PROCESADORES DE LENGUAJES

TRABAJO PRÁCTICO

HITO 1 MODIFICADO

Integrantes del grupo:

Diego Sepúlveda Millán

Pablo Valle Nieto

Pedro Rodríguez Viñuales

David Rivera Concepción

Miguel Ángel Matas Rubio

Enlace repositorio Git-Hub:

https://github.com/PedroRoViUni/Procesador-de-Lenguaje

1.- Presentación del Problema y justificación del uso de los conocimientos de la materia.

Con el fin de construir un programa para la simulación de planos de hogares inteligentes, abordando el problema desde la asignatura Procesadores de Lenguajes, vemos necesaria la creación de una gramática libre de contexto que nos permita estandarizar la representación de un hogar y especificar los diferentes actuadores y sensores propios de una casa del futuro, así como facilitar lo máximo posible su programación.

Dicha gramática servirá para la construcción de un procesador el cual nos dé como salida una simulación a través de comandos, los cuales indicarán diferentes estados y situaciones posibles por los que el usuario irá interactuando. Todo ello acompañado de una representación 2D a tiempo real del estado de los actuadores, sensores y posición del usuario.

2.- EBNF del lenguaje.

PROGR ::= newh id “{“ L_HAB “;” accesos “:” L_ACC “;" REGLAS “}”

L_HAB ::= “(“ HAB ”)” { “,” “(“ HAB ”)” }

HAB ::= id “,” DIM “;” SENS “;” ACTUAS

L_ACC ::= ACC {“,” ACC}

ACC ::= “(“ id “-“ id “)”

DIM ::= “(“ num “,” num ”)”

SENS ::= SEN {“,” SEN}

SEN ::= SEN_LUM | SEN_TEM | SEN_PRE | SEN_GAS | SEN_FUE | SEN_HUM

SEN_LUM ::= id lum “=” num

SEN_TEM ::= id tem “=“ num

SEN_PRE ::= id pre “=” (True | False)

SEN_GAS ::= id gas “=” (True | False)

SEN_FUE ::= id fue “=” (True | False)

SEN_HUM::= id hum “=“ num

ACTUAS::= ACTUA {, ACTUA}

ACTUA ::= ACTUA_CALE | ACTUA_AIR | ACTUA_PERS |ACTUA_ROCI | ACTUA_ALARM

ACTUA_CALE ::= id cale “=” num

ACTUA_AIR ::= id air “=” num

ACTUA_PERS ::= id pers “=” (subir | bajar | parar)

ACTUA_ROCI ::= id roci “=” (True | False)

ACTUA_ALAR ::= id alar “=” (True | False)

REGLAS ::= IFF {“;” REGLAS}

IFF ::= if “(“ CONDS “)” “{“ CONSE “}”

CONDS ::= CONDI { (and | or) CONDI}

CONDI ::= CONDIB | CONDIN

CONDIB ::= id COMPAB (True | False)

CONDIN ::= id COMPA num

COMPA ::= menor| mayor | igual

COMPAB ::= igualc | distin

CONSE ::= id “=” (num | (True|False) |(subir | bajar | parar))

id =[a-z]+[0-9]+'

num =[0-9]{1,2}'

2.1 Semántica EBNF del lenguaje.

PROGR : Símbolo inicial de la gramática, se declara un nombre para el hogar, después las habitaciones, los acceso y las reglas que existirán entre sensores y actuadores.

L_HAB : Lista de habitaciones, contendrá de una a más producciones de HAB.

HAB : Cada habitación tendrá asignado un id y las variables DIM, SENS y ACTUAS.

L_ACC : Lista de accesos, tendrá de una a más producciones de ACC.

ACC : Un acceso estará compuesto por dos id de habitaciones separados de un guión.

DIM : Las dimensiones estarán compuestas de dos números enteros separados por una coma. Representarán los metros cuadrados de la habitación en cuestión.

SENS : La lista de sensores, compuesta de una a más producciones de SEN.

SEN : Tipos de sensores.

SEN_LUM: Los sensores lumínicos o numéricos tendrán un id asociado la palabra reservada lum y un número.

SEN_TEM: Los sensores de temperatura o numéricos tendrán un id asociado la palabra reservada tem y un número entre 0 y 50.

SEN_PRE: Los sensores de presencia o booleanos tendrán un id asociado, la palabra reservada pre y un valor boolean.

SEN_GAS: Los sensores de gas o booleanos tendrán un id asociado, la palabra reservada gas y un valor boolean.

SEN_FUE: Los sensores de fuego o booleanos tendrán un id asociado, la palabra reservada fue y un valor boolean.

SEN_HUM: Los sensores humedad o numéricos tendrán un id asociado la palabra reservada hum y un número.

ACTUAS : Lista de actuadores, que contiene de una a más producciones de ACTUA.

ACTUA: Tipos de actuadores.

ACTUA_CALE: Los actuadores de calefacción o numéricos están compuestos de un id, la palabra reservada cale y un número entre 0 y 50.

ACTUA_AIR: Los actuadores de aire acondicionado o numéricos están compuestos de un id, la palabra reservada air y un número entre 0 y 50.

ACTUA_PERS: Los actuadores de persianas están compuesto de un id, la palabra reservada pers y un valor (subir | bajar | parar).

ACTUA_ROCI: Los actuadores de rociadores de incendios o booleanos están compuestos de un id, la palabra reservada roci y un valor booleano.

ACTUA_ALAR: Los actuadores de alarmas booleanos están compuestos de un id, la palabra reservada roci y un valor booleano.

REGLAS: Las reglas tendrán producciones de IFF separadas por punto y coma.

IFF: Comenzarán con la palabra reservada if o si, continuada por la producción CONDS entre paréntesis y le seguirá entre llaves las consecuencias con la producción CONSE.

CONDS: Lista de condiciones, separadas entre sí por las palabras reservadas and u or.

CONDI: Puede ser o CONDB o CONDIN siendo estas condiciones o booleanas o numéricas.

CONDIB: Está compuesto de un id que corresponde con un id de sensor, una comparación expresada con la producción COMPAB y un valor booleano.

CONDIN: Está compuesto de un id que corresponde con un id de sensor, una comparación expresada con la producción COMPA y un valor booleano.

COMPA: Representa los símbolos de comparación menor, mayor e igual que.

COMPAB: Representa los símbolos de comparación igual o distinto que.

CONSE: Una consecuencia compuesta por un id, un igual y un valor del tipo del sensor que representa el id inicial.

3.- ¿Cómo se va a construir el procesador del lenguaje diseñado? Emplea los diagramas en forma de T estudiados en clase.

Usamos nuestro propio lenguaje de dominio para representar cada hogar, usando python3 como lenguaje objeto y de implementación, además del uso de la librería PLY (Python Lex-Yacc) para la codificación de la gramática. Se ha creado la clase hogar con sus respectivos atributos como sensor, actuador, habitación… representados en distintas clases de tal manera que al hacer el análisis sintáctico se creen objetos con el fin de que al finalizar el análisis se disponga el objeto hogar el cual sirve de entrada a la clase simulación que será la que nos permite interactuar con el hogar descrito.