

Aufgabe 2b. Parallele Spezifikationen

Trafficlightprotokoll_03

% Definition der Richtungen

sort

CardinalDirection = struct north | east | south | west;

% Definition der Farben

sort

Colour = struct red | yellow | green;

map

nextColour: Colour -> Colour; % Der Wert des nextColour als Colour deklariert

% Initialisierung die Werte der deklarierten Colourstypen

eqn

nextColour(red) = green;

nextColour(green) = yellow;

nextColour(yellow) = red;

map

isCrossingSafe: Colour # Colour # Colour # Colour -> Bool; % Der Wert des isCrossingSafe als Boolean deklariert

var

n,e,s,w : Colour;

% isCrossingSafe gibt boolean true zurück, wenn alle die gegebene Bedingungen erfüllt sind

eqn

isCrossingSafe(n,e,s,w) = if(((n == red && w == red && s == red && e == red) || (n == red && w == green && s == red && e == green) || (n == red && w == yellow && s == red && e == yellow) || (n == red && w == green && s == red && e == yellow) || (n == red && w == yellow && s == red && e == green) || (n == green && w == red && s == green && e == red) || (n == yellow && w == red && s == yellow && e == red) || (n == green && w == red && s == yellow && e == red) || (n == yellow && w == red && s == green && e == red) || (n == red && w == red && s == red && e == green) || (n == red && w == red && s == red && e == yellow) || (n == red && w == red && s == green && e == red) || (n == red && w == red && s == yellow && e == red) || (n == red && w == green && s == red && e == red) || (n == red && w == yellow && s == red && e == red) || (n == green && w == red && s == red && e == red) || (n == yellow && w == red && s == red && e == red))),true,false);

% Definition der Aktionen und Prozesse

act

show: CardinalDirection # Colour;

seeColour: CardinalDirection # Colour;

colourSeen: CardinalDirection # Colour;

% Es wird immer durch eine Farbe die nächste Farbe gebracht

proc

TrafficLight(dir : CardinalDirection, colour : Colour) =
show(dir,colour).TrafficLight(colour = nextColour(colour));

% Der Monitor im Zentrum guckt in vier Richtungen und verhindert die unsichere Kombinationen von Ampelwerten

```
Monitor(n: Colour, e: Colour, s:Colour, w:Colour) =sum dir:CardinalDirection.sum
colour:Colour.(dir == north)-> (isCrossingSafe(colour,e,s,w) ->
(seeColour(dir,colour).Monitor(n = colour))) <>(dir == east)->
(isCrossingSafe(n,colour,s,w) -> (seeColour(dir,colour).Monitor(e = colour))) <>(dir ==
south) -> (isCrossingSafe(n,e,colour,w) -> (seeColour(dir,colour).Monitor(s = colour)))
<>(dir == west)-> (isCrossingSafe(n,e,s,colour) -> (seeColour(dir,colour).Monitor(w =
colour))));
```

% Allow nur die Aktion colourseen

% Die Aktionen show und seeColour werden zusammen verbunden,und in multi-Prozesse TrafficLight mit der Aktion colourseen kommunizieren

% TrafficLight Prozesse laufen parallel, und werden auf Red initialisiert

% Monitor wird auch auf red initialisiert

```
CrossRoad =allow({colourSeen},comm({show|seeColour ->
colourSeen},{TrafficLight(north, red) || TrafficLight(east, red) || TrafficLight(south, red) ||
TrafficLight(west, red) || Monitor(red,red,red,red)}));
```

% Es wird mit dem Prozess CrossRoad begonnen

init

CrossRoad;

Beschreibung der Probleme :

1. Die Ampelfarbe muss in alle Richtungen durch den Monitor überprüft werden.
2. Der Monitor verhindert die unsichere Kombinationen von Ampelwerten.
3. Allow nur das Prozess colourseen.
4. Die Prozesse show und seeColour werden in multi-Prozesse TrafficLight mit dem Prozess colourseen kommunizieren.

Lösungen der Probleme :

1. A- Eqn $\text{isCrossingSafe}(n,e,s,w)$ = Alle die zulässige Bedingungen; Es wird mit eqn $\text{isCrossingSafe}(n,e,s,w)$ überprüft, ob alle die Ampelfarben in alle Richtungen die gegebene Bedingungen erfüllen oder nicht. $\text{Monitor}(n: \text{Colour}, e: \text{Colour}, s: \text{Colour}, w: \text{Colour}) = \sum \text{dir:CardinalDirection.} \sum \text{colour:Colour.} (\text{dir} == \text{north}) \rightarrow (\dots);$.Prozess Monitor erhält vier Colourtypen für alle die Richtungen in seinem Constructor.
2. $\text{Monitor}(n: \text{Colour}, e: \text{Colour}, s: \text{Colour}, w: \text{Colour}) = \sum \text{dir:CardinalDirection.} \sum \text{colour:Colour.} (\text{dir} == \text{north}) \rightarrow (\text{isCrossingSafe}(\text{colour}, e, s, w) \rightarrow (\text{seeColour}(\text{dir}, \text{colour}). \text{Monitor}(n = \text{colour}))) \langle \rangle$ so weiter für alle die andere Richtungen; .Alle die Richtungen werden durch den Monitor überprüft, dann wird durch die Funktion $\text{isCrossingSafe}(\text{colour}, e, s, w)$ überprüft, ob die Kombination der Ampelfarben sicher ist oder nicht, wenn nicht geht das Prozess nicht weiter nach der Aktion $\text{seeColour}()$.
3. $\text{allow}\{\{\text{colourSeen}\}\}$ nur die Aktionen colourseen ist zulässig.
4. $\text{comm}\{\{\text{show} \mid \text{seeColour} \rightarrow \text{colourSeen}\}\}$