ARFF

PROYECTO LEX

JULIO SANCHEZ MIRON 75936406C

Introducción

En este proyecto se ha implementado un reconocedor del formato ARFF ampliamente utilizado en IA. Para ello se ha utilizado Lex para el proceso de análisis léxico de cadenas de caracteres como input.

Desarrollo

El principal objetivo del programa desarrollado es datar de un error de formato, en caso de que se produzca, en una línea determinada para que pueda corregirse y utilizarse en los algoritmos de IA.

ARFF consta de dos secciones distintas. En primer lugar, la sección de cabecera (HEADER), la cual es seguida por la sección de datos (DATA)

La cabecera contiene el nombre de la relación, una lista de atributos y sus respectivos tipos.

La sección de datos, contiene en un orden prefijado, cada uno de los valores de los tipos especificados en los atributos.

Para resolver la tarea, tengo una estructura de datos básica en la que almaceno el tipo de dato de cada atributo que se va definiendo y, cuando voy leyendo datos, voy comprobando que efectivamente éste corresponde con el tipo de dato prefijado en la definición del atributo.

Cómo ejecutar el analizador

Hay varios detalles a tener en cuenta:

- Siguiendo los ejemplos publicados en Weka, se ha implementado que en la primera parte de la definición de los atributos nominales, debe aparecer el nombre de la relación del fichero ARFF. Las especificaciones del atributo nominal class han de ser <irisp>-<especificacion1>

```
@RELATION irisp

@ATTRIBUTE sepallength  NUMERIC
@ATTRIBUTE sepalwidth   NUMERIC
@ATTRIBUTE petallength  NUMERIC
@ATTRIBUTE petalwidth   NUMERIC
@ATTRIBUTE class  {irisp-uno,irisp-dos}
```

  - Comentar que no se ha conseguido implementar nombres de relaciones con espacios, las cuales deberían de aparecer con simples o dobles comillas.

- Respecto a las fechas que pueden ser usadas se han restringido a dos tipos:
  - “yyyy-MM-dd HH:mm:ss”
  - “dd-MM-yyyy HH:mm:ss”

```
@RELATION prueba
@ATTRIBUTE numero NUMERIC
@ATTRIBUTE sepallength  {prueba-ejemplo,prueba-ejemplo1,prueba-ejemplo2}
@ATTRIBUTE otroNumero NUMERIC
@ATTRIBUTE unaprueba STRING
@ATTRIBUTE pruebafecha DATE "yyyy-MM-dd HH:mm:ss"

@data
1,prueba-ejemplo,4.2,'jejeasa- -as .',2016-04-23 13:22:12
```

Este es un ejemplo de funcionamiento donde se usa el formato “yyyy-MM-dd HH:mm:ss” y posteriormente se proporciona como dato la fecha 2016-04-23 13:22:12. Hay que resaltar que **no** se reconocen fechas con años mayores a 2017 (Esto se podría cambiar fácilmente en la zona dónde se han definido los alias, pero no resulta de nuestro interés datos que se produzcan en años superiores a 2017).

Resultados

En caso de éxito, el analizador devolverá el siguiente mensaje por consola:

```
YECTO$ ./prog arffP
------------------------------------------------------------
----El fichero dado como entrada contiene formato ARFF----
------------------------------------------------------------
```

Este se puede probar con el fichero denominado “arffP” contenido en el zip entregado.

Los casos de errores pueden ser por una gran variedad de motivos, entre los que destacamos:

- Redeclaración de secciones como @RELATION , @DATA

```
13 tacion/PROYECTO$ ./prog arffP
14
15 Error en linea 17, seccion @DATA ya se había leído
16 @DATA
17 @DATA
18 2.2,string,2014-11-20 10:10:02,0.2,Iris-atributo1
19 4.9,'otro string',2000-11-20 09:12:02,0.2,Iris-atributo2
```

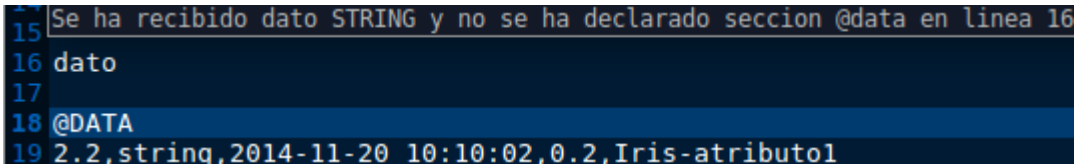
- Declaración de sección @ATTRIBUTE sin previamente haber declarado sección @RELATION, obligatoria y necesaria. Se puede apreciar que @RELATION está comentado en la imagen siguiente:

```
6 Error: Declaracion de seccion @ATTRIBUTE sin haber declarado sección @RELATION en linea 10
7 %
8 % @RELATION Iris
9
10 @ATTRIBUTE sepallength  NUMERIC
11 @ATTRIBUTE sepalwidth   STRING
```

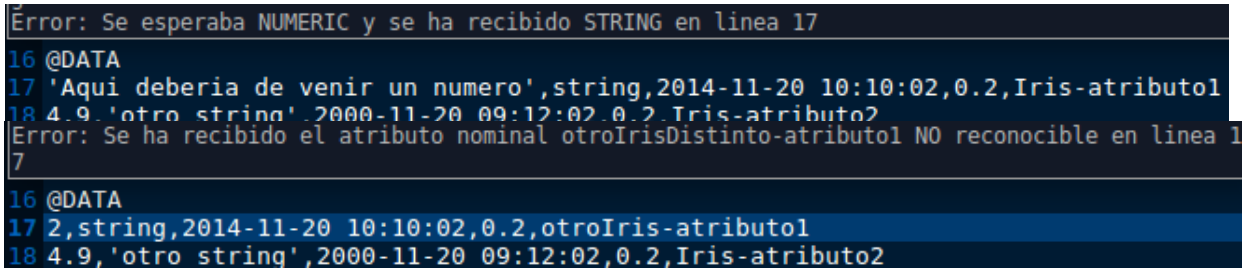
```
6
7 Error: El nombre de la relacion no corresponde con el atributo nominal en linea 14
8 @RELATION Iris
9
10 @ATTRIBUTE sepallength  NUMERIC
11 @ATTRIBUTE sepalwidth   STRING
12 @ATTRIBUTE pruebafecha DATE "yyyy-MM-dd HH:mm:ss"
13 @ATTRIBUTE petalwidth   NUMERIC
14 @ATTRIBUTE nominal   {Iris-atributo1,otroIris-atributo2,Iris-atributo3}
```

- Cuando difiere el valor de la sección @RELATION declarada y un atributo nominal como se puede apreciar que @RELATION tiene valor Iris y el segundo atributo nominal es definido con ‘otroIris-atributo2’

- Cuando por basura presente en el fichero, se recibe algún carácter (que tendría que pertenecer a algún dato y sin embargo no se ha declarado aún sección @DATA). Se puede apreciar que el string “dato” está en tierra de nadie y no significa nada:



- Cuando se esperaba recibir un tipo de dato y sin embargo se recibe otro distinto (recordemos que los tipos de datos a recibir van en orden y vienen predefinidos en función del orden de la declaración de los atributos).



- Cuando se recibe como dato un atributo nominal que difiere a los previamente declarados (otroIris-atributo1) no se ha definido como posible valor de atributo nominales

- Cuando el número de datos proporcionados en una línea es inferior a los que se corresponden

Por último, si un dato se omite, éste se puede sustituir por ? Sin alterar resultado del análisis de ARFF. Vemos el éxito del fichero que contiene valores omitidos y sustituidos por ?