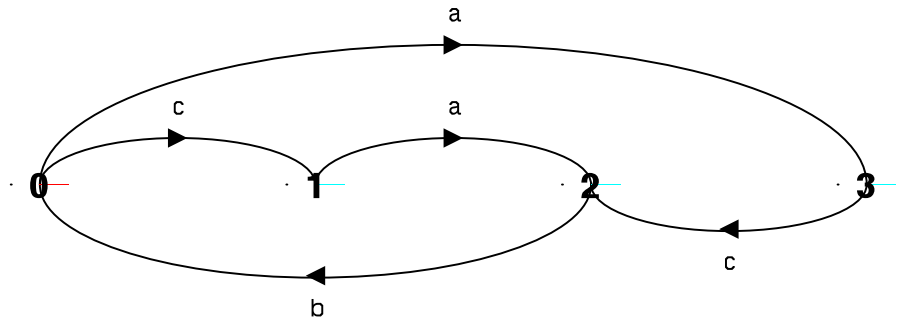
Chapter 3 - solutions

3.1 Both S1 & S2 have the same LTS although S2 must be minimised.



3.2

```
ELEMENT=(up->down->ELEMENT).

||FOUR = ({a,b,c,d}:ELEMENT)
 /{a.down/b.up,b.down/c.up,c.down/d.up,
   up/a.up, down/d.down}
 @{up,down}.

Or more generally, for different values of N

const N =4
ELEMENT=(up->down->ELEMENT).
||FOUR = (el[1..N]:ELEMENT)
         /{el[i:1..N-1].down/el[i+1].up,
           up/el[1].up, down/el[N].down}
         @{up,down}.
```

3.3

```
CLIENT = (call->wait->continue->CLIENT).
SERVER = (request->service->reply->SERVER).
||CLIENT_SERVER = (a:CLIENT || b:CLIENT
                   || {a,b}::SERVER)
            /{a.call/a.request, a.reply/a.wait,
              b.call/b.request, b.reply/b.wait ,
              service/{a.service,b.service}}.
```

Or, using relational relabeling

```
CLIENT = (call->wait->continue->CLIENT).
SERVER = (request->service->reply->SERVER).
||CLIENT_SERVER = (a:CLIENT || b:CLIENT
                    || SERVER)
           /{ {a.call,b.call}/request,
              {a.wait,b.wait}/reply}.
```

3.4

```
CLIENT = (call->(wait->continue->CLIENT
                |timeout->CLIENT)).
```

As a result, the system deadlocks:

```
Trace to DEADLOCK:
      a.call
      a.timeout
      a.service
```