



INSTITUTO FEDERAL DE MINAS GERAIS

Bacharelado em Ciência da Computação

Disciplina: Matemática Discreta

Trabalho Prático 02

Professor: Diego Mello da Silva

Formiga-MG
21 de dezembro de 2015

Sumário

1	Informações Gerais	1
2	O problema	1
2.1	Redes Semânticas	1
2.2	Relações $R \subseteq A \times A$	4
2.3	Redes Semânticas como Relações	5
3	Especificação	9
3.1	Requisito: Entrada de Dados via Arquivo	10
3.2	Requisito: <i>Lookup Table</i> para Armazenar Objetos, Conceitos e Idéias da Rede	12
3.3	Requisito: <i>Lookup Table</i> para Armazenar Tipos de Relacionamentos	12
3.4	Requisito: Matriz de Incidência da Rede Semântica	13
3.5	Requisito: Mapeamento de ID para Objeto, Conceito ou Idéia	13
3.6	Requisito: Mapeamento de ID para tipo de Relacionamento	13
3.7	Requisito: Estatísticas da Rede no Terminal	13
3.8	Requisito: Saída de Dados no Formato Dot	15
3.9	Requisito: Dataset com Redes Semânticas Mapeadas	15
3.10	Requisito: Documentação de Código	15
3.11	Requisito: Corretude dos Resultados	15
4	Barema de Correção	16
5	Bibliografia	16

1 Informações Gerais

Este documento descreve a especificação do Trabalho Prático 02 da disciplina Matemática Discreta, e deve ser seguido de forma a contemplar os itens considerados na avaliação por parte do professor da disciplina. O trabalho deve ser feito em grupo de até 3 alunos e tem o valor de 30 pontos. A data limite do trabalho deverá ser combinada em sala de aula. O trabalho deverá ser entregue até as 23:59 hs da data limite. Trabalhos entregues após este prazo serão desconsiderados e não serão corrigidos. O documento é organizado como segue. Na Seção 2 será apresentado o problema que este trabalho prático pretende resolver; na Seção 3 serão apresentados os requisitos de *software* que, se implementados, permitirão resolver o problema proposto; na Seção 4 serão apresentados os critérios de avaliação do trabalho e respectiva pontuação. **Trabalhos plagiados ou copiados valerão zero.**

2 O problema

O presente documento especifica uma aplicação que implementa um tipo de representação de conhecimento em inteligência artificial denominada de **rede semântica**.

2.1 Redes Semânticas

Uma rede semântica é uma um grafo que contém nós e arcos direcionados. Os nós de uma rede semântica são usados para modelar **objetos**, **situações** e **conceitos**; já os arcos da rede modelam **relacionamentos** binários entre eles. Tanto os nós da rede quanto os arcos da rede são rotulados para identificar os objetos e relacionamentos que representam. Em geral, redes semânticas fornecem um modo intuitivo de representar conhecimento sobre objetos e relacionamentos entre eles. O sentido do arco indica quem se relaciona com quem.

Para exemplificar, seja o seguinte conjunto de declarações que representam conhecimento acerca de um determinado domínio: ‘Fido é um cachorro que caça Fang, um gato’, ‘Bob é construtor, come queijo, e é dono de Fido’ e ‘Fang caça ratos que comem queijos’. Uma possível representação para este conhecimento é dado na Fig 1, extraído do livro ‘Inteligência Artificial’, de Ben Coppin, Editora LTC, 2012 (ISBN: 978-85-216-1729-7).

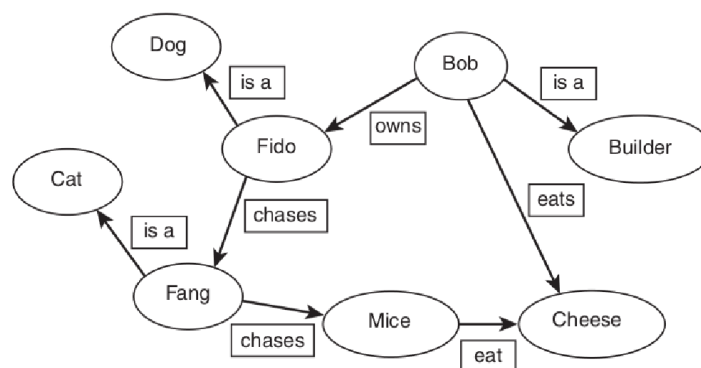


Figura 1: Exemplo de Rede Semântica

Como arcos representam relacionamentos entre objetos da rede semântica, não existe um conjunto bem definido de relações. É comum representar uma propriedade ou conceito (tal como ‘caça’, ‘come’, ‘possui’) e herança de propriedades (tal como ‘é um’, ‘é parte de’, ‘é um tipo de’ (AKO: *a kind of*), etc).

Outros exemplos de redes semânticas, obtidas das fontes: <http://docplayer.com.br/5130528-Representacao-de-conhecimento-redes-semanticas-e-frames.html> (Fig 2 e Fig 3) e <http://slideplayer.com.br/slide/390385/> (Fig 4). No caso da Fig 2 a rede semântica modela dois animais e suas propriedades: um cão e um pássaro. Pelos arcos da rede semântica modelada observa-se que tanto o cão quanto o pássaro são animais (pois o cão é **um tipo de** mamífero, que é por sua vez **um tipo de** animal; o pássaro é **um tipo de** animal). De acordo com a rede, tanto o pássaro quanto o cão comem; no entanto apenas o cão tem pêlos pois é mamífero, e o pássaro não.

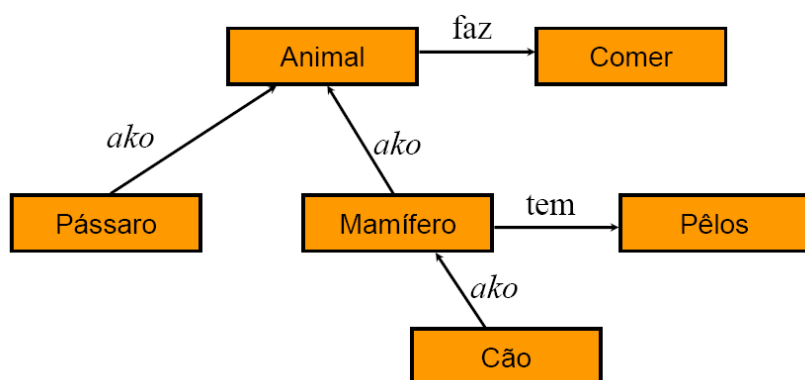


Figura 2: Exemplo de Rede Semântica

No exemplo apresentado pelas figuras Fig 3 e Fig 4 observa-se também como é possível instanciar casos específicos em uma rede semântica. Observe na rede da Fig 3 os nós ‘Ana’ e ‘Cadeira 27’ são instâncias, respectivamente, de objetos da classe ‘Pessoa’ e ‘Cadeira’. ‘Cadeira’ por sua vez é **um tipo de** ‘Mobília’.

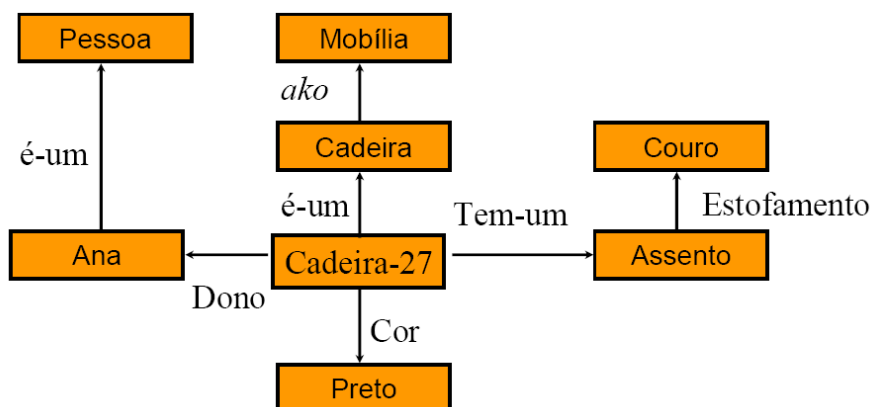


Figura 3: Exemplo de Rede Semântica

A presença de instâncias de classes também pode ser observada na Fig 4. No exemplo, ‘carro1’ é uma instância de ‘automóvel’, e ‘Maria’ é uma instância de ‘pessoa’. Nota-se também que o modelador preocupou-se em considerar que cada automóvel da rede seja formado por ‘rodas’, ‘carroceria’ e ‘motores’.

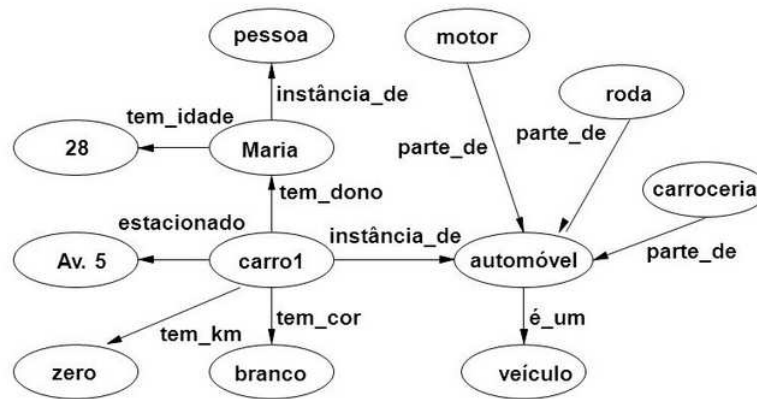


Figura 4: Exemplo de Rede Semântica

A representação de redes semânticas, embora simples, tem limitações. Não é permitido por exemplo modelar situações tais como negações, disjunções (declarações do tipo ‘ou’) ou quantificações. Existe ainda uma relação forte entre redes semânticas e lógica proposicional, sendo possível converter um tipo de representação na outra. Tal assunto está fora do escopo deste trabalho e, portanto, não será coberto neste documento.

Apesar das limitações apresentadas é possível fazer algum tipo de inferência sobre o conhecimento representado pela rede semântica. Para exemplificar, é possível inferir (i) classificação de descrições, (ii) herança de informação a partir de conceitos gerais para conceitos mais específicos, (iii) seguir conexões entre objetos, seus pares, seus nomes, propriedades típicas, usos, papéis, etc, e (iv) explorar tudo o que se sabe sobre um determinado objeto mapeado na rede semântica.

Em alguns casos a inferência em redes semânticas pode ser realizada por meio da aplicação de algoritmos de busca em grafos. Técnicas de busca podem ser usadas em redes semânticas como ferramenta explicativa, para explorar exhaustivamente um tópico, ou para encontrar relacionamento entre dois objetos. Para exemplificar, seja o exemplo dado na Fig 2. Se a busca no grafo for usada

- Como ferramenta explicativa: poderíamos usar a busca como ferramenta explicativa se quiséssemos provar a declaração ‘cães comem’. Buscando a partir do nó ‘cão’, tenta-se verificar pela busca se em algum momento chega-se ao nó ‘comer’. Existe uma sequência neste exemplo que mostra este fato: ‘cão é um tipo de mamífero’, ‘mamífero é um tipo de animal’, ‘animal faz comer’, que prova portando que ‘cães comem’.
- Para explorar exhaustivamente um tópico: usa-se busca em largura a partir do nó ‘cão’. Tudo o que for relacionado com o nó ‘cão’ será derivado pela

busca, que segue o sentido da relação com outros nós. Derivaríamos que ‘cão é mamíferos’, ‘cão tem pêlos’, ‘cão é animais’ e ‘cão comem’.

- Para relacionar tópicos da rede semântica: se aplicarmos busca em largura a partir do nó ‘cão’, e outra busca em largura a partir do nó ‘pássaro’, tudo o que existir na intersecção da busca que parte destes dois tópicos relaciona cães e pássaros. Neste caso sabe-se que ‘tanto o cão quanto o pássaro são animais’, e que ‘tanto o cão quanto o pássaro comem’.

Embora busca possa ser usada para realizar inferências sobre a rede semântica, este assunto está fora do escopo da disciplina de Matemática Discreta e não será tratado neste trabalho. O foco deste projeto será na representação da rede semântica usando grafos (e, portanto, relações sobre conjuntos) e não no mecanismo de inferência sobre o mesmo. A seção 2.2 apresentará brevemente o que é uma relação sob o ponto de vista matemático; na seção 2.3 o tema será retomado, desta vez unindo ambos os assuntos.

2.2 Relações $R \subseteq A \times A$

Uma relação R consiste em um subconjunto de pares ordenados do tipo (x, y) obtidos a partir do produto cartesiano $A \times A$. Formalmente, $R \subseteq A \times A$. Uma relação R é do tipo 1-para-1 se os componentes x e y do par ordenado aparecem apenas uma vez em R ; ela é do tipo 1-para- N se existe algum componente x do par ordenado aparece mais de uma vez em R ; ela é N -para-1 se algum componente y aparece mais de uma vez em R ; por fim, ela é N -para- N se existem componentes x e y do par ordenado que aparecem mais de uma vez em R . A Figura 5 apresenta um diagrama com os tipos de relações mencionadas.

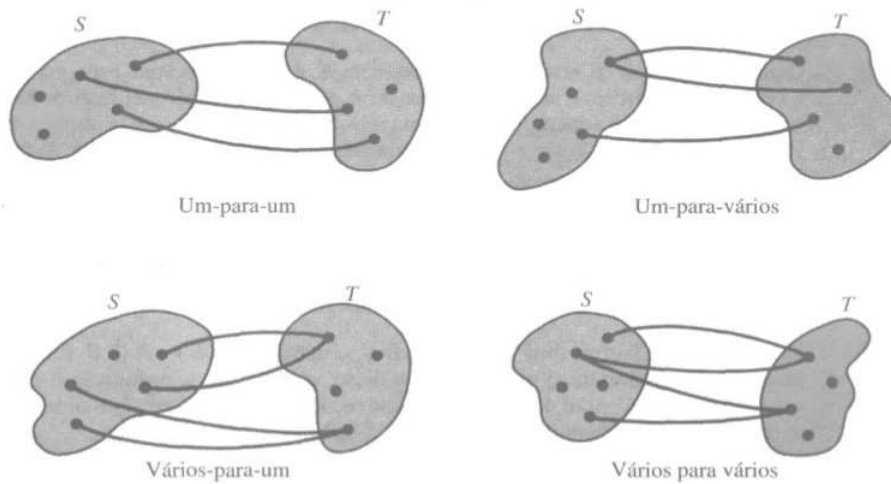


Figura 5: Tipos de Relações

Matematicamente, uma relação R em um conjunto A é chamada de **reflexiva** se $(x, x) \in R$ para todo elemento $x \in A$, ou seja, $\forall x \in A((x, x) \in R)$. Em outras palavras, R é reflexiva se contiver todos os pares do tipo (x, x) , onde x é elemento do domínio A . Uma relação R em um conjunto A é chamada de **simétrica** se $(y, x) \in R$

sempre que $(x, y) \in R$, ou seja, $\forall x \forall y ((x, y) \in R \rightarrow (y, x) \in R)$. Novamente, uma relação será simétrica desde que, para cada par ordenado (x, y) existente em R também exista o par ordenado (y, x) correspondente. Por fim, uma relação é denominada **transitiva** quando $\forall x \forall y \forall z ((x, y) \in R \wedge (y, z) \in R \rightarrow (x, z) \in R)$.

Relações podem ser representadas graficamente, por meio de grafos. Um grafo $G = (V, E)$ consiste em um par ordenado que contém um conjunto de vértices V e arcos $E \subset V \times V$. Uma forma simples e fácil de representar relações computacionalmente como grafos é utilizar o conceito de **matriz de incidência**. Uma matriz de incidência é uma estrutura de dados do tipo matricial, que contém valores booleanos que representam se dois elementos quaisquer da matriz estão ou não relacionados segundo R . A interpretação da matriz segue a convenção de que as linhas quando percorridas referem-se aos elementos x do par ordenado $(x, y) \in R$, e que as colunas quando percorridas referem-se aos elementos y do mesmo par ordenado, com $x, y \in A$. Para exemplificar, seja a relação R sobre $A = \{1, 2, 3, 4\}$ dada por:

$$R = \{(1, 1), (3, 2), (2, 2), (1, 3), (4, 4), (3, 3), (2, 3), (2, 1), (3, 1), (1, 2)\}$$

A representação de R por matriz de incidência é dada por:

$$R = \begin{bmatrix} 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Da mesma forma outras relações podem ser construídas e ter suas propriedades verificadas desde que se utilize alguma estrutura de dados para representar a relação entre os elementos do conjunto considerado.

2.3 Redes Semânticas como Relações

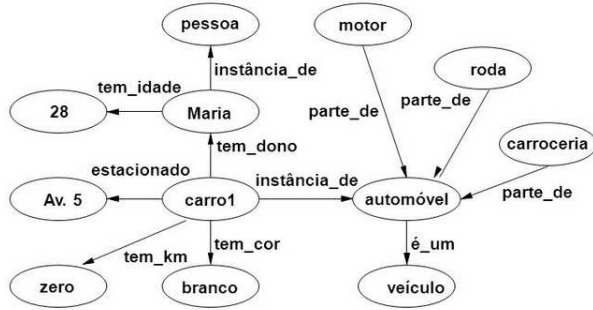
Nas últimas duas seções deste documento foram apresentados os fundamentos básicos de redes semânticas (muito usadas para representação do conhecimento em inteligência artificial) e relações sobre conjuntos. Nesta seção ambos os conceitos serão unificados.

Da mesma forma que uma relação $R \subseteq A \times A$ é um conjunto de pares ordenados do tipo (x, y) , onde $x \in A$ se relaciona com $y \in A$ quando o conjunto A é numérico, podemos também representar relações entre objetos que não são números usando a estrutura de incidência apresentada na seção anterior para representar, por exemplo, um conhecimento mapeado sob a forma de rede semântica. Para objetos numéricos, cada nó representa um número e uma matriz binária pode ser usada para indicar quando um nó se relaciona com outro. Para tratar objetos não numéricos como relações será preciso usar estruturas de dados auxiliares, que serão apresentadas nesta seção. Tratam-se de *lookup tables*, estruturas de dados de acesso rápido que usam vetores para indexar itens pré-calculados previamente armazenados.

A primeira delas provê um mapeamento de **tipos de relacionamentos** contidos na rede semântica. Para cada tipo diferente de mapeamento atribui-se um número natural não nulo, de forma que a matriz de incidência apresentada acima seja estendida para indicar não apenas se existe ou não relacionamento entre seus nós, mas sim qual é o tipo de relacionamento (entrada nula na linha i para coluna

j indica que não existe relacionamento; caso contrário o número indica o tipo de relacionamento).

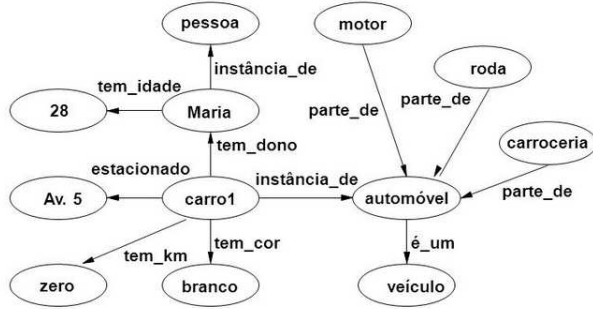
Para exemplificar, seja a *lookup table* que mapeia os relacionamentos do exemplo dado na Fig 4, que modela a situação de um automóvel zero km branco estacionado na Avenida 5 que pertence à Maria, uma jovem de 28 anos de idade.



Índice	Tipo de Relacionamento
1	instância_de
2	parte_de
3	é_um
4	tem_cor
5	tem_km
6	estacionado
7	tem_dono
8	tem_idade

Todos os relacionamentos da rede semântica foram mapeados por números naturais não nulos, consecutivos (i.e., sem lacunas). É importante que a atribuição de números seja sequencial e sem lacunas.

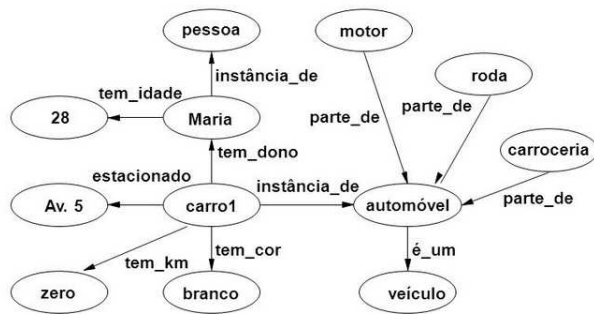
De forma semelhante, outra *lookup table* provê um mapeamento de conceitos, objetos e idéias descritos pelos nós da rede semântica. Uma forma de fazer esse mapeamento é construir um registro, estrutura ou algo semelhante que seja capaz de atribuir a cada conceito um número natural diferente. O conceito será o objeto que o nó da rede semântica representa, e o número associado será o índice da matriz correspondente à esse nó. Para exemplificar, seja o exemplo apresentado na



Índice	Conceito, Objeto ou Idéia
1	automóvel
2	carro1
3	Maria
4	motor
5	pessoa
6	28 anos
7	Avenida 5
8	zero Km
9	branco
10	roda
11	carroceria
12	veículo

Para cada nó da rede semântica original há identificador diferente mapeado. Por fim, ambas as estruturas de dados são usadas para construir uma matriz de incidência contendo nas linhas e colunas os índices dos objetos, conceitos e idéias mapeados, e em cada entrada (i, j) da matriz um valor nulo se não houver relação de i para j , ou o número natural não-nulo da *lookup table* construído no mapeamento dos tipos de relacionamentos.

Para exemplificar, a matriz de incidência do exemplo apresentado na Fig 4 é dada a seguir. Observe a matriz de incidência e os valores de sua entrada, relacionando os objetos da rede semântica e determinando o tipo de relacionamento.



0	0	0	0	0	0	0	0	0	0	0	0	3
1	0	7	0	0	0	6	5	4	0	0	0	0
0	0	0	0	1	8	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0

Índice	Tipo de Relacionamento
1	instância_de
2	parte_de
3	é_um
4	tem_cor
5	tem_km
6	estacionado
7	tem_dono
8	tem_idade

Índice	Conceito, Objeto ou Idéia
1	automóvel
2	carro1
3	Maria
4	motor
5	pessoa
6	28 anos
7	Avenida 5
8	zero Km
9	branco
10	roda
11	carroceria
12	veículo

Estas estruturas de dados são suficientes para representar a rede semântica.

Como busca na rede está fora do escopo deste trabalho, optou-se por gerar um arquivo de saída no formato `.dot`¹, uma linguagem de descrição de grafos que permite descrevê-los em um arquivo no formato ASCII para posterior renderização no pacote GraphViz, que contém os aplicativos `dot` e `neato`. Um exemplo de descrição de grafo para a rede semântica apresentada na Fig 4 e armazenada usando os códigos numéricos das *lookup tables* exemplificadas acima é dado a seguir.

```
digraph Rede {
  1 [label="automóvel"];
  2 [label="carro1"];
  3 [label="Maria"];
  4 [label="motor"];
  5 [label="pessoa"];
  6 [label="28"];
  7 [label="Av. 5"];
  8 [label="zero"];
  9 [label="branco"];
  10 [label="roda"];
  11 [label="carroceria"];
  12 [label="veículo"];
  1 -> 12 [label="é_um"];
  2 -> 1 [label="instância_de"];
  2 -> 3 [label="tem_dono"];
  2 -> 7 [label="estacionado"];
  2 -> 8 [label="tem_km"];
  2 -> 9 [label="tem_cor"];
  3 -> 5 [label="instância_de"];
  3 -> 6 [label="tem_idade"];
  4 -> 1 [label="parte_de"];
  10 -> 1 [label="parte_de"];
  11 -> 1 [label="parte_de"];
}
```

O arquivo no formato `.dot` acima pode ser compilado pelo aplicativo `dot`, parte integrante do pacote GraphViz, via linha de comando no Linux por meio da seguinte sintaxe via terminal, que resulta na Fig 6:

```
user@machine$ dot -Tpdf rede.dot -o rede.pdf
```

¹Ver artigo em <http://www.graphviz.org/Documentation/dotguide.pdf>

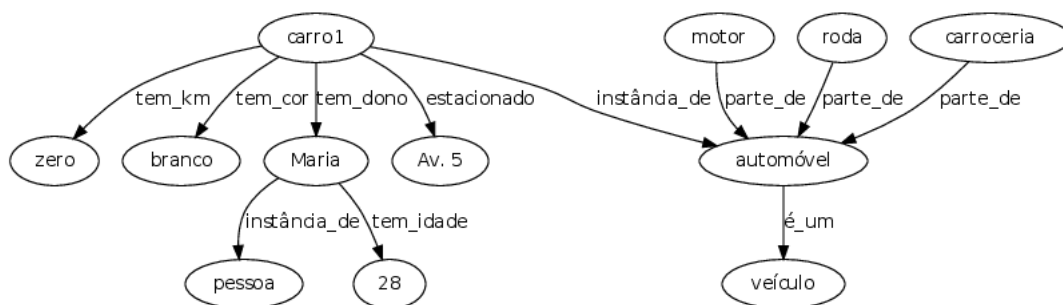


Figura 6: Saída de arquivo .dot gerada pelo aplicativo dot do GraphViz

O objetivo deste documento é descrever um aplicativo que seja capaz de receber dados para a construção de uma rede semântica por meio de arquivos. O aplicativo deverá ler o arquivo de entrada com os dados da rede, armazenar as informações sobre objetos, conceitos e idéias da rede semântica assim como o tipo de relacionamento entre eles usando *lookup tables* e matrizes de incidência, e gerar um arquivo de saída descritivo com o grafo da relação no formato .dot apresentado acima. Para facilitar o entendimento da demanda, assim como permitir uma experiência de modelagem e abstração a seção 3 detalhará os principais requisitos de *software* que a aplicação deverá conter.

3 Especificação

Esta seção apresenta uma especificação técnica de elementos que devem ser implementados pelo grupo para atender ao *software* capaz de representar uma rede semântica, especificado neste trabalho. A aplicação deverá ser escrita em linguagem C ou Free Pascal, usando os compiladores gcc para linguagem C ou fpc para a linguagem Free Pascal. A aplicação será do tipo console (linha de comando), sendo que toda a entrada e saída de dados ocorrerá mediante uso de teclado e monitor.

O código fonte da aplicação será compilado no ambiente operacional Linux, distribuição Ubuntu. Certifique-se que o seu código fonte é compatível com tal ambiente. **Códigos que não compilarem no ambiente mencionado serão desconsiderados.** Toda a implementação deverá ser entregue em um único arquivo de nome semantic-net.c se implementado em C, ou semantic-net.pas se implementado em Free Pascal. **O código fonte deverá possuir um cabeçalho em comentários contendo os nomes e matrículas dos integrantes do grupo.** O trabalho deverá ser enviado ao professor em um único arquivo compactado denominado pratico-dm-2015.zip contendo (i) o código fonte da aplicação que constrói a rede semântica e (ii) pelo menos 5 arquivos de entrada contendo especificações de redes semânticas diferentes das apresentadas neste documento.

As próximas sub-seções detalham cada requisito da aplicação. **De maneira geral, cada requisito deve ser implementado em um procedimento ou função separada**, podendo-se modularizar ainda mais cada requisito em pedaços menores desde que documentados e que facilitem a leitura do código fonte.

3.1 Requisito: Entrada de Dados via Arquivo

Este requisito trata da entrada de dados da aplicação **semantic-net**. Conforme mencionado, a entrada de dados será feita mediante arquivo ASCII cujo formato será conhecido a priori. Arquivos ASCII são facilmente criados a partir de editores de texto sem formatação, como Bloco de Notas e **Notepad++** da plataforma Windows, e **gedit** da plataforma Linux.

O nome do arquivo de entrada deverá ser informado por linha de comando, como argumento juntamente com o nome da aplicação como sugere o exemplo abaixo, para execução da aplicação após compilação na plataforma Linux:

```
user@machine$ ./semantic-net <arquivo-entrada> <arquivo-saida>
```

onde **<arquivo-entrada>** corresponde ao nome de um arquivo de entrada no formato ASCII informado pelo usuário no terminal que conterá os objetos, conceitos e idéias da rede semântica com seus tipos de relacionamento, no seguinte formato, sendo cada linha iniciada por um caracter de controle:

```
# <string: linha de comentario, ignorada até o CR>
N <inteiro: número de objetos, conceitos ou idéias>
n <natural: id> <string: nome do objeto, conceito ou idéia>
n <natural: id> <string: nome do objeto, conceito ou idéia>
n <natural: id> <string: nome do objeto, conceito ou idéia>
...
K <inteiro: número de tipos de relacionamento>
k <inteiro: id> <string: nome do relacionamento>
k <inteiro: id> <string: nome do relacionamento>
k <inteiro: id> <string: nome do relacionamento>
...
r <inteiro: inicial> <inteiro: final> <inteiro: tipo>
...
f
```

onde

- N: indica o número total de nós da rede semântica
- n: identifica unicamente cada nó da rede, com ID e descrição
- K: indica o número total de tipos de relacionamento da rede
- k: identifica unicamente cada tipo de relacionamento (arco) por meio de ID e descrição
- r: cria um relacionamento na rede semântica, unindo dois nós pelos respectivos IDs segundo um tipo de relacionamento segundo ID

- f: fim de arquivo

Para exemplificar, considere o exemplo apresentado na seção sec:semantic-relations. O arquivo de entrada que descreve os elementos da rede semantica em questão no formato do arquivo de entrada é dado por:

```
#####
#           Exemplo de rede semantica, retirada da especificacao           #
#####

# Numero de nos da rede semantica
N 12

# Descricao dos nos da rede semantica
n 1 Automovel
n 2 Carro1
n 3 Maria
n 4 Motor
n 5 Pessoa
n 6 28Anos
n 7 Avenida5
n 8 ZeroKm
n 9 Branco
n 10 Roda
n 11 Carroceria
n 12 Veiculo

# Total de tipos de relacionamento da rede semantica
K 8

# Descricao dos tipos de relacionamento
k 1 InstanciaDe
k 2 ParteDe
k 3 EhUm
k 4 TemCor
k 5 TemKm
k 6 Estacionado
k 7 TemDono
k 8 TemIdade;

# Relacionandos nós por meio de tipos de relacionamento
r 1 12 3
r 2 1 1
r 2 3 7
r 2 7 6
r 2 8 5
r 2 9 4
r 3 5 1
r 3 6 8
r 4 1 2
r 10 1 2
r 11 1 2

# Fim de arquivo
f
```

Suponha que o arquivo foi salvo com o nome `rede01.txt`. Para gerar o arquivo de descrição no formato `.dot` com nome `rede01.dot` deve-se usar a seguinte linha de comando:

```
user@machine$ ./semantic-net rede01.txt rede01.dot
```

Observe que o nome do arquivo é informado como argumento de linha de comando, e não mediante digitação pelo usuário. A leitura de argumentos por linha de comando deverá ser pesquisada pelo grupo, mas sugere-se buscar pelas funções do **Free Pascal** denominadas `ParamStr()` e `ParamCount()`. A leitura de caracteres e inteiros de arquivos texto (ASCII) também devem ser pesquisadas a parte pelo grupo para conclusão deste requisito. Sugere-se procurar compreender o funcionamento das funções `Assign()`, `Reset()`, `Read()` e `Close()`.

O grupo deverá pensar em situações problemas que devem ser tratadas na entrada de dados. Caso alguma delas ocorra a aplicação deverá ser abortada, exibindo uma mensagem de erro para o usuário no terminal. A entrada de dados deverá ser implementada em uma função auxiliar que percorrerá o arquivo lendo seu conteúdo.

3.2 Requisito: *Lookup Table* para Armazenar Objetos, Conceitos e Idéias da Rede

Neste requisito deve-se construir uma *lookup table* para armazenar os conceitos, idéias e objetos da rede semântica com seu respectivo número associado. O grupo deverá pensar em como armazenar essa associação pensando que a quantidade de nós da rede é conhecida e informada por arquivo de entrada; logo a estrutura de dados que irá mapear o par $\langle \text{identificador}, \text{objeto} \rangle$ deverá ter sua dimensão determinada em tempo de execução (o mesmo vale para a *lookup table* descrita no requisito da seção ??). Deve-se observar que nenhum identificador de objeto, conceito ou idéia deve exceder a quantidade máxima de nós da rede semântica.

3.3 Requisito: *Lookup Table* para Armazenar Tipos de Relacionamentos

Neste requisito deve-se construir uma *lookup table* para armazenar todos os tipos de relacionamentos permitidos na rede semântica. Os tipos serão usados para mapear os arcos da rede, indentificando quando existe relacionamento entre dois nós e qual é o tipo de tal relacionamento. Deve-se cuidar na construção desta estrutura que nenhum tipo de relacionamento seja mapeado com o natural zero, pois este valor é reservado na matriz de incidência descrito no requisito da seção 3.4 para indicar que não existe relacionamento entre os nós i e j quando não houver arco (i, j) no grafo que representa a rede semântica. Também deve-se garantir que nenhum identificador de tipo de relacionamento exceda o limite máximo determinado pela quantidade de tipos de relação descritas no arquivo de entrada.

3.4 Requisito: Matriz de Incidência da Rede Semântica

Neste requisito deve-se construir uma matriz de incidência para armazenar os relacionamentos entre nós da rede. Deve-se seguir o exemplo apresentado na seção 2.3, cuidando para que as relações documentadas não excedam o limite de nós da rede semântica (como por exemplo, se no arquivo de entrada estiver mapeado um relacionamento para objeto, conceito ou idéia não reconhecido na leitura dos dados). As dimensões da matriz são conhecidas apenas em tempo de execução, durante a leitura do arquivo; conseqüentemente a matriz deverá ser dimensionada usando alguma estrutura dinâmica.

3.5 Requisito: Mapeamento de ID para Objeto, Conceito ou Idéia

Neste requisito deve-se construir uma função que receba como entrada um identificador de nó (que mapeia objeto, conceito ou idéia) e que busque na *lookup table* o texto do objeto correspondente, obtido via arquivo de entrada.

3.6 Requisito: Mapeamento de ID para tipo de Relacionamento

Neste requisito deve-se construir uma função que receba como entrada um identificador de tipo de relacionamento e que busque na *lookup table* o texto do tipo correspondente, obtido via arquivo de entrada.

3.7 Requisito: Estatísticas da Rede no Terminal

Neste requisito pede-se que o grupo implemente um método capaz de extrair e imprimir no console da aplicação algumas estatísticas básicas sobre o grafo (e relação) que representa a rede semântica. Dentre as estatísticas selecionadas pede-se:

- A quantidade de nós da rede, obtida a partir da leitura do arquivo;
- A densidade de arcos do grafo: calculada como o percentual de arcos existentes na rede, dada pela razão $\frac{\#arcos}{V^2}$ onde V é o número de nós da rede;
- A quantidade de arcos da rede, obtida a partir da leitura do arquivo;
- A quantidade de vezes que cada relacionamento ocorreu no grafo;
- O grau de entrada de cada nó da rede;
- O grau de saída de cada nó da rede;
- Quais nós são sorvedouros;
- Quais nós são nascedouros;

Para exemplificar, a execução da aplicação sobre o arquivo de entrada apresentado na seção 3.1 que apresenta o requisito de entrada retornaria o que segue no terminal:

```
user@machine$ ./semantic-net rede01.txt rede01.dot
```

```
Statistics:
```

```
=====
```

1. General

- Number of Nodes : 12
- Number of Edges : 11
- Density : 7.63%

2. By Types of Relationship

- InstanciaDe: 2/11
- ParteDe: 3/11
- EhUm: 1/11
- TemCor: 1/11
- TemKm: 1/11
- Estacionado: 1/11
- TemDono: 1/11
- TemIdade: 1/11

3. By Nodes

- Automovel: In 3, Out 1
- Carro1: In 0 (it's a source), Out 5
- Maria: In 1, Out 2
- Motor: In 0 (it's a source), Out 1
- Pessoa: In 1, Out 0 (it's a sink)
- 28Anos: In 1, Out 0 (it's a sink)
- Avenida5: In 1, Out 0 (it's a sink)
- ZeroKm: In 1, Out 0 (it's a sink)
- Branco: In 1, Out 0 (it's a sink)
- Roda: In 0 (it's a source), Out 1
- Carroceria: In 0 (it's a source), Out 1
- Veiculo: In 1, Out 0 (it's a sink)

```
End of processing.
```

Observação: cada estatística solicitada acima deve necessariamente ser implementada em uma função separada. Embora tal restrição exija mais linhas de código para ser implementada, na prática a complexidade assintótica dos métodos não se altera. Todas as estatísticas podem ser obtidas a partir de algoritmos simples projetados para manipular a matriz de incidência.

3.8 Requisito: Saída de Dados no Formato Dot

Neste requisito o grupo deverá construir um procedimento capaz de ler a matriz de incidência carregada a partir de arquivo de entrada, e com auxílio das *lookup tables* de mapeamento de objetos e de tipos de relacionamento construir um arquivo de saída no formato ASCII contendo a descrição do dígrafo que representa a rede semântica. Para detalhes sobre como construir redes usando a linguagem `.dot` consultar o artigo indicado na seção 2.3.

Observação: a geração de imagens a partir do arquivo `.dot` somente ocorre após uso de aplicativos do pacote GraphViz. Tanto na plataforma Windows quanto na plataforma Linux é possível instalar o pacote, que é gratuito e pode ser executado via linha de comando em terminal do sistema correspondente. É de total responsabilidade do grupo homologar a instalação do GraphViz e garantir que o arquivo gerado seja devidamente interpretado pelos aplicativos do pacote.

3.9 Requisito: Dataset com Redes Semânticas Mapeadas

Neste requisito o grupo deverá providenciar 03 (três) arquivos de entrada no formato estabelecido no requisito da seção 3.1 que deverá ser enviado juntamente com o código fonte da aplicação. Nos arquivos deve-se mapear ao menos 15 objetos, conceitos ou idéias. As redes semânticas mapeadas nos arquivos podem ser redes existentes, desde que se indique a fonte (livro, *link* da internet, etc) ou podem ser criadas pelo grupo. É importante que a rede expresse algo real, ou seja, que ela cumpra seu papel como estrutura de representação de conhecimento. Para detalhes, consultar o livro do Ben Coppin.

3.10 Requisito: Documentação de Código

A documentação de código é importante em qualquer implementação computacional e será considerada neste trabalho. Pede-se que o arquivo que contem o código fonte possua ao menos (i) cabeçalho inicial, contendo nome do aplicativo, membros do grupo com nome e matrícula, instruções de compilação, ambiente de desenvolvimento, data e objetivo do arquivo; (ii) cabeçalho das funções auxiliares e procedimentos implementados no trabalho; (iii) comentários nos principais trechos de código de cada algoritmo, explicando resumidamente o que está sendo codificado a seguir.

3.11 Requisito: Corretude dos Resultados

O requisito mais importante do trabalho é aquele que lida com resultados corretos. Desta forma, é um requisito essencial do trabalho que além da aplicação ser implementada respeitando-se os requisitos anteriores é importante que ela retorne resultados de qualidade. Em virtude disso o grupo deverá testar exaustivamente a aplicação, gerando arquivos de entrada para diferentes redes semânticas. Os arquivos de entrada usados no teste devem ser compactados junto com o código fonte e

enviados para o professor da disciplina no e-mail `diego.silva@ifmg.edu.br` até a data limite do trabalho.

4 Barema de Correção

Conforme mencionado, o trabalho prático tem valor de 10 pontos. A correção seguirá o barema apresentado a seguir, que lista os requisitos do trabalho prático e suas respectivas pontuações.

Requisito	Pontos
Entrada de Dados via Arquivo	5
Lookup Table para Armazenar Objetos, Conceitos e Idéias	1
Lookup Table para Armazenar Tipos de Relacionamentos	1
Matriz de Incidência da Rede Semântica	2
Mapeamento de ID para Objeto, Conceito ou Idéia	1
Mapeamento de ID para tipo de Relacionamento	1
Estatísticas da Rede no Terminal	5
Saída de Dados no Formato Dot	5
Dataset com Redes Semânticas Mapeadas	3
Documentação de Código	1
Corretude dos Resultados	5
TOTAL	30.0 pts

Caso haja necessidade ou suspeita de plágio é reservado o direito à arguição por parte do professor para o grupo ou cada um de seus membros em separado.

5 Bibliografia

- Principles of Semantic Networks - Explorations in Representation of Knowledge. Morgan Kaufmann Publisher. ISBN: 1-55860-088-4.
- Inteligência Artificial. Ben Coppin, Editora LTC, 2012 (ISBN: 978-85-216-1729-7)