

Knowledge Based Agents with CP0 Logic

Práctica 1

Aprenentatge i raonament automàtic

Universitat de Lleida

Autor: Marc Vivas Baiges

DNI: 78101067J

Fecha de inicio: 9 de abril del 2022

Fecha de finalización: 11 de abril de 2022





Índice

Introducción	2
Diseño	2
Variables	2
Fórmula	3
Comentarios sobre la implementación y el diseño	4
Pequeña optimización	4
Referencias	4

Introducción

En esta práctica se va a intentar resolver las tareas que se proponen en el enunciado. La práctica ha sido hecha de manera individual. La explicación del código se ha hecho con los comentarios en cada función y clase creada usando javadoc. En este documento se van a explicar algunos aspectos que se creían que necesitaban una explicación un poco más extensa.

Diseño

Variables

A continuación, las variables que se utilizarán:

$\forall x,y \in [1,n] \times [1,n]$:

$e_{x,y}^{t-1}$ Ciertos si el sobre se encontraba en la posición x,y .

$e_{x,y}^{t+1}$ Ciertos si el sobre estará en la posición x,y .

$sensor1_{x,y}^t$ Ciertos si el sensor 1 detecta un sobre en al menos una de las siguientes posiciones. $\{(x+1, y), (x-1, y), (x, y-1), (x, y+1)\}$

$sensor2_{x,y}^t$ Ciertos si el sensor 2 detecta un sobre en al menos una de las siguientes posiciones. $\{(x-1, y-1), (x+1, y-1), (x-1, y+1), (x+1, y+1)\}$

$sensor3_{x,y}^t$ Ciertos si el sensor 3 detecta un sobre en la posición (x,y) .

$Variables = \{e_{x,y}^{t-1}, e_{x,y}^{t+1}, sensor1_{x,y}^t, sensor2_{x,y}^t, sensor3_{x,y}^t : \forall x,y \in [1, n] \times [1, n]\}$

Fórmula Γ

$$\Gamma = (e_{1,1}^{t-1} \vee e_{1,2}^{t-1} \vee \dots e_{n,n}^{t-1}) \wedge (e_{1,1}^{t+1} \vee e_{1,2}^{t+1} \vee \dots e_{n,n}^{t+1}) \wedge$$

$$\forall x, y \in [1, n] \times [1, n]:$$

$$sensor3_{x,y}^t \wedge \neg sensor1_{x,y}^t \wedge \neg sensor2_{x,y}^t \rightarrow \neg e_{x',y'}^{t+1} \forall x', y' \in \{(x+1, y), (x-1, y),$$

$$(x, y-1), (x, y+1), (x-1, y-1), (x+1, y-1), (x-1, y+1),$$

$$(x+1, y+1)\} \wedge$$

$$sensor2_{x,y}^t \wedge \neg sensor1_{x,y}^t \wedge \neg sensor3_{x,y}^t \rightarrow \neg e_{x',y'}^{t+1} \forall x', y' \in \{(x+1, y), (x-1, y),$$

$$(x, y-1), (x, y+1), (x, y)\} \wedge$$

$$sensor1_{x,y}^t \wedge \neg sensor2_{x,y}^t \wedge \neg sensor3_{x,y}^t \rightarrow \neg e_{x',y'}^{t+1} \forall x', y' \in \{(x-1, y-1), (x+1, y-1),$$

$$(x-1, y+1), (x+1, y+1), (x, y)\} \wedge$$

$$sensor1_{x,y}^t \wedge sensor2_{x,y}^t \rightarrow \neg e_{x',y'}^{t+1} \forall x', y' \in \{(x, y)\} \wedge$$

$$sensor1_{x,y}^t \wedge sensor3_{x,y}^t \rightarrow \neg e_{x',y'}^{t+1} \forall x', y' \in \{(x-1, y-1), (x+1, y-1),$$

$$(x-1, y+1), (x+1, y+1)\} \wedge$$

$$sensor2_{x,y}^t \wedge sensor3_{x,y}^t \rightarrow \neg e_{x',y'}^{t+1} \forall x', y' \in \{(x+1, y), (x-1, y), (x, y-1), (x, y+1)\} \wedge$$

$$\neg sensor1_{x,y}^t \wedge \neg sensor2_{x,y}^t \wedge \neg sensor3_{x,y}^t \rightarrow \neg e_{x',y'}^{t+1} \forall x', y' \in \{(x+1, y), (x-1, y),$$

$$(x, y-1), (x, y+1), (x-1, y-1), (x+1, y-1), (x-1, y+1),$$

$$(x+1, y+1), (x, y)\} \wedge$$

$$\neg e_{x,y}^{t-1} \rightarrow \neg e_{x,y}^{t+1}$$

Comentarios sobre la implementación y el diseño

Pequeña optimización

Cuando se pregunta lo siguiente:

$$\Gamma \cup E \models Q$$

Si resulta ser consecuencia lógica, se guarda esa cláusula Q en un Set para después si se vuelve a hacer la misma pregunta, no haga falta llamar al SAT solver otra vez ni volver a añadir su respectiva variable $t-1$ en la próxima iteración.

Referencias

No se ha consultado nada, ya que se tenían los suficientes conocimientos para realizar la práctica.

Sin embargo, se tuvieron algunos problemas con Maven, saltaba error al utilizar los comandos propuestos en el enunciado. Para solucionarlo, en la carpeta `/usr/share/maven/lib` tuve que borrar los archivos *guava.jar* y *guice.jar*, luego me descargué los siguientes archivos y los puse en `/usr/share/maven/lib`.

<https://github.com/google/guava/wiki/Release22>

<https://github.com/google/guice/releases>