Beweren en Bewijzen Leertaak 8

16 april 2017

Opgave 1

a) Het juiste antwoord is A.

Toelichting:

Jack is getrouwd George is niet getrouwd Anne weten we niet van of zij getrouwd is of niet, dus er zijn twee scenario's namelijk:

Anne is getrouwd: Anne kijkt naar George, volgens de beschrijving, dit zou betekenen dat er een getrouwd persoon naar een ongetrouwd persoon kijkt omdat George niet getrouwd is.

Anne is niet getrouwd: Jack kijkt naar Anne, volgens de beschrijving, dit zou betekenen dat er een getrouwd persoon naar een ongetrouwd persoon kijkt omdate Anne nu dus niet getrouwd is.

In beide gevallen kijkt er een getrouwd persoon naar een ongetrouwd persoon dus het antwoord is A, Ja.

b) Het juiste antwoord is B.

Opgave 2

a) Dit zijn de gevraagde functies en predikaten.

```
(* Functies *)
Variable bestemmingsPlan (* x *) : PN -> BT.
  (* Bij perceelnummer x hoort een bestemmingsplan *)

Variable perceelBijBouw (* x *) : BP -> PN .
  (* Bouwplan x gaat over perceelnummer y *)

Variable gewenstBebouwing (* x *) : BP -> BT.
  (* Bij bouwplan x hoort bouwtype y *)


(* Predikaten *)
Variable isGebouw (* x *) : PN -> Prop .
  (* Er staat een gebouw op perceelnummer x *)
  (* Opzoeken in de administratie van de gemeente Nijmegen *)

Variable verbouwing (* x *) : BP -> Prop .
  (* Bouwplan x betreft een verbouwing *)
  (* Kijken in bouwplan x *)
```

b) Dit zijn de gevraagde hulppredikaten:

```
Definition Past (bp:BP) :=
    gewensteBebouwing bp
  =
    bestemmingsPlan (perceelBijBouw bp)
.

Definition vergunningWordtAfgegeven (bp:BP) :=
    Past bp
  \/
    (
        isGebouw (perceelBijBouw bp)
      /\
       verbouwing bp
  )
.
```

c) De eindige verzameling van bouwtypes kunnen we in Coq zo definiëren:

```
Inductive BT' : Set := Wonen | Bedrijventerrein | Horeca | Groenvoorziening.
```

Opgave 3

a) Ik kreeg als icoontje het vinkje.

b) Dit is de definitie van `magBewegen`:

```
Definition magBewegen (t:T) :=
(* Het systeem mag bewegen op tijdstip t als er t1 seconden geleden op de
   veiligheidsknop gedrukt is en er in het interval tussen t1 seconden
   geleden en t geen obstructie is en geen noodstop is ingedrukt. *)

exists t1:T,
        (t1 < t)
    /\
        isVeiligknop t1
    /\
        (
            forall d:T,
                    (d > t1)
                /\
                    (d < t)
                ->
                    ~noodstop d
        )

.
```

Toelichting: Als er een tijdstip t1 bestaat die kleiner is dan t en waarop de veiligknop is ingedrukt en er in de tussentijd niet op de noodstop is gedrukt dan mag het systeem bewegen.

c) Dit is de definitie van `allesBereikbaar`:

```
Definition allesBereikbaar :=
(* Als er op een knop voor een kast gedrukt wordt en er mag
```

```
   30 seconden bewogen worden dan is 30 seconden later bij de
   bijbehorende kast ruimte aanwezig. *)
forall s:S,
    forall t:T,
            knopGedrukt t s
        /\
            (
                forall d:T,
                        (d > t)
                    /\
                        (d <= t+30)
                    ->
                        ~noodstop d
            )
        ->
            ruimteAanwezig (t+30) (knopKast s)

.
```

Toelichting: Als er op een bepaald moment op de knop is gedrukt en er in 30 seconden niet op de noodstop wordt gedrukt dan is er 30 seconden later ruimte aanwezig bij de kast die bij de knop hoort.

d) Dit is de specificatie van het geheel, genaamd `diepvriespakhuis`:

```
Definition diepvriespakhuis :=
forall t:T,
    forall s:S,
            knopGedrukt t s
        ->
            (
                exists d:T,
                        (d>t)
                    /\
                        ruimteAanwezig d (knopKast s)
            )
.
```

Toelichting: Als er op een bepaald moment op de knop is gedrukt dan bestaat er een moment waarop er ruimte aanwezig is bij de kast die bij de knop hoort.