

Avaliação #1

Leitor de Atributos ARFF

Você já ouviu falar do **formato de arquivos ARFF** (*Attribute-Relation File Format*)? Esse formato de arquivo foi desenvolvido no contexto da solução WEKA (*Waikato Environment for Knowledge Analysis*), dedicada à análise de dados e *machine learning*.

O objetivo do formato ARFF é oferecer uma estrutura padronizada e versátil para armazenar dados em formato de tabela. Nele, cada linha representa uma instância ou registro, enquanto cada coluna representa um atributo. Além disso, o formato ARFF permite definir informações detalhadas sobre os atributos, incluindo seus nomes, tipos de dados e possíveis valores. Essa flexibilidade torna o ARFF uma escolha popular para a importação e exportação de conjuntos de dados em diversas ferramentas de aprendizado de máquina e análise de dados.

Um arquivo ARFF é dividido em duas seções: **a de atributos e a de dados**. A seção de atributos é a primeira do arquivo. Nela todos os atributos (colunas) são definidas através do seu nome e tipo de dados. Cada atributo é definido em uma linha iniciada pela *string* “@attribute”; um espaço a frente existe o nome do atributo, definido como uma *string* que suporta qualquer caractere digitável, exceto espaços; por fim, um espaço a frente existe o tipo do atributo.

Existem três tipos de atributo, sendo eles:

- (i) Tipo numérico: denotado pela palavra “**numeric**”, indica um atributo definido através de um número real;
- (ii) Tipo textual: denotado pela palavra “**string**”, indica um atributo definido através de uma *string*;
- (iii) Tipo categórico: denotado por abre e fecha chaves, com as categorias possíveis (*strings*) separadas por vírgula entre as chaves (por exemplo, “{cat1,cat2,cat3}”).

Sendo assim, um exemplo de seção de dados de um arquivo no formato ARFF é:

```
@attribute Nome string
@attribute Idade numeric
@attribute Grau {BC,MC,Bacharel,Mestre,Doutor}
```

A seção de atributos se encerra quando o *token* da seção de dados é encontrado. Tal *token* é inserido em uma linha isolada e é denotado pela *string* “@data”.

Além disso, cabe ressaltar que **linhas em branco** antes dos atributos, entre atributos e entre um atributo e o *token* “@data” devem ser simplesmente **ignoradas**.

Ou seja, você pode encontrar um arquivo ARFF com o seguinte conteúdo (exemplo):

```
@attribute Nome string
@attribute Idade numeric
@attribute Grau {BC,MC,Bacharel,Mestre,Doutor}
@data
```

Assim como você pode encontrar um arquivo ARFF com o seguinte conteúdo (exemplo):

```
@attribute Nome string
```

```
@attribute Idade numeric
```

```
@attribute Grau {BC,MC,Bacharel,Mestre,Doutor}
```

```
@data
```

O seu objetivo é implementar funções que (i) abre um arquivo no formato ARFF; (ii) lê a sua seção de atributos; (iii) para cada atributo, armazena o seu rótulo, tipo e categorias em uma estrutura como a definida a seguir:

```
typedef struct {  
    char *rotulo;  
    char *tipo;  
    char *categorias;  
} atributo;
```

Se o tipo for numérico ou textual, em “tipo” deve constar, respectivamente, “numeric” e “string”, e em “categorias” deve existir zero (*NULL*); se o tipo for categórico, em “tipo” deve constar “categoric” e em categorias deve constar a *string* com todas as categorias (no exemplo, seria “{BC,MC,Bacharel,Mestre,Doutor}”).

A função de criação deve retornar um **vetor de struct atributo** contendo cada atributo informado no arquivo ARFF fornecido como entrada. Além disso, **o ordenamento dos atributos no vetor deve coincidir com a ordem de aparição dos atributos no arquivo.**

Por fim, se existir **qualquer problema na definição da seção de atributos (ou ausência do token @data)** do arquivo ARFF fornecido como entrada, **a função deve retornar zero (NULL).**

VOCÊ DEVE UTILIZAR O ARQUIVO main.c FORNECIDO PARA IMPLEMENTAR A SUA SOLUÇÃO. LEIA ATENTAMENTE OS COMENTÁRIOS NO ARQUIVO PARA QUE ESTE ALCANCE O RESULTADO ESPERADO.

Para facilitar, **considere que:**

- > Nenhuma linha de atributo terá mais do que 1024 caracteres;
- > Haverá exatamente um espaço entre um elemento e outro em uma linha de atributo;
- > Não haverão espaços no início e no final de uma linha de atributos;
- > Atributos categóricos terão seus valores sempre definidos corretamente.

Também, **não esqueça que:**

- > Uma linha de atributo pode conter menos ou mais elementos, além dos necessários (o programa deve tratar esse possível erro!);
- > Uma linha de atributo não iniciada com “@attribute” deve ser indicada como erro.

```
#####  
IMPORTANTE: A ENTREGA DO TRABALHO DEVE CONTEMPLAR OS REQUISITOS  
ELENCADOS NO PLANO DA DISCIPLINA.  
#####
```