Sammanfattning
Föreläsning 2 - Digitalteknik

I boken: avsnitt 2.1-2.2 (-)

Tillståndsgrafer och maskiner

En *graf* är ett par av mängder $(\mathcal{V}, \mathcal{E})$, där $e \in \mathcal{E}$ är av formen $e = (v_1, v_2)$, $v_1, v_2 \in \mathcal{V}$. Elementen i $\mathcal{V}$ kallar vi för *noder* (hörn, punkter) och elementen i $\mathcal{E}$ kallar vi för bågar (kanter, övergångar). Bågarna kan ses som en övergång från en nod till en annan (möjligen samma) nod. En *riktad graf* har bågarna ordnade, dvs $(a, b) \neq (b, a)$.

Def. 2.2: En *tillståndsmaskin* (maskin, automat) består av mängderna $\mathcal{I}$=alla möjliga insignaler; $\mathcal{Z}$=alla möjliga utsignaler; $\mathcal{S}$=alla möjliga tillstånd; samt funktionerna $\delta : \mathcal{S} \times \mathcal{I} \to \mathcal{S}$ (*nästa-tillstånds funktion*) och $\lambda : \mathcal{S} \times \mathcal{I} \to \mathcal{Z}$ (*utsignalfunktion*). Till detta kan vi lägga ett *starttillstånd* s_0.

Ett önskat beteende hos ett digitalt system kan modelleras genom en tillståndsmaskin. Man måste då bestämma tillståndsmaskinen. Då mängderna normalt sätt är ändliga kan vi specificera maskinen i tabellform. Det är dock betydligt enklare att först specificera maskinen genom att ta fram en *tillståndsgraf*.

En tillståndsgraf (tillståndsdiagram) är den riktade graf som får då $\mathcal{S}$ är noderna i grafen. Mängden bågar är de $(s, \delta(s, i))$ som får då s, i löper igenom alla värden i $\mathcal{S}, \mathcal{I}$. Varje båge i grafen har dessutom en etikett $i/\lambda(s, i)$. Innebörden är att i givet tillstånd med given insignal i så byter vi tillstånd till nästa tidsenhet genom att följa den båge från tillståndet som har etikett $i/\lambda(s, i)$ och vi ändrar dessutom genast utsignalen till $\lambda(s, i)$. Följande gäller då $s(t)$ är tillståndet vid tidpunkt t, $i(t)$ är insignalen vid tid t, och $z(t)$ är utsignalen vid tid t:

$$s(t+1) = \delta(s(t), i(t)) \quad (1)$$

$$z(t) = \lambda(s(t), i(t)). \quad (2)$$

Det finns en skillnad mellan händelsestyrda maskiner och klock-kontrollerade maskiner. I en händelsestyrd maskin får vi en ny händelse varje tidsenhet. Exempelvis om vi processar en sekvens med bitar i en händelsestyrd maskin så kommer en ny bit varje tidsenhet. I en klock-kontrollerad maskin kan insignalen vara fix vid ett värde under många tidsenheter, innan den ändrar värde och blir fix vid det nya värdet under ett antal tidsenheter, osv. Vi måste tänka på detta när vi gör tillståndsgrafen.

Slutligen finns begreppet *trellis*. En trellis är en riktad graf som är uppdelad i sektioner. Bågar existerar endast mellan noder i sektion i och $i+1$, $i \geq 0$. Beteendet hos en maskin kan illustreras med en trellis. Varje väg i trellisen svarar mot en insignalsekvens och mängden av vägar i trellisen ger mängden av möjliga utsignalsekvenser.

Under föreläsningen studerar vi ett antal exempel på problemställningar och försöker modellera det önskade beteendet i en tillståndsgraf. Se exempel i boken och på slides.

Avancerat exempel: En (d, k)- runlength-limited sequence är en bitsekvens som uppfyller regeln att antalet nollor mellan två ettor i sekvensen alltid är minst d och högst k. Exemplevis 01000100100 är en $(1, 3)$-RLL sekvens men inte en $(1, 2)$-RLL sekvens. I CD-skivor använder man $(2, 10)$-sekvenser. Antag att vi har en vanlig bitsekvens och vi skulle vilja översätta den till en $(1, 4)$-RLL sekvens. Tanken är att vi då stoppar in extra nollor eller ettor när de behövs. Konstruera tillståndsgrafen för den maskin som gör översättningen. Utsignalalfabetet är $\mathcal{Z} = \{0, 1, [10], [010]\}$.

Lösning: Först får vi komma fram till lämpligt beteende enligt regeln **1.** efter en 1, stoppa in extra 0. **2.** efter 0000, stoppa in extra 10.

Sedan fås tillståndsgrafen för kodaren.

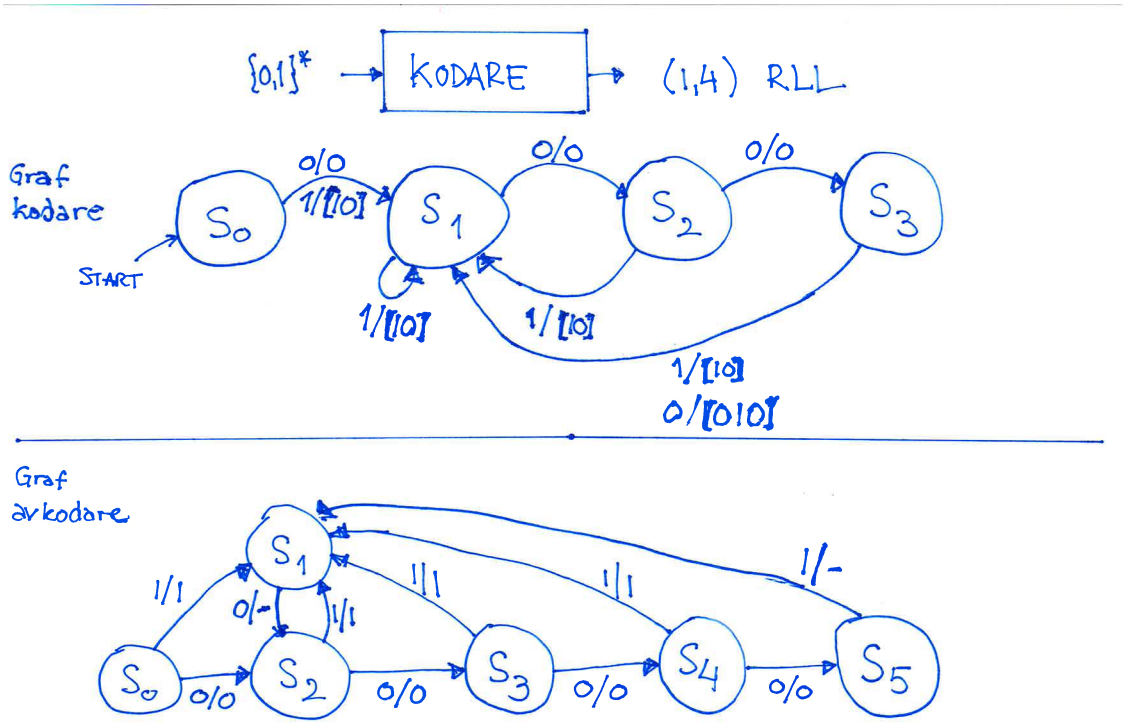

Figure 1: RLL exempel

En kodning kan då vara

$$10110000011... \rightarrow [10]0[10][10]00[010]00[10][10]...$$

Låt oss sedan göra motsvarande graf för avkodare, dvs, en maskin som tar en $(1, 4)$-RLL sekvens konstruerad enligt ovan regel, och levererar den bitsekvens som var insignalsekvens från början. Vi använder då utsignalalfabet $\mathcal{Z} = \{0, 1, -\}$, där $-$ innebär tomrum. Grafen är given ovan. Exempelvis

$$1000101000010001010... \rightarrow 1 - 01 - 1 - 000 - -001 - 1 - ....$$