

Representación del conocimiento

Tarea 5

Raúl Daniel García Ramón
raul.d.garcia95@gmail.com
zs22000520@estudiantes.uv.mx

7 de junio de 2023

Índice

1. Agentes especialistas.	2
2. Acción interna.	7
3. Interacción entre especialistas.	9
4. Justificación.	13

En la base de conocimientos que implementaron en Prolog para el protocolo del Covid-19, se puede ver que el conocimiento proviene de diversas fuentes o especialistas.

1. Agentes especialistas.

Separe su base de conocimientos en al menos dos agentes Jason que implementan dichos especialistas. [30/100]

Los dos agentes que funcionan como los especialistas de la base de conocimiento son un doctor y un epidemiólogo. El doctor es el encargado de tratar con los agentes pacientes, los cuales le dicen los síntomas que tienen en sus creencias y con base en esa información determina si son un caso sospechoso o no, y si lo son le indica al epidemiólogo que les haga una prueba. A continuación, el epidemiólogo revisa en su base de conocimiento que tipo de prueba le corresponde al paciente dependiendo de su estado y le indica al doctor el resultado de la prueba. Finalmente, el doctor le indica al paciente el resultado de su prueba.

A continuación se muestra el código del agente doctor:

```
1  /* Agent doctor in project tarea5*/
2
3  /* Initial beliefs and rules */
4  /*lista de enfermedades*/
5  enfermedadrespiratoria(aguda).
6  enfermedadrespiratoria(leve).
7  enfermedadrespiratoria(grave).
8
9  /*lista de contactos*/
10 contacto(casoconfirmado).
11 contacto(bajoinvestigacion).
12
13 /*lista de paises*/
14 viaje(china).
15 viaje(hongkong).
16 viaje(coreadelsur).
17 viaje(japon).
```

```

18 viaje(italia).
19 viaje(iran).
20 viaje(singapur).
21
22 /*si el paciente tiene enfermedad respiratoria y
23 tuvo contacto hace menos de 14 dias es caso sospechoso*/
24 casosospechoso(Enfermedadrespiratoria,Contacto,_,Dias) :-
25     enfermedadrespiratoria(Enfermedadrespiratoria) &
26     contacto(Contacto) &
27     Dias < 15.
28
29 /*si el paciente tiene enfermedad respiratoria y
30 viajo a un pais con transmision local hace menos de 14 dias
31 es caso sospechoso*/
32 casosospechoso(Enfermedadrespiratoria,_,Viaje,Dias):-
33     enfermedadrespiratoria(Enfermedadrespiratoria) &
34     viaje(Viaje) &
35     Dias < 15.
36
37 /*lista de tipos de muestras*/
38 tipodemuestra(normal,exudadonasofaringeoyfaringeo).
39 tipodemuestra(defuncion,biopsiapulmonar).
40 tipodemuestra(intubado,lavadobronquioalveolar).
41
42 /* Initial goals */
43
44 !start.
45
46 /* Plans */
47
48 +!start : true <- .print("Hola.").
49
50 /*si la enfermedad respiratoria esta en la lista
51 pregunta si tuvo contacto con alguien
52 si la enfermedad no esta en la lista le indica que
53 es caso no sospechoso
54 y agrega a sus creencias que el paciente Y es caso no sospechoso*/
55 +enfermedadrespiratoria(X)[source(Y)] : Y \== self <-
56     Listaenfermedadrespiratoria = [aguda,leve,grave];
57     if (.member(X,Listaenfermedadrespiratoria))

```

```

58     {
59         .send(Y,askOne,contacto(Z));
60         .print(Y, " ¿Ha estado en contacto con alguien enfermo o
61         bajo investigación?");
62     }
63     else
64     {
65         .send(Y,tell,case(nosospechoso));
66         .send(self,tell,case(Y,nosospechoso));
67         .print(Y, " usted no es caso sospechoso.");
68     }.
69
70     /*si el contacto es peligroso pregunta hace cuantos dias fue
71     si el contacto no es peligroso
72     pregunta si viajo a un pais con contagio local*/
73     +contacto(X)[source(Y)] : Y \== self <-
74         Listacontacto = [bajoinvestigacion,caseconfirmado];
75         if (.member(X,Listacontacto))
76         {
77             .send(Y,askOne,dias(Z));
78             .print(Y, " ¿Hace cuantos días?");
79         }
80     else
81     {
82         .send(Y,askOne,viaje(W));
83         .print(Y, " ¿Ha viajado a algún pais?");
84     }.
85
86     /*si viajo a un pais con contagio local pregunta hace cuantos dias fue
87     si no viajo a un pais con contagio local
88     indica que no es caso sospechoso
89     y agrega a sus creencias que el paciente Y es caso no sospechoso*/
90     +viaje(X)[source(Y)] : Y \== self <-
91         Listapaises = [china,hongkong,coreadelsur,japon,italia,iran,singapur];
92         if (.member(X,Listapaises))
93         {
94             .send(Y,askOne,dias(Z));
95             .print(Y, " ¿Hace cuantos días?");
96         }
97     else

```

```

98     {
99         .send(Y,tell, caso(nosospechoso));
100        .send(self,tell, caso(Y,nosospechoso));
101        .print(Y, " usted no es caso sospechoso.");
102    }.
103
104    /*si el contacto o el viaje fueron menor a 15 dias es caso sospechoso
105    si no es caso no sospechoso
106    y agrega a sus creencias si es o no caso sospechoso*/
107    +dias(X)[source(Y)] <-
108        if (X < 15)
109        {
110            .send(Y,tell, caso(sospechoso));
111            .send(self,tell, caso(Y,sospechoso));
112            .print(Y, " usted es caso sospechoso.");
113        }
114        else
115        {
116            .send(Y,tell, caso(nosospechoso));
117            .send(self,tell, caso(Y,nosospechoso));
118            .print(Y, " usted no es caso sospechoso.");
119        }.
120
121    /*si recibe un caso sospechoso le indica que se ponga cubrebocas*/
122    +caso(X,sospechoso)[source(_)] <-
123        .print(X, " lo vamos a tener que aislar.");
124        .send(X,tell, aislado);
125        .print(X, " pongase cubrebocas.");
126        .send(X,achieve, colocarcubrebocas);
127        .send(X,askOne, status(Y)).
128
129    /*le dice al epidemiologo que le realice la prueba al paciente
130    y le indica el estatus en el que esta el paciente*/
131    +status(Y)[source(X)] <-
132        .send(epidemiologo,achieve, realiceprueba(X,Y));
133        .print(X, " el epidemiologo le hara la prueba de Covid.").
134
135    /*regla para colocar cubrebocas*/
136    +!colocarcubrebocas(boca,nariz)[source(_)] <-
137        .send(self,tell, colocarcubrebocas(boca,nariz)).

```

```

138
139  /*se le informa al paciente el resultado de su prueba
140  y se borra de las creencias que el caso era sospechoso*/
141  +caso(X,Resultado)[source(_)] <-
142      .send(X,untell,caso(sospechoso));
143      .send(X,tell,caso(Resultado));
144      .print(X , " su prueba dio " , Resultado, ".");
145      .send(self,untell,caso(X,sospechoso)).
146
147  /*si un paciente manda toda la informacion junta
148  se hace un querie para ver si es sospechoso o no*/
149  +state(Enfermedadrespiratoria,Contacto,Viaje,Dias)[source(X)] : X \== self <-
150      ?casosospechoso(Enfermedadrespiratoria,Contacto,_,Dias);
151      .send(X,tell,caso(sospechoso));
152      .send(self,tell,caso(X,sospechoso));
153      .print(X, " usted es caso sospechoso. ").

```

El siguiente agente es el epidemiologo:

```

1  /* Agent epidemiologo in project */
2
3  /* Initial beliefs and rules */
4  /* tipos de muestra dependiendo el estatus del paciente*/
5  tipodemuestra(normal,exudadonasofaringeooyfaringeo).
6  tipodemuestra(defuncion,biopsiapulmonar).
7  tipodemuestra(intubado,lavadobronquioalveolar).
8
9  /* Initial goals */
10
11 /* Plans */
12 /*cuando el doctor le dice que le realice la prueba al paciente X
13 con el estatus Y, revisa en sus creencias
14 que tipo de prueba le corresponde*/
15 +!realiceprueba(X,Y)[source(doctor)] <-
16     ?tipodemuestra(Y,Z);
17     .print("La prueba de " , X , " sera " , Z , ".");
18     .send(self,tell,prueba(X,Y,Z));
19     .send(self,achieve,prueba(X,Y,Z)).
20

```

```

21  /*suponemos que la prueba da positiva
22  elimina de sus creencias que tipo de prueba hara
23  y guarda que tipo de prueba fue y el resultado
24  y le informa al doctor que la prueba fue positiva*/
25  +!prueba(X,Y,Z)[source(_)] <-
26      .send(self,untell,prueba(X,Y,Z));
27      .send(self,tell,prueba(X,Y,Z,positiva));
28      .send(doctor,tell,case(X,positivo)).

```

2. Acción interna.

Extienda estos agentes con una acción interna, p. ej. algún cálculo matemático, acceso a base de datos, etc. [30/100]

Dentro de las acciones internas que realizan los agentes se encuentra el revisar si los días de contacto o viaje son menores a 15, además de las revisiones de sí algún elemento forma parte de una lista con la función member. A continuación, se muestran algunos ejemplos de las acciones realizadas por el agente doctor:

```

1  /*si la enfermedad respiratoria esta en la lista
2  pregunta si tuvo contacto con alguien
3  si la enfermedad no esta en la lista le indica que
4  es caso no sospechoso
5  y agrega a sus creencias que el paciente Y es caso no sospechoso*/
6  +enfermedadrespiratoria(X)[source(Y)] : Y \== self <-
7      Listaenfermedadrespiratoria = [aguda,leve,grave];
8      if (.member(X,Listaenfermedadrespiratoria))
9      {
10         .send(Y,askOne,contacto(Z));
11         .print(Y, " ¿Ha estado en contacto con alguien enfermo o
12         bajo investigación?");
13     }
14     else
15     {
16         .send(Y,tell,case(nosospechoso));
17         .send(self,tell,case(Y,nosospechoso));
18         .print(Y, " usted no es caso sospechoso.");
19     }.

```

```

20
21  /*si el contacto es peligroso pregunta hace cuantos dias fue
22  si el contacto no es peligroso
23  pregunta si viajo a un pais con contagio local*/
24  +contacto(X)[source(Y)] : Y \== self <-
25      Listacontacto = [bajoinvestigacion, casoconfirmado];
26      if (.member(X, Listacontacto))
27      {
28          .send(Y, askOne, dias(Z));
29          .print(Y, " ¿Hace cuantos días?");
30      }
31      else
32      {
33          .send(Y, askOne, viaje(W));
34          .print(Y, " ¿Ha viajado a algún país?");
35      }.
36
37  /*si viajo a un pais con contagio local pregunta hace cuantos dias fue
38  si no viajo a un pais con contagio local
39  indica que no es caso sospechoso
40  y agrega a sus creencias que el paciente Y es caso no sospechoso*/
41  +viaje(X)[source(Y)] : Y \== self <-
42      Listapaises = [china, hongkong, coreadelsur, japon, italia, iran, singapur];
43      if (.member(X, Listapaises))
44      {
45          .send(Y, askOne, dias(Z));
46          .print(Y, " ¿Hace cuantos días?");
47      }
48      else
49      {
50          .send(Y, tell, caso(nosospechoso));
51          .send(self, tell, caso(Y, nosospechoso));
52          .print(Y, " usted no es caso sospechoso.");
53      }.
54
55  /*si el contacto o el viaje fueron menor a 15 dias es caso sospechoso
56  si no es caso no sospechoso
57  y agrega a sus creencias si es o no caso sospechoso*/
58  +dias(X)[source(Y)] <-
59      if (X < 15)

```



```

60     {
61         .send(Y,tell, caso(sospechoso));
62         .send(self,tell, caso(Y,sospechoso));
63         .print(Y, " usted es caso sospechoso.");
64     }
65     else
66     {
67         .send(Y,tell, caso(nosospechoso));
68         .send(self,tell, caso(Y,nosospechoso));
69         .print(Y, " usted no es caso sospechoso.");
70     }.

```

3. Interacción entre especialistas.

Diseñe un protocolo de interacción entre sus especialistas, basado en actos de habla. [20/100]

Los actos de habla implementados en el código son el tell donde el paciente le dice al doctor que síntomas tiene, además el doctor puede realizar preguntar con el askOne para obtener más información necesaria, de igual manera dar órdenes con el achieve. También se simuló que al paciente pancho se le indicara ponerse cubrebocas y él pregunte mediante askHow las instrucciones para colocárselo correctamente. Esta interacción se puede reflejada en la consola, donde se ve las preguntas que realiza el doctor y las indicaciones que les va dando a los pacientes y al epidemiólogo. A continuación se muestran los códigos de los pacientes, Pancho y Fernanda, al igual que la consola donde se aprecia la conversación que se tiene y las creencias finales de cada agente.

Código Pancho:

```

1  /* Agent pancho in project tarea5*/
2
3  /* Initial beliefs and rules */
4  enfermedadrespiratoria(aguda).
5  contacto(no).
6  viaje(china).
7  dias(10).
8  status(normal).

```

```

9
10  /* Initial goals */
11
12  /* Plans */
13
14  /*le informa al doctor que tiene tiene enfermedad respiratoria*/
15  +enfermedadrespiratoria(X) : true <-
16      .send(doctor,tell,enfermedadrespiratoria(X));
17      .print("Doctor tengo enfermedad respiratoria " , X , ".").
18
19  /*si se le pide que se ponga cubrebocas pregunta como
20 una vez que sabe como lo ejecuta*/
21  +!colocarcubrebocas[source(X)] :true <-
22      .send(X,askHow, {+!colocarcubrebocas(_,_) [source(_)]});
23      .print("¿Como se coloca el cubrebocas?");
24      .wait(1);
25      .print("Se coloca: ");
26      .list_plans({+!colocarcubrebocas(_,_) [source(_)]});
27      .print;
28      .send(self,achieve,colocarcubrebocas(_,_) [source(_)]).

```

Código Fernanda:

```

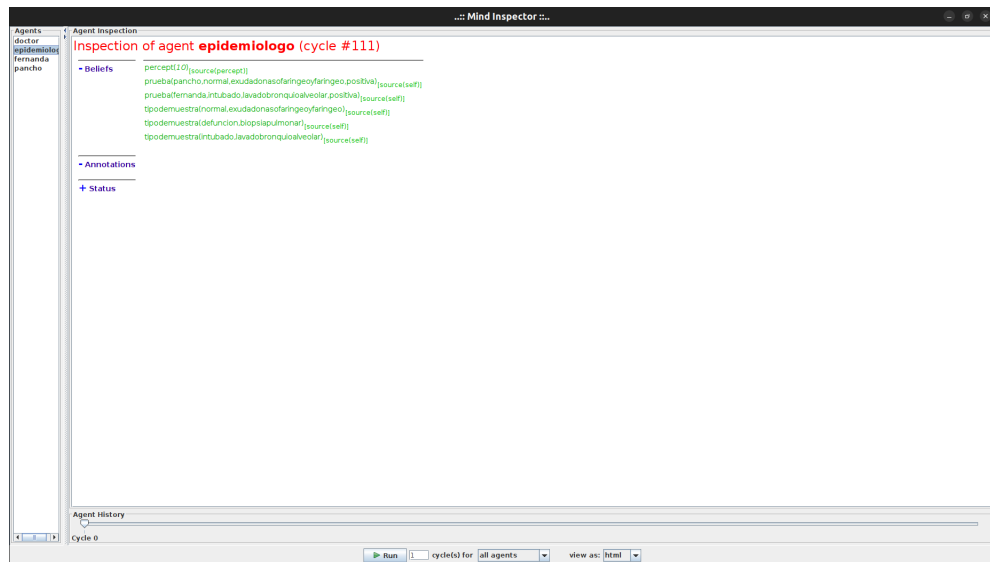
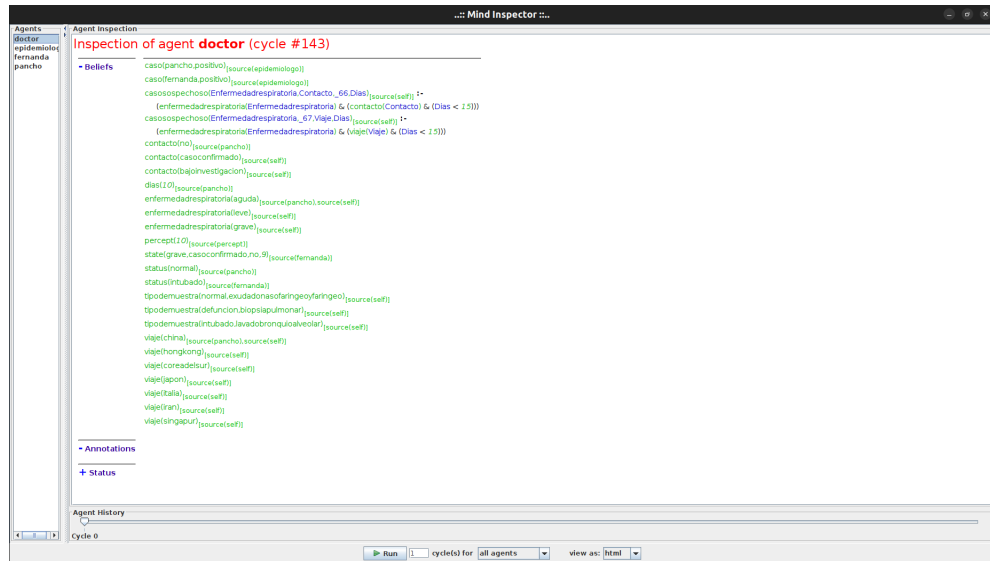
1  /* Agent fernanda in project */
2
3  /* Initial beliefs and rules */
4  state(grave,casoconfirmado,no,9).
5  status(intubado).
6  /* Initial goals */
7
8  /* Plans */
9  /*el paciente manda toda la informacion junta*/
10 +state(X,Y,Z,W) <-
11     .send(doctor,tell,state(X,Y,Z,W));
12     .print("Doctor tengo sintomas respiratorios " ,X,
13         " estuve en contacto con " , Y, " hace " ,W, " dias.").
14
15 /*regla para colocar cubrebocas*/
16 +!colocarcubrebocas[source(_)] <-
17     .send(self,tell,colocarcubrebocas(boca,nariz)).

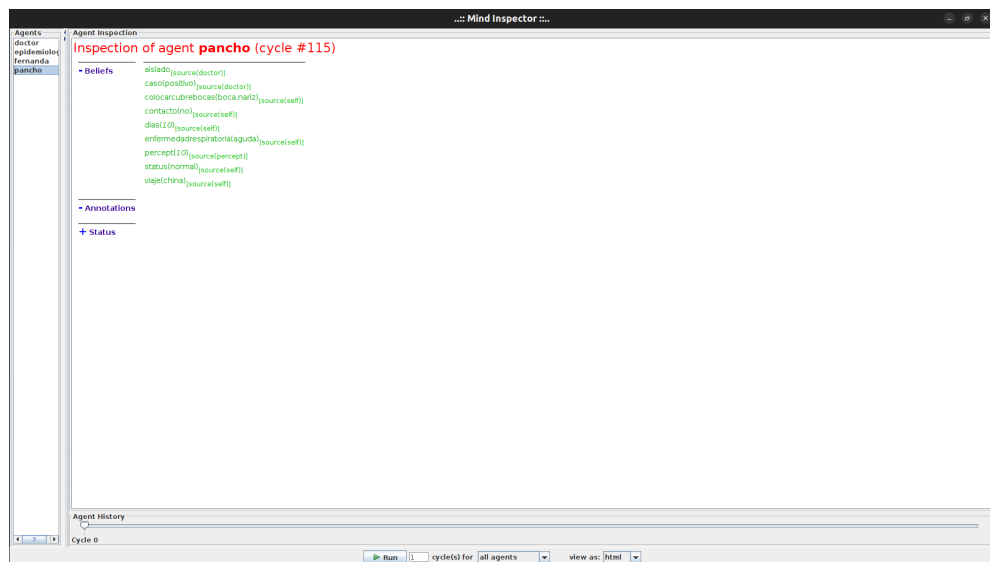
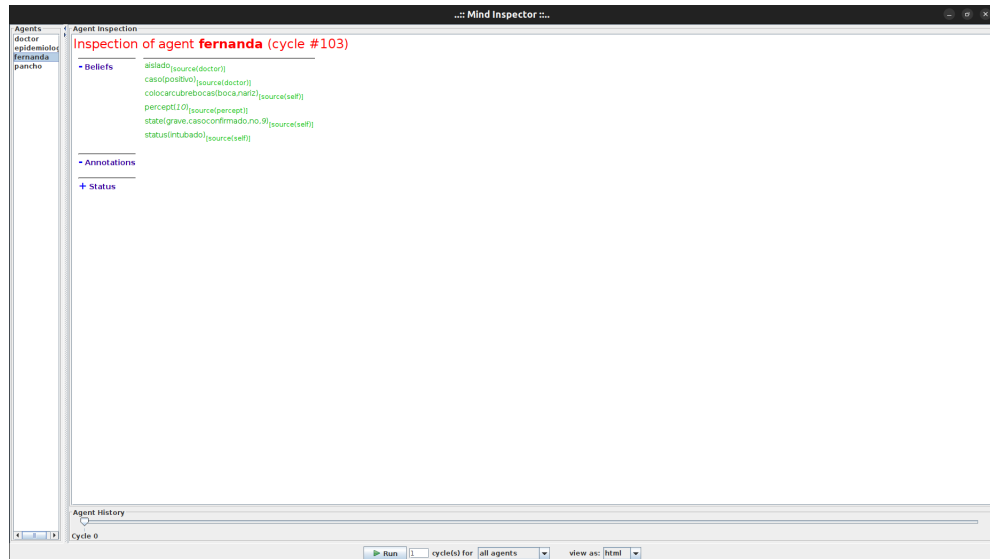
```

Consola de Conversación:

```
1 Runtime Services (RTS) is running at 127.0.1.1:42957
2 Agent mind inspector is running at http://127.0.1.1:3272
3 [pancho] Doctor tengo enfermedad respiratoria aguda.
4 [doctor] Hola.
5 [fernanda] Doctor tengo sintomas respiratorios grave estuve en contacto
6 con casoconfirmado hace 9 dias.
7 [doctor] fernanda usted es caso sospechoso.
8 [doctor] fernanda lo vamos a tener que aislar.
9 [doctor] pancho ¿Ha estado en contacto con alguien enfermo o bajo investigación?
10 [doctor] fernanda pongase cubrebocas.
11 [doctor] pancho ¿Ha viajado a algún pais?
12 [doctor] fernanda el epidemiologo le hara la prueba de Covid.
13 [epidemiologo] La prueba de fernanda sera lavadobronquioalveolar.
14 [doctor] pancho ¿Hace cuantos días?
15 [doctor] pancho usted es caso sospechoso.
16 [doctor] pancho lo vamos a tener que aislar.
17 [doctor] pancho pongase cubrebocas.
18 [doctor] fernanda su prueba dio positivo.
19 [pancho] ¿Como se coloca el cubrebocas?
20 [doctor] pancho el epidemiologo le hara la prueba de Covid.
21 [epidemiologo] La prueba de pancho sera exudadonasofaringeoyfaringeo.
22 [doctor] pancho su prueba dio positivo.
23 [pancho] Se coloca:
24 [pancho] @p__14[source(doctor),url("file:src/agt/doctor.asl")]
25 +!colocarcubrebocas(boca,nariz)[source(_104)] <-
26 .send(self,tell,colocarcubrebocas(boca,nariz)).
27 [pancho]
```

Creencias de cada agente al final de la ejecución:





4. Justificación.

Justifique las performativas usadas con este propósito. [20/100]

La idea con la manera en que se realiza la interacción de habla entre

los agentes, es tratar de simular como es una consulta médica en la vida real, donde algunos pacientes llegan y directamente le indican todo el padecimiento al doctor y él solo se encarga de hacer el diagnóstico, como lo es con el caso de Fernanda.

Por otro lado, en el caso de Pancho se buscó simular la consulta más común, que es donde el paciente indica un malestar y el doctor va realizando una serie de preguntas para poder encontrar el padecimiento específico, además de Pancho preguntar como se realiza una actividad que él no tiene en su conocimiento como llevar a cabo, en este caso el colocarse un cubrebocas.

Finalmente, el doctor pasa parte de la información recabada al epidemiólogo para que él sepa, basado en el estatus del paciente, qué tipo de prueba va a tener que realizar y posteriormente se simula como el epidemiólogo pasa el resultado al doctor y el doctor se encarga de pasar ese resultado a cada paciente por separado.

Referencias

- Bratko, I. (2012). *Prolog programming for Artificial Intelligence*. Pearson, fourth edition.
- Clocksin, W. F. and Melish, C. S. (2003). *Programming in Prolog, using the ISO standard*. Springer-Verlag, Berlin-Heidelberg, Germany.
- Nilsson, U. and Maluszynski, J. (2000). *Logic, Programming and Prolog*. John Wiley & Sons Ltd, 2nd edition.
- Rafael H. Bordini, Jomi Fred Hübner, M. W. (2007). *Programming Multi-Agent Systems in AgentSpeak using Jason*. Wiley Interscience.