Murphi-Class2

June 2, 2018

DeadLock

No next state

- no rule to execute
- have one rule

Running Example: Mutual Exclusion Protocol

N symmetric processors, behaviour of processor i is described by:

- $try(i) := a[i] = I \to a[i]' = T$
- $crit(i) := (a[i] = T \wedge x = true \to a[i]' = C \wedge x' = false)$
- $exit(i) := a[i] = C \to a[i]' = E$
- $idle(i) := a[i] = E \to a[i]' = I \wedge x' = true$

Initial states: $x = true$ and $a[i] = I$ for all i

Invariant property (where we assume parameters are pairwise disjoint):
$\neg(a[i] = C \wedge a[j] = C)$

Reachable state set

T

he set of reachable states for a protocol $\mathcal{P} = (I, R)$, denoted as $\mathrm{RS}(\mathcal{P})$, can be defined inductively:

- a state s is in $\mathrm{RS}(\mathcal{P})$ if there exists a formula $f \in I$ such that $s \models f$;
- a state s' is in $\mathrm{RS}(\mathcal{P})$ if there exists a state s and a guarded command $r \in R$ such that $s \in \mathrm{RS}(\mathcal{P})$ and $s \xrightarrow{r} s'$.

Important properties–Safety Properties

- Bad things never happen $\Box P$.
- Invariants properties of a protocol: mutual exclusion $\neg(a[i] = C \wedge a[j] = C)$
- Data Coherence: $(ExGntd = false \rightarrow MemData = AuxData)$ $\forall i \in NODE.Cache[i].State! = I \rightarrow Cache[i].Data = AuxDataend;$
- No deadLock.

Important properties–Liveness Properties

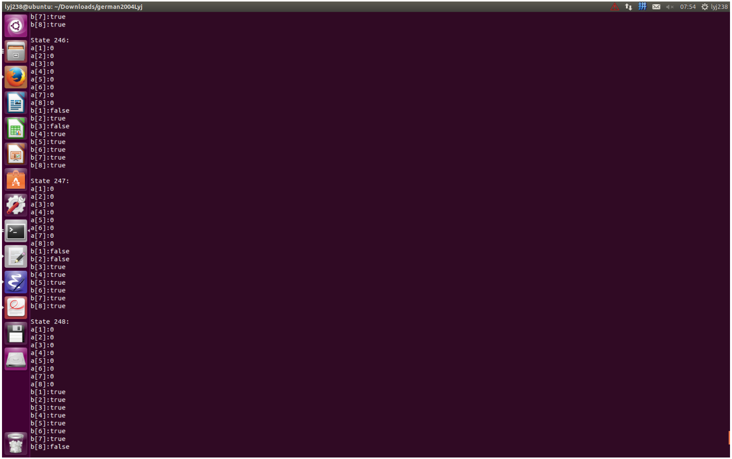
- Good things eventually happen
- A request eventually is served $\Box(P \rightarrow \Diamond Q)$
- A process is eventually scheduled
- A Loop is terminated

Use Murphi to Compute Reachable state set

./mutualEx -ta -d ./

Use python to create the table from the trace

Output Result



```
b[7]:true
b[8]:true

State 246:
a[1]:0
a[2]:0
a[3]:0
a[4]:0
a[5]:0
a[6]:0
a[7]:0
a[8]:0
b[1]:false
b[2]:true
b[3]:false
b[4]:true
b[5]:true
b[6]:true
b[7]:true
b[8]:true

State 247:
a[1]:0
a[2]:0
a[3]:0
a[4]:0
a[5]:0
a[6]:0
a[7]:0
a[8]:0
b[1]:false
b[2]:false
b[3]:true
b[4]:true
b[5]:true
b[6]:true
b[7]:true
b[8]:true

State 248:
a[1]:0
a[2]:0
a[3]:0
a[4]:0
a[5]:0
a[6]:0
a[7]:0
a[8]:0
b[1]:true
b[2]:true
b[3]:true
b[4]:true
b[5]:true
b[6]:true
b[7]:true
b[8]:false
```

A Table to illustrate a reachable state set

Table: a data table transformed from reachable state set

n[1]	n[2]	x
I	I	TRUE
T	I	TRUE
I	T	TRUE
C	I	FALSE
T	T	TRUE
I	C	FALSE
E	I	FALSE
C	T	FALSE
T	C	FALSE
I	E	FALSE
E	T	FALSE
T	E	FALSE