

a)

1= true 0 = false

Not gate 1 blir til 0

|   |   |
|---|---|
| 0 | 1 |
| 1 | 0 |

And gate: Kun input 1 AND 1 = true

| Input | Output |
|-------|--------|
| 00    | 0      |
| 01    | 0      |
| 10    | 0      |
| 11    | 1      |

Or gate: 1 or 1 is true output is true

| Input | Output |
|-------|--------|
| 00    | 0      |
| 01    | 1      |
| 10    | 1      |
| 11    | 1      |

NOR gate = 1 or 1 is not true

| Input | Output |
|-------|--------|
| 00    | 1      |
| 01    | 0      |
| 10    | 0      |
| 11    | 0      |

**XOR 1 og 1 = true bortsett fra nå det er 11**

| Input | Output |
|-------|--------|
| 00    | 0      |
| 01    | 1      |
| 10    | 1      |
| 11    | 0      |

**b)**

Operasjonene den gjør med utgangspunkt i illustrasjonen er å lese, endre symbol, flytte den til høyre, flytte til venstre, endre state og stoppe.

**Tilstand 0, skrive blank,111B**

**flytter tape til høyre.**

**Neste tilstand 1. Symbol er 1, skriver derfor 0. Flytter tape til høyre.**

**110B**

**Går til tilstand 1, skriver 0. Flytter til høyre igjen. Går til tilstand 1.**

**100B**

**Går på nytt til tilstand 1, symbol er 1, skriver 0, flytter tape til høyre.**

**000B**

**Symbol blank, skriver 1, flytter til venstre.**

**1000B**

**Går over til tilstand 2, symbol er 0, skriver 0,**

**1000B**

**Flytter til venstre, skriver 0**

**1000B**

**Flytter til venstre, skriver 0**

**1000B**

**Flytter til venstre, skriver blank, stopper tilstand**

**1000B**

1000B står på tape ved stopp tilstand.

Leser: 9

Skriver : 9

Flytter til venstre: 4

Flytter til høyre: 4

Tilstandsendringer: 2

Stopper: 1

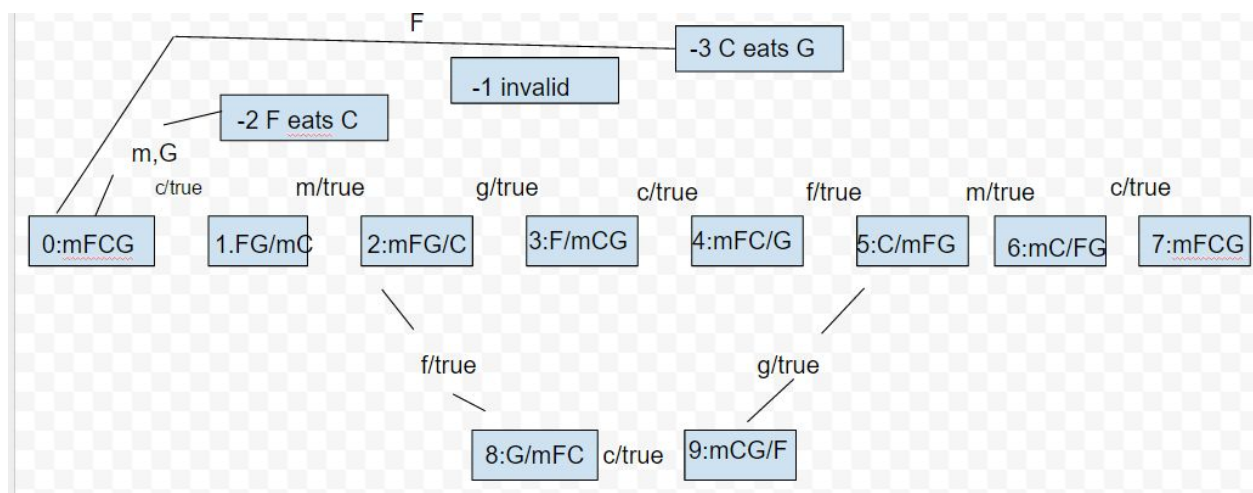
Operasjoner 29.

c)

Oppsummering av steg for å komme over elven.

1. Take the chicken across
2. Come back with the boat empty besides yourself
3. Take the grain (or the fox) across
4. Take the chicken back
5. Take the fox (or the grain) across
6. Come back with the boat empty besides yourself
7. Take the chicken across

Her ser du tilstandsdiagrammet vårt, med definerte states og inputs, og en mulighet til å gå en annen vei til å nå målet mellom state 2 og 5.



*Inputs som ikke er true går til false. For eksempel dersom tilstanden er 3 så er input g og c true, og F vil være false, da blir det tilstand -1 og output invalid.*

*“s3 i = C og G, O = mFCG/C og mFC/G true, IM false” definert eksempel. Mer definisjoner ligger under og baserer seg på language acceptor prinsippene fra lesestoffet for uke 09.*

**Language acceptor/elvekryssningsp**

**S = {0, -2, -1, -1, 1, 3, 4, 5, 6, 7, 8, 9}**

$I = \{C, mG, F, M, G\}$

$O = \text{true, false}$  (outputs kan være true eller false (invalid move) basert på hvilke inputs som anvendes i en viss state)

*starting point:*

$mFCG$  if  $s_0 I = c, mG$  og  $F$ .  $O(\text{valid}) = FG|mC$ , Fox eats Chicken og Chicken eats grain  
True, Invalid Move false input

if  $s_1 I = C$  og  $m$ ,  $O = mFGC|$  og  $mFG|C$  true, invalid move false

$s_2 I = F$  og  $G$ ,  $O = G|mFC$  og  $F|mCG$  true, IM false

$s_3 i = C$  og  $G$ ,  $O = mFGC|C$  og  $mFC|G$  true, IM false

$s_4 i = c$  og  $F$ ,  $O = F|mCG$  og  $C|MFG$  true, IM false

$s_5 i = G$  og  $m$ ,  $O = mCFG|F$  og  $mC|FG$  true, IM false

$s_6 i = m$  og  $C$ ,  $O = C|mFG$  og  $|mFCG$  true, IM false

$s_7 i = C$ ,  $O = MC|FG$  true, IM false, (victory state)

$s_8 I = F$  og  $C$ ,  $O = mFG|C$  og  $mCG|F$  true, IM false

$s_9 = C$  og  $G$ ,  $O = G|mFC$  og  $mCG|F$  true, IM false

If  $s_{-2}$   $O = \text{Fox eats Chicken, (game over)}$

If  $s_{-3}$   $O = \text{Chicken eats grain, (game over)}$

“if  $s_{-1}$   $o = \text{Invalid move, false input comitted}$ ”

**Forskjell med 4.1.3 og videoen:**

I videoen er det ikke brukt sensor. I videoen er løsningen prediktiv, har angitt states på forhånd. Prediktiv vil si at den beskriver hvordan omgivelsene fungerer. I 4.1.3 er løsningen analytisk, dvs at den beskriver handlingene som en kombinasjon av kontroll på systemet og området den kontrollerer. Input er en enkel beskjed til hele systemet, mens output måler systemet.

**Kilder:**

<http://www.dreamincode.net/forums/blog/91/entry-3332-beginner-series-finite-state-machine/>

[https://fronter.com/uia/links/files.phtml/1924884411\\$936476613\\$/Fagstoff/Uke+09/teori\\_uke09.pdf+\\_prcent\\_28les+introduksjon+++studer+n\\_prcent\\_F8ye+4.1.3\\_prcent\\_29](https://fronter.com/uia/links/files.phtml/1924884411$936476613$/Fagstoff/Uke+09/teori_uke09.pdf+_prcent_28les+introduksjon+++studer+n_prcent_F8ye+4.1.3_prcent_29)