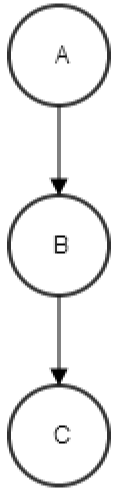
Equivalência Relacional Gramatical

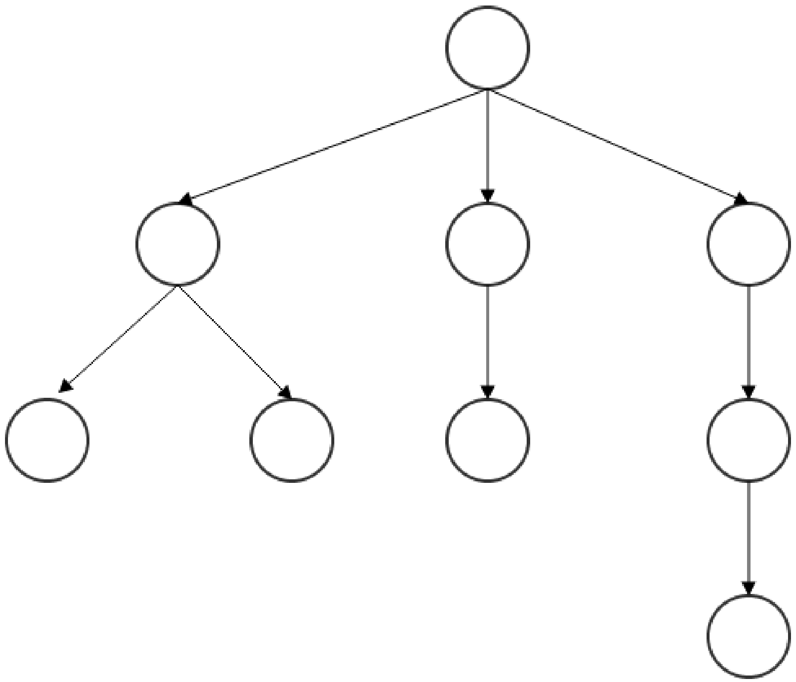
Pedro Henrique Pamplona Savarese

Motivação

- Algoritmo de Blondel:

- Equivalência a partir de um *role model*

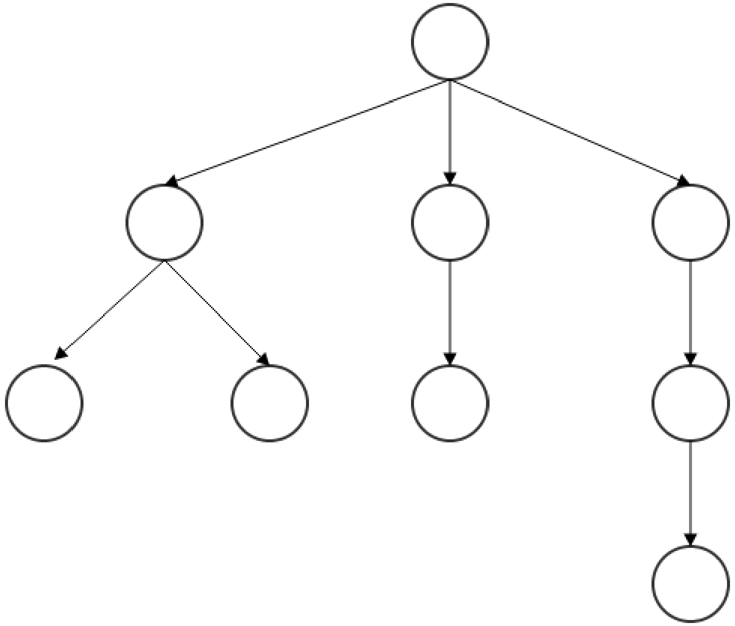




Motivação

- *Role Model* também pode ser dado de forma gramatical:

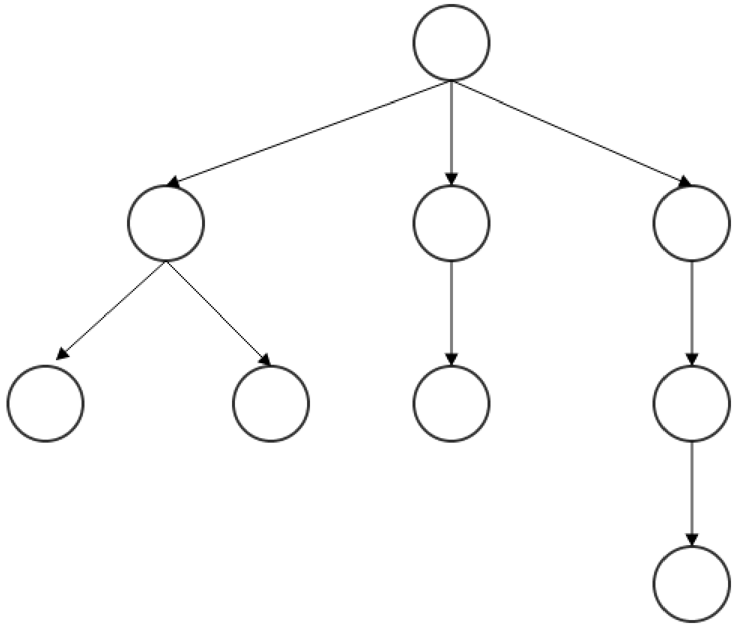
$$A \to B, \qquad B \to C$$



Gramáticas

- Problema: de onde começar?
  - *Sink-Driven*: $A \to B,\ B \to B,\ B \to \blacksquare$
  - *Origin-Driven:* $* \to B,\ B \to B, B \to C$

  Onde $\blacksquare$ é um simbolo terminal e $*$ é o símbolo de início



Gramáticas

- Problema: de onde começar?

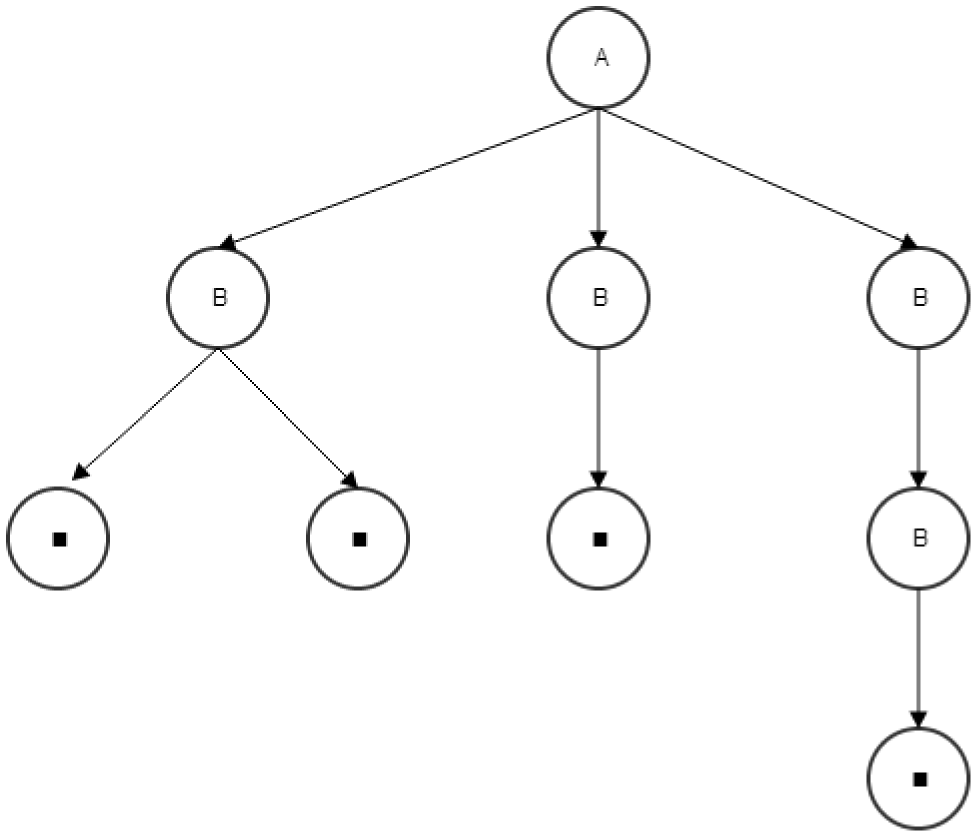
  - *Sink-Driven*: $A \rightarrow B,\ B \rightarrow B,\ B \rightarrow \blacksquare$

  - *Origin-Driven:* $* \rightarrow B,\ B \rightarrow B, B \rightarrow C$

  Onde $\blacksquare$ é um simbolo terminal e $*$ é o símbolo de início

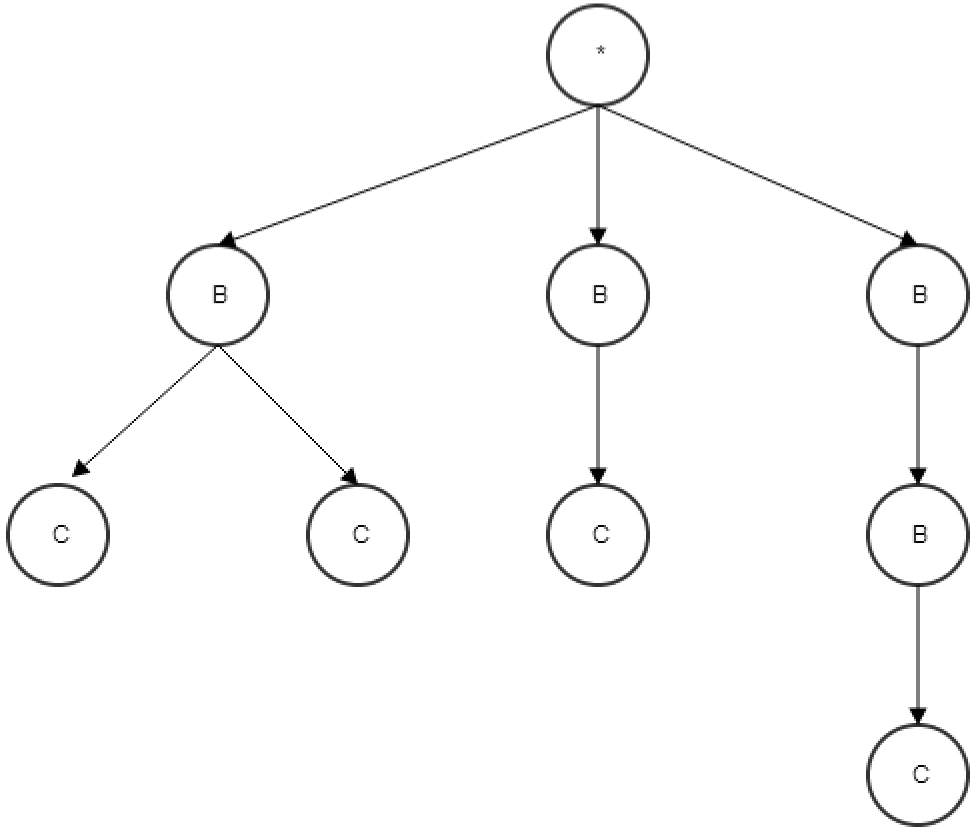
Exemplo

Sink-Driven: $A \rightarrow B,\ B \rightarrow B,\ B \rightarrow \blacksquare$



Exemplo

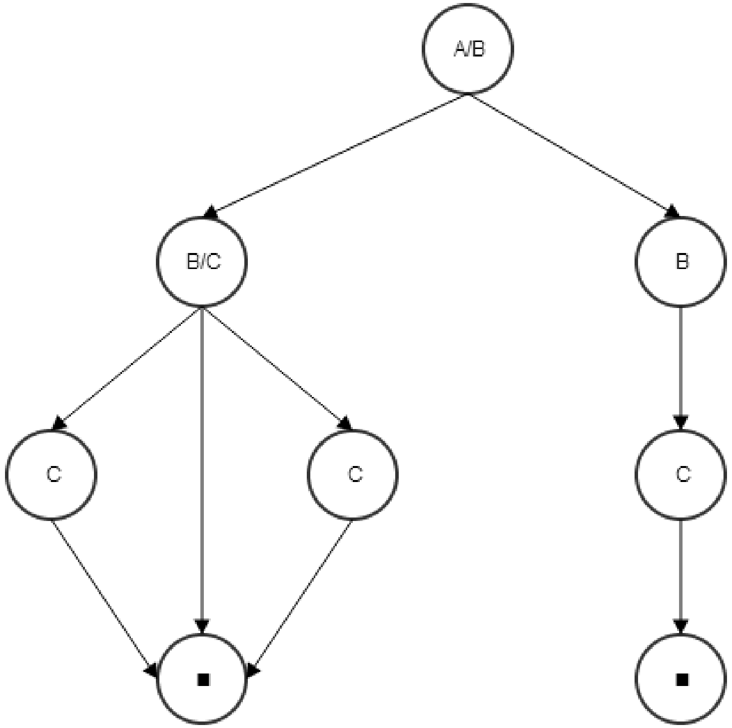
Origin-Driven: $* \rightarrow B,\ B \rightarrow B, B \rightarrow C$



Múltiplos papéis

- Dependendo da gramática e da rede, podemos ter múltiplos papéis

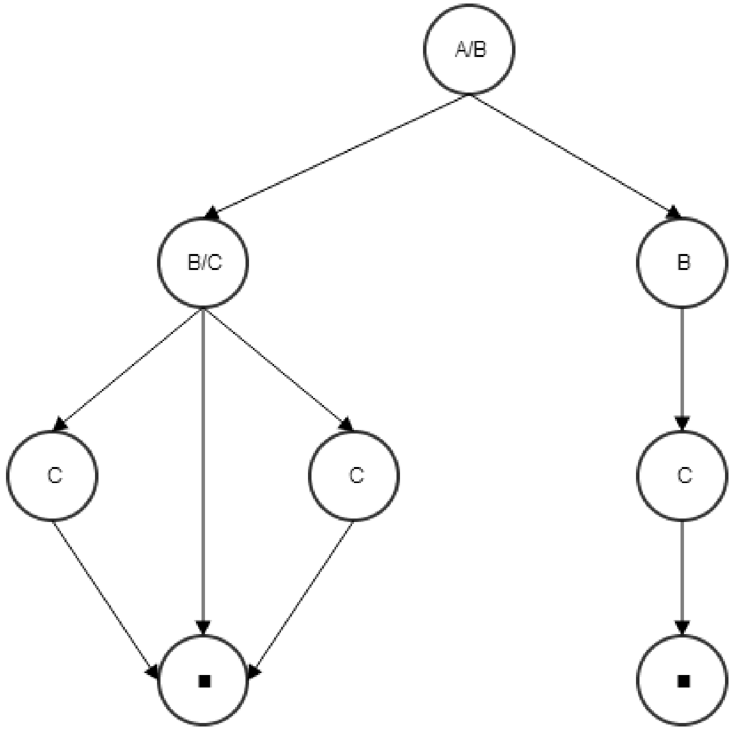
$$A \rightarrow A, \mathrm{A} \rightarrow B, B \rightarrow C, C \rightarrow \blacksquare$$



Múltiplos papéis

- Podemos dizer que o nó “B/C” é “duas vezes mais B que C”
- Representação vetorial:

(0, 2, 1, 0) -> pai de dois C’s e um ■



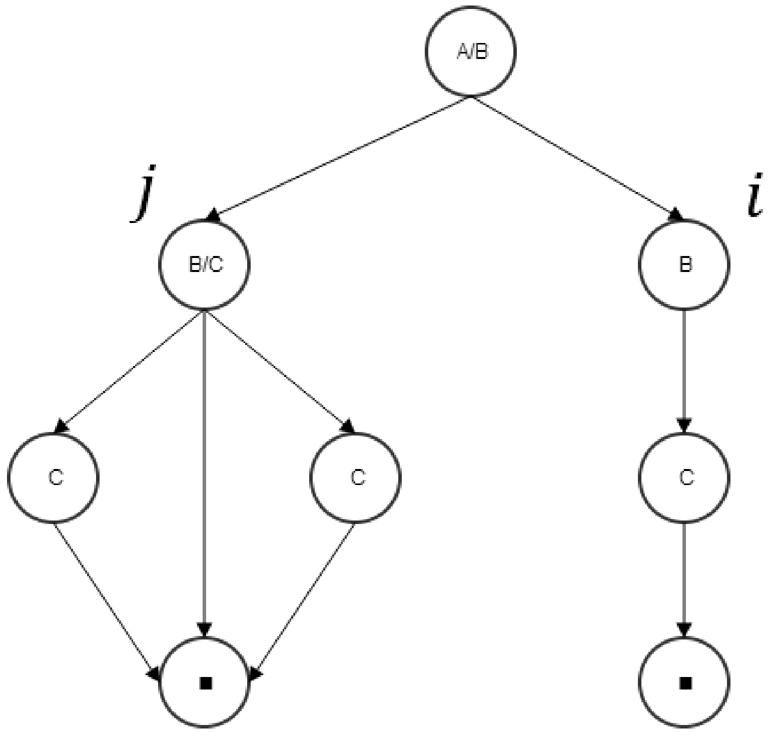
Múltiplos papéis

- Representação vetorial nos possibilita comparar nós

- $N_i = (0,1,0,0)$
- $N_j = (0,2,1,0)$

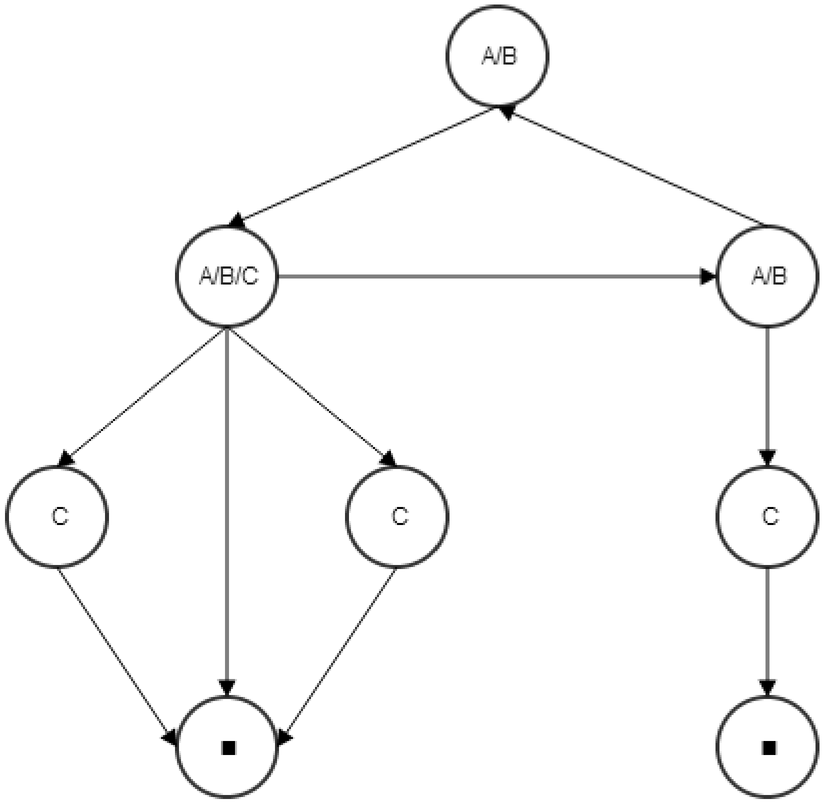
- $Sim(i,j) = \widehat{N}_i \,.\, \widehat{N}_j = \frac{2}{3}$

- Onde $\widehat{N}_j = \frac{N_j}{|N_j|} = (0, \frac{2}{3}, \frac{1}{3}, 0)$



Ciclos

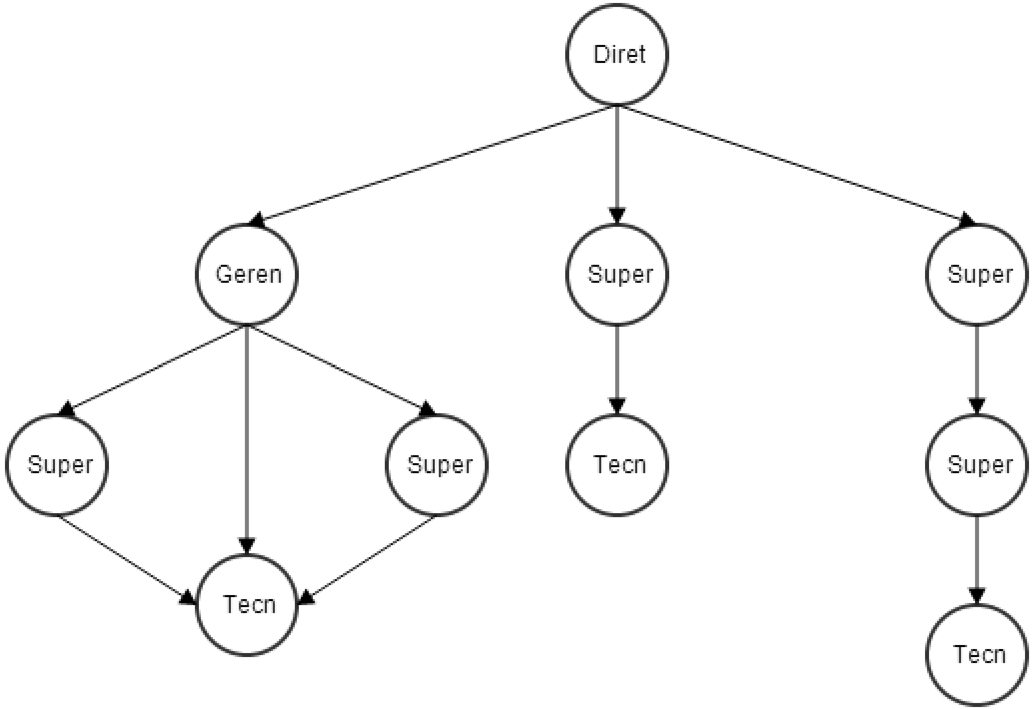
- *Sink-Based:* percorrer apenas arestas do destino para a origem, excluíndo-as
- Estratégia para ordenar arestas



Regras de Aglomeração

Exemplo *sink-based*, $Tecn = \blacksquare$

- $Super \rightarrow Tecn$
- $Geren \rightarrow Super^{x}\ Tecn^{y}, x + y \geq 2$
- $Diret \rightarrow Gen^{x}\ Super^{y}, x \geq 1$

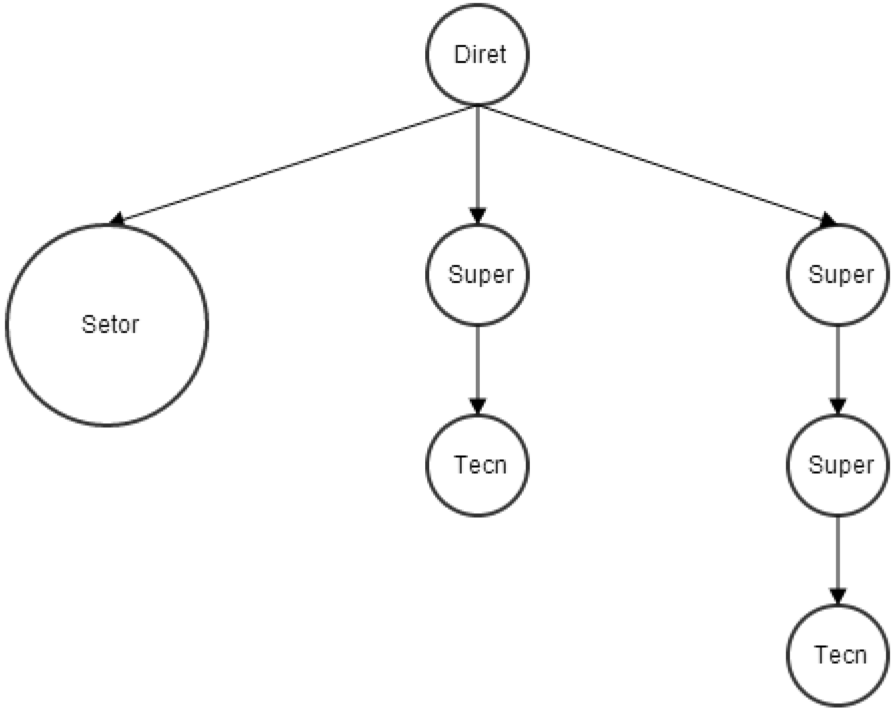


Regras de Aglomeração

Exemplo *sink-based*, $Tecn = \blacksquare$

- $Super \rightarrow Tecn$
- $Geren \rightarrow Super^x\ Tecn^y, x + y \geq 2$
- $Diret \rightarrow Gen^x\ Super^y, x \geq 1$

- Outro tipo de regra:
- $(Gerer \rightarrow Super \rightarrow Tecn) \Rightarrow Setor$



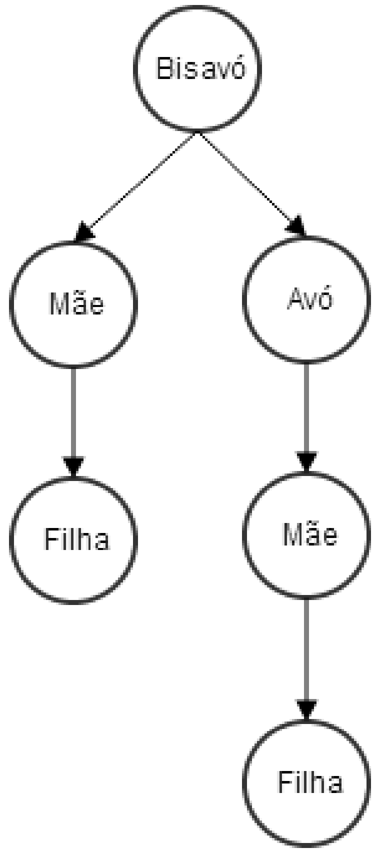
Regras com pesos

- Podemos priorizar regras atribuindo pesos

$$Bisavó \rightarrow Avó, w = 3$$
$$Avó \rightarrow Mãe, w = 2$$
$$Mãe \rightarrow Filha, w = 1$$
$$Filha = \blacksquare$$

Maximização dos pesos diminui papéis multiplos

Dá prioridade à certas relações



Problemas

- *Sink-based* necessita de nós sem grau de saída
- *Origin-based* necessita de um nó escolhido a priori (ou um nó sem grau de entrada)
- Ciclos: resultado depende da estratégia de escolha de arestas, ou requer iterações até convergência (converge?)
- Regras com pesos: algoritmo polinomial? Restringir escopo?
- Regras de aglomeração: casos muito específicos, pode-se generalizar?
- Conjunto de regras precisa conter uma auto-indução (mas qual?)
- Útil?