

# ToFoLo2 Project 3: Informatica

Benjamin De Leeuw

*Departement Zuivere Wiskunde en Computer Algebra*  
Galglaan 2, B-9000 Gent  
Benjamin.DeLeeuw@UGent.be  
Gent University, Belgium

December 15, 2006

## 1 Inleiding

Dit project gaat over Hoare Calculus en Prolog, zoals we beide hebben besproken in de oefeningenlessen. Er zijn twee opdrachten: een Hoare Calculus bewijs, en een prolog programma.

## 2 Oplossing

De oplossing van *Oefening Hoare* bedraagt een .pdf file met daarin het Hoare Calculus bewijs. Gebruik LaTeX<sup>1</sup> om de .pdf file te compileren. Het bewijs bestaat uit twee delen:

1. Een Hoare Calculus bewijs van het gestelde, waarbij alle precondities en postcondities van de programmaregels, in volgorde worden verbonden door Hoare Calculus regels. Vermeld bij elke stap de gebruikte regel (uit het op Minerva ter beschikking gestelde document “Hoare Calculus en Natuurlijke Deductie”, Hoare.pdf). Implicatiebewijzen worden genummerd, maar hier niet bewezen (zie 2.).
2. De bewijzen van de genummerde implicatieregels (uit 1.). De gehanteerde stijl voor deze bewijzen is bij voorkeur de natuurlijke deductiestijl van Fitch. Eventueel mag ook een calculatonele stijl (waarbij zinnen of termen met respectievelijk equivalenties of gelijkheid aan elkaar worden gekoppeld tot een bewijs, zie ToFoLo1) van bewijzen gebruikt worden. Wees steeds zo duidelijk mogelijk. Indien het bewijs onverstaanbaar of onduidelijk is, zullen geen punten worden toegekend voor dit deel van de oefening.

De oplossing van *Oefening Prolog* bedraagt een prolog file: *Exercise Prolog.pl*. Zip alle files in een bestand met als naam *GroepXX\_P3.zip*, met *XX* uw groepsnummer op *Minerva*. Indienen gebeurt via de *Dropbox* van *Minerva*.

## 3 Opgaven

### 3.1 Oefening Hoare

- Een specificatie van een functie  $n!$  :

---

<sup>1</sup>Voor de installatie en het gebruik van LaTeX, verwijzen we naar de website <http://www.latex-project.org/>.

$$1! = 1$$

$$k! = (k-1)! \cdot k$$

Volgende programma implementeert deze functie, bewijs deze claim in Hoare Calculus:

```

natural faculteit(natural n)
{
  natural fac := 1;
  natural count := n;
  while(count > 0)
  {
    fac := fac * count;
    count := count - 1;
  }
  return fac;
}

```

Zoek een invariant voor de while lus (inspiratie kan gevonden worden in de voorbeelden uit de oefeningenles van 06-12-2006). Zet het programma om naar de taal, zoals gehanteerd in het document “Hoare Calculus en Natuurlijke Deductie”, Hoare.pdf, en bewijs dat de postconditie van het programma  $\{fac = n!\}$  bedraagt, onder preconditie  $\{n > 0\}$ .

### 3.2 Oefening Prolog

Een prolog feitendatabank wordt met volgende feiten geïnitialiseerd:

- werelden: *werelden*([*w1*, *w2*, ..., *wn*]).
- toegangsrelaties: *relaties*([(*w1*, *w2*), ..., (*w2*, *w1*)]).
- valuaties: *valuaties*([(*w1*, [*p*, *q*, *r*]), (*w2*, [*p*]), ..., (*wn*, [*q*, *r*])]).

De *werelden* zijn gespecificeerd als een lijst van *atomen*. Het *relaties* predikaat bestaat uit een lijst van koppels van werelden. Het *valuaties* predikaat is een lijst van koppels van werelden en lijsten van *atomen*. Deze atomen staan voor de propositiesymbolen die waar zijn in de wereld uit het eerste argument. Elk feit komt slecht éénmaal voor in de feitendatabank. De feiten worden in het programma zelf gespecificeerd. Een voorbeeld feitendatabank (uit de oefeningenles van 06-12-2006):

```

werelden([1, 2, 3, 4]).
relaties([(1, 1), (1, 2), (1, 4), (2, 3), (3, 2), (3, 4), (4, 4)]).
valuaties([(1, [p, q]), (2, [p]), (3, [p]), (4, [q])]).

```

Implementeer een prolog programma dat, gegeven deze of gelijkaardige feiten, de modale waarheidswaarde nagaat van een modaal logische formule, bij de vraag

*waarde*(< *modaleformule* >, < *wereld* >, < *waarheidswaarde* >).

Volgende modaallogische connectieven kunnen gebruikt worden in < *modaleformule* >:

- p “en” q : *a*(*p*, *q*)
- p “of” q : *o*(*p*, *q*)
- p “impliceert” q : *i*(*p*, *q*)

- “niet”  $p : n(p)$
- “noodzakelijk”  $p : nd(p)$
- “mogelijk”  $p : mg(p)$

Voorbeelden van  $\langle \text{modale formule} \rangle$ s (uit de oefeningenles van 06-12-2006)

$$\begin{array}{ll} mg(a(p, q)) & \text{voor } \Diamond(p \wedge q) \\ nd(i(p, nd(p))) & \text{voor } \Box(p \rightarrow \Box p) \\ i(mg(nd(p)), nd(mg(p))) & \text{voor } \Box \Diamond p \rightarrow \Diamond \Box p \end{array}$$

Een  $\langle \text{wereld} \rangle$  is een wereldatoom uit de feitendatabank. Er zijn slechts twee mogelijke waarden voor  $\langle \text{waarheidswaarde} \rangle$ , namelijk 1, voor “waar” en 0, voor “vals”. Een voorbeeld van de werking van het  $\text{waarde}/3$  predikaat: op de vraag

$$\text{waarde}(nd(i(p, nd(p))), 3, 1).$$

wordt, gegeven de voorbeeldfeitendatabank boven, geantwoord met

*Yes.*

Op de vraag

$$\text{waarde}(nd(i(p, nd(p))), 3, W).$$

wordt, gegeven de voorbeeldfeitendatabank boven, geantwoord met

$$W = 1;$$

*No.*

Op de vraag

$$\text{waarde}(nd(i(p, nd(p))), 4, 0).$$

wordt, gegeven de voorbeeldfeitendatabank boven, geantwoord met

*No.*

Indien geen antwoord kan gegeven worden, omdat er feiten ontbreken, produceert het  $\text{waarde}/3$  predikaat eveneens een *No*. Het predikaat  $\text{waarde}/3$  moet alle mogelijke feitendatabanken, samen met alle mogelijke  $\langle \text{modale formule} \rangle$ s uit de propositionele modale logica ondersteunen, en steeds een correct antwoord geven op de gestelde vraag.

Veel succes!