

UNIVERSITAT DE LLEIDA
Escola Politècnica Superior
Grau en Enginyeria Informàtica
Aprenentatge i Raonament Automàtic

Práctica de agentes inteligentes con lógica proposicional

Sergi Simón Balcells, Joaquim Picó Mora
21040111X, 49383707Q
PraLab2

Professorat : R. Bejar
Data : Dijous 23 d'abril

Contents

1	Explicació de la fórmula	1
2	Canvis en el disseny	2
2.1	Tests	3

1 Explicació de la fórmula

Es defineixen les variables com:

$$Vars = \{tr_{i,j}^{t-1}, p_j^t, u^t, d_{i,j,k}^t, tr_{i,j}^{t+1}\}$$

1. $tr_{i,j}^{t-1}$ i $tr_{i,j}^{t+1}$ informació sobre l'estat del taulell.
2. $d_{i,j,k}^t$ variable per modelitzar les diferents respostes del sensor, hi ha k variables on k és el nombre de respostes que pot donar el sensor:

$$k \in \{0, 1, 2, 3\}$$

3. p_j^t variable per controlar si ens trobem un pirata en l'altura j .
4. u^t , variable per a la resposta que ens dona el pirata.

El nombre de variables és:

$$f(n) = n^2 + n + 1 + 4n^2 + n^2$$

Llavors, la fórmula Γ es defineix amb les següents normes:

Clàusula inicial de l'agent, existeix al menys un tresor. Serveix per si es vol preguntar a l'agent si en la posició (x, y) hi ha un tresor.

$$\bigcup (tr_{i,j}^{t-1}) \quad (1)$$

Quan el sensor ens respon amb un 0 vol dir que el tresor no es troba en un quadrat de 2 caselles al voltant de l'agent.

$$d_{i,j,0}^t \rightarrow \neg tr_{i',j'}^{t+1}, \text{ t.q. } \max(|i - i'|, |j - j'|) < 3 \quad (2)$$

En el cas que ens respongui amb un 1 voldrà dir que el tresor no es trobarà en cap altra posició que no sigui l'actual de l'agent.

$$d_{i,j,1}^t \rightarrow \neg tr_{i',j'}^{t+1}, \text{ t.q. } i \neq i' \cup j \neq j' \quad (3)$$

Si la resposta fos un dos, el tresor no es troba en cap de les posicions adjacents, contant les diagonals.

$$d_{i,j,2}^t \rightarrow \neg tr_{i',j'}, t.q.max(|i - i'|, |j - j'|) \neq 1 \quad (4)$$

I finalment quan el sensor ens respon amb el 3, voldrà dir que el tresor es troba en una casella la distància de la qual és exactament 2, del qual podem treure la norma:

$$d_{i,j,3}^t \rightarrow \neg tr_{i',j'}, t.q.max(|i - i'|, |j - j'|) \neq 2 \quad (5)$$

Si hem trobat al pirata i ens ha respost que el tresor es troba a les caselles que estan per sobre d'ell podem descartar aquelles que estiguin per sota.

$$p_j^t \wedge u^t \rightarrow \neg tr_{i,j'}^{t+1}, t.q.j' \leq j \quad (6)$$

De la mateixa forma, si ens trobem al pirata i ens diu que el tresor està a dalt, podem descartar les posicions que estan per sobre.

$$p_j^t \wedge \neg u^t \rightarrow \neg tr_{i,j'}^{t+1}, t.q.j' > j \quad (7)$$

Les variables en $t + 1$ actualitzen els valors que existeixen en $t - 1$, i les variables que estan a $t - 1$ es mantenen.

2 Canvis en el disseny

Els canvis del disseny més superficials han estat en funcions on utilitzaven parelles d'enters per a descriure la posició, que s'han canviat a utilitzar **Position**, per a facilitar el pas de dades. També s'ha afegit a aquesta classe un mètode per a calcular la distància entre dos punts, que fa més llegible la creació de les normes del detector.

També s'ha afegit un mètode a **TFState** per a retornar un **HashSet** de totes les posicions on no hi ha "X", que facilita la inferència. A més a més, s'han generat **toString**, **equals** i **hash** de totes les classes menys **TreasureFinder**, ja que es va creure que s'utilitzaria.

2.1 Tests

S'han afegit tests unitaris de totes les classes, inclús s'ha afegit més tests a la classe de `TFinder`, ja que amb funcions més petites era més fàcil veure els errors. Els primers tests són molt superficials, on es mira si el nombre de clàusules i el número de variables generats en crear Γ són correctes, i els següents miren si es fa correctament la inferència de variables. Els altres tests van servir per familiaritzar-se amb el codi donat, mentre es creaven mètodes per a ajudar. Seguint la guia d'estil de Java, tots els mètodes creats no privats s'han fet amb javadoc. Els mètodes privats que s'ha cregut que amb el nom i els paràmetres no s'entendria s'ha fet un comentari.