

# Solution Guide for Mandatory Assignment 1

Mikkel Bundgaard

February 25, 2005

## Part 1

### Question

Write a CCS expression  $T(\text{start}, \text{stop}, \text{join})$  denoting the LTS:

### Solution

---

```
T          = Created
Created    = start.Alive + stop.Dead
Alive      = t.Alive + stop.Dead + t.Dead
Dead       = join.Dead
```

---

### Question

Write the expression in the syntax of Mobility Workbench, and verify that the syntax is correct.

### Solution

---

```
agent T(start, stop, join)      = Created<start, stop, join>
agent Created(start, stop, join) = start.Alive<start, stop, join> +
                                   stop.Dead<start, stop, join>
agent Alive(start, stop, join)  = t.Alive<start, stop, join> +
                                   stop.Dead<start, stop, join> + t.Dead<start, stop, join>
agent Dead(start, stop, join)   = join.Dead<start, stop, join>
```

---

## Part 2

### Question

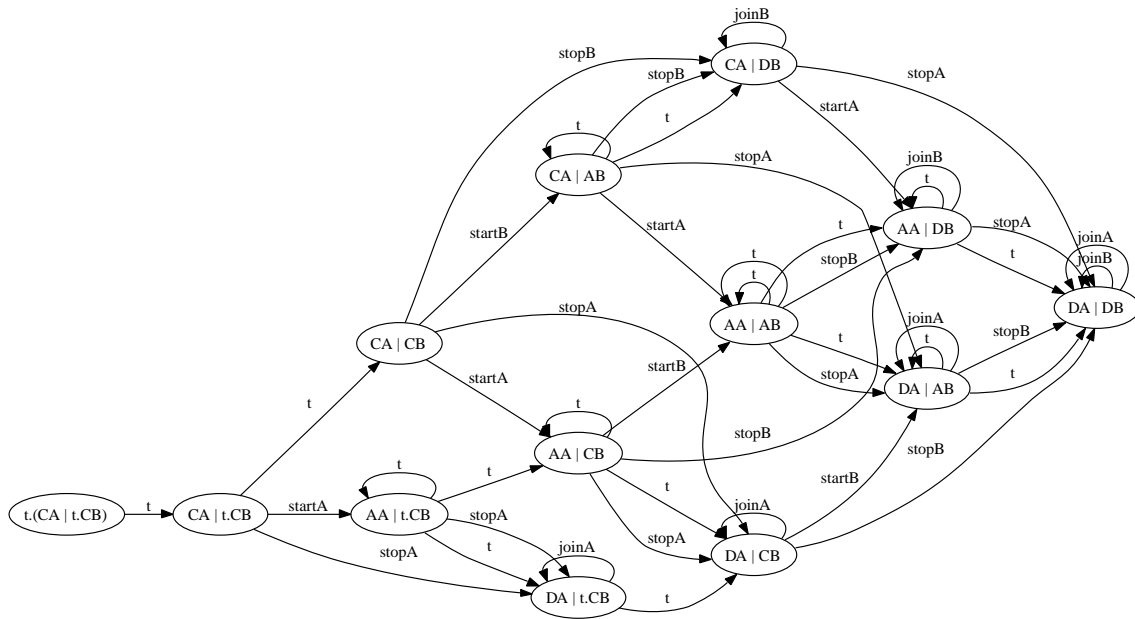
Draw the LTS for the CCS process

---

```
TwoThreads(startA, stopA, joinA, startB, stopB, joinB) =
t.(T<startA, stopA, joinA> | t.(T<startB, stopB, joinB>))
```

---

## Solution



## Question

How many states would the LTS have if 3 threads were created?

## Solution

- For 1 thread we have 4 states (including the initial state)
- For 2 threads we have 13 states (including the initial state) = 4 (1+3) states where thread B has not been created yet and 9 ( $3 \times 3$ ) states where both threads are created
- For 3 threads we have 40 states = 13 (1+3+3\*3) states where not all threads have been created and 27 ( $3 \times 3 \times 3$ ) states where all threads are created
- ...
- For  $k$  states the solution is  $\frac{3^{k+1}-1}{2}$

## Part 3

### Question

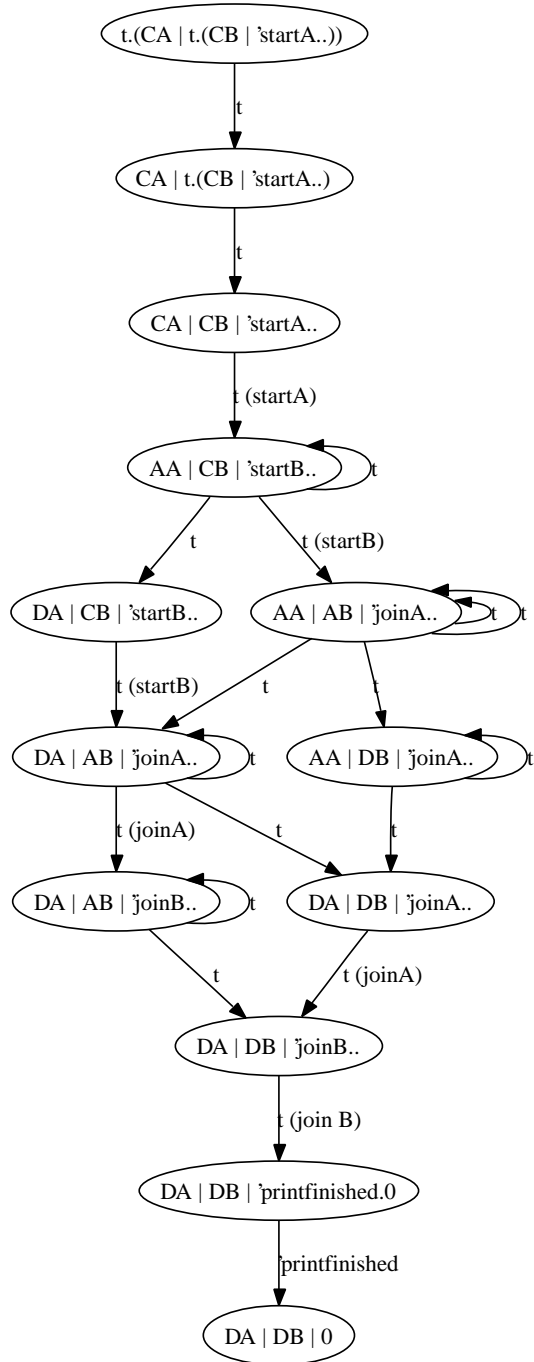
Draw the LTS for the CCS process

---

```
TwoThreads(printfinished) = (~startA, startB, joinA, joinB, stopA, stopB)
(t.(T<startA, stopA, joinA> | t.(T<startB, stopB, joinB> |
'startA.'startB.'joinA.'joinB.'printfinished.0)))
```

---

## Solution



## Question

Give a formal inference of two of the transitions of the process, at least one that uses the SYNC rule.

## Solution

We'll take the easy one first, and try to derive the single  $\tau$  transition of

---

```
(^startA, startB, joinA, joinB, stopA, stopB)
(t.(T<startA, stopA, joinA> | t.(T<startB, stopB, joinB> |
' startA.' startB.' joinA.' joinB.' printfinished.0)))
```

---

We let  $L = startA, startB, joinA, joinB, stopA, stopB$ ,  
let  $'startA.. = ' startA.' startB.' joinA.' joinB.' printfinished.0$ ,  
and let  $TA = T\langle startA, stopA, joinA \rangle$  and  $TB = T\langle startB, stopB, joinB \rangle$

$$\text{RES} \frac{\text{ACT} \frac{t.(TA | t.(TB | 'startA..)) \xrightarrow{\tau} TA | t.(TB | 'startA..)}{(\wedge L)(t.(TA | t.(TB | 'startA..))) \xrightarrow{\tau} (\wedge L)(TA | t.(TB | 'startA..))} (\tau \notin L)}$$

And the other inference that must use the SYNC rule.

---

```
(^startA, startB, joinA, joinB, stopA, stopB)
(Dead<startA, stopA, joinA> | Dead<startB, stopB, joinB> |
' joinB.' printfinished.0)
```

---

Again, we use the following shorthands  $L = startA, startB, joinA, joinB, stopA, stopB$ ,  
let  $'joinB.. = ' joinB.' printfinished.0$ ,  
and let  $DeadA = Dead\langle startA, stopA, joinA \rangle$  and  $DeadB = Dead\langle startB, stopB, joinB \rangle$

$$\begin{array}{c}
\text{ACT} \frac{}{} \\
\text{CON} \frac{\text{ACT} \frac{}{} \text{joinB}. \text{DeadB} \xrightarrow{\text{joinB}} \text{DeadB}}{} \\
\text{COM2} \frac{\text{CON} \frac{}{} \text{DeadB} \xrightarrow{\text{joinB}} \text{DeadB}}{\text{DeadA} \mid \text{DeadB} \xrightarrow{\text{joinB}} \text{DeadA} \mid \text{DeadB}} \quad \text{ACT} \frac{}{} \\
\text{SYNC} \frac{\text{COM2} \frac{}{} \text{DeadA} \mid \text{DeadB} \xrightarrow{\text{joinB}} \text{DeadA} \mid \text{DeadB} \quad \text{ACT} \frac{}{} \text{'joinB}.. \xrightarrow{\text{joinB}} \text{'printfinished}.0}}{\text{DeadA} \mid \text{DeadB} \mid \text{'joinB}.. \xrightarrow{\tau} \text{DeadA} \mid \text{DeadB} \mid \text{'printfinished}.0} \\
\text{RES} \frac{\text{SYNC} \frac{}{} \text{DeadA} \mid \text{DeadB} \mid \text{'joinB}.. \xrightarrow{\tau} \text{DeadA} \mid \text{DeadB} \mid \text{'printfinished}.0}}{(\hat{L})(\text{DeadA} \mid \text{DeadB} \mid \text{'joinB}..) \xrightarrow{\tau} (\hat{L})(\text{DeadA} \mid \text{DeadB} \mid \text{'printfinished}.0)} \quad (\tau \notin L)
\end{array}$$