. Jämför de två strängarna till höger om de #-symboler som markerats genom att flytta skrivhuvudet fram och tillbaka. Refusera strängen om dom är lika.
4. Flytta den högraste av de två markörerna till nästa #-symbol. Om vi inte hittar något sådan flyttar vi den vänstraste markören till nästa # och placerar den högraste markören på # symbolen direkt till höger. Om vi den här gången inte hittar ett lämpligt # för den andra symbolen så har vi jämfört all strängarna och kan acceptera,
5. men lyckas vi flytta markörerna så har vi arbete kvar. Gå till steg 3.

Typer av språk

- Mängden strängar som M accepterar är språket som M känner igen och skrivs $L(M)$.
- Ett språk är *Turing-igenkännbart* om det finns en Turingmaskin som känner igen det.
- Vi säger att en Turingmaskin *avgör* ett språk om den stannar och svarar ja eller nej på alla inputsträngar.
- Ett språk är *Turing-avgörbart* om det finns en Turingmaskin som avgör det.

Varianter av Turingmaskinen

- Det finns många olika varianter av Turingmaskinen, men de är alla lika kraftfulla,
- så vi säger att Turingmaskinen är en *robust* beräkningsmodell eftersom den måste ändras väldigt mycket innan dess beräkningskraft påverkas.
- Vi ska nu titta på några olika varianter som kan vara praktiska att känna till.
- T.ex. vad händer om vi förutom förflyttningarna L och R även tillåter att en TM står still när den skrivit ett tecken?

Flerbandsmaskiner

- En *flerbands* TM är är som en vanlig TM, men den har fler än ett band.
- Varje band har sitt eget skrivhuvud, och
- maskinen startar med inputsträngen på det första bandet. De övriga banden är initialt tomma.
- Transitionsfunktionen tillåter att man parallellt läser, skriver och flyttar skrivhuvudet på samtliga band.
- Formellt,

$$\delta : Q \times \Gamma^k \mapsto Q \times \Gamma^k \times \{L, R, S\}^k \quad .$$

Ekvivalensen mellan enbandsTM och flerbandsTM

Teorem 2.13 Varje flerbandsTM har en ekvivalent enbandsTM.

Ickedeterminisitiska TM

- Precis som för NFA:er är δ nu en relation.
- Mer formellt, $\delta : Q \times \mathcal{P}(\Gamma \mapsto Q \times \Gamma \times \{L, R\})$
- Beräkningen i en NTM förgrenar sig, och om någon gren leder till q_{ja} accepteras inputsträngen.
- Varje NTM har en ekvivalent DTM.

Fler varianter

Påverkas beräkningskraften hos en vanlig TM om ...

- den har ett band som är oändligt i båda riktningarna?
- den får läsa och skriva k sammanhängde celler samtidigt?
- den har tillgång till en k dimensionell array av minnesceller?
- den får ändra innehållet i varje cell max två gånger?
- den får ändra innehållet i varje cell max en gång?
- bandet ersätts med två heltalsregister som har operationerna `isZero?`‘, `inc`, och `dec`?
- den bara får röra sig åt höger?

Rekursivt uppräkningsbara språk

- En synonym till Turing-igenkännbara språk är *rekursivt uppräkningsbara språk*.
- Namnet kommer från en variant av Turingmaskinen som kallas *uppräknare*; i princip en TM med en printer som kan skriva ut strängar. *tavlan*
- En uppräknare tar ingen input, och om den inte avbryter sin beräkning kan den skriva ut en oändlig lista med strängar.
- Mängden strängar som uppräknaren skriver ut är dess språk. Strängarna kan komma i vilken ordning som helst och med uppräkningar.

Ekvivalens mellan uppräknare och Turingmaskin

Teorem 3.21 Ett språk är Turing-igenkännbart om och endast om det finns en uppräknare som skriver ut alla strängar i språket. *bevisskiss*

Church-Turing tesen

Alla algoritmer kan implementeras som en Turingmaskin.