

## Práctico N° 2 Razonamiento

### LINEAMIENTOS GENERALES

Se deberá entregar:

- El código fuente de todos los programas, dominios y bases de conocimiento desarrollados, junto con archivos de datos (si fuera necesario)
- Un informe que incluya
  - Documentación del algoritmo y parámetros utilizados en el ejercicio n° 1 (funciones de pertenencia, particiones borrosas, mecanismos de inferencia, reglas, etc). Incluya gráficas y tablas con el rendimiento del sistema en distintos escenarios y configuraciones del controlador difuso
  - Documentación de la base de conocimientos y modelo de planning de los ejercicios 2 y 3. En caso de desarrollar modelos en dominios diferentes a los propuestos, incluir una descripción de los mismos. También presentar todo otro elemento de documentación que considere relevante

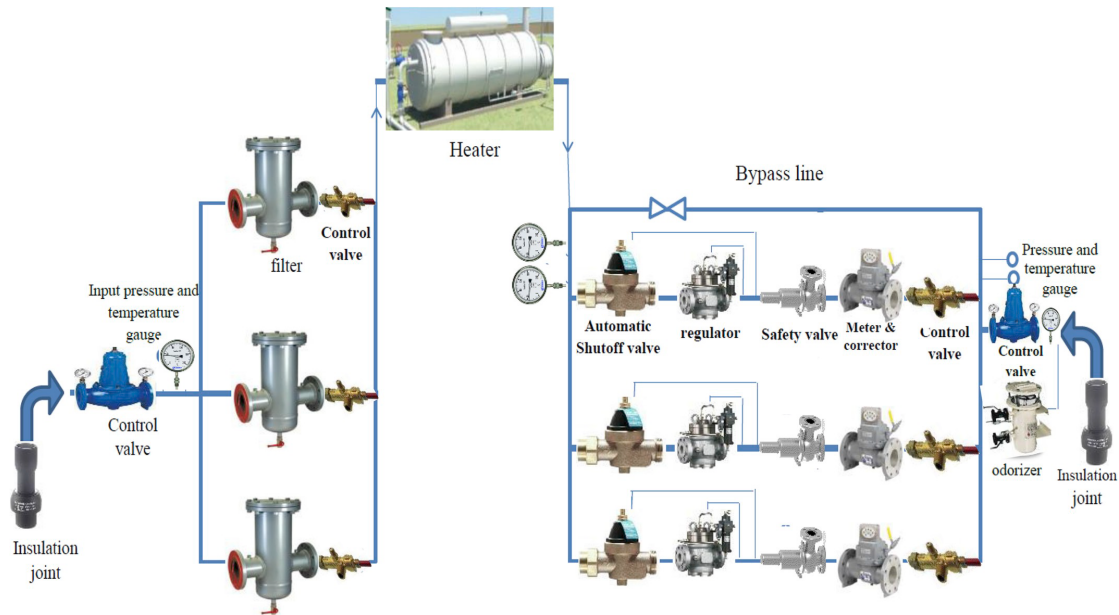
### EJERCICIOS

1. **Desarrolle una base de conocimientos (KB, Knowledge Base) en PROLOG para alguna de las alternativas que se presentan más abajo. Además de la base de conocimientos, plantee preguntas que el sistema puede contestar, incluyendo preguntas cerradas (verdadero/falso) y preguntas abiertas (ej: qué debe hacerse?). Además de las reglas axiomáticas, incluya algunos ground facts (hechos) necesarios para codificar una instancia del problema.**

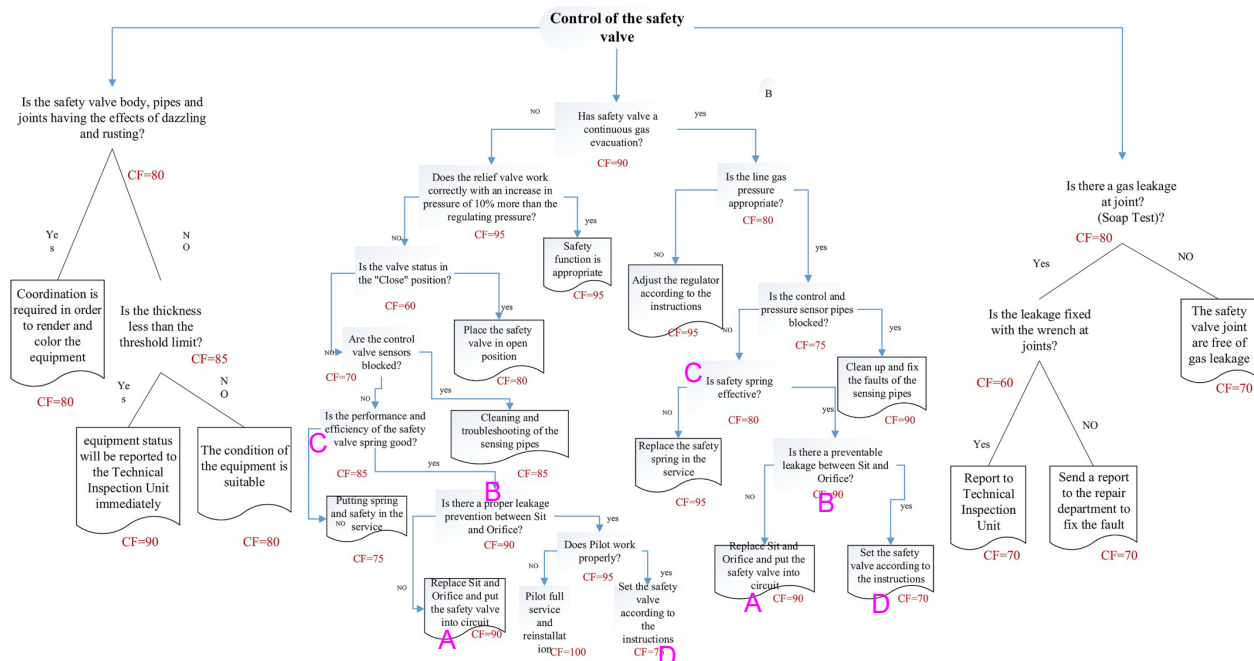
**Alternativas:**

- Un dominio a su elección
- Evaluación y mantenimiento de una válvula de seguridad de una estación de reducción de presión de gas.

Un esquema de una estación de reducción de presión de gas natural puede verse en la siguiente figura:



El proceso de revisión y mantenimiento (simplificado) puede describirse mediante el árbol de decisión que se presenta a continuación. Complete las reglas axiomáticas y los ground facts necesarios para que la KB pueda funcionar, contestar preguntas e indicar acciones de mantenimiento a realizar. Por ejemplo, si se debe verificar el espesor del cuerpo de la válvula con respecto a un cierto umbral o threshold, se deben incluir ground facts para expresar cuál es el espesor de la válvula y cuál es ese threshold, ya sea por inclusión directa en la KB (ground fact), o por inferencia (ej: tomando información de catálogo para diversas marcas/modelos de válvula, y luego incluyendo un ground fact que indique qué marca/modelo de válvula está instalada). Los CF (Confidence Factors) no deben tenerse en cuenta en este ejercicio.



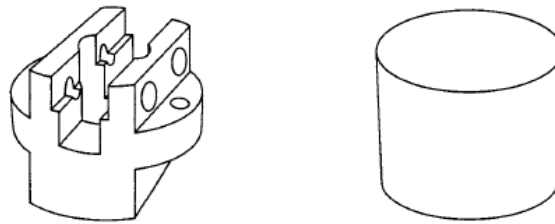
Se podría poner un predicado que se resuelva ejecutando otro programa que permita sensor de forma electrónica una variable del mundo real

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

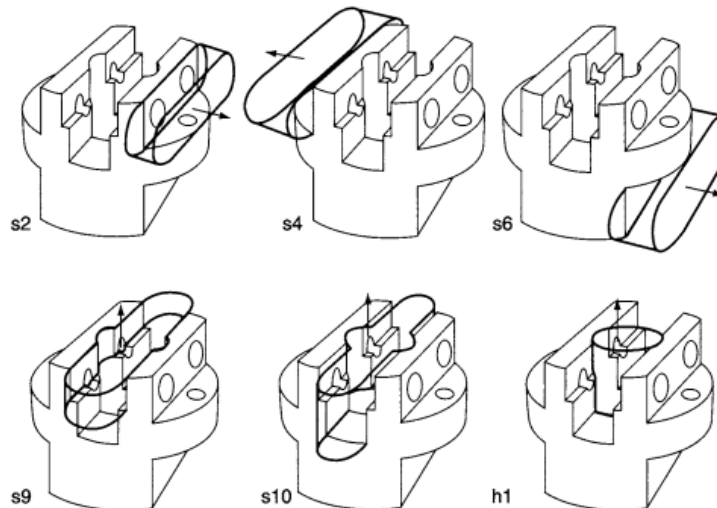
- a. Modelar en PDDL el dominio de transporte aéreo de cargas y definir algunas instancias del problema
- b. Modelar en PDDL y definir al menos una instancia del problema para alguna de las siguientes opciones
  - Un dominio a su elección
  - Planificación de Procesos Asistida por Computadora (CAPP, Computer-Aided Process Planning).

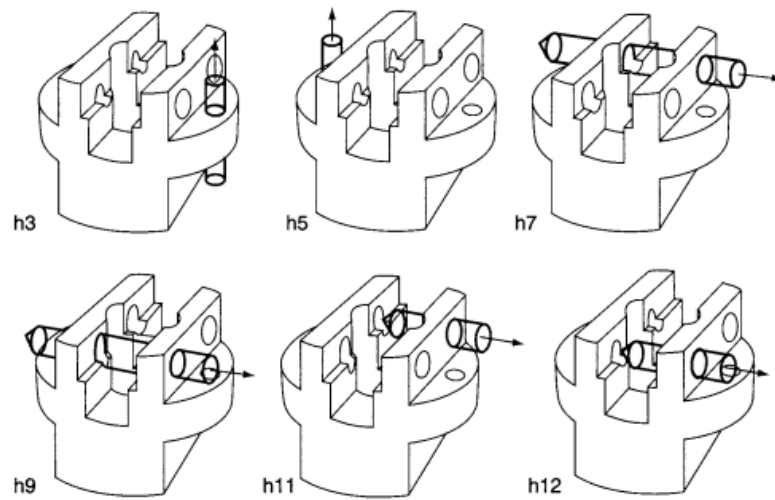
CAPP puede describirse como la definición de los procesos de manufactura necesarios para producir una parte o un producto, junto con su secuencia, setups, máquinas, herramientas y fijaciones requeridas, así como los parámetros de manufactura (generalmente maquinado, como ser velocidades, avances, etc). Una parte está compuesta por diversas *características o features* (por ejemplo, agujeros pasantes, agujeros ciegos, chaflanes, etc), las cuales pueden ser maquinadas mediante distintos procesos (ejemplo taladrado, fresado, etc) utilizando distintas combinaciones de máquinas, herramientas y orientaciones de la pieza (el denominado “setup”).

Por ejemplo, en la siguiente figura se muestra a la izquierda una pieza terminada, y a la derecha la materia prima de la cual se parte.

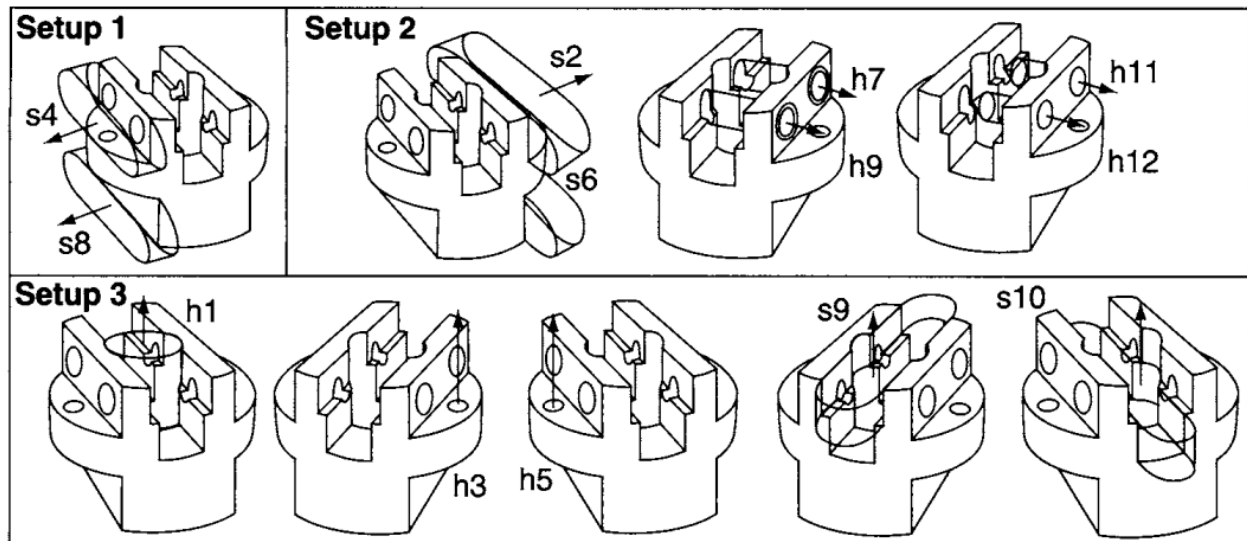


La CAPP consiste en la definición de todos los procesos de fabricación para llegar desde la materia prima al producto terminado, incluyendo setups de máquina, y considerando restricciones de precedencia. Siguiendo con el ejemplo de la pieza anterior, en la siguiente figura se muestran algunas de las features que deben maquinarse.





El modelo del dominio debe contener acciones para producir distintos setups de máquina (cambiar la orientación de la pieza, montar una herramienta en la máquina, etc), y ejecutar procesos de maquinado. En la siguiente figura se observan algunos setups, particularmente los relacionados con la orientación de la pieza (pero no los necesarios para cambiar la herramienta, u otros), necesarios para fabricar las features.



Asuma que el estado inicial contiene la lista de características que deben producirse, junto con restricciones de precedencia.

### 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

$$\ddot{\theta} = \frac{g \sin \theta + \cos \theta \left( \frac{-F - ml\dot{\theta}^2 \sin \theta}{M + m} \right)}{l \left( \frac{4}{3} - \frac{m \cos^2 \theta}{M + m} \right)}$$

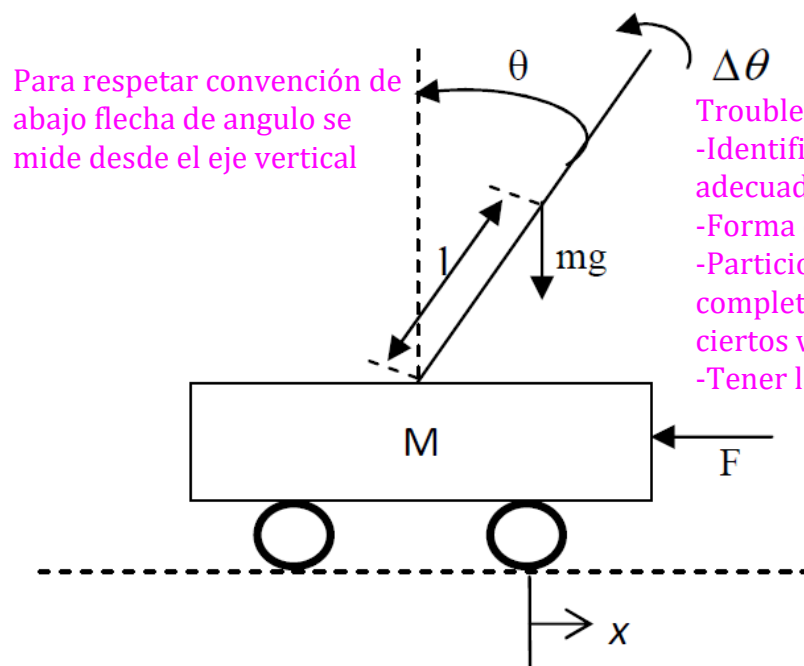
$$\theta' = \theta' + \theta'' \Delta t$$

$$\theta = \theta + \theta' \Delta t + (\theta'' \Delta t^2) / 2$$

-Validar modelo de forma separada a la del controlador.

-Si la implementación y las ec es son correctas pasas a modelar la FAM o las reglas eq.

-Podes usar la FAM como modelo para luego armar las reglas



Para respetar convención de abajo flecha de angulo se mide desde el eje vertical

Troubleshooting:

-Identificar si algunas reglas no son adecuadas.

-Forma de funciones de pertenencia.

-Particiones borrosas deben ser completas. Sino no sabe qué hacer para ciertos valores.

-Tener log de todo lo que sucedió.

Asumimos:

theta, theta' --> Positivos sentido aguja del reloj

theta=0 vertical, pos a la der y neg a la izq.

F positivo hacia la izquierda

### Base de reglas



- Una herramienta de modelado: Fuzzy Abstract Machine (FAM)

FAM

Theta	NG	NP	Z	PP	PG
Theta'	NG	NP	Z	PP	PG
NG	?	?	PG	PP	?
NP	?	?	?	Z	?
Z	?	?	Z	?	NG
PP	?	?	NP	NG	NG
PG	?	?	?	NG	?

### Reglas equivalentes

Este caso es menos gráfico, pero permite representar combinaciones de más variables.

...  
Theta is PP and Theta' is NG then F is PP

...  
Theta is PG and Theta' is PP then F is NG

Como tenemos que programarlo hay que usar las reglas en el código

- Procesamiento de reglas con el mismo consecuente **Hacer este preprocesamiento**  $\text{MAX}(\text{MIN}(A1, A2), \text{MIN}(B1, B2))$  or entre reglas, and dentro de la regla

Variable lingüísticas:

- Ángulo ( $\theta$ )
- Velocidad Angular ( $\theta'$ )
- Fuerza ( $F$ )

Particiones Borrosas:

- NG,NP,Z,PP,PG con set\_width de:
  - $\theta: 60^\circ$
  - $\theta' = 1 \text{ rad/s}$
  - $F = 170 \text{ N}$

Operaciones Borrosas:

- Conjunción:  $\min$  (entre  $f$  de pertenencias)
- Disyunción:  $\max$
- Implicación:  $\min$  entre conjunto borroso de salida y valor del antecedente. Da conjunto

Reglas de Inferencia:

- Se deducen de la FAM