Pontificia Universidad Católica de Chile
Escuela de Ingeniería
Departamento de Ciencia de la Computación

Tarea 3 – Transporte de Productos en Bodegas
IIC2613 - Inteligencia Artificial
Primer Semestre, 2019
Entrega: 12 de junio, hasta las 18:59

1. Objetivo

El objetivo de esta tarea es que usted practique conceptos de programación en lógica, aplicado a la inteligencia artificial. En particular veremos cómo es posible representar problemas de búsqueda en los que n agentes cooperan para lograr un objetivo común en una situación de buena aplicabilidad práctica.

Importante: Ambas partes son independientes. Comience por aquella que más le motive. Las mejoras de la parte 1 son aplicables a la parte 2.

2. Parte 1: Mejorando el Rendimiento del Ejemplo de Clases

Tal como fue visto clases, el mundo de las bodegas se puede representar como una grilla rectangular, donde cada celda está libre o es un obstáculo. En la versión simplificada vista en clases, el problema que se resuelve usando el programa en lógica consiste en desplazar n robots desde una posición inicial a una posición objetivo. En cada instante de tiempo, los robots pueden ejecutar 1 de 5 acciones posibles en $\{right, left, up, down, wait\}$.

En el siguiente URL:
`https://github.com/IIC2613/Syllabus/tree/master/Ejemplos/Logic%20Programming/Bodegas/base_t3`

podrá encontrar una modificación del ejemplo de clases, que agrega dos elementos que no se vieron durante la clase. Primero, una restricción que impide que los robots se intercambien 'pasando uno sobre el otro'. Esta restricción es muy importante para que las soluciones tengan sentido físico. Segundo, se agrega una manera sencilla de expresar que todos los robots deben llegar a destino. Estas nuevas sentencias están marcadas como `(NUEVO)` en el archivo base.

Como sabemos, el solver de ASP en su primera etapa realiza una instanciación (*grounding*). La representación instanciada es luego pasada a un solver, que es finalmente el que calcula el modelo. Dependiendo de cómo se define el modelo, la representación instanciada puede ser más o menos grande. Por otro lado, el tiempo empleado por el solver es sensible al tamaño de la representación instanciada. En resumen, no da lo mismo cómo uno modela un problema porque al momento de resolverlo se obtiene mejor o peor rendimiento dependiendo de cómo es la modelación.

En nuestro archivo de ejemplo, hay dos restricciones (marcadas como R1 y R2) que al ser instanciadas generan un número de restricciones que es **cuadrático** en el número de agentes (mire el archivo hasta convencerse de la veracidad de esta afirmación). En esta parte de la tarea, usted debe proponer una forma de reemplazar estas restricciones de tal manera que, al instanciar, se genere un número de restricciones que sea **lineal** en el número de agentes.

Para ello, utilice 5 predicados nuevos:

- $up(X, Y, T)$: representa que algún agente se movió desde (X, Y) hasta $(X, Y + 1)$ en tiempo T.

- $down(X, Y, T)$, $left(X, Y, T)$, $right(X, Y, T)$ son análogas a la anterior.
- $stay(X, Y, T)$ es verdadero cuando un agente que estaba en (X, Y) ejecutó un *wait* en el tiempo T.

Defina adecuadamente la dinámica de estos predicados y reemplace las dos reglas cuadráticas por nuevas reglas que usen estos predicados. Demuestre que la instanciación resultante es lineal en el número de agentes. Ahora **evalúe** su nuevo modelo con al menos 10 problemas aleatorios en los cuales se vean diferencias. Use una grilla de 20×20 con un 10 % de obstáculos, y compare los tiempos de ejecución de ambos modelos a medida que aumenta el número agentes.

3. Parte 2: Misión Transporte de Productos

Modelaremos ahora una variante del problema original. Pensaremos ahora que los robots tienen la habilidad de recoger productos, acarrearlos y luego depositarlos en un lugar diseñado para empaque. Para simplificar nuestro modelo, supondremos que los productos inicialmente se encuentran en celdas libres (no obstáculo) del mapa y que, como antes, los robots pueden desplazarse libremente sobre las celdas que no son obstáculos, incluyendo las celdas que contienen productos. Cada producto tiene asociada una celda destino. El objetivo es que los robots lleven a destino todos los productos. Para modelar esta parte considere lo siguiente.

Adicionalmente a las acciones que ya hemos definido, deberá agregar:

- $pick(P)$: Un robot R puede ejecutar esta acción cuando se encuentra en una celda (X, Y) en la cual está el paquete P. El efecto es que el robot R está acarreando el producto P.
- $deliver(P)$: Un robot R puede ejecutar esta acción cuando está acarreando el producto P y está sobre la celda (X, Y) que es el destino final del producto P. Tiene como efecto que el producto P está entregado.

Necesitará representar las siguientes propiedades:

- $holding(R, P, T)$: representa que el robot R está acarreando al producto P en el tiempo T.
- $at(P, X, Y, T)$: representa que el producto P está en la celda (X, Y). Sólo es verdadera cuando P no está siendo acarreado por ningún robot. Esto último es importante para evitar problemas en su modelación.
- $holding(R, P)$: representa que el producto P está siendo acarreado por el robot R.
- $delivery\_station(P, X, Y)$: representa que la celda (X, Y) es el destino final del producto P.
- $delivered(P)$: representa que el producto P está entregado.

Suponga que cada robot puede acarrear a lo más 3 productos.

Una vez funcionando, evalúe en una grilla de 10×10 sobre al menos 20 problemas con distinto número de agentes y productos. Reporte los tiempos obtenidos, saque concluisiones y escríbalas.

4. Bonus

Investigue (en la documentación/tutoriales de clingo) cómo sería posible que sus soluciones minimicen el número de acciones que no son *wait*. (En la ayudantía solo se vió como minimizar el número de pasos). ¿Qué tanto más difícil es este nuevo problema para clingo? Intente encontrar escenarios en donde se vean diferencias.

5. Qué entregar

- **Código**. Entregue dos archivos separados para la parte 1 y parte 2. Llámelos `{parte1.lp} y \verb`parte2.lp. En ambas partes, puede agregar más predicados si lo considera necesario, pero **no cambie** los nombres de los predicados ni acciones que han sido mencionados en este enunciado. Su tarea podrá ser corregida por un programa y su ayudante podría decidir aplicar descuentos si usted cambia nombres.
- **Informe**. Para la parte 1 debe incluir la demostración de que la instanciación es lineal y la comparación experimental que realizó. Para la parte 2 debe incluir su evaluación experimental, describiendo las configuraciones que usó.