

Informe de trabajo Práctico N°2: Razonamiento

Olguin Nahuel

Molina Matías

Maccapa Luis

1. Desarrollo y análisis de una base de conocimiento para una planta de reducción de presión de gas

Se ingresan las reglas axiomáticas y los ground facts.

Los ground facts son los hechos que se dan en una instancia específica del dominio. Estos contienen toda la información necesaria para recorrer el árbol, se basan en cada pregunta que contiene el árbol. Estos se programaron de tal manera que se pueden modificar dinámicamente con assert y retract de acuerdo a lo que se modifique en la planta.

2. Utilizando el planificador [Fast Downward](#)

a. Modelar en PDDL el dominio de transporte aéreo de cargas y definir algunas instancias del problema

Para la realización de este ejercicio se partió del código de Carga aérea subido en aulabierta ; del mismo se utilizó el dominio tal cual estaba y se realizó un código de problema nuevo con varias instancias de cada elemento(aviones,aeropuertos,cargas).

El resultado de la corrida del programa dio el siguiente resultado.

```
(cargar yerba aa01 pss)
(volar aa01 pss brc)
(cargar chocolates aa01 brc)
(descargar yerba aa01 brc)
(volar aa01 brc mdz)
(descargar chocolates aa01 mdz)
(cargar vino aa01 mdz)
(volar aa01 mdz pss)
(descargar vino aa01 pss)
(cargar fertilizante aa04 aep)
(volar aa04 aep rsa)
(descargar fertilizante aa04 rsa)
(cargar soja aa04 rsa)
(volar aa04 rsa aep)
(descargar soja aa04 aep)
(cargar semillas aa05 eze)
(volar aa05 eze rsa)
(descargar semillas aa05 rsa)
; cost = 18 (unit cost)
```

b. Modelar en PDDL y definir al menos una instancia del problema para alguna de las siguientes opciones

- **Planificación de Procesos Asistida por Computadora (CAPP, Computer-Aided Process Planning).**

Para la realización de este ejercicio se partió del código de CAPP subido en aulabierta ; del mismo se utilizó el dominio tal cual estaba y en el problema se agregaron las instancias de cada elemento faltante.

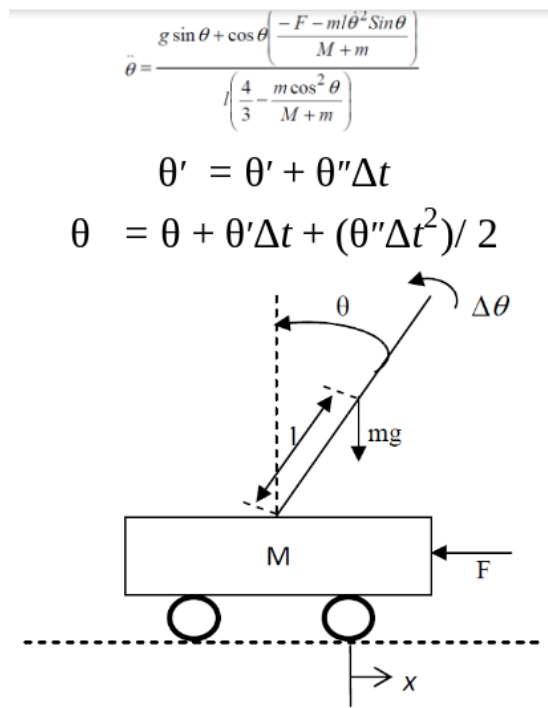
En este problema se observó que si se instanciaba varios elementos el problema no convergia.

Una de las corridas del programa dio el siguiente resultado.

(op-fresado orientacion-x s4 slot fresado)
 (setup-orientacion orientacion-x orientacion+x)
 (op-fresado orientacion+x s2 slot fresado)
 (op-fresado orientacion+x s6 slot fresado)
 (setup-orientacion orientacion+x orientacion+z)
 (op-fresado orientacion+z h1 blind-hole fresado)
 (op-fresado orientacion+z s10 slot fresado)
 (op-fresado orientacion+z s9 slot fresado)
 ; cost = 8 (unit cost)

3. Implementar un sistema de inferencia difusa para controlar un péndulo invertido

- Asuma que el carro no tiene espacio restringido para moverse
- Definir variables lingüísticas de entrada y salida, particiones borrosas, operaciones borrosas para la conjunción, disyunción e implicación, reglas de inferencia (cubrir todas las posibles combinaciones de valores borrosos de entrada en la base de reglas)
- Utilice el siguiente modelo del sistema carro-péndulo



Para la implementación del problema el mismo se dividió en 3 códigos

1. Fborrosificador :

En este código se desarrollaron todas las funciones y métodos para poder pasar del dominio nítido al dominio borroso y también su inversa (de dominio borroso a dominio nítido) de el desplazamiento angular, la velocidad angular y la fuerza.

2. FAM:

En este código se desarrolló como tratar las variables de entrada : desplazamiento angular y velocidad angular ,para obtener la salida: fuerza.

3. menu principal:

Este código utiliza las funciones de los dos códigos anteriormente mencionados y a partir de los datos de entrada nítidos y el valor de la FAM deseado , obtener la fuerza necesaria para mantener el carro en equilibrio.