UNIVERSIDAD
DE GRANADA

Ingeniería del Conocimiento
grado en ingeniería informática

PRÁCTICA FINAL

SBC para el riego automático de cultivos y plantas

Autor
Vladislav Nikolov Vasilev

DNI
X8743846M

Rama
Computación y Sistemas Inteligentes

Escuela Técnica Superior de Ingenierías Informática y de Telecomunicación

Curso 2018-2019

Índice

1. Resumen del funcionamiento del sistema **2**

2. Descripción del proceso de creación del sistema **3**
2.1. Procedimiento seguido para el desarrollo de la base de conocimiento 3
2.2. Procedimiento de validación y verificación del sistema 4
2.2.1. Verificación 4
2.2.2. Validación 5

3. Descripción del sistema **6**
3.1. Variables de entrada del problema 6
3.2. Variables de salida del problema 7
3.3. Conocimiento global del sistema 8
3.4. Especificación de los módulos desarrollados 10
3.5. Hechos y reglas de cada módulo 11
3.5.1. Reglas generales 12
3.5.2. Módulo Riego 14
3.5.3. Módulo CalculoSiguienteLluvia 15
3.5.4. Módulo Vaporizacion 16

4. Manual de uso del sistema **17**

Referencias **19**

1. Resumen del funcionamiento del sistema

El sistema ha sido implementado utilizando **CLIPS** y **Python3**, más concretamente la versión 3.7.1 de Python. Para poder utilizar el sistema, se recomienda tener instalada una versión de Python igual o superior a la 3.7.1, además de tener instalados en el equipo los módulos *numpy* [1] y *scikit-fuzzy* [2], los cuáles son utilizados por el programa.

El sistema implementado en CLIPS es el sistema experto principal. Éste utiliza un pequeño programa escrito en Python que se encarga de realizar las deducciones correspondientes al riego de las plantas mediante un sistema basado en *lógica difusa*. Los dos programas se comunican enviándose la información necesaria para realizar las deducciones y las deducciones mismas mediante archivos.

El sistema principal se encuentra en *sistema.clp*, mientras que *sensores-init.clp* contiene información inicial relativa a los sensores (para no tener que insertar unos datos iniciales a mano) y *previsiones.clp* contiene una serie de previsiones del tiempo que se recomienda encarecidamente cargar en el sistema para que pueda funcionar de forma correcta, ya que el sistema supone que contiene información completa sobre la previsión del tiempo para las siguientes horasr. Estas previsiones pueden ser modificadas, tal y como ser indicará posteriormente. El archivo *fuzzy.py* contiene el sistema basado en lógica difusa que se encarga de realizar las deducciones para el riego de las plantas.

En general, el funcionamiento del sistema es bastante simple. El sistema experto va recibiendo valores de entrada de los sensores relativos a la **humedad**, **temperatura** y **luminosidad** de las plantas, además de valores de predicciones del tiempo. Con estos datos, el sistema decide si activar el **vaporizador** en aquellas plantas que considere oportunas con tal de reducir su temperatura o si activar el sistema de **riego** de las plantas, utilizando para ello los valores de los sensores. El sistema experto principal le envía los datos al sistema difuso para que se encargue de decidir cuál es el mejor curso de acción. Una vez decidido, envía los resultados de vuelta, y el sistema experto manipula esta información utilizando las predicciones de las próximas 8 horas con el objetivo de determinar cuál es el mejor curso de acción a tomar.

El sistema también permite pasar las horas una a una, simulando el paso del tiempo y actualizando el conocimiento de forma correspondiente, deduciendo por ejemplo si en la hora actual está lloviendo según la previsión que había para esa hora.

2. Descripción del proceso de creación del sistema

2.1. Procedimiento seguido para el desarrollo de la base de conocimiento

Para el desarrollo del sistema, se han ido satisfaciendo los requisitos especificados uno a uno. Estos requisitos son los siguientes:

1. Ser consciente de las necesidades de humedad de cada cultivo. Se ha de considerar, al menos, tres tiestos con rangos diferentes de humedad ideal.
2. Activar el riego automático de cada tiesto independientemente cuando dicho cultivo lo necesite.
3. Desactivar el riego de cada tiesto cuando se llegue a la humedad deseada.
4. Realizar un riego más inteligente en función del momento del día, considerando la posible evaporación.
5. Evitar, siempre que se pueda, el riego en situaciones de altas temperaturas y sol. No se podrá posponer en presencia de valores de humedad críticamente bajos.
6. Activar vaporizadores para refrescar el ambiente de cada tiesto y reducir altas temperaturas.
7. Posibilitar la combinación del uso de vaporizadores con el riego en momentos de altas temperaturas, cuando sea estrictamente necesario.
8. Utilizar la información meteorológica de la zona local para gestionar el riego de manera más eficiente, como posponer o regar con menor intensidad en caso de lluvia, según su intensidad y la certeza de la predicción.

Primero se ha comenzado especificando las plantas a utilizar, que en este caso han sido un **cactus**, un **tulipán** y una **palmera**, y se han especificado los valores mínimos y máximos de humedad, temperatura y luminosidad, además de los valores críticos de humedad y temperatura.

Sobre estos datos se ha añadido la capacidad de activar y desactivar el riego de forma automática para cada planta según su humedad. Posteriormente, se introdujo el sistema difuso escrito en Python3 que se encargaba de determinar si regar las plantas o no y con qué intensidad se deben regar en función de la información proporcionada por los sensores. Estas salidas se verán más adelante. Además de eso, se explicarán qué reglas ha seguido el sistema difuso para obtener las salidas.

Posteriormente, se añadió la capacidad al sistema de activar y desactivar los vaporizadores de las plantas, tal y cómo se hacía con el sistema de riego. Los vaporizadores tienen precedencia al riego, ya que si se usan antes de regar para reducir las temperaturas, se pueden conseguir unos riegos más eficaces, ya que se podrá regar más.

Finalmente, para poder trabajar con el tiempo atmosférico, se introdujo en el sistema la capacidad de representar el tiempo y las predicciones de tiempo según la hora, además del tiempo actual (si llueve o no y con qué intensidad lo hace). También se modificó el funcionamiento del sistema para poder manipular las salidas que le proporcionaba el sistema difuso para que, según el pronóstico de lluvias de las próximas 8 horas, pueda ser capaz de deducir si se debe regar menos de lo que indica la salida o si se debería proceder tal y como ha indicado el sistema difuso. En secciones posteriores veremos cómo se han representado estos hechos y cómo se han ido deduciendo las cosas en el sistema.

2.2. Procedimiento de validación y verificación del sistema

La validación y verificación del sistema ha sido un proceso que se ha ido haciendo a lo largo de todo el desarrollo, desde las etapas iniciales hasta las finales.

2.2.1. Verificación

En cuanto a la verificación, se ha comprobado que el sistema fuese el correcto en cada etapa. A medida que se iba añadiendo funcionalidad al sistema se iba comprobando que ésta funcionase correctamente. Por ejemplo, si se incluía una nueva regla, se probaba de forma interactiva añadiendo hechos a la base de conocimiento con el objetivo de ver si esa regla nueva era capaz de activarse o no. También se ha ido comprobando a medida que se iba construyendo el sistema que no hubiesen **errores sintácticos** ni **inconsistencias**, como por ejemplo reglas que nunca se llegasen a disparar o que produjesen resultados contradictorios entre ellas. También se ha comprobado que no se produjesen inconsistencias semánticas con el fin de mantener la coherencia dentro del sistema. Se ha ido comprobando, por ejemplo, si los valores deducidos se correspondían con el tipo que debían tener.

Otra cosa que se ha hecho para validar el sistema es simular el comportamiento de un usuario para ver que podía trabajar con normalidad sin que se produjesen errores o inconsistencias.

Con todo el proceso de verificación se cree que se han eliminado la mayoría o casi todos los errores, ya que se ha ido haciendo de forma progresiva. Sin embargo,

nunca podemos estar seguros del todo, con lo cuál es posible que aún exista algún que otro pequeño error que no hayamos podido detectar debido a que, por motivos de tiempo, es imposible simular toda la actividad que realizaría un usuario, ya que existen infinitos casos que se pueden dar.

2.2.2. Validación

En cuanto a la validación, se ha ido comprobando en cada fase que se cumplían los requisitos especificados. Además, se ha comprobado que los resultados obtenidos fuesen correctos según las entradas que se le iba proporcionando al sistema. Aparte de esto, se ha comprobado que la comunicación entre el sistema principal y el sistema difuso fuese correcta, que fuese un sistema relativamente rápido y que fuese capaz de funcionar en tiempo real y que, además de todo esto, el sistema explicase suficientemente bien todo lo que sucede en el sistema, explicando al usuario todas las deducciones que va haciendo de tal forma que éste tenga en todo momento suficiente información para saber qué es lo que está ocurriendo.

Se cree que, en general, los resultados que ofrece el sistema son buenos. Sin embargo, al no haber extraído el conocimiento de un experto si no de nosotros mismos, ya que nos hemos considerado los expertos, puede que hayamos cometido algún error a la hora de, por ejemplo, especificar con qué intensidad se debe regar una planta en función de la información proporcionada por los sensores de entrada. Esto es inevitable, ya que no tenemos una fuente real de conocimiento experto. Sin embargo, el sistema está hecho de tal forma que se pueda modificar el funcionamiento fácilmente.

Por ejemplo, si no se está de acuerdo con algún resultado obtenido de, por ejemplo, el riego que se debe realizar, se puede revisar el sistema difuso y modificar las reglas del riego para que produzcan unos resultados que se ajusten más a los que deberían obtenerse realmente o a los deseados por el usuario.

3. Descripción del sistema

3.1. Variables de entrada del problema

Las entradas al sistema son los valores que miden los sensores de humedad, temperatura y luminosidad para cada planta y la previsión de lluvia para una determinada hora.

Las entradas de los sensores tienen la siguiente forma:

```
(valor Tipo Planta Valor-Medido)
```

donde:

- *Tipo* representa Humedad, Temperatura o Luminosidad.
- *Planta* hace referencia a una de las tres plantas (Cactus, Tulipan o Palmera).
- *Valor-Medido* es el valor medido por el sensor. Todos son valores enteros y toman los siguients valores:
  - En el caso de la **humedad**, los valores posibles están en el rango $[0, 1023]$, donde 0 indica humedad máxima y 1023 indica lo más seco posible.
  - En el caso de **temperatura**, los valores posibles son todos los enteros, tanto positivos como negativos. Sin embargo, el sistema difuso solo acepta valores en el rango $[0, 99]$, ya que no parece muy posible que las plantas que se cuiden sobrevivan a temperaturas superiores o inferiores a esas.
  - En el caso de **luminosidad**, los valores van desde 0 a cualquier valor, no hay límite. Sin embargo, el sistema difuso asume que la luminosidad más grande es 1000. De ser necesario, se puede ampliar.

Las entradas de las previsiones tienen la siguiente forma:

```
(prevision_lluvia Hora Intensidad)
```

donde:

- *Hora* es un valor entero en el rango $[0, 23]$ que representa a qué hora hace referencia la previsión.
- *Intensidad* es un valor real que indica cuál es la intensidad de la lluvia en mm/h.

Estos datos de entrada son procesados para facilitarle el funcionamiento al sistema posteriormente. Para cada **valor** de entrada, se generan estos dos hechos:

```
(valor_registrado Tiempo Tipo Planta Valor-Medido)
(utlimo_registro Tipo Planta Tiempo)
```

donde *Tiempo* es un valor que representa un tiempo desde que se inició el sistema. Esta es una forma de identificar a los valores registrados y hacerlos únicos. Así, desde el **ultimo_registro** se puede acceder fácilmente al *Valor-Medido* utilizando *Tiempo*.

Para las previsiones, se obtienen los siguientes datos de entrada procesados:

```
(LluviaPrevista Hora Tipo Intensidad)
```

donde *Tipo* va en función de la intensidad. Es decir:

- Si *Tipo* es **no** es que no va a llover, lo cuál significa que *Intensidad* está en el rango $[0, 0.2)$.
- Si *Tipo* es **debil** es que habrá lluvia de intensidad débil, lo cuál significa que *Intensidad* está en el rango $[0.2, 6.5)$.
- Si *Tipo* es **moderada** es que habrá lluvia de intensidad moderada, lo cuál significa que *Intensidad* está en el rango $[6.5, 15)$.
- Si *Tipo* es **fuerte** es que habrá lluvia de intensidad fuerte, lo cuál significa que *Intensidad* está en el rango $[15, 100)$.
- Si *Tipo* es **torrencial** es que habrá lluvia de intensidad torrencial, lo cuál significa que *Intensidad* está en el rango $[100, \infty)$.

3.2. Variables de salida del problema

Como tal, el sistema no produce ningún valor de salida. Sin embargo, tiene que interactuar con los sistemas de riego y vaporización. Para ello, el sistema tiene que activar/desactivar estos sistemas, con lo cuál usará una serie de hechos.

Para el caso del riego, se tiene el siguiente hecho:

```
(regar Planta Estado [HumedadObjetivo])
```

donde se tiene que:

- *Planta* es la planta asociada a dicho sistema de riego.

- *Estado* indica si el sistema de riego está encendido o apagado mediante los valores on/off.
- *HumedadObjetivo* es un valor que solo aparece si *Estado* está en on. Este valor indica cuál es la humedad que se quiere conseguir con el riego.

En cuanto a la vaporización, tenemos el siguiente hecho:

```
(vaporizador Planta Estado [TemperaturaObjetivo])
```

donde se tenemos:

- *Planta* es la planta asociada a dicho sistema de vaporización.
- *Estado* indica si el sistema de vaporización está encendido o apagado mediante los valores on/off.
- *TemperaturaObjetivo* es un valor que solo aparece si *Estado* está en on. Este valor indica cuál es la temperatura a conseguir con la vaporización.

3.3. Conocimiento global del sistema

El sistema tiene una serie de hechos que son gloables a todo el sistema y que se cargan al inicio. A continuación se van a describir brevemente cuáles son estos hechos, explicando como se llaman y la funcionalidad que tienen:

- **Tiesto**: Contiene los tipos de plantas especificados. Inicialmente hay 3, tal y como se mencionó anteriormente: un cactus, un tilipán y una palmera.
- **HumedadMin**, **HumedadMax** y **HumedadCrit**: Valores enteros que representan las humedades mínima, máxima y crítica de la planta. Esto significa que la humedad ideal o media de la planta planta está entre el mínimo y el máximo. Si está por encima del máximo será una humedad alta, y si está por debajo del mínimo será baja. Adicionalemnte, si la humedad es muy próxima al valor crítico o baja de este, significará que obligatoriamente se tiene que regar la planta[1]. Un ejemplo de este conocimiento es el siguiente hecho que representa un valor de HumedadMin (para los otros casos solo cambia el primer valor, pero la estructura es la misma):

[1]Si se desean modificar estos valores, se le da la libertad al usuario para hacerlo. El sistema difuso funciona con estos conceptos de humedad por intervalos. Si se desean modificar los valores que permiten crear los conjuntos difusos, se tiene la libertad de modificar el programa en Python3 que contiene los valores para crear estos conjuntos.

```
(HumedadMin Cactus 800)
```

- **TemperaturaMin**, **TemperaturaMax** y **TemperaturaCrit**: Valores enteros que representan la temperatura mínima, máxima y crítica de la planta, respectivamente. Al igual que en el caso anterior, se considerará que la temperatura ideal o media se encuentra entre los valores mínimo y máximo. Una temperatura baja se encontrará por debajo del mínimo y una alta por encima del máximo[2]. La temperatura crítica indica un valor de temperatura alta que ya no puede ser tolerado por la planta, y que por tanto, debe ser **vaporizada** si se excede este valor. A continuación se muestra un ejemplo que representa la temperatura mínima, aunque para los otros hechos la estructura es la misma:

```
(TemperaturaMin Cactus 15)
```

- **LuminosidadMin** y **LuminosidadMax**: Valores enteros que representan la luminosidad mínima y máxima de una planta. Como antes, la luminosidad ideal o media se encuentra entre los valores máximos y mínimos. Por debajo del mínimo se considera que hay una baja luminosidad, mientras que por encima del máximo que hay una alta luminosidad [3]. La luminosidad será una especie de indicativo de la cantidad de luz que recibe la planta, y por ende, es una representación del momento del día en el que nos encontramos. La luminosidad influirá a la hora de regar más o menos las plantas. A continuación se muestra un ejemplo que representa la luminosidad mínima, y para el caso de la máxima, la estructura es la misma solo que cambia el primer valor:

```
(LuminosidadMin Cactus 300)
```

- **Regado**: Valor entero que indica cuánto incrementa la humedad de la planta por cada unidad de tiempo que se pasa regando. Se representa de la siguiente forma (el valor asociado puede cambiarse):

```
(Regado 100)
```

- **Vaporizado**: Valor entero que indica cuánto se decrementa la temperatura de la planta por cada unidad de tiempo que se pasa vaporizando. Se representa de la siguiente forma (el valor puede cambiar):

```
(Vaporizado 2)
```

- **hora**: Valor entero que representa la hora actual. Toma valores en el rango $[0, 23]$. Inicialmente se indica que son las 8 de la mañana:

[2]Tal y como se dijo antes, estos valores se pueden modificar. El sistema difuso también utiliza rangos para las temperaturas para definir los conjuntos difusos, así que dichos valores se pueden modificar de ser necesario.

[3]De nuevo, estos valores se pueden modificar. Los valores que utiliza el sistema difuso para crear los conjuntos difusos también se pueden modificar para adaptarlos a las necesidades.

```
(hora 8)
```

- **LluviaActual**: Hecho que representa si actualmente está lloviendo o no, dependiendo de la previsión que había para esa hora en concreto. Inicialmente se indica que para la hora actual no está lloviendo. Los posibles valores de LLuviaActual que se pueden tener son **no**, **debil**, **moderada**, **fuerte** y **torrencial**. A continuación se muestra el valor inicial, tal y como se ha mencionado anteriormente:

```
(LluviaActual no)
```

Adicionalmente, tal y como se indicó anteriormente, hay dos archivos extra, *sensores-init.clp* y *previsiones.clp* que contienen hechos iniciales relacionados con los sensores de los tiestos y las previsiones del tiempo. Estos valores siguien la misma estructura que los datos de entrada mostrados en la sección 3.1. Se recomienda cargar estos hechos, ya que evitan tener que introducirlos a mano. Este conocimiento inicial se ha separado del resto del sistema para que sea más fácil localizarlo y modificarlo y añadirle lo que se necesite de ser necesario. Además, puede que el usuario no quiera cargar estos hechos iniciales y que los quiera definir él mismo.

3.4. Especificación de los módulos desarrollados

El sistema experto utiliza 3 módulos distintos. Aparte de ellos, el sistema tiene una serie de reglas que pueden funcionar independientemente del módulo activo, como por ejemplo leer las entradas de los sensores o actualizar la hora y las previsiones de tiempo.

A continuación se describe cada uno de los módulos:

- **Módulo Riego**: Este módulo se encarga de decidir el riego de las plantas utilizando las entradas de los sensores, las previsiones de tiempo y los resultados proporcionados por el sistema difuso. Cada vez que llega una nueva entrada a cualquiera de los sensores o se cambia la hora se comprueba si se puede regar dicha planta o no. El módulo utiliza el resultado obtenido por el sistema difuso para saber si debe regar o no y se lo comunica al usuario. En caso de decidir regar, activa los sistemas de riego de la planta y se encarga de llegar a la humedad ideal. Antes de utilizar este módulo se activa un módulo que veremos un poco más adelante que se encarga de deducir si habrá alguna lluvia en las próximas 8 horas con el objetivo de modificar los resultados que ha proporcionado el sistema difuso.

- **Módulo Vaporizacion**: Este módulo se activa cuando el sistema ha decidido realizar la vaporización de una planta debido a las altas temperaturas que se han detectado. Una vez activado, se encarga de la vaporización de la planta hasta que los sensores de temperatura detecten que se ha llegado al valor especificado.
- **Módulo CalculoSiguienteLluvia**: Este módulo es utilizado justo antes de utilizar el módulo **Riego** y se encarga de deducir si en las próximas 8 horas se va a producir una lluvia, o si en el caso contrario, no se producirá. Para ello, utiliza información referente a las previsiones de tiempo. Para que el módulo funcione correctamente, se tienen que tener las previsiones para todas las horas, ya que se asume que se tiene información completa sobre el tiempo, aunque esta pueda cambiar posteriormente.

El sistema va activando y desactivando uno u otro módulo en función de lo que vaya deduciendo. Si decide regar es que ha deducido que es necesario realizar un riego, y por tanto primero activaría el módulo **CalculoSiguienteLluvia** que se encarga de deducir si va a llover o no y posteriormente activaría el módulo **Riego** para decidir si debe o no regar la planta con la información de la que dispone. Si por ejemplo deduce que debe vaporizar una planta, el sistema activará el módulo **Vaporizacion** para realizar la vaporización de la planta, y una vez que ha terminado, la desactivaría.

Es importante destacar que la vaporización, tal y como se mencionó anteriormente, tiene una mayor prioridad que el riego. Es decir, el sistema dará preferencia a la deducción de la necesidad de vaporizar a la de regar, y por tanto, activará antes el módulo **Vaporizacion** que el módulo **Riego**.

3.5. Hechos y reglas de cada módulo

A continuación se van a presentar de forma superficial las reglas que utiliza cáda módulo y se va a explicar muy brevemente su funcionamiento. Si se quiere obtener más información sobre el funcionamiento concreto de las reglas, se recomienda consultar la implementación del sistema.

Antes de ver las reglas, vamos a mostrar cuáles son las reglas que utiliza el sistema difuso para decidir si debe o no realizar el riego. Se recuerda que no somos expertos en el riego de las plantas, pero que las reglas representan lo que haríamos:

$$\text{si } Humedad(Baja) \wedge Luminosidad(Baja) \wedge Temperatura(Baja) \rightarrow \text{ regar media}$$
$$\text{si } Humedad(Baja) \wedge Luminosidad(Baja) \wedge Temperatura(Media) \rightarrow \text{ regar alta}$$
$$\text{si } Humedad(Baja) \wedge Luminosidad(Baja) \wedge Temperatura(Alta) \rightarrow \text{ regar media}$$
$$\text{si } Humedad(Baja) \wedge Luminosidad(Media) \wedge Temperatura(Baja) \rightarrow \text{ regar media}$$
$$\text{si } Humedad(Baja) \wedge Luminosidad(Media) \wedge Temperatura(Media) \rightarrow \text{ regar alta}$$
$$\text{si } Humedad(Baja) \wedge Luminosidad(Media) \wedge Temperatura(Alta) \rightarrow \text{ regar poca}$$
$$\text{si } Humedad(Baja) \wedge Luminosidad(Alta) \wedge Temperatura(Baja) \rightarrow \text{ regar poca}$$
$$\text{si } Humedad(Baja) \wedge Luminosidad(Alta) \wedge Temperatura(Media) \rightarrow \text{ regar media}$$
$$\text{si } Humedad(Baja) \wedge Luminosidad(Alta) \wedge Temperatura(Alta) \rightarrow \text{ regar poca}$$
$$\text{si } Humedad(Media) \vee Humedad(Alta) \rightarrow \text{ no regar}$$

Las salidas que devuelve son con qué intensidad regar en los casos en los que se debe hacer (**poca**, **media** o **alta** intensidad) o si no debe hacerse. Poca intensidad significa regar la planta hasta alcanzar una humedad un poco superior al mínimo. Intensidad media significa regar las plantas para conseguir una humedad un poco por debajo del máximo. Alta intensidad significa regar las plantas para conseguir una humedad un poco superior a la del máximo.

Con esto dicho, vamos a ver cómo serían las reglas del sistema.

3.5.1. Reglas generales

Primero vamos a ver las reglas generales, desde donde se activan los distintos módulos:

- **nuevo_registro**: Registra un nuevo valor detectado por los sensores, añadiendo un hecho (valor_registrado Tiempo Tipo Planta Valor) y un (ultimo_registro Tipo Planta Tiempo).
- **ultimo_reg**: Elimina de la base del conocimiento el anterior ultimo_registro y se queda con el más reciente.
- **eliminarEsperarHumedad**: Esta regla elimina el valor de (esperar_humedad Planta Tiempo) que había en el conocimiento al llegarle un nuevo valor de humedad mediante el sensor correspondiente. El hecho de esperar_humedad indica que la planta ha sido regada recientemente, y por tanto, que el sistema no necesita deducir si debe o no ser regada de nuevo.

- **siguienteHora**: Esta regla permite adelantar la hora actual en 1 deduciendo cuál es la siguiente hora. Para activarla, se necesita un hecho que sea (siguiente_hora) en la base del conocimiento.
- **calcularNoLluvia**: Regla que deduce si no se va a llover durante una determinada hora que viene dada por el hecho (prevision_lluvia Hora Intensidad). Se considera que no va a llover si la intensidad está en el rango $[0, 0.2)$. Actualiza el conocimiento añadiendo un hecho que es (LluviaPrevista Hora no Intensidad).
- **calcularLluviaDebil**: Regla que deduce si la lluvia prevista a una hora dada por el hecho (prevision_lluvia Hora Intensidad) es débil, es decir, si la intensidad está en el rango $[0.2, 6.5)$. Actualiza el conocimiento añadiendo un hecho que es (LluviaPrevista Hora debil Intensidad).
- **calcularLluviaMedio**: Regla que deduce si la lluvia prevista a una hora dada por el hecho (prevision_lluvia Hora Intensidad) es moderada o media, es decir, si la intensidad está en el rango $[6.5, 15)$. Actualiza el conocimiento añadiendo un hecho que es (LluviaPrevista Hora moderada Intensidad).
- **calcularLluviaFuerte**: Regla que deduce si la lluvia prevista a una hora dada por el hecho (prevision_lluvia Hora Intensidad) es fuerte, es decir, si la intensidad está en el rango $[15, 100)$. Actualiza el conocimiento añadiendo un hecho que es (LluviaPrevista Hora fuerte Intensidad).
- **calcularLluviaTorrencial**: Regla que deduce si la lluvia prevista a una hora dada por el hecho (prevision_lluvia Hora Intensidad) es torrencial, es decir, si la intensidad es de 100 o más. Actualiza el conocimiento añadiendo un hecho que es (LluviaPrevista Hora torrencial Intensidad).
- **modificarPrediccionLluvia**: Regla que permite modificar el conocimiento asociado a las predicciones de tiempo cuando se detecta que en la base de conocimiento hay un hecho que es (modificar_prevision Hora NuevaIntensidad), actualizando el conocimiento del sistema sobre la predicción de lluvia a esa hora de la forma correspondiente.
- **comenzarLluvia**: Regla que sirve para deducir que en la hora actual va a llover según la predicción que había para la hora actal. Se activa al cambiar la hora. Elimina la previsión que había para la nueva hora y añade a la base del conocimiento un hecho de tipo (LluviaActual Tipo), donde Tipo es débil, moderada, fuerte o torrencial.
- **pararLluvia**: Regla que deduce que la lluvia ha parado o que no llueve durante la hora actual de forma parecida a la que se hace en la regla anterior.
- **calcularRegarPlanta**: Regla que deduce si se debe regar o no la planta utilizando la información de los sensores y pasándosela al sistema difuso. Luego

activa el módulo **CalculoSiguienteLluvia** para deducir si en las próximas 8 horas se producirá lluvia o no. No se puede regar si se está vaporizando o está lloviendo.

- **calcularVaporizarPlanta**: Regla que deduce si se debe vaporizar la planta a partir de los valores de temperaturas que han detectado los sensores. Se encarga de activar el módulo **Vaporizacion**. No se puede vaporizar si se está regando o está lloviendo.

3.5.2. Módulo Riego

A continuación se describirán las reglas que utiliza el módulo Riego:

- **riegoNo**: Regla que muestra información al usuario sobre lo que ha determinado el sistema de no regar la la planta. Además de eso, sale del módulo Riego, ya que no se va a regar la planta.
- **modificarRiegoLluviaDebil**: Regla para modificar la intensidad del riego si se ha deducido que en las próximas 8 horas se va a producir una lluvia débil. En este caso, se ha supuesto que la lluvia débil no va a modificar en mucha cantidad la humedad de la planta, así que se deja la intensidad del riego se deja tal cuál. Sirve más para ofrecer información al usuario sobre las decisiones que va tomando el sistema en función de lo que va deduciendo.
- **modificarRiegoAltoLluviaModerada**: Regla para modificar la intensidad del riego si se ha deducido que en las próximas 8 horas se va a producir una lluvia moderada. En este caso, si el riego que se iba a hacer un riego de alta intensidad se reduce a uno de media, ya que la lluvia aportará algo de humedad. Todos los demás casos se dejan igual.
- **modificarRiegoAltoLluviaFuerte**: Regla para modificar la intensidad del riego si se ha deducido que en las próximas 8 horas se va a producir una lluvia fuerte. Si se deduce que se producirá una lluvia fuerte y se tiene que el sistema difuso había determinado que se tiene que regar con intensidad fuerte, se reduce a débil, ya que la lluvia tendrá un gran aporte de humedad.
- **modificarRiegoMedioLluviaFuerte**: Regla para modificar la intensidad del riego si se ha deducido que en las próximas 8 horas se va a producir una lluvia fuerte. Si se deduce que se producirá una lluvia fuerte y se tiene que el sistema difuso había determinado que se tiene que regar con intensidad media, se reduce a débil, ya que la lluvia tendrá un gran aporte de humedad.
- **modificarRiegoLluviaTorrencialNo**: Regla para modificar la intensidad del riego si se ha deducido que en las próximas 8 horas se va a producir una lluvia

torrencial. Si el sistema deduce que la humedad no está por debajo del nivel crítico corregirá el conocimiento eliminando las deducciones que había hecho sobre regar las plantas y saldrá del módulo Riego, ya que no se va a regar la planta. Esto se debe a que el sistema no considera necesario regar una planta a menos que su humedad esté por debajo del nivel crítico si ha detectado que se va a producir una lluvia torrencial.

- **modificarRiegoLluviaTorrencialBajo**: Regla para modificar la intensidad del riego si se ha deducido que en las próximas 8 horas se va a producir una lluvia torrencial. Si el sistema deduce que la humedad está por debajo del nivel crítico eliminará la intensidad del riego deducida anteriormente y determinará que se debe realizar un riego de baja intensidad, ya que la lluvia torrencial hará un gran aporte de humedad y no será necesario regar más.
- **riegoBajo**: Regla que, una vez deducido que el tipo de riego es de intensidad baja, activa el riego de intensidad baja con la humedad objetivo que ha determinado el sistema difuso (un poco por encima del mínimo).
- **riegoMedio**: Regla que, una vez deducido que el tipo de riego es de intensidad media, activa el riego de intensidad media con la humedad objetivo que ha determinado el sistema difuso (un poco por debajo del máximo).
- **riegoAlto**: Regla que, una vez deducido que el tipo de riego es de intensidad alta, activa el riego de intensidad alta con la humedad objetivo que ha determinado el sistema difuso (un poco por encima del máximo).
- **regarPlanta**: Regla para regar la planta. Cuando se ha decidido regar la planta, se aplica esta regla para ir regando la planta y aumentando su humedad hasta llegar al valor objetivo de humedad.
- **detenerRegarPlanta**: Regla para detener el regado de la planta una vez que se ha llegado al valor objetivo, desactivando por tanto el módulo Riego.

3.5.3. Módulo CalculoSiguienteLluvia

A continuación se describirán las reglas que utiliza el módulo CalculoSiguienteLluvia:

- **contador**: Regla contador. Cuando se actibe este módulo y el sistema inserte el hecho (Contador HoraContador ValorContador) se activa esta regla y se empiezan a recorrer las previsiones en busca de la primera lluvia hasta que el ValorContador llegue a 0, incrementando cada vez en 1 la HoraContador.

- **contadorSiguienteLluvia**: Regla que se dispara cuando el contador se consigue llegar a una hora en la que hay lluvia. Añade el hecho (SiguienteLluvia Tipo Hora) a la base del conocimiento, donde Tipo se deduce segúna la previsión de lluvia que haya a esa hora, y sale del módulo **CalculoSiguienteLluvia**, activando ahora el módulo **Riego**.
- **contadorNoLluvia**: Regla que se dispara cuando el contador llega a 0. Se deduce que no se va a producir ninguna lluvia ya que no se ha encontrado ninguna previsión de lluvia utilizando el contador. Por tanto, añade el hecho deducido (SiguienteLluvia no Hora), desactiva el módulo **CalculoSiguienteLluvia** y activa el módulo **Riego**.

3.5.4. Módulo Vaporizacion

A continuación se describirán las reglas que utiliza el módulo Vaporizacion:

- **vaporizarPlanta**: Regla para vaporizar la planta. Una vez activo el módulo, se aplica esta regla para ir vaporizando la planta y reduciendo su temperatura hasta llegar al valor objetivo de temperatura.
- **detenerVaporizadoPlanta**: Regla para detener la vaporización de la planta una vez que se ha llegado al valor objetivo, desactivando por tanto el módulo Vaporizacion.

4. Manual de uso del sistema

En esta sección se va a explicar brevemente cómo trabajar con el sistema.

Lo primero de todo es, una vez inicializado CLIPS, cargar el sistema. Opcionalmente se pueden cargar los valores iniciales de los sensores y las previsiones del tiempo. Si no se quiere hacer, se tienen que introducir manualmente. Sin embargo, si se toma este segundo curso de acción, se tienen que introducir, al menos, previsiones del tiempo para las 8 siguientes horas.

A continuación se puede ver un ejemplo de cómo se podrían cargar todos los archivos. Si no se quiere alguno de ellos, basta con simplemente no introducirlo:

```
(load sistema.clp)
(load sensores-init.clp)
(load previsiones.clp)
(reset)
(run)
```

Una vez cargados el sistema y los datos, se pueden insertar valores de los sensores de la siguiente forma:

```
(assert (valor Tipo Planta ValorSensor))
```

donde *Tipo* es Temperatura, Luminosidad o Humedad; *Planta* es alguna de las plantas y *ValorSensor* es el valor medido por el sensor. Un ejemplo de esto sería el siguiente (el (run) se utiliza para insertar los hechos en la base del conocimiento):

```
(assert (valor Temperatura Tulipan 24))
(run)
```

Para ir insertando previsiones del tiempo, se puede hacer de la siguiente forma:

```
(assert (prevision_lluvia Hora Intensidad))
```

donde *Hora* es un valor en el rango $[0, 23]$ e *Intensidad* es un valor real que representa la intensidad de la lluvia en mm/h. Un ejemplo seria el siguiente:

```
(assert (prevision_lluvia 11 45))
(run)
```

En caso de que el usuario se haya equivocado o la previsión del tiempo para una determinada hora cambie, se puede hacer lo siguiente para modificar la previsión:

```
(assert (modificar_prevision Hora NuevaIntensidad))
```

donde *Hora* y *NuevaIntensidad* tienen los mismos significados que antes. Por ejemplo, si quieremos modificar la previsión que anteriormente teníamos de las 11 para que tenga una intensidad de 14, debemos hacer los siguiente:

```
(assert (modificar_prevision 11 14))
(run)
```

El sistema también permite pasar las horas. Para hacerlo, basta con simplemente hacer lo siguiente:

```
(assert (siguiente_hora))
(run)
```

Si se necesita modificar algo del sistema difuso, se recomienda echar un vistazo a este ejemplo de sistema difuso [3] y a la documentación de las funciones implementadas [4].

Referencias

[1] NumPy
`https://www.numpy.org/`

[2] GitHub. *scikit-fuzzy*
`https://github.com/scikit-fuzzy/scikit-fuzzy`

[3] Ejemplo de sistema difuso
`https://pythonhosted.org/scikit-fuzzy/auto_examples/plot_tipping_problem.html#example-plot-tipping-problem-py`

[4] Manual de las funciones utilizadas de *scikit-fuzzy*
`https://pythonhosted.org/scikit-fuzzy/api/skfuzzy.html`