

a)

función Agente-BC(percepción) devuelve una acción

variables estáticas: BC, una base de conocimiento

Decir(BC, Construir-Sentencia-De-Percepción(percepción))

acción ← Preguntar(BC, Pedir-Acción())

Decir(BC, Construir-Sentencia-De-Acción(acción))

devolver acción

**REAS**

Rendimiento: -1 por cada acción. -1000 por caer en un hoyo. +300 por limpiar habitación

Entorno: una matriz de 4x4 habitaciones donde puede haber hoyos y suciedad.

Actuadores: actuadores para moverse a otra casilla y para aspirar suciedad

Sensores: sensores para detectar la brisa, un muro, y la suciedad.

**Base de conocimiento**

*H*x,y: Hay un hoyo en la casilla (x,y)

*P*x,y: El agente está posicionado en la casilla (x,y)

*B*x,y: Hay brisa en la casilla (x,y)

*S*x,y: Hay suciedad en la casilla (x,y)

R1: ᆨ*H*1,1

R2: *P*1,1

R3: *B*1,1 ↔ (*H*1,2 v *H*2,1)

R4: *B*1,2 ↔ (*H*1,1 v *H*2,2 v *H*1,3)

R5: *B*1,3 ↔ (*H*1,2 v *H*2,3 v *H*1,4)

R6: *B*1,4 ↔ (*H*1,3 v *H*2,4)

R7: *B*2,1 ↔ (*H*1,1 v *H*3,1 v *H*2,2)

R8: *B*2,2 ↔ (*H*2,1 v *H*1,2 v *H*2,3 v *H*3,2)

R9: *B*2,3 ↔ (*H*2,2 v *H*1,3 v *H*3,3 v *H*2,4)

R10: *B*2,4 ↔ (*H*1,4 v *H*2,3 v *H*3,4)

R11: *B*3,1 ↔ (*H*2,1 v *H*3,2 v *H*4,1)

R12: *B*3,2 ↔ (*H*3,1 v *H*2,2 v *H*4,2 v *H*3,3)

R13: *B*3,3 ↔ (*H*2,3 v *H*3,2 v *H*3,4 v *H*4,3)

R14: *B*3,4 ↔ (*H*2,4 v *H*3,3 v *H*4,4)

R15: *B*4,1 ↔ (*H*3,1 v *H*4,2)

R16: *B*4,2 ↔ (*H*4,1 v *H*4,3 v *H*3,2)

R17: *B*4,3 ↔ (*H*4,2 v *H*3,3 v *H*4,4)

R18: *B*4,4 ↔ (*H*4,3 v *H*3,4)

b)

función Agente-BC(percepción) devuelve una acción

variables estáticas: BC, una base de conocimiento

Decir(BC, Construir-Sentencia-De-Percepción(percepción))

acción ← Preguntar(BC, Pedir-Acción())

Decir(BC, Construir-Sentencia-De-Acción(acción))

devolver acción

**REAS**

Rendimiento: -1 por cada acción. -100 por moverse a una habitación limpia. -1000 por caer en un hoyo. +500 por limpiar la habitación.

Entorno: una matriz de 4x4 habitaciones donde puede haber hoyos y suciedad.

Actuadores: actuadores para moverse a otra casilla y para aspirar suciedad

Sensores: sensores para detectar la brisa, un muro, y la suciedad de cualquier habitación.

**Base de conocimiento**

*H*x,y: Hay un hoyo en la casilla (x,y)

*P*x,y: El agente está posicionado en la casilla (x,y)

*B*x,y: Hay brisa en la casilla (x,y)

*S*x,y: Hay suciedad en la casilla (x,y)

R1: ᆨ*H*1,1

R2: *P*1,1

R3: *B*1,1 ↔ (*H*1,2 v *H*2,1)

R4: *B*1,2 ↔ (*H*1,1 v *H*2,2 v *H*1,3)

R5: *B*1,3 ↔ (*H*1,2 v *H*2,3 v *H*1,4)

R6: *B*1,4 ↔ (*H*1,3 v *H*2,4)

R7: *B*2,1 ↔ (*H*1,1 v *H*3,1 v *H*2,2)

R8: *B*2,2 ↔ (*H*2,1 v *H*1,2 v *H*2,3 v *H*3,2)

R9: *B*2,3 ↔ (*H*2,2 v *H*1,3 v *H*3,3 v *H*2,4)

R10: *B*2,4 ↔ (*H*1,4 v *H*2,3 v *H*3,4)

R11: *B*3,1 ↔ (*H*2,1 v *H*3,2 v *H*4,1)

R12: *B*3,2 ↔ (*H*3,1 v *H*2,2 v *H*4,2 v *H*3,3)

R13: *B*3,3 ↔ (*H*2,3 v *H*3,2 v *H*3,4 v *H*4,3)

R14: *B*3,4 ↔ (*H*2,4 v *H*3,3 v *H*4,4)

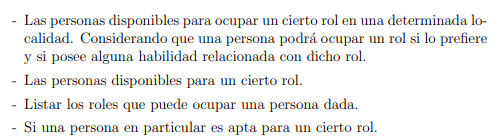
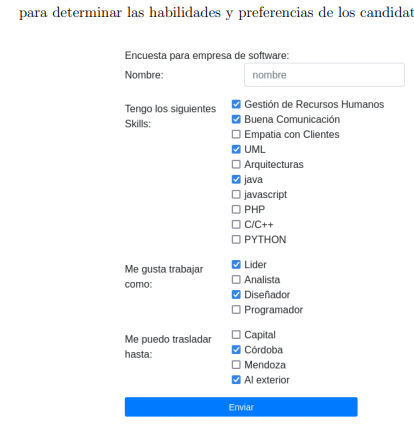
R15: *B*4,1 ↔ (*H*3,1 v *H*4,2)

R16: *B*4,2 ↔ (*H*4,1 v *H*4,3 v *H*3,2)

R17: *B*4,3 ↔ (*H*4,2 v *H*3,3 v *H*4,4)

R18: *B*4,4 ↔ (*H*4,3 v *H*3,4)

R19: S(x,y), con x∈{1,2,3,4} e y∈{1,2,3,4}



**REAS**

Rendimiento: asistir en consulta +100

Entorno: un listado de información de los candidatos (hechos de Prolog)

Actuadores: una consola para mostrar los resultados

Sensores: un input donde el usuario ingresa la consulta Prolog

**Predicados:**

se\_puede\_trasladar(Persona, Localidad)

prefiere\_rol(Persona, Rol)

posee\_skill(Persona, Skill)

es\_para(Skill, Rol)

**Hechos:**

posee\_skill(alejo, php).

posee\_skill(alejo, uml).

posee\_skill(lautaro, java).

posee\_skill(lautaro, buena\_comunicacion).

posee\_skill(fermin, empatia\_con\_clientes).

posee\_skill(fermin, javascript).

es\_para(php, diseniador).

es\_para(uml, diseniador).

es\_para(java, programador).

es\_para(buena\_comunicacion, lider).

es\_para(javascript, programador).

es\_para(empatia\_con\_clientes, analista).

prefiere\_rol(lautaro, lider).

prefiere\_rol(lautaro, programador).

prefiere\_rol(alejo, programador).

prefiere\_rol(fermin, analista).

prefiere\_rol(fermin, diseniador).

se\_puede\_trasladar(lautaro, al\_exterior).

se\_puede\_trasladar(lautaro, cordoba).

se\_puede\_trasladar(fermin, al\_exterior).

se\_puede\_trasladar(alejo, capital).

se\_puede\_trasladar(alejo, mendoza).

Reglas:

**R1:** es\_apto(Persona, Rol) :-

prefiere\_rol(Persona, Rol),

posee\_skill(Persona, Skill),

es\_para(Skill, Rol).

**R2:** cumple\_rol(Persona, Rol) :-

posee\_skill(Persona, Skill),

es\_para(Skill, Rol).

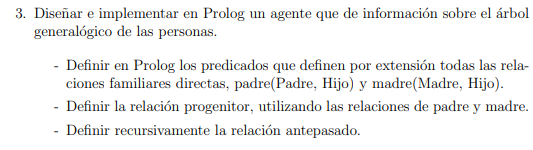
**R3:** roles(Persona, Rol) :-

es\_apto(Persona, Rol).

**R4:** esta\_disponible\_en(Persona, Rol, Localidad) :-

se\_puede\_trasladar(Persona, Localidad),

es\_apto(Persona, Rol).



**Predicados:**

padre(Padre, Hijo).

madre(Madre, Hijo).

mujer(Persona).

hombre(Persona).

**Hechos:**

padre(alfredo, alejo).

padre(viviana, josesito).

padre(jose, lautaro).

padre(carlos, jose).

padre(juan, karina).

padre(gaston, fermin).

padre(hector, lorena).

padre(pedro, gaston).

padre(alfredo, melisa).

madre(viviana, ignacio).

madre(karina, lautaro).

madre(viviana, raquel).

madre(martina, jose).

madre(maria, karina).

madre(lorena, fermin).

madre(marta, lorena).

madre(viviana, alejo).

madre(alicia, gaston).

mujer(melisa).

hombre(ignacio).

**Reglas:**

**R1:** progenitor(Persona, Hijo) :-

padre(Persona, Hijo); madre(Persona, Hijo).

**R2:** antepasado(PersonaHijo, PersonaAntepasado) :-

progenitor(PersonaAntepasado, PersonaHijo);

padre(Padre, PersonaHijo), antepasado(Padre, PersonaAntepasado);

madre(Madre, PersonaHijo), antepasado(Madre, PersonaAntepasado).

**R3:** abuelo(Persona, Abuelo) :-

progenitor(Progenitor, Persona), padre(Abuelo, Progenitor).

**R4:** abuela(Persona, Abuela) :-

progenitor(Progenitor, Persona), madre(Abuela, Progenitor).

**R5:** hermano(Persona, Hermano) :-

hombre(Hermano),

progenitor(Progenitor, Persona),

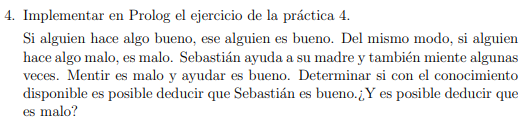
progenitor(Progenitor, Hermano).

**R6:** hermana(Persona, Hermana) :-

mujer(Hermana),

progenitor(Progenitor, Persona),

progenitor(Progenitor, Hermana).



**Predicados:**

hace(Persona, Accion)

esBueno(Accion)

**Hechos:**

hace(sebastian, miente).

hace(sebastian, ayuda).

hace(fermin, ayuda).

hace(lautaro, miente).

esBueno(ayuda).

esMalo(miente).

**Reglas:**

**R1:** personaBuena(Persona) :-

hace(Persona, Accion), esBueno(Accion).

**R2:** personaMala(Persona) :-

hace(Persona, Accion), esMalo(Accion).

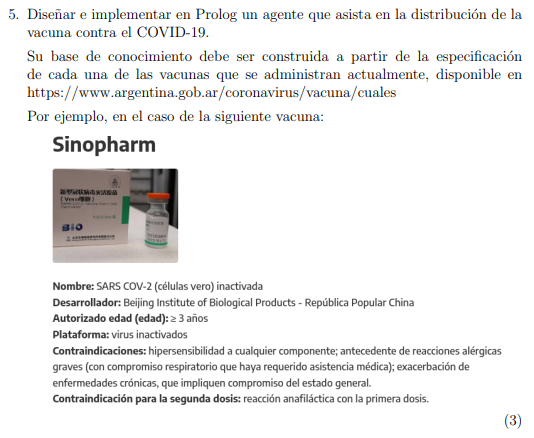
El determinar si sebastian es bueno o no depende de cómo se plantee la regla, por ejemplo, en nuestra implementación, alcanza con que la persona haga una buena acción para ser buena persona, pero se podría plantear de otra manera, como:

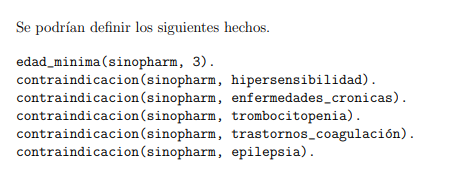
**R1:** personaBuena(Persona) :-

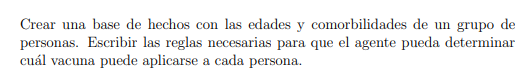
forall(hace(Persona, Accion), esBueno(Accion)).

En este caso todas las acciones de una persona deben ser buena para que la persona sea buena.

El mismo caso se da con una mala persona.







**Predicados:**

edad\_minima(Vacuna, Edad).

contraindicacion(Vacuna, Contra).

edad(Persona, Edad).

comorbilidad(Persona, Comorbilidad).

**Hechos:**

vacuna(sputnikv).

vacuna(covishield).

vacuna(sinopharm).

vacuna(astrazeneca).

vacuna(moderna).

vacuna(convidecia).

vacuna(comirnaty).

edad\_minima(sputnikv, 18).

edad\_minima(covishield, 18).

edad\_minima(sinopharm, 3).

edad\_minima(astrazeneca, 18).

edad\_minima(moderna, 12).

edad\_minima(convidecia, 18).

edad\_minima(comirnaty, 12).

contraindicacion(sputnikv, hipersensibilidad).

contraindicacion(sputnikv, enfermedad\_cronica).

contraindicacion(sputnikv, alergia).

contraindicacion(covishield, hipersensibilidad).

contraindicacion(covishield, alergia).

contraindicacion(sinopharm, hipersensibilidad).

contraindicacion(sinopharm, alergia).

contraindicacion(sinopharm, enfermedad\_cronica).

contraindicacion(astrazeneca, hipersensibilidad).

contraindicacion(astrazeneca, alergia).

contraindicacion(moderna, hipersensibilidad).

contraindicacion(moderna, alergia).

contraindicacion(moderna, enfermedad\_cronica).

contraindicacion(moderna, enfermedad\_grave).

contraindicacion(convidecia, hipersensibilidad).

contraindicacion(convidecia, alergia).

contraindicacion(convidecia, enfermedad\_cronica).

contraindicacion(convidecia, epilepsia).

contraindicacion(convidecia, enfermedad\_neurologica).

contraindicacion(comirnaty, hipersensibilidad).

contraindicacion(comirnaty, alergia).

contraindicacion(comirnaty, enfermedad\_cronica).

contraindicacion(comirnaty, enfermedad\_grave).

edad(alejo, 22).

edad(ignacio, 10).

edad(lautaro, 22).

edad(francisco, 8).

edad(fermin, 22).

edad(pedro, 80).

comorbilidad(alejo, enfermedad\_cronica).

comorbilidad(lautaro, enfermedad\_grave).

comorbilidad(francisco, hipersensibilidad).

comorbilidad(fermin, enfermedad\_grave).

comorbilidad(pedro, epilepsia).

**Reglas:**

**R1:** noHayProblema(Comorbilidad, Vacuna) :-

forall(contraindicacion(Vacuna, Contraindicacion),

Comorbilidad \= Contraindicacion).

**R2:** recibe(Persona, Vacuna) :-

edad(Persona, EdadP),

vacuna(Vacuna),

edad\_minima(Vacuna, EdadV),

EdadP >= EdadV,

forall(comorbilidad(Persona, Comorbilidad),

noHayProblema(Comorbilidad, Vacuna)).