

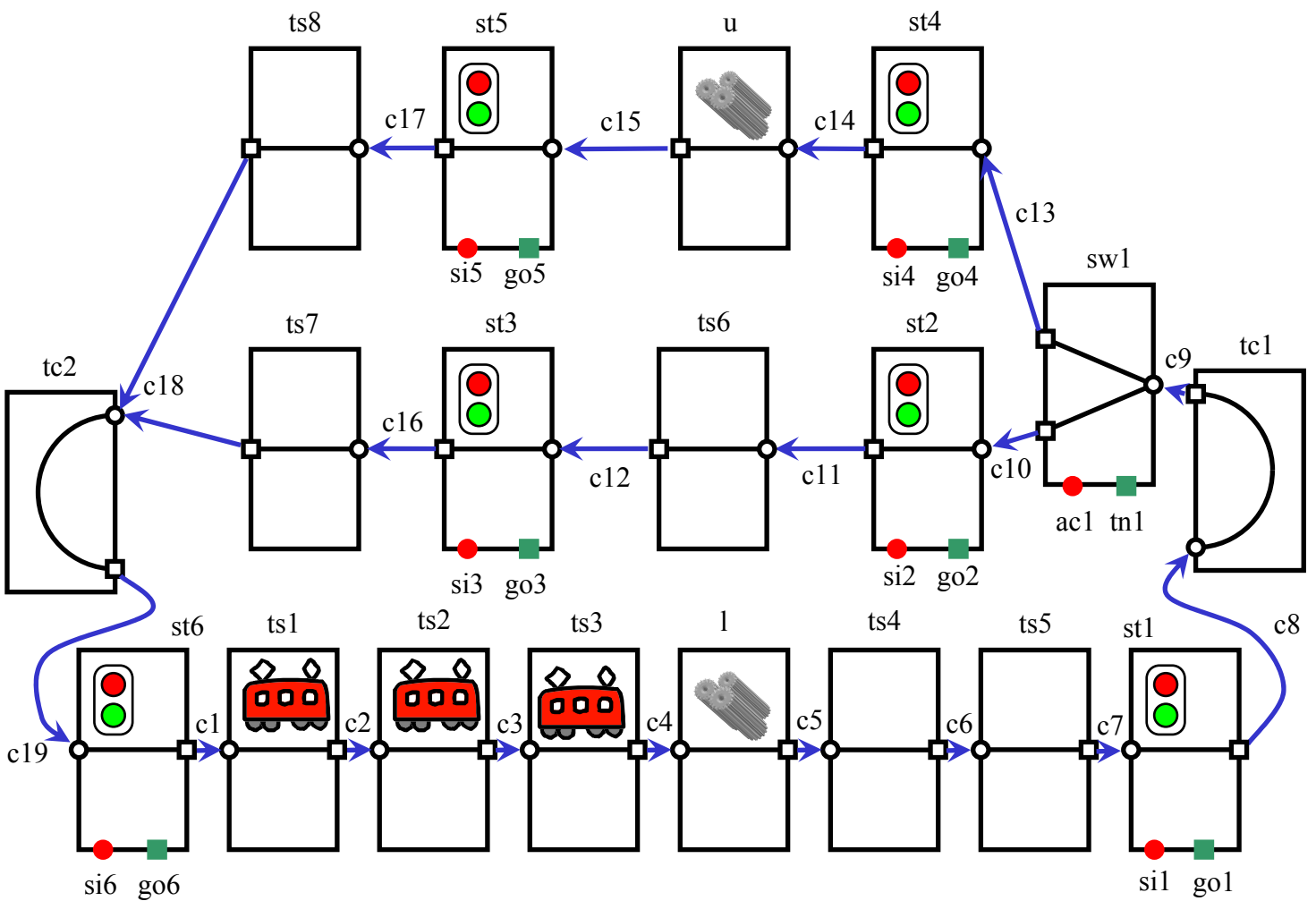
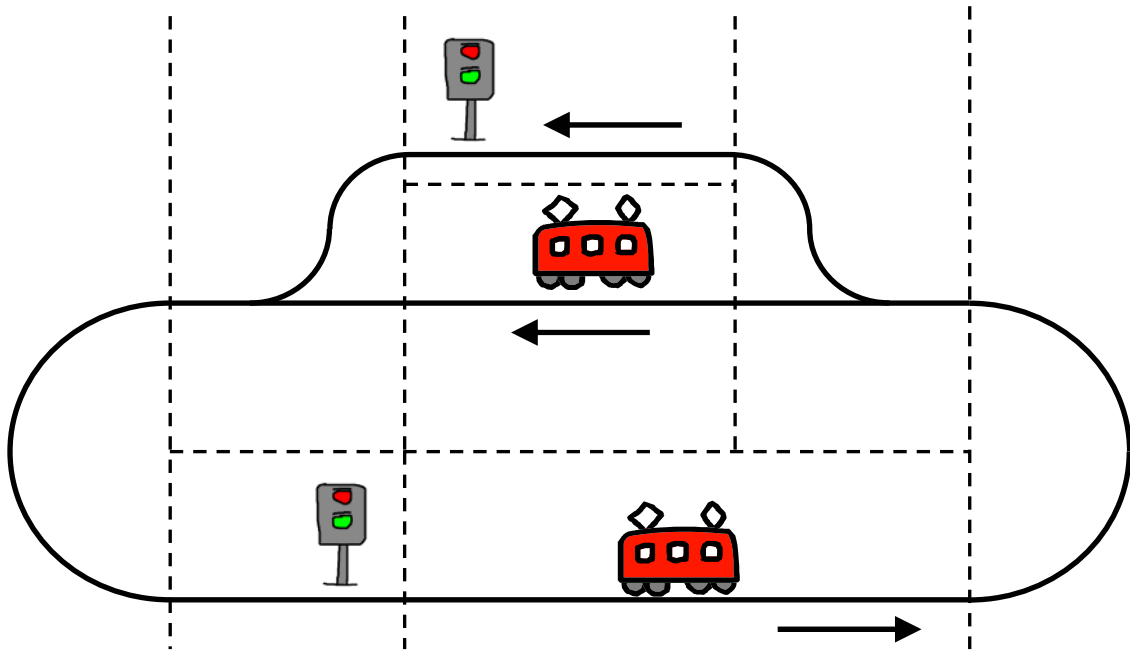
Beispiel zur Lehrveranstaltung

Modelchecking

WS 2003/04

Ekkart Kindler
Universität Paderborn

MonTrac-Anlage und Steuerung



```

MODULE track_s(start,end,initial)

VAR
  shuttle : { none, loaded, unloaded };

ASSIGN
  init(shuttle):= initial;
  init(start):= none;

  next(shuttle):=
    case
      shuttle = none & start != none : start;
      shuttle != none & end = none   : none;
      1                               : shuttle;
    esac;

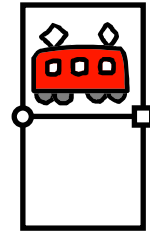
  next(start):=
    case
      shuttle = none & start != none : none;
      1                               : start;
    esac;

  next(end):=
    case
      shuttle != none & end = none : shuttle;
      1                             : end;
    esac;

FAIRNESS running

SPEC AG AF shuttle = none;
SPEC AG EF shuttle != none;

```



```

MODULE track_c(start,end)

VAR
  shuttle : { none, loaded, unloaded };

ASSIGN
  init(shuttle):= none;
  init(start):= none;

  next(shuttle):=
    case
      shuttle = none & start != none   : start;
      shuttle != none & end = none     : none;
      1                                  : shuttle;
    esac;

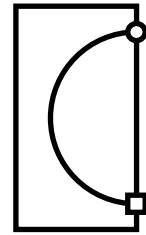
  next(start):=
    case
      shuttle = none & start != none : none;
      1                               : start;
    esac;

  next(end):=
    case
      shuttle != none & end = none : shuttle;
      1                             : end;
    esac;

FAIRNESS running

SPEC AG AF shuttle = none;
SPEC AG EF shuttle != none;
SPEC AG ( start!= none  -> shuttle = none );

```



```
MODULE switch(start,end_l,end_r,turn,ack)
```

```
VAR
```

```
  shuttle  : { none, loaded, unloaded };  
  direction : { no, left, right };
```

```
ASSIGN
```

```
  init(start) := none;  
  init(shuttle) := none;  
  init(direction) := left;  
  init(ack) := left;
```

```
  next(shuttle) :=
```

```
    case  
      shuttle = none & start != none           : start;  
      shuttle != none & end_l = none & direction = left : none;  
      shuttle != none & end_r = none & direction = right : none;  
      1                                             : shuttle;  
    esac;
```

```
  next(start) :=
```

```
    case  
      shuttle = none & start != none           : none;  
      1                                         : start;  
    esac;
```

```
  next(end_l) :=
```

```
    case  
      shuttle != none & end_l = none & direction = left : shuttle;  
      1                                                  : end_l;  
    esac;
```

```
  next(end_r) :=
```

```
    case  
      shuttle != none & end_r = none & direction = right : shuttle;  
      1                                                  : end_r;  
    esac;
```

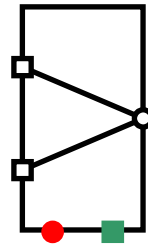
```
  next(direction) :=
```

```
    case  
      turn != no : turn;  
      1          : direction;  
    esac;
```

```
  next(ack) := direction;
```

```
FAIRNESS running
```

```
SPEC AG AF shuttle = none;  
SPEC AG EF shuttle != none;  
SPEC AG direction != no;  
SPEC AG ( turn != no -> shuttle = none );
```



```
MODULE stopper(start,end,status,go)
```

```
VAR
```

```
  shuttle   : { none, loaded, unloaded };
```

```
ASSIGN
```

```
  init(shuttle) := none;
```

```
  init(status) := none;
```

```
  init(start) := none;
```

```
  next(shuttle) :=
```

```
    case
```

```
      shuttle = none & start != none   : start;
```

```
      shuttle != none & end = none & go : none;
```

```
      1                                  : shuttle;
```

```
    esac;
```

```
  next(status) :=
```

```
    case
```

```
      shuttle = none & start != none   : start;
```

```
      shuttle != none & end = none & go : none;
```

```
      1                                  : shuttle;
```

```
    esac;
```

```
  next(start) :=
```

```
    case
```

```
      shuttle = none & start != none   : none;
```

```
      1                                  : start;
```

```
    esac;
```

```
  next(end) :=
```

```
    case
```

```
      shuttle != none & end = none & go : shuttle;
```

```
      1                                  : end;
```

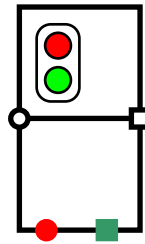
```
    esac;
```

```
FAIRNESS running
```

```
SPEC AG AF shuttle = none;
```

```
SPEC AG EF shuttle != none;
```

```
SPEC AG ( ( shuttle = none & go ) -> A[ shuttle = none U !go] )
```



```

MODULE load(start,end)

VAR
  shuttle    : { none, loaded, unloaded };

ASSIGN
  init(shuttle) := none;
  init(start) := none;

  next(shuttle) :=
    case
      shuttle = none & start != none : start;
      shuttle != none & end = none    : none;
      1                                : shuttle;
    esac;

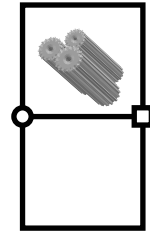
  next(start) :=
    case
      shuttle = none & start != none : none;
      1                                : start;
    esac;

  next(end) :=
    case
      shuttle != none & end = none    : { loaded, unloaded };
      1                                : end;
    esac;

FAIRNESS running

SPEC AG AF shuttle = none;
SPEC AG EF shuttle != none;
SPEC AG shuttle != loaded;

```



```

MODULE unload(start,end)

VAR
  shuttle   : { none, loaded, unloaded };

ASSIGN
  init(shuttle) := none;
  init(start) := none;

  next(shuttle) :=
    case
      shuttle = none & start != none : start;
      shuttle != none & end = none    : none;
      1                                : shuttle;
    esac;

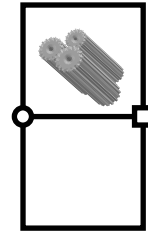
  next(start) :=
    case
      shuttle = none & start != none : none;
      1                                : start;
    esac;

  next(end) :=
    case
      shuttle != none & end = none    : unloaded;
      1                                : end;
    esac;

FAIRNESS running

SPEC AG AF shuttle = none;
SPEC AG EF shuttle != none;
SPEC AG shuttle != unloaded;

```



```
MODULE control(si1,go1,si2,go2,si3,go3,si4,go4,si5,go5,si6,go6,tn1,ac1)
```

```
VAR
```

```
  statel : { idle, switching, leaving, pending, arriving };
```

```
  state2 : { idle, leaving_l, leaving_r, pending, arriving };
```

```
  prio   : { left, right };
```

```
ASSIGN
```

```
  init(go1) := 0;
```

```
  init(go2) := 0;
```

```
  init(go3) := 0;
```

```
  init(go4) := 0;
```

```
  init(go5) := 0;
```

```
  init(go6) := 0;
```

```
  init(tn1) := no;
```

```
  init(statel) := idle;
```

```
  init(state2) := idle;
```

```
-----  
next(statel) :=
```

```
  case
```

```
    statel = idle      & si1 != none           : switching;
```

```
    statel = switching & ac1 = tn1             : leaving;
```

```
    statel = leaving  & si1 = none            : pending;
```

```
    statel = pending  & ( si2 != none | si4 != none ) : arriving;
```

```
    statel = arriving & ( si2 = none & si4 = none ) : idle;
```

```
    1                                                         : statel;
```

```
  esac;
```

```
next(tn1) :=
```

```
  case
```

```
    statel = idle      & si1 = loaded   : right;
```

```
    statel = idle      & si1 = unloaded : left;
```

```
    statel = switching & ac1 = tn1     : no;
```

```
    1                                                         : tn1;
```

```
  esac;
```

```
next(go1) :=
```

```
  case
```

```
    statel = switching & ac1 = tn1 : 1;
```

```
    statel = leaving  & si1 = none : 0;
```

```
    1                                                         : go1;
```

```
  esac;
```

```
next(go2) :=
```

```
  case
```

```
    si2 != none : 1;
```

```
    si2 = none  : 0;
```

```
  esac;
```

```
next(go4) :=
```

```
  case
```

```
    si4 != none : 1;
```

```
    si4 = none  : 0;
```

```
  esac;
```

```
-----
```

```

next(state2) :=
  case
    state2 = idle      & si3 != none & prio = left  : leaving_l;
    state2 = idle      & si3 != none & si5 = none   : leaving_l;
    state2 = idle      & si5 != none & prio = right : leaving_r;
    state2 = idle      & si5 != none & si3 = none   : leaving_r;
    state2 = leaving_l & si3 = none                  : pending;
    state2 = leaving_r & si5 = none                  : pending;
    state2 = pending   & si6 != none                 : arriving;
    state2 = arriving  & si6 = none                  : idle;
    1                                                           : state2;
  esac;

```

```

next(prio) :=
  case
    state2 = leaving_l : right;
    state2 = leaving_r : left;
    1                   : prio;
  esac;

```

```

next(go3) :=
  case
    state2 = idle      & si3 != none & prio = left  : 1;
    state2 = idle      & si3 != none & si5 = none   : 1;
    state2 = leaving_l & si3 = none                 : 0;
    1                                                           : go3;
  esac;

```

```

next(go5) :=
  case
    state2 = idle      & si5 != none & prio = right : 1;
    state2 = idle      & si5 != none & si3 = none   : 1;
    state2 = leaving_r & si5 = none                 : 0;
    1                                                           : go5;
  esac;

```

```

next(go6) :=
  case
    si6 != none          : 1;
    si6 = none           : 0;
  esac;

```

MODULE main

VAR

```
ctrl: control(si1,go1,si2,go2,si3,go3,si4,go4,si5,go5,si6,go6,tn1,ac1);

c1 : { none, loaded, unloaded };
ts1 : process track_s(c1,c2,unloaded);

c2 : { none, loaded, unloaded };
ts2 : process track_s(c2,c3,unloaded);

c3 : { none, loaded, unloaded };
ts3 : process track_s(c3,c4,unloaded);

c4 : { none, loaded, unloaded };
l : process load(c4,c5);

c5 : { none, loaded, unloaded };
ts4 : process track_s(c5,c6,none);

c6 : { none, loaded, unloaded };
ts5 : process track_s(c6,c7,none);

c7 : { none, loaded, unloaded };
si1 : { none, loaded, unloaded };
go1 : boolean;
st1 : process stopper(c7,c8,si1,go1);

c8 : { none, loaded, unloaded };
tc1 : process track_c(c8,c9);

c9 : { none, loaded, unloaded };
tn1 : { no, left, right };
ac1 : { no, left, right };
sw1 : process switch(c9,c10,c13,tn1,ac1);

c10 : { none, loaded, unloaded };
si2 : { none, loaded, unloaded };
go2 : boolean;
st2 : process stopper(c10,c11,si2,go2);

c11 : { none, loaded, unloaded };
ts6 : process track_s(c11,c12,none);

c12 : { none, loaded, unloaded };
si3 : { none, loaded, unloaded };
go3 : boolean;
st3 : process stopper(c12,c16,si3,go3);

c13 : { none, loaded, unloaded };
si4 : { none, loaded, unloaded };
go4 : boolean;
st4 : process stopper(c13,c14,si4,go4);

c14 : { none, loaded, unloaded };
u6 : process unload(c14,c15);

c15 : { none, loaded, unloaded };
si5 : { none, loaded, unloaded };
go5 : boolean;
st5 : process stopper(c15,c17,si5,go5);
```

```
c16 : { none, loaded, unloaded };
ts7 : process track_s(c16,c18,none);

c17 : { none, loaded, unloaded };
ts8 : process track_s(c17,c18,none);

c18 : { none, loaded, unloaded };
tc2 : process track_c(c18,c19);

c19 : { none, loaded, unloaded };
si6 : { none, loaded, unloaded };
go6 : boolean;
st6 : process stopper(c19,c1,si6,go6);
```

```
DEFINE
  running:= 1;
```