

Práctica 5: Razonamiento Basado en Casos. Documento de Diseño

GRUPO: 10

Estructura de los casos*Enumerar los tributos de la descripción/consulta y el tipo de datos de cada uno.**Representación del resultado del caso (positivo, negativo, valor entre [-1.,1], etc.)*

MSPACMAN:

Variable	Tipo
<i>Menor distancia fantasma por cada lado</i>	<i>Integer[]</i>
<i>Menor distancia fantasma comestible por cada lado</i>	<i>Integer[]</i>
<i>Menor distancia pills por cada lado</i>	<i>Integer[]</i>
<i>Menor distancia power pills por cada lado</i>	<i>Integer[]</i>
<i>Direcciones válidas</i>	<i>String</i>
<i>Puntuación</i>	<i>Integer</i>
<i>Vidas</i>	<i>Integer</i>

Resultado del caso: *Integer***Solución:** *MOVE*

GHOSTS:

Variable	Tipo
Distancia de fantasmas por cada lado del Mspacman	Int []
Distancia a Mspacman por cada lado del fantasma	Int []
Fantasmas son comestibles	Int [] : 0 no comestible , e.o.c el edibletime restante
MsPacman comió al fantasma	Boolean
El fantasma se comió a MsPacman	Boolean
Score	int
Distancia de fantasma a la powerpill más cercana de Ms Pacman	Int
Distancia powerpill más cercana de Ms Pacman	int
Número de muertes de MsPacMan	int
Last move ghost	MOVE
Last move pacman	MOVE

Resultado del caso: Double

Solución: MOVE

Persistencia

Describir si se utilizará una única base de casos o algún tipo de partición por estado edible, oponente, nivel, etc (que puede coincidir o no con la indexación en memoria)

Indicar si la base de casos será genérica y/o especializada para los distintos contrincantes.

Se usará un conjunto de bases de casos, una para cada oponente.

Organización en memoria

Describir si se utilizará una organización lineal en memoria o algún tipo de indexación o subdivisión de la base de casos.

MSPACMAN:

Los casos estarán indexados en base a los movimientos válidos según la forma del mapa y el último movimiento.

Por ejemplo, hay una subdivisión que contiene todo caso en el que MsPacman solo se puede mover UP y LEFT.

GHOSTS:

Se utilizará una organización lineal de memoria de la base de casos, almacenando de las n entradas (teniendo en cuenta que un movimiento no es válido si es el contrario a tu dirección a la hora de recuperar).

Recuperación

Describir el proceso de recuperación según la organización en memoria. Indicar si para la recuperación se compararán todos los atributos de la consulta con los casos o solo los más relevantes.

Indicar si se utilizará pesos para comparar los atributos.

Especificar las medidas de similitud, tanto local como global que se aplicará a cada atributo.

Indicar el número de vecinos más cercanos a recuperar.

Indicar cuándo se recuperarán los casos (cambio de posición, nueva intersección, ...)

MSPACMAN:

En la recuperación se comparan todos los atributos excepto Puntuación, Vidas, y Movimientos Válidos.

En la comparación se usan pesos y la medida de similitud local es la de Intervalo:

$$1 - (|x - y| / \text{intervalo})$$

La medida de similitud global es la media de las similitudes locales aplicando los pesos.

Se recupera solo el primer vecino más cercano (da mejores resultados que con 4NN)

GHOSTS:

Se comparan todos los atributos excepto
MsPacManEatMe, eatenMsPacman, ghost, MsPacmanDeaths y *Score*.

Para comparar los atributos daremos un peso del 50% al atributo *lastMoveGhost*, los demás se reparten el peso restante por igual y se calcula sus similitudes en base a la función local definida :

-Interval para comparar distancias $(1 - (|x - y| / \text{intervalo}))$

-Equal para comparar movimientos

Se recuperarán los 5 vecinos más cercanos y se hará cada vez que pacman o que un fantasma vayan a ser comidos. Para eso el fantasma recuerda la última intersección antes de llegar a la situación actual.

Cada vecino tendrá un peso diferente sobre la votación, dependiendo de lo similares que sean y lo bueno o malo que haya sido ese movimiento; sumará (*similarity*score*) si es bueno y le restará si es malo a la opción de realizar dicho movimiento.

REUTILIZACIÓN

Indicar cómo se adaptará la solución o si se aplicará directamente. Especificar cómo se combinan las soluciones de los casos recuperados (votación, votación ponderada, ...) y si se tendrá en cuenta el resultado del caso para realizar dicha adaptación.

MSPACMAN:

La solución se aplica si el caso más similar tiene una similitud superior a 0.9. En caso contrario se elige una solución aleatoria.

GHOSTS:

La solución será una votación ponderada de los 5 vecinos más cercanos como explicamos en el siguiente punto.

Se aplicará directamente la solución y no se adaptará.

REVISIÓN

Indicar la función de cálculo del resultado del caso y cuando se calcularía (p.e. tras 5 intersecciones se calcula el cambio en el score).

MSPACMAN:

Resultado: score aumentado tras N intersecciones - 500 * vidas perdidas

GHOSTS:

Resultado:

- Suma 5000 si se come al pacman.
- Suma 2500 si ayuda a comer a pacman
- Resta 5000 si ha sido comido por pacman

RECUERDO

Indicar cual es la política de recuerdo de los casos (todos, solo los que no tengan vecinos muy similares, ...), y cuando se realiza dicho almacenamiento en ficheros (se recomienda que los nuevos casos se guarden al finalizar, aunque esto no es totalmente obligatorio, pero penaliza el tiempo de ciclo).

MSPACMAN:

Se guardan los casos con resultado superior a 700, o con resultado superior a 300 puntos, y similitud menor a 0.95.

Si hay más de 2000 casos, se dejan de guardar para evitar problemas de rendimiento.

GHOSTS:

**Solo se guardan los casos que no tengan vecinos muy similares (<0.95 de similitud).
Un caso malo solo se guarda si el número de casos malos $\leq 0.4 \times$ número de casos buenos.**

Si hay más de 2000 casos, se dejan de guardar para evitar problemas de rendimiento.

Guardar los nuevos casos: Al finalizar cada partida