



# Raciocínio Probabilístico

Raciocínio sobre os padrões de interrupção para sistemas de trens  
usando Rede de Bayesian e Prolog

Alexandre Lacerda Moura dos Santos



# Objetivo

O sistema ferroviário da Indonésia enfrenta diariamente várias interrupções nas linhas.

Os trens da Indonésia são os elétricos ou também conhecidos como os Commuter Line.

Tendo isso o objetivo desse trabalho era identificar as interrupções onde e quando iriam acontecer usando a Rede Bayesianas.



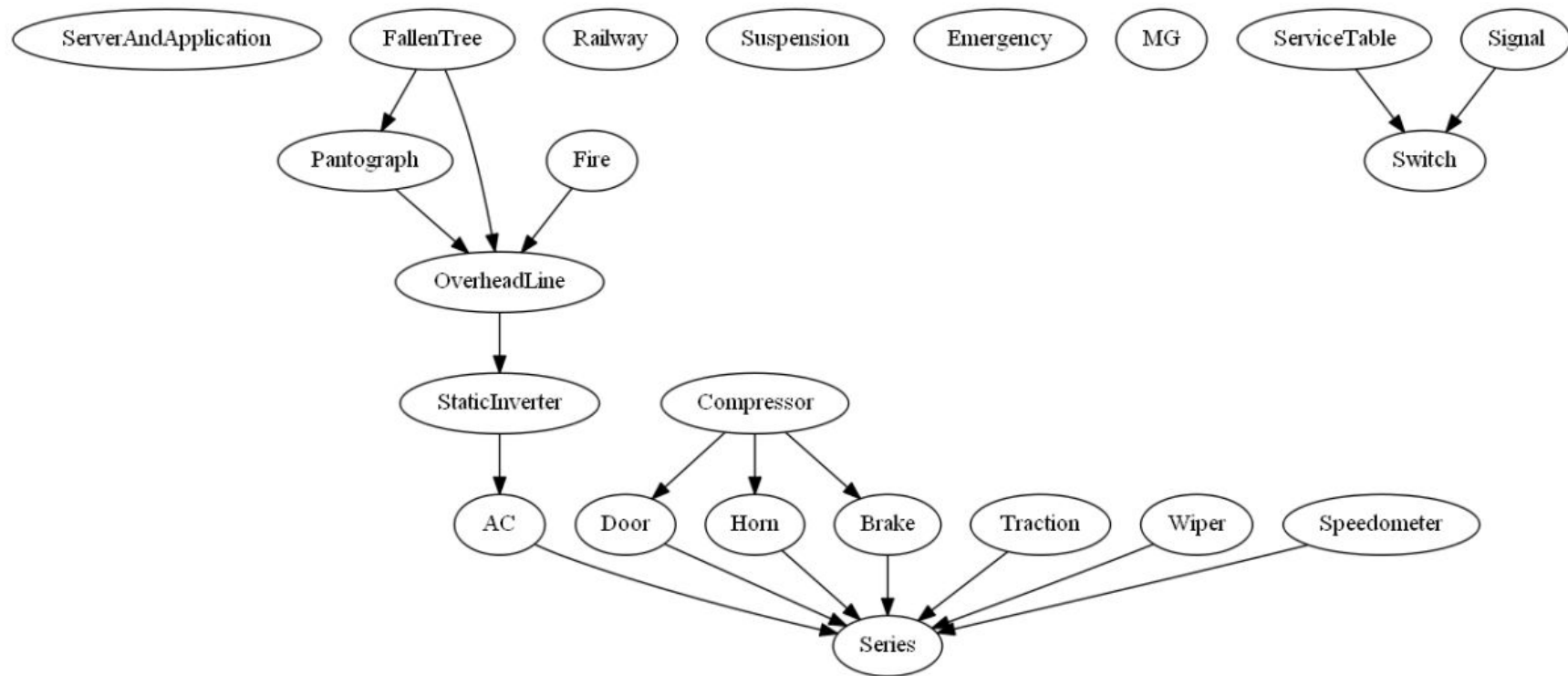
# Os tipos de interrupções

As interrupções que fazem a linha parar:

- (1.) Ar Condicionado (AC).
- (2.) Freio.
- (3.) Compressor.
- (4.) Porta.
- (5.) Emergência.
- (6.) Árvore caída
- (7.) Fogo.
- (8.) Buzina.
- (9.) Motor gerador (MG).
- (10.) Linha aérea
- (11.) Pantógrafo.

As interrupções que fazem a linha parar:

- (12.) Ferrovia.
- (13.) Série.
- (14.) Mesa de serviço.
- (15.) Servidor e aplicação.
- (16.) Sinal.
- (17.) Velocímetro
- (18.) Inversor Estático
- (19.) Suspensão.
- (20.) Interruptor.
- (21.) Tração.
- (22.) Limpador.



# Especificações das consultas no prolog

## Q1

Calcule a probabilidade de uma certa interrupção das outras interrupções.

## Q2

Encontre caminhos de uma interrupção para outras interrupções a partir de uma determinada probabilidade condicional do problema.

## Q3

Encontre o caminho de interrupção de uma interrupção para outra interrupção que contém e não contém certa perturbação.

# Especificações das consultas no prolog (cont.)

## Q4

Encontre as interrupções desencadeadas comuns de duas outras interrupções. Esta consulta tem como objetivo encontrar interrupções que levaram diretamente a duas outras interrupções. Para qualquer interrupção  $D_i$ ,  $D1$  é a interrupção desencadeada comum de  $D2$  e  $D3$  se  $D2$  e  $D3$  tiverem arestas diretas para  $D1$ .

## Q5

Encontre as interrupções desencadeantes comuns que levam a outras interrupções. Esta consulta tem como objetivo encontrar interrupções que podem desencadear diretamente duas outras interrupções. Para qualquer interrupção  $D_i$ ,  $D1$  é a interrupção de disparo comum de  $D2$  e  $D3$  se  $D1$  tiver uma borda direta para  $D2$  e  $D3$ .

---

# Especificações das consultas no prolog (cont.)

## Q6

Encontre a interrupção mais desencadeante na rede.

## Q7

Encontre a interrupção mas desencadeada na rede.

# Definição dos fatos do prolog (Q2)

```
adjacent(A, B ,[A,B]) :- parent(A, B).  
path(A, B, [A,B]) :- adjacent(A,B, [A,B]).  
path(A, B, [A | Rest]) :-  
    adjacent(A, X, [A,X]),  
    path(X, B, Rest).  
  
path_probability(A, B, Path, Prob) :-  
    path(A, B, Path), prob(B, [A], P),  
    Prob is P.
```



# Definição dos fatos do prolog (Q3)

```
show_path(A,A,T,P) :- reverse([A|T],P).  
show_path(A,Z,T,P) :- edge(A,B), not(member(A,T)), show_path(B,Z,[A|T],P).  
show_path(A,B,P) :- show_path(A,B,[],P).
```

```
causing(A, B, Path, Cause) :-  
    show_path(A, B, Path), subset([Cause], Path).
```

```
match(L1,L2) :- member(X,L1),member(X,L2).  
avoid(L1,L2) :- not(match(L1,L2)).
```

```
notCausing(A,B,Path,Avoid) :-  
    show_path(A, B, Path), avoid([Avoid], Path).
```

## Definição dos fatos do prolog (Q4)

```
com_triged_dis(X,Y,Z):-  
    ancestor(X,Z), ancestor(Y,Z).  
show_com_triged_dis(X, Y, Com_Triged_Dis) :-  
    findall(Z, com_triged_dis(X,Y,Z), L), sort(L, Com_Triged_Dis).
```

## Definição dos fatos do prolog (Q5)

```
com_triging_dis(X,Y,Z):-  
    ancestor(Z,X), ancestor(Z,Y), X \= Y.  
show_com_triging_dis(Z, Com_Triging_Dis) :-  
    findall([X,Y], com_triging_dis(X,Y,Z), L), sort(L, Com_Triging_Dis).
```

# Definição dos fatos do prolog (Q6)

```
descendant(X,Y) :- parent(X,Y).
descendant(X,Y) :- parent(X,Z), descendant(Z,Y).

descendants([Node, Descendant, Length]) :-
    node(Node), findall(A, descendant(Node,A), L),
    sort(L, Descendant), length(Descendant, Length).

all_descendant(Result) :-
    findall([Node, Descendant, Length],
        descendants([Node, Descendant, Length]), Result).

compare_descending('<', [-, -, X], [-, -, Y]) :- X > Y, !.
compare_descending('>', -, _N).

sort_descendant_descending(Sorted) :-
    all_descendant(Result),
    predsor(sort_descending, Result, Sorted).
max_descendant(L) :-
    sort_descendant_descending(Result), nth0(0, Result, L).
detail_max_descendant([Node, Descendant, Length]) :-
    max_descendant(L),
    nth0(0, L, Node),
    nth0(1, L, Descendant),
    nth0(2, L, Length).
```

# Definição dos fatos do prolog (Q7)

```
ancestor(X,Y) :- parent(X,Y).
ancestor(X,Y) :- parent(X,Z), ancestor(Z,Y).
ancestors([Node, Ancestor, Length]) :-
    node(Node), findall(A, descendant(A, Node), L),
    sort(L, Ancestor), length(Ancestor, Length).

all_ancestor(Result) :-
    findall([Node,Ancestor, Length], ancestors([Node,Ancestor, Length]), Result).
sort_ancestor_descending(Sorted) :-
    all_ancestor(Result),
    predsor(sort(compare_descending, Result, Sorted)).
max_ancestor(L) :-
    sort_ancestor_descending(Result), nth0(0, Result, L).
detail_max_ancestor([Node, Ancestor, Length]) :-
    max_ancestor(L),
    nth0(0, L, Node),
    nth0(1, L, Ancestor),
    nth0(2, L, Length).
```

# Resultados

A interrupção mais acionada (usando a rede Bayesiana) é a interrupção em série. Pode levar a cinco outras interrupções: AC, linha aérea, pantógrafo, inversor estático e interrupção em série.

# Conclusão

O sistema proposto pode ser usado para raciocinar casos relacionados a padrões de dados e relação parte-todo no sistema ferroviário. Colocamos várias questões para todas as redes bayesianas sobre a probabilidade, causalidade entre cada interrupção para outras interrupções e as interrupções mais desencadeadas e desencadeadas em cada rede

The image features two thin, dark horizontal lines. The top line starts with a curved segment on the left side, and the bottom line ends with a curved segment on the right side.

Obrigado

# Referências

- Artigo científico Reasoning about the disruption patterns for train system using Bayesian Network and Prolog