Práctico Nº 2
Razonamiento

Alumnos:

- **Luciano Esperlazza**
- **Tejerina Nicolás Ezequiel**

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 nº 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

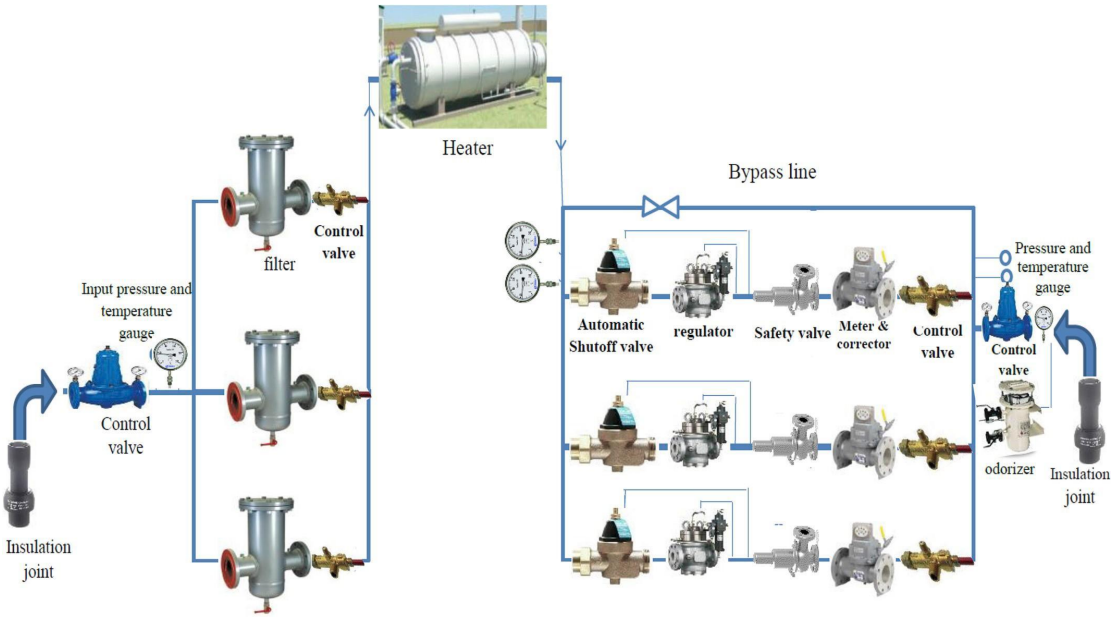
1. **Desarrolle una base de conocimientos (KB, Knowledge Base) en PROLOG para alguna de las alternativas que se presentan más abajo. Además de la base de conocimientos, plantee preguntas que el sistema puede contestar, incluyendo preguntas cerradas (verdadero/falso) y preguntas abiertas (ej: qué debe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

Alternativas:

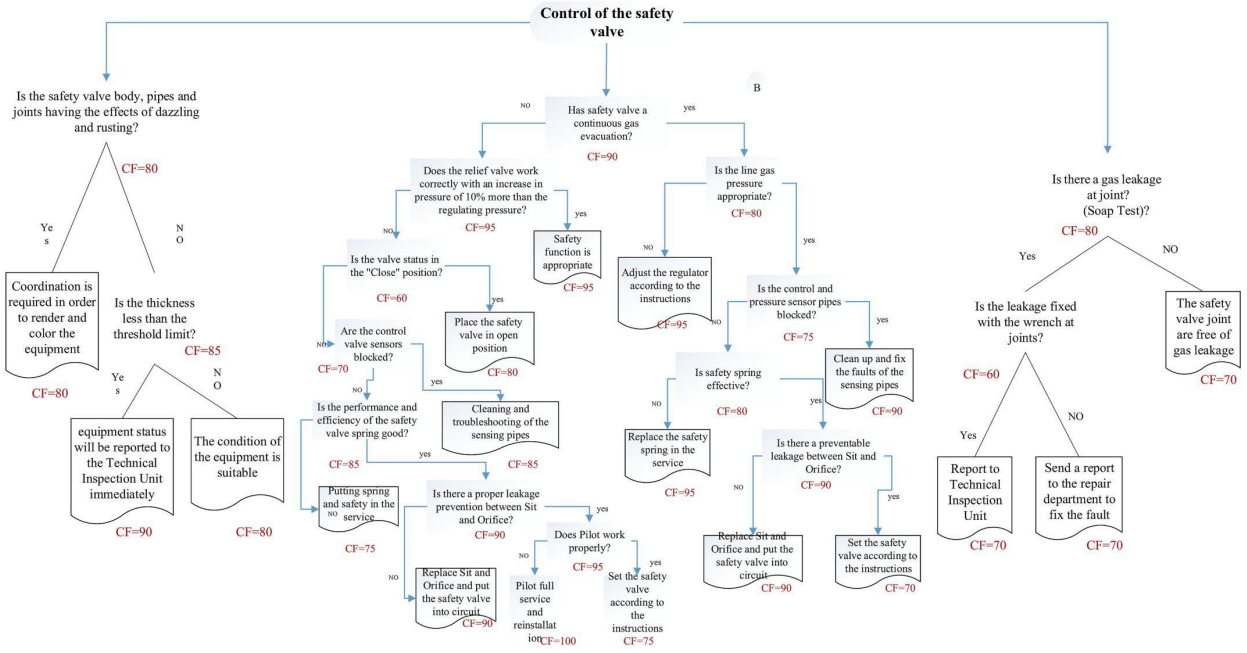
- Un dominio a su elección

- 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.



2. **Utilizando el planificador [Fast Downward]**
   a. **Modelar en PDDL el dominio de transporte aéreo de cargas y definir algunas instancias del problema**

Para modelar estos problemas usamos Fast-Downward Planning on the Web

Para el modelado del transporte aéreo de cargas tenemos:

Tenemos como estado inicial:

```
(:init
    (avion LA01)
    (avion LA02)
    (avion LA03)
    (avion AA01)
    (avion AA02)
    (avion AA03)
    (avion FB01)
    (avion FB02)
    (avion FB03)
    (aeropuerto MDZ)
    (aeropuerto AEP)
    (aeropuerto COR)
    (aeropuerto SFN)
    (carga FERTILIZANTE)
    (carga TELA-GRANIZO)
    (carga COSECHADORA)
    (carga AUTOPARTES)
    (en LA01 MDZ)
    (en LA02 AEP)
    (en LA03 COR)
    (en AA01 SFN)
    (en AA02 MDZ)
    (en AA03 AEP)
    (en FB01 COR)
    (en FB02 AEP)
    (en FB03 SFN)
    (en FERTILIZANTE AEP)
    (en TELA-GRANIZO SFN)
    (en COSECHADORA MDZ)
    (en AUTOPARTES COR)
)
```

(En el estado inicial declaramos que es cada cosa, y donde se encuentra)

Como estado objetivo:

```
(:goal
    (and
        (en FERTILIZANTE SFN)
        (en TELA-GRANIZO MDZ)
        (en COSECHADORA COR)
        (en AUTOPARTES AEP)
    )
)
```

Vemos que se ha encontrado una solución en un tiempo muy rápido.

```
[t=0.0167858s, 34608 KB] Expanded until last jump: 0 state(s).
[t=0.0167858s, 34608 KB] Reopened until last jump: 0 state(s).
[t=0.0167858s, 34608 KB] Evaluated until last jump: 1 state(s).
[t=0.0167858s, 34608 KB] Generated until last jump: 0 state(s).
[t=0.0167858s, 34608 KB] Number of registered states: 406
[t=0.0167858s, 34608 KB] Int hash set load factor: 406/512 = 0.792969
[t=0.0167858s, 34608 KB] Int hash set resizes: 9
[t=0.0167858s, 34608 KB] Search time: 0.00913139s
[t=0.0167858s, 34608 KB] Total time: 0.0167858s
Solution found.
Peak memory: 34608 KB
Remove intermediate file output.sas
search exit code: 0
```

Y las operaciones que se obtienen son:

Plan

```
(cargar autopartes la03 cor)
(volar la03 cor aep)
(cargar fertilizante la03 aep)
(descargar autopartes la03 aep)
(volar la03 aep sfn)
(cargar tela-granizo la03 sfn)
(descargar fertilizante la03 sfn)
(volar la03 sfn mdz)
(descargar tela-granizo la03 mdz)
(cargar cosechadora la03 mdz)
(volar la03 mdz cor)
(descargar cosechadora la03 cor)
; cost = 12 (unit cost)
```

Luego para definir otras instancias del problema, se agregaron pasajeros con su respectivo lugar de origen y hacia dónde se dirigen. También se agregó un taller de reparación al cual debe viajar un avión para su mantenimiento/arreglo. Se agregaron las acciones para volar al taller, y para que suban y bajen los pasajeros.

Como estado inicial:

```
(:init

    (avion LA01)
    (avion LA02)
    (avion LA03)
    (avion AA01)
    (avion AA02)
    (avion AA03)
    (avion FB01)
    (avion FB02)
    (avion FB03)
    (aeropuerto MDZ)
    (aeropuerto AEP)
    (aeropuerto COR)
    (aeropuerto SFN)
    (taller AVIOCENTRO-SERVICIOS-DE-AVIACION)
    (carga FERTILIZANTE)
    (carga TELA-GRANIZO)
    (carga COSECHADORA)
    (carga AUTOPARTES)
    (pasajeros PASAJEROS-HACIA-MDZ)
    (pasajeros PASAJEROS-HACIA-SFN)

    (en LA01 MDZ)
    (en LA02 AEP)
    (en LA03 COR)
    (en AA01 SFN)
    (en AA02 MDZ)
    (en AA03 AEP)
    (en FB01 COR)
    (en FB02 AEP)
    (en FB03 SFN)
    (en FERTILIZANTE AEP)
    (en TELA-GRANIZO SFN)
    (en COSECHADORA MDZ)
    (en AUTOPARTES COR)
    (en PASAJEROS-HACIA-MDZ AEP)
    (en PASAJEROS-HACIA-SFN COR)
)
```

Tenemos como objetivo:

```
(:goal
    (and
        (en FERTILIZANTE SFN)
        (en TELA-GRANIZO MDZ)
        (en COSECHADORA COR)
        (en AUTOPARTES AEP)
        (en PASAJEROS-HACIA-MDZ MDZ)
        (en PASAJEROS-HACIA-SFN SFN)
        (en LA01 AVIOCENTRO-SERVICIOS-DE-AVIACION)


    )
```

Se encontró una solución (esta vez demoró un poco más aunque sigue siendo una búsqueda muy rápida):

```
[t=0.0292142s, 34740 KB] Int hash set load factor: 875/1024 = 0.854492
[t=0.0292142s, 34740 KB] Int hash set resizes: 10
[t=0.0292142s, 34740 KB] Search time: 0.0201911s
[t=0.0292142s, 34740 KB] Total time: 0.0292142s
Solution found.
Peak memory: 34740 KB
Remove intermediate file output.sas
search exit code: 0
```

Y las operaciones obtenidas son:

Plan

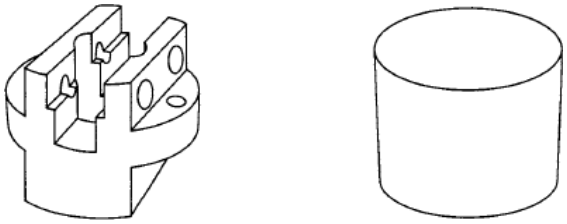
```
(subir_pasajeros pasajeros-hacia-sfn la03 cor)
(volar la03 cor aep)
(subir_pasajeros pasajeros-hacia-mdz la03 aep)
(volar la03 aep sfn)
(cargar tela-granizo la03 sfn)
(bajar_pasajeros pasajeros-hacia-sfn la03 sfn)
(volar la03 sfn mdz)
(descargar tela-granizo la03 mdz)
(bajar_pasajeros pasajeros-hacia-mdz la03 mdz)
(cargar cosechadora la03 mdz)
(volar la03 mdz cor)
(descargar cosechadora la03 cor)
(cargar autopartes la03 cor)
(volar la03 cor aep)
(cargar fertilizante la03 aep)
(descargar autopartes la03 aep)
(volar la03 aep sfn)
(descargar fertilizante la03 sfn)
(volar_al_taller la01 mdz aviocentro-servicios-de-aviacion)
; cost = 19 (unit cost)
```

Las cuales verifican los estados objetivos.

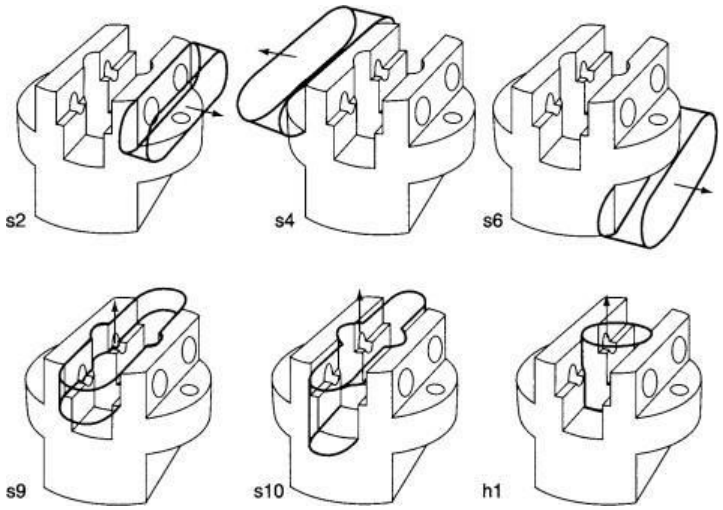
b. Modelar en PDDL y definir al menos una instancia del problema para alguna de las siguientes opciones
   - **Un dominio a su elección**
   - **Planificación de Procesos Asistida por Computadora (CAPP, Computer-Aided Process Planning).**

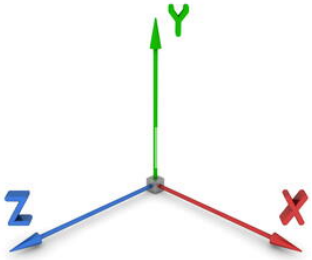
CAPP puede describirse como la definición de los procesos de manufactura necesarios para producir una parte o un producto, junto con su secuencia, setups, máquinas, herramientas y fijaciones requeridas, así como los parámetros de manufactura (generalmente maquinado, como ser velocidades, avances, etc). 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”).

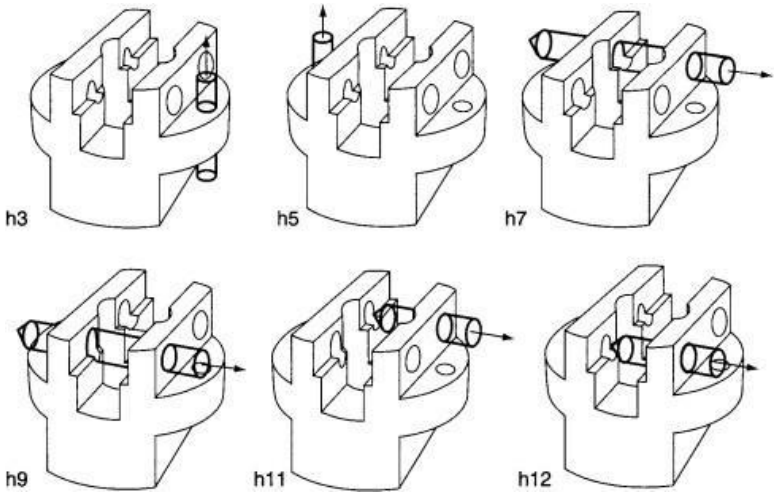
Por ejemplo, en la siguiente figura se muestra a la izquierda una pieza terminada, y a la derecha la materia prima de la cual se parte.



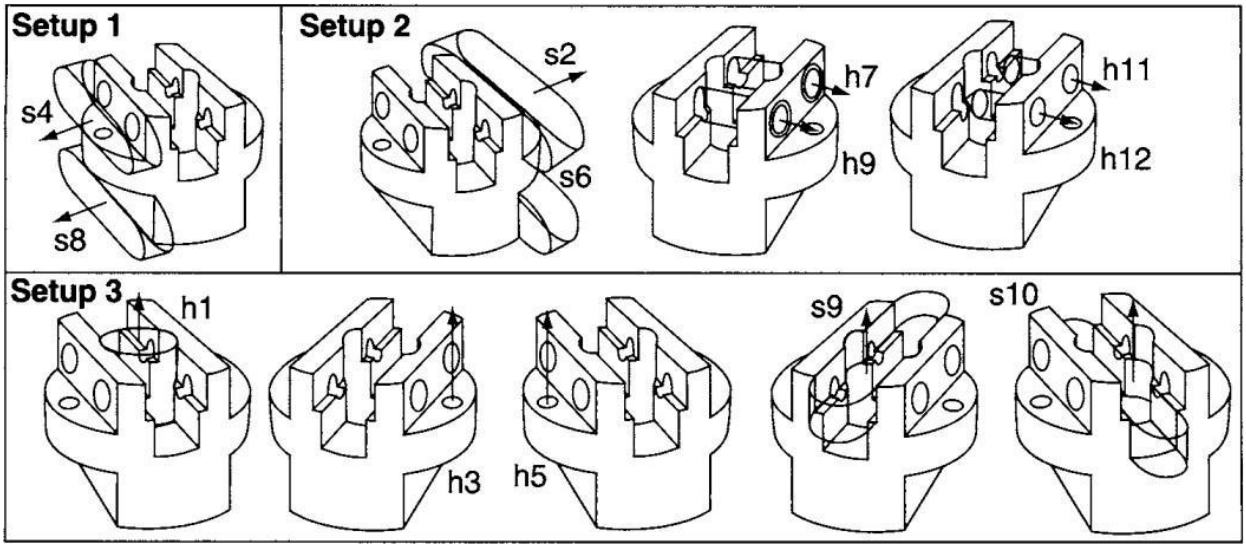
La CAPP consiste en la definición de todos los procesos de fabricación para llegar desde la materia prima al producto terminado, incluyendo setups de máquina, y considerando restricciones de precedencia. Siguiendo con el ejemplo de la pieza anterior, en la siguiente figura se muestran algunas de las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

Tenemos como estado inicial:

```
(orientacion-pieza orientacion-x)
(orientacion-feature s2 orientacionx)
(orientacion-feature s4 orientacion-x)
(orientacion-feature s6 orientacionx)
(orientacion-feature s9 orientacionz)
(orientacion-feature s10 orientacionz)
(orientacion-feature h1 orientacionz)
(orientacion-feature h3 orientacionz)
(orientacion-feature h5 orientacionz)
(orientacion-feature h7 orientacionx)
(orientacion-feature h9 orientacionx)
(orientacion-feature h11 orientacionx)
(orientacion-feature h12 orientacionx)
(fabricable slot fresado)
(fabricable through-hole taladrado)
```

Como meta:

```
(:goal
    (and
        (fabricada s2)
        (fabricada s4)
        (fabricada s6)
        (fabricada s9)
        (fabricada s10)
        (fabricada h1)
        (fabricada h3)
        (fabricada h5)
        (fabricada h7)
        (fabricada h9)
        (fabricada h11)
        (fabricada h12)
    )
)
```

Observamos que se encontró solución y el tiempo de búsqueda:

```
[t=0.00643333s, 34476 KB] Generated until last jump: 0 state(s).
[t=0.00643333s, 34476 KB] Number of registered states: 98
[t=0.00643333s, 34476 KB] Int hash set load factor: 98/128 = 0.765625
[t=0.00643333s, 34476 KB] Int hash set resizes: 7
[t=0.00643333s, 34476 KB] Search time: 0.00210042s
[t=0.00643333s, 34476 KB] Total time: 0.00643333s
Solution found.
Peak memory: 34476 KB
Remove intermediate file output.sas
search exit code: 0
```

Las operaciones realizadas son:

Plan

```
(op-fresado orientacion-x s4 slot fresado)
(setup-orientacion orientacion-x orientacionx)
(op-fresado orientacionx s2 slot fresado)
(op-fresado orientacionx s6 slot fresado)
(op-taladrado orientacionx h11 through-hole
taladrado)
(op-taladrado orientacionx h12 through-hole
taladrado)
(op-taladrado orientacionx h7 through-hole
taladrado)
(op-taladrado orientacionx h9 through-hole
taladrado)
(setup-orientacion orientacionx orientacionz)
(op-fresado orientacionz s10 slot fresado)
(op-fresado orientacionz s9 slot fresado)
(op-taladrado orientacionz h1 through-hole
taladrado)
(op-taladrado orientacionz h3 through-hole
taladrado)
(op-taladrado orientacionz h5 through-hole
taladrado)
; cost = 14 (unit cost)
```

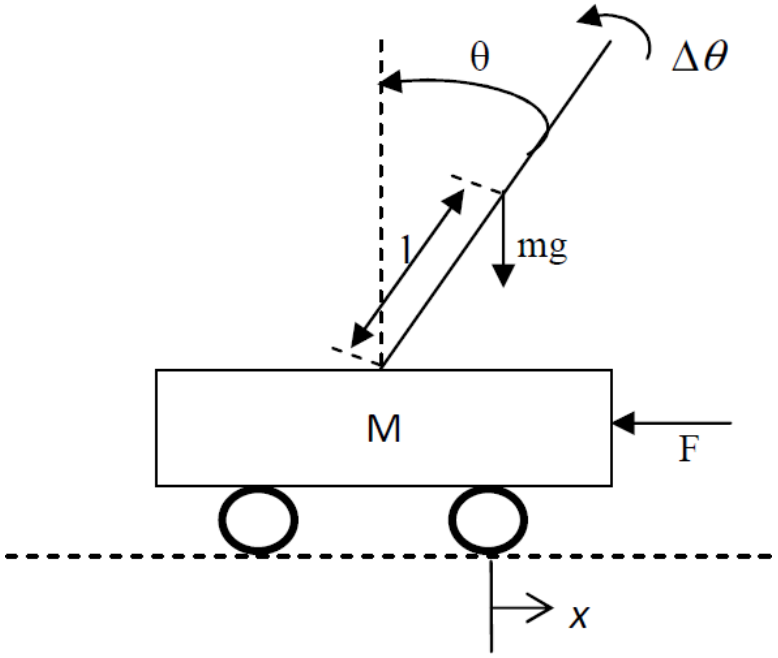
Explicando brevemente el plan:
Inicialmente al estar ya orientada en -x, hace el fresado s4
luego se reorienta para poner en posición x
hace los fresados s2,s6, los taladrados h11,h12,h7yh9
se reorienta a la posición z
hace los fresados s10, s9, y finalmente los taladrados h1,h3,h5

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

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

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

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



Para las tres variables (entradas: ángulo theta, velocidad ; salidas: fuerza) utilizamos los mismos conjuntos borrosos:
NG: negativo grande
NP: negativo pequeño
Z: cero
PP: positivo pequeño
PG: positivo grande

Definimos los conjuntos superpuestos en un 50%

Universo del discurso:

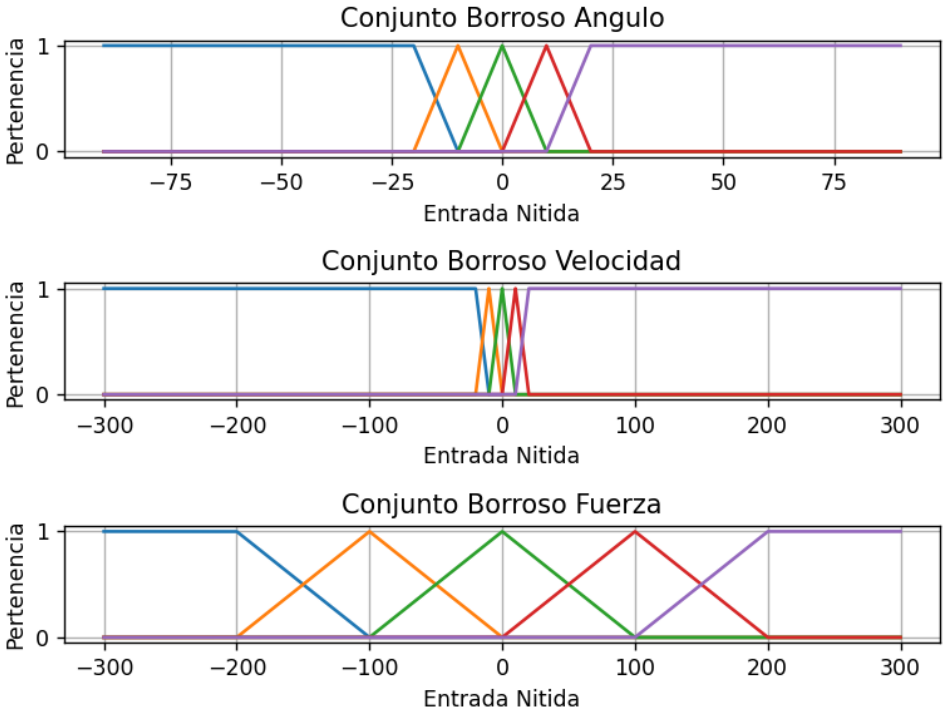
Ángulo: [-90 , 90]
Velocidad: [-300 , 300]
Fuerza: [-300 , 300]

Las reglas de inferencia utilizadas fueron:

Theta / Velocidad	NG	NP	Z	PP	PG
NG	NG	NG	NG	NP	Z
NP	NG	NG	NP	Z	PP
Z	NG	NP	Z	PP	PG
PP	NP	Z	PP	PG	PG
PG	Z	PP	PG	PG	PG

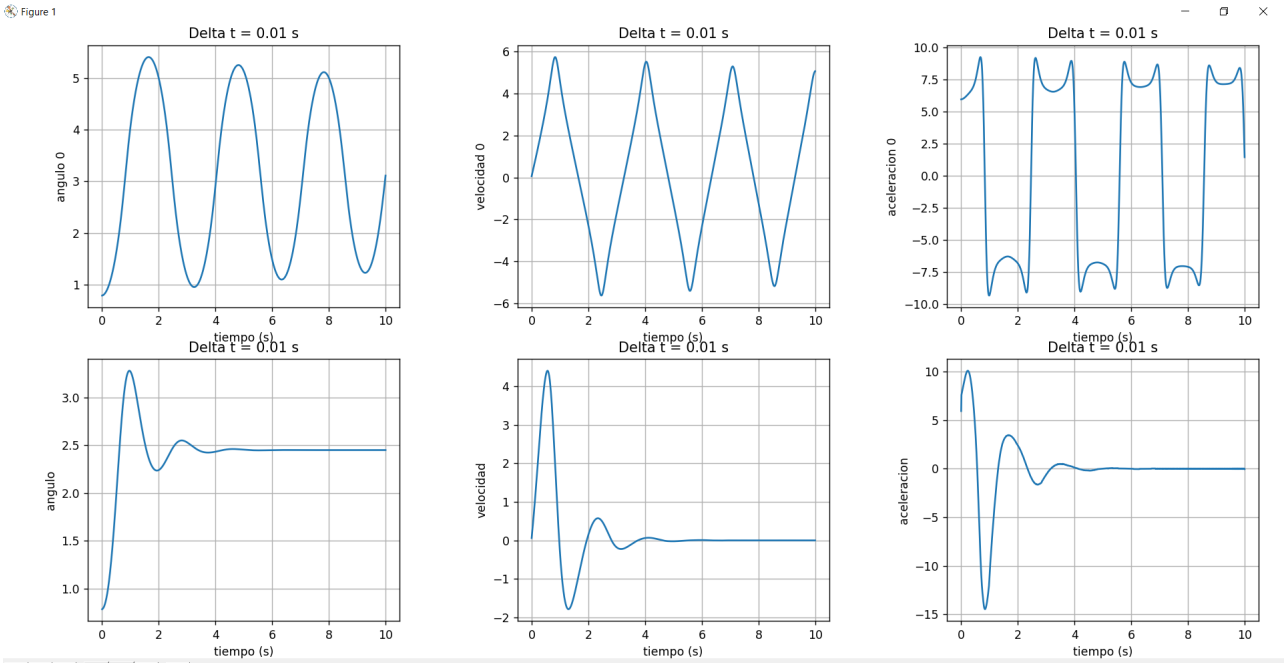
Donde tenemos como entradas a la velocidad y el ángulo como entradas, y la fuerza como salida.

Utilizamos una función borrosificador ( ) que recibe como parámetros los universos del discurso de las 3 variables, y los intervalos borrosos.



A la velocidad y el ángulo los creamos como objetos con los atributos “inicio”, que nos indica el valor inicial del universo del discurso (lo utilizamos para calcular la pertenencia), una lista llamaba “conborroso” que almacena los valores luego de aplicar el borrosificador, y otra lista llamada “pertenencia” que alberga los valores de pertenencia para cada valor del conjunto. Además tenemos un método llamado “obtenerPertenencia” que se encarga de calcular el valor de pertenencia para cada valor de ángulo y velocidad que vamos teniendo durante la ejecución del programa.
Tenemos también la clase Salida, para la cual creamos el objeto Fuerza, con los mismos atributos de las entradas, pero sin el método para obtener la pertenencia, dado que este dato lo vamos a calcular utilizando la función baseConocimiento ( ) donde aplicamos las reglas de inferencia mostradas anteriormente.
Por último presentamos el desborrosificador ( ), utilizando la media de centros, con la siguiente fórmula:

$$y = \frac{\sum_{l=1}^{l=M} \bar{y}^{l} \mu_{B'}(\bar{y}^{l})}{\sum \mu_{B'}(\bar{y}^{l})}$$



Vemos en la parte superior las salidas sin aplicar el sistema de inferencia difusa, y debajo vemos la acción del mismo al amortiguar el movimiento.