UNIVERSIDADE FEDERAL DE PERNAMBUCO

Cálculo de Conexões e a Regra de Corte

Dimas Melo Filho

Roteiro

- Introdução
- Definições
- Cálculo de Conexões
- Redundância
- Folding Up e Folding Down
- Anti-Lemmata
- Conclusão

Introdução

- Cálculo de Conexões (com Tableau)
  - Extensão do Tableau Calculus
  - Guiado por Conexões
- Conexões
  - Idéia Introduzida por Andrews e Bibel
  - Utiliza Literais COMPLEMENTARES para guiar a prova.

Diferenças

- Cálculo de Conexões (Com Tableau)
  - Utiliza estrutura de Tableau
  - Usa prova por refutação com a FNC
- Método de Conexões
  - Utiliza estrutura de Matrizes
  - Usa prova direta com a FND

- São diferentes, mas utilizam princípios semelhantes do **CÁLCULO DE CONEXÕES**

FNC vs FND

- FNC: Forma Normal Conjuntiva

$$(L_1 \vee L_2 \vee L_3) \wedge (L_4 \vee L_5 \vee L_6) \wedge (L_7 \vee L_8 \vee L_9) \ldots$$

- FND: Forma Normal Disjuntiva

$$(L_1 \wedge L_2 \wedge L_3) \vee (L_4 \wedge L_5 \wedge L_6) \vee (L_7 \wedge L_8 \wedge L_9) \ldots$$

- De forma geral a FNC é usada em provas por refutação e a FND em provas diretas.

FNC vs FND

- Deseja-se deduzir α a partir de KB
  - $KB=(K_1 \wedge K_2 \wedge ... \wedge K_n)$ conjunção de condições
  - $\alpha=(L_1 \wedge L_2 \wedge ... L_n)$ conjunção de consequências
- Prova por Refutação: $\neg(KB \models \alpha)$ é falso
  - Equivale a: $K_1 \wedge K_2 \wedge ... \wedge K_n \wedge \neg(L_1 \wedge L_2 \wedge ... \wedge L_m)$
  - Ou seja: $K_1 \wedge K_2 \wedge ... \wedge K_n \wedge (\neg L_1 \vee \neg L_2 \vee ... \vee \neg L_m)(FNC)$
- Prova Direta: $KB \models \alpha$ é verdadeiro
  - Equivale a: $\neg(K_1 \wedge K_2 \wedge ... \wedge K_n) \vee (L_1 \wedge L_2 \wedge ... \wedge L_m)$
  - Ou seja: $\neg K_1 \vee \neg K_2 \vee ... \vee \neg K_n \vee (L_1 \wedge L_2 \wedge ... \wedge L_m)(FND)$

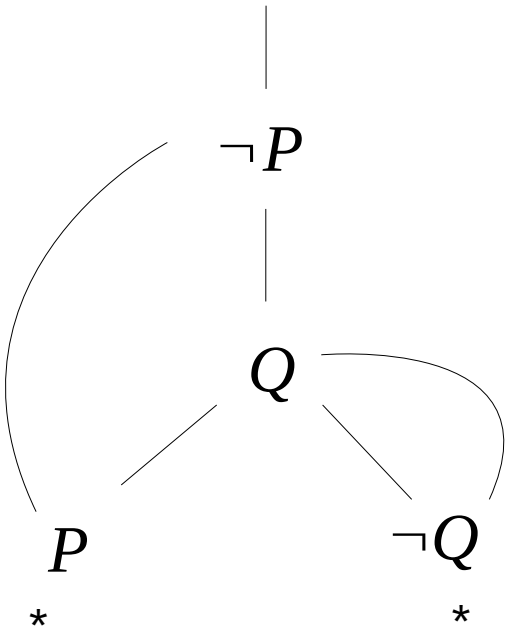
Conexão

- A conexão é a base do cálculo de conexões
  - Uma conexão consiste de um par de literais complementares. $\langle L, \neg L \rangle$

- Uma conexão na FNC resulta em $\bot$
  - Pois $L \wedge \neg L \models \bot$
- Uma conexão na FND resulta em $\top$
  - Pois $L \vee \neg L \models \top$

Conexões - Exemplo FNC

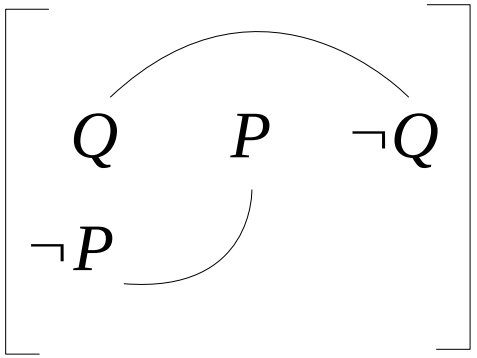
- Considere: $(P \vee \neg Q), \neg P \models \neg Q$
- Prova por Refutação (FNC): $(P \vee \neg Q) \wedge \neg P \wedge Q$
- Tableau:



Conexões – Exemplo FND

- Considere: $(P \vee \neg Q), \neg P \models \neg Q$
- Prova Direta (FND): $(\neg P \wedge Q) \vee P \vee \neg Q$

- Matriz:



(tautologia)

Conexões - Objetivo

- O principal objetivo das conexões é guiar o cálculo, independente do método ou da forma normal usada.

- Sem as conexões um mecanismo automático teria que buscar todas as combinações de cláusulas possíveis.

Cálculo de Conexões

- Consiste das seguintes operações:
  - Expansão
  - Redução
  - Extensão
  - Início
  - Axioma
- Para os exemplos a seguir considere as cláusulas:

$$(P \vee R) \wedge (\neg P \vee Qx) \wedge \neg Qy \models R$$

$$FNC(Refutação): (P \vee R) \wedge (\neg P \vee Qx) \wedge \neg Qy \wedge \neg R$$
$$FND(Direto): (\neg P \wedge \neg R) \vee (P \wedge \neg Qx) \vee Qy \vee R$$

Cálculo de Conexões - Expansão

- Tableau
  - Expandir o Tableau com uma variante de uma cláusula do conjunto de fórmulas.
- Matriz
  - Equivale a passar por uma coluna.
- Fórmula
  - Não é utilizada no cálculo com fórmula.

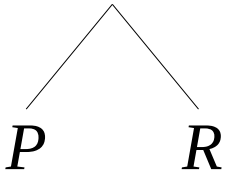
Expansão – Exemplo Tableau

$$FNC: (P \vee R), (\neg P \vee Qx), \neg Qy, \neg R$$

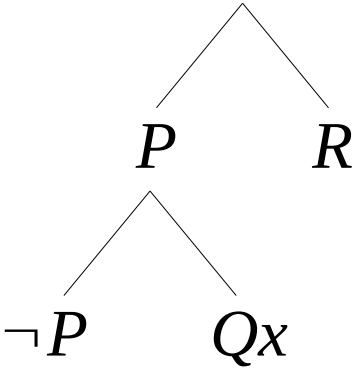
Cláusula Inicial

Cláusula para Expandir

Antes da Expansão



Após a Expansão



Expansão – Exemplo Matriz

$$FND: (\neg P \wedge \neg R), (P \wedge \neg Qx), Qy, R$$

Cláusula Inicial

Cláusula para Expandir

Antes da Expansão

$$\begin{bmatrix} \neg P & P & Qy & R \\ \neg R & \neg Qx & & \end{bmatrix}$$

Após a Expansão

$$\begin{bmatrix} \neg P & P & Qy & R \\ \neg R & \neg Qx & & \end{bmatrix}$$

Andou para a próxima coluna

Cálculo de Conexões - Redução

- Tableau
  - Fechar um caminho com uma conexão.
- Matriz
  - Fechar um caminho com uma conexão.
- Fórmula

$$\frac{C, M, P \cup \{L_2\}}{C \cup \{L_1\}, M, P \cup \{L_2\}}$$

$C = Cláusula\ objetivo$

$M = Cláusulas\ disponíveis\ (Matriz)$

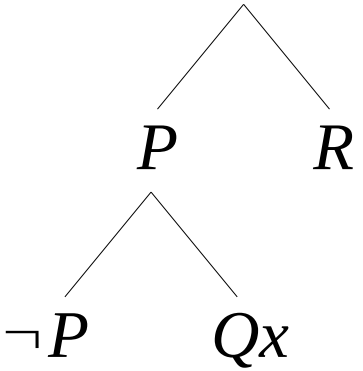
$P = Caminho$

$L_x = Literais \qquad \sigma L_1 = \sigma \overline{L_2}$

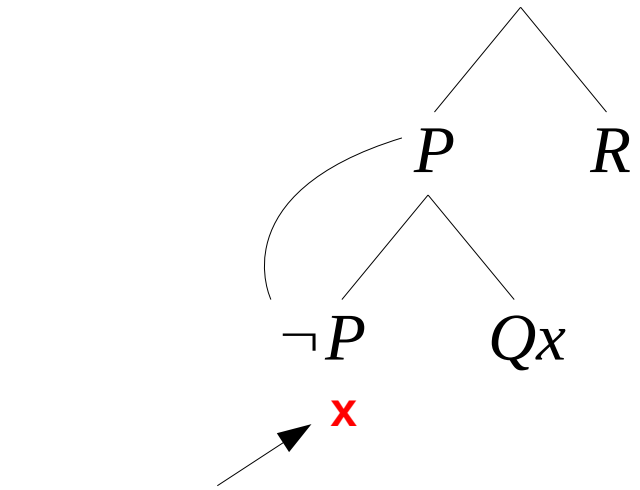
Redução – Exemplo Tableau

$$FNC: (P \vee R), (\neg P \vee Qx), \neg Qy, \neg R$$

Antes da Redução



Após a Redução

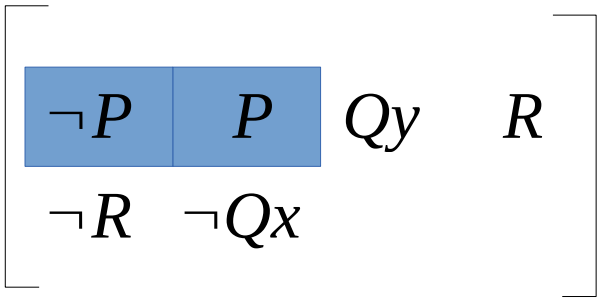


Caminho Fechado

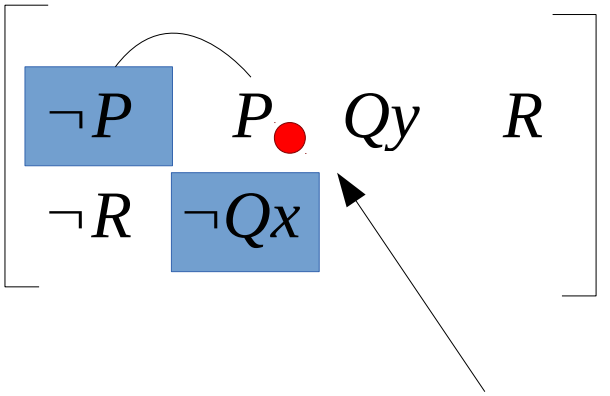
Redução – Exemplo Matriz

$$FND: (\neg P \wedge \neg R), (P \wedge \neg Qx), Qy, R$$

Antes da Redução



Após a Redução



Fechou caminho com a conexão

Cálculo de Conexões - Extensão

- Tableau
  - Expansão seguida de Redução
- Matriz
  - Expansão seguida de Redução
- Formula

$$\frac{C_2 \setminus \{L_2\}, M, P \cup \{L_1\} \qquad C_1, M, P}{C_1 \cup \{L_1\}, M, P}$$

$C_1, C_2 = cláusulas$

$M = Cláusulas\ disp.$

$P = Caminho$

$L_1, L_2 = Literais$

$\sigma L_1 = \sigma \overline{L_2}$

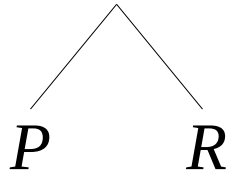
$L_2 \in C_2$

Extensão – Exemplo Tableau

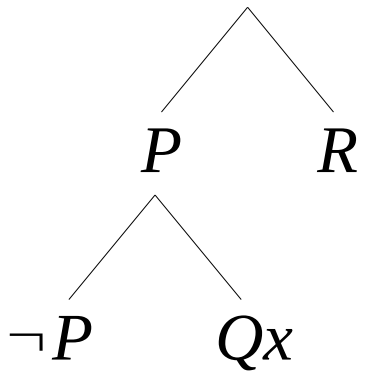
$$FNC: (P \vee R), (\neg P \vee Qx), \neg Qy, \neg R$$

L_1 C_1 L_2 C_2

Antes da Extensão



Após a Extensão



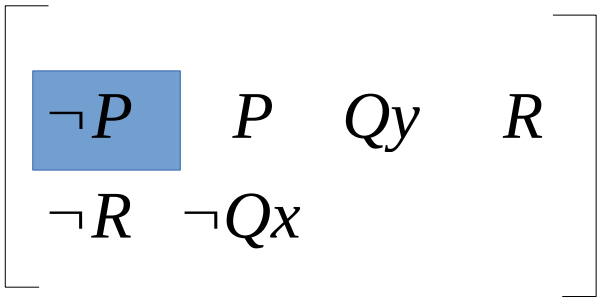
Caminho Fechado

Extensão – Exemplo Matriz

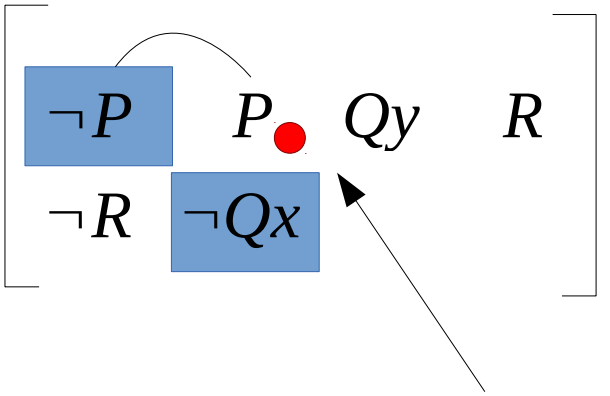
$$FND: (\neg P \wedge \neg R), (P \wedge \neg Qx), Qy, R$$

L_1 C_1 L_2 C_2

Antes da Extensão



Após a Extensão



Fechou caminho com a conexão

Cálculo de Conexões – Início

- Tableau
  - Expandir a cláusula inicial.
- Matriz
  - Selecionar a cláusula inicial.
- Fórmula

$$\frac{C_1, M, \{\}}{\varepsilon, M, \varepsilon}$$

$$C_1 = cláusula \in M$$
$$M = Cláusulas\ disp.$$
$$\varepsilon = Vazio$$

Cálculo de Conexões - Axioma

- Tableau
  - Quando todos os caminhos estão fechados.
- Matriz
  - Quando não é possível percorrer nenhum caminho novo, há uma conexão bloqueando cada caminho possível.
- Fórmula

$$\overline{\{\}, M, P}$$

$$M = Cláusulas\ disp.$$
$$P = Caminho$$

Redundância

- Um dos problemas mais comuns no raciocínio automático é a redundância.
  - Quando se gera ou utiliza a mesma cláusula mais de uma vez.
  - Quando se é necessário provar a mesma coisa mais de uma vez antes de chegar a prova final.
  - Quando se tem cláusulas com literais repetidos.
- Esses fatores aumentam o espaço de busca.
- Existem estratégias para diminuir a redundância.

Regularidade

- Característica de não repetição de um literal em partes de uma prova.
- Tableau
  - Nenhum ramo da árvore pode conter literais repetidos em seus caminhos.
- Matriz
  - O caminho ativo nunca pode conter o mesmo literal mais de uma vez.

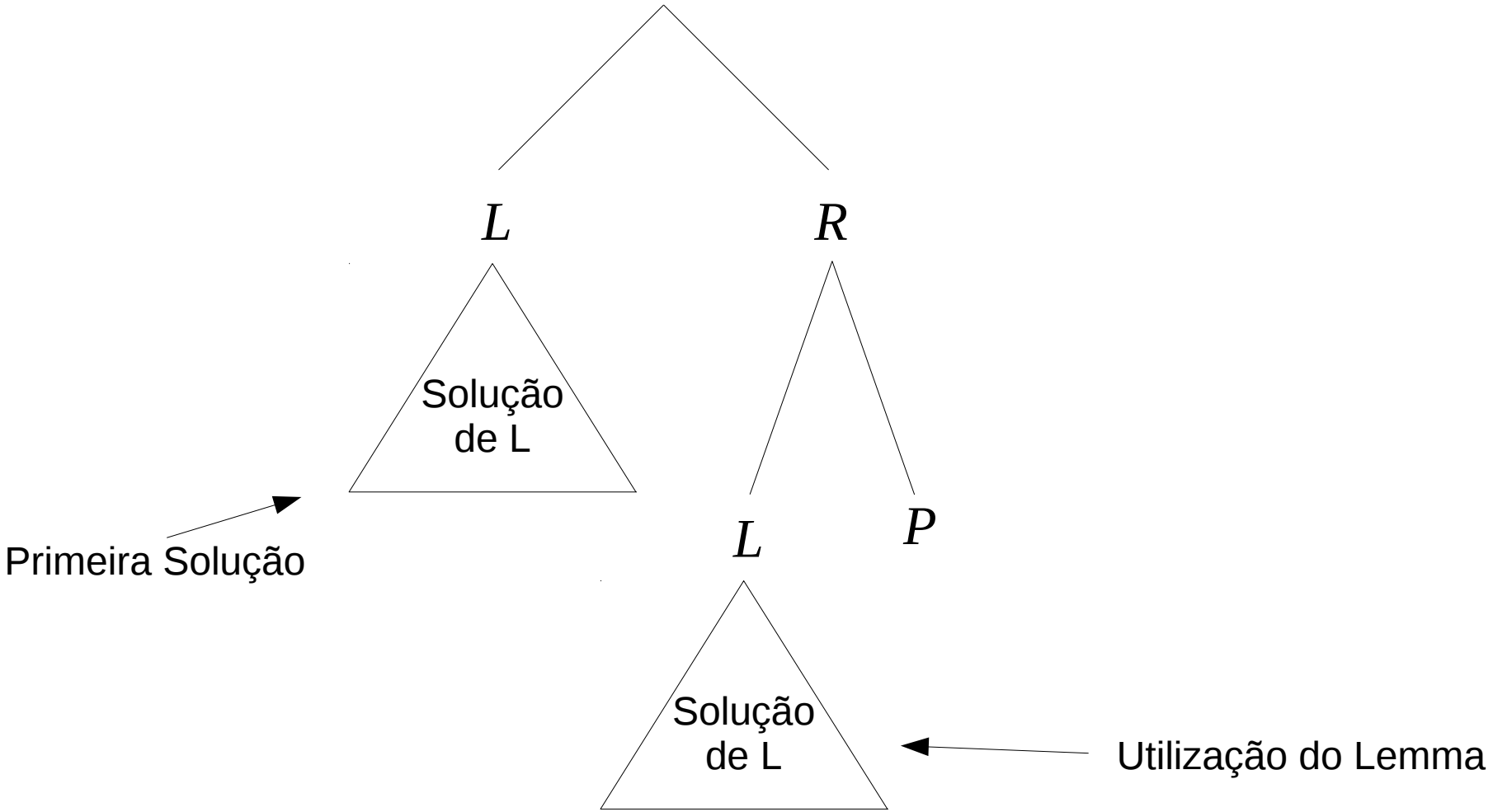
Regularidade Forte

- Não repetição de um literal N em sua prova.
- Tableau
  - Para qualquer sub-árvore do Tableau partindo de N, o literal N não aparece nenhuma vez.
- Matriz
  - Nenhum caminho percorrido pode passar pelo mesmo literal mais de uma vez.

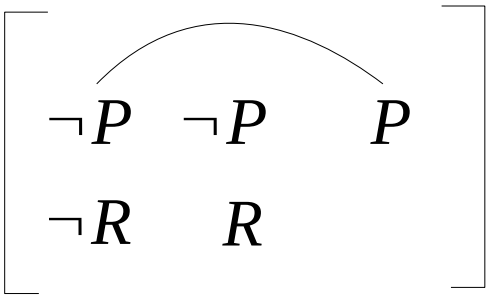
Lemmata

- Lemmas são provas já realizadas.
- Em provas muito grandes, é comum ter de provar a mesma coisa mais de uma vez.
- Os provadores modernos utilizam caches de lemmas, evitando fazer a mesma prova mais de uma vez.

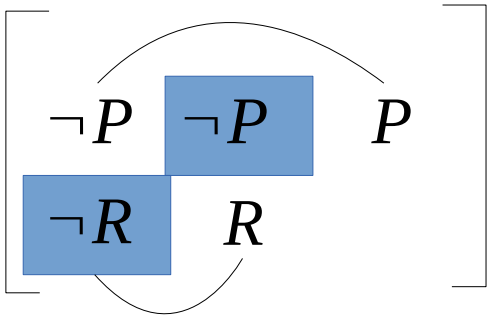
Lemmata – Exemplo Tableau



Lemmata – Exemplo Matriz



Provou $\neg P$, vira lemma para o vizinho $\neg R$



Ao tentar provar $\neg R$, $\neg P$ já foi provado e pode ser ignorado.

Lemmata

- Lemmata no Tableau, também chamado de Fatorização, pode ser otimista ou pessimista.
- Abordagem pessimista
  - só considera um lemma depois de prová-lo.
- Abordagem otimista
  - Considera como lemma um literal se o mesmo aparece em outra posição ainda a ser explorada na árvore.
- Deve-se ter o cuidado de não criar ciclos de dependência.

Lemmata

- No cálculo de conexões, pode-se utilizar lemmas através da fórmula:

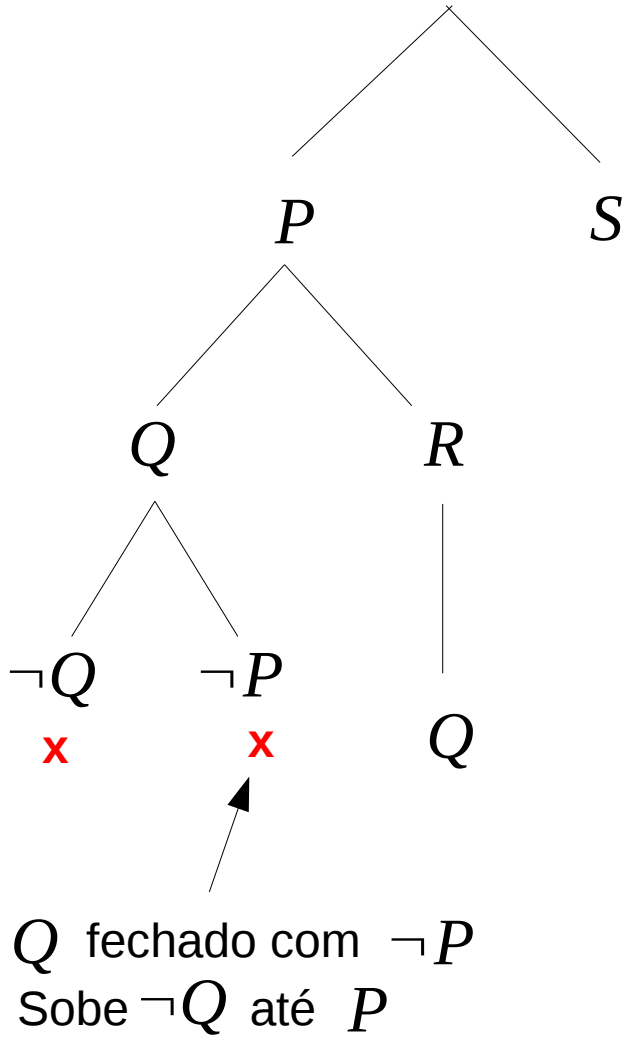
$$\frac{C, M, P, Lem \cup \{L_2\}}{C \cup \{L_1\}, M, P, Lem \cup \{L_2\}}$$

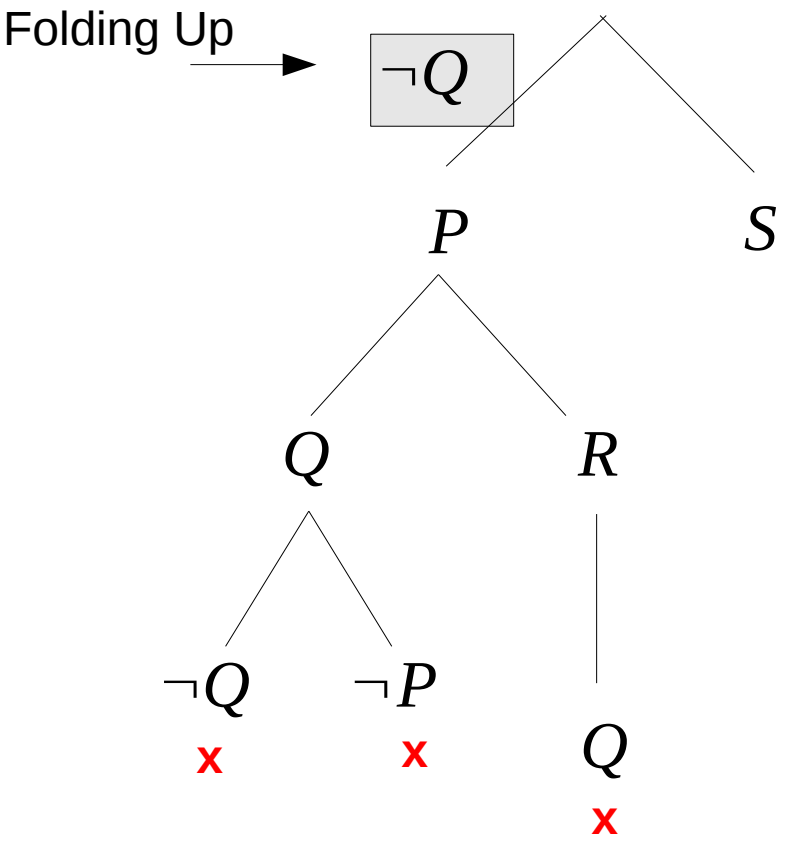
$C = Cláusula\ objetivo$
$M = Cláusulas\ disp.$
$P = Caminho$
$L_1, L_2 = Literais$
$\sigma L_1 = \sigma L_2$
$Lem = Conj.\ de\ Lemmas$

Folding Up

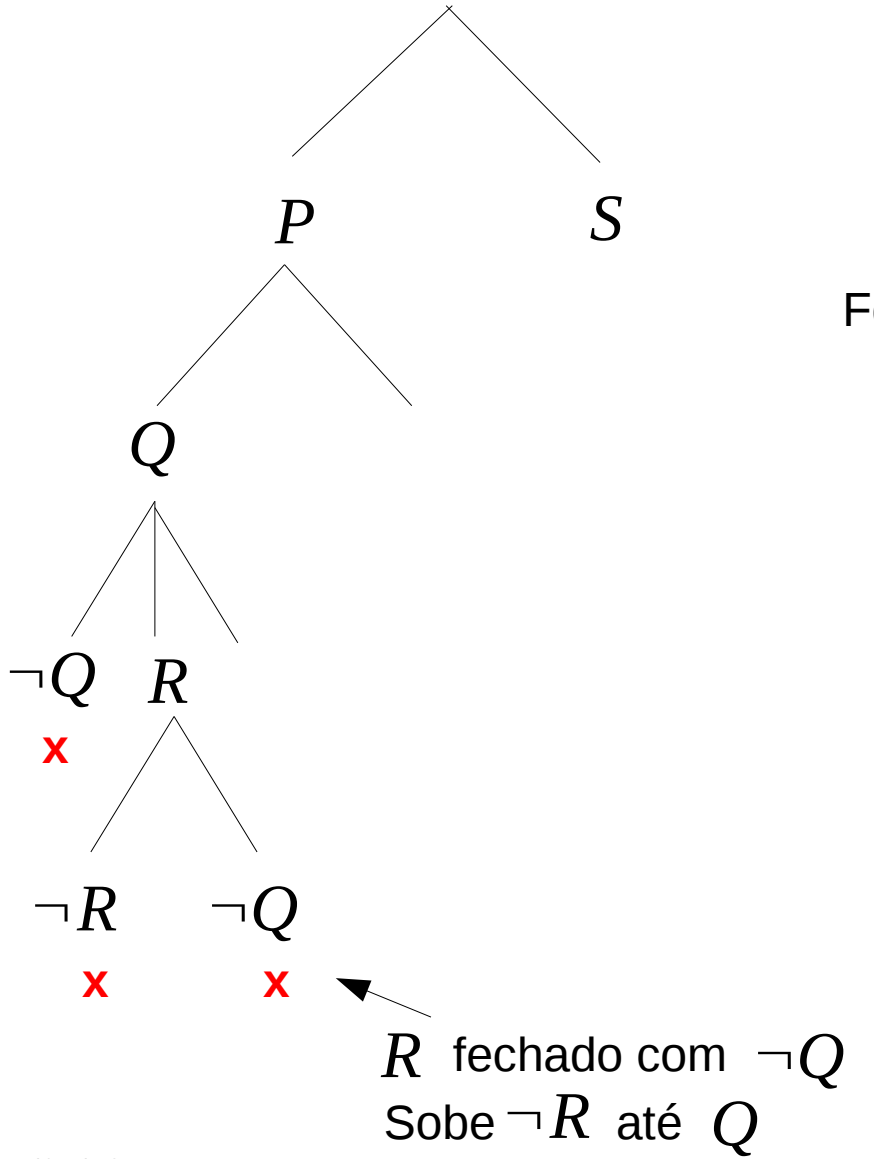
- Regra que funciona como um Lemmata local. De forma semelhante a uma Fatorização Pessimista.
- Em uma árvore T, qualquer coisa provada abaixo de um nó N usando seu complemento ~N pode ser usada para provar outras partes de N.
- Quando um nó é fechado com seu próprio complemento, sem dependências, o complemento pode ser usado para provar qualquer outra parte de N.

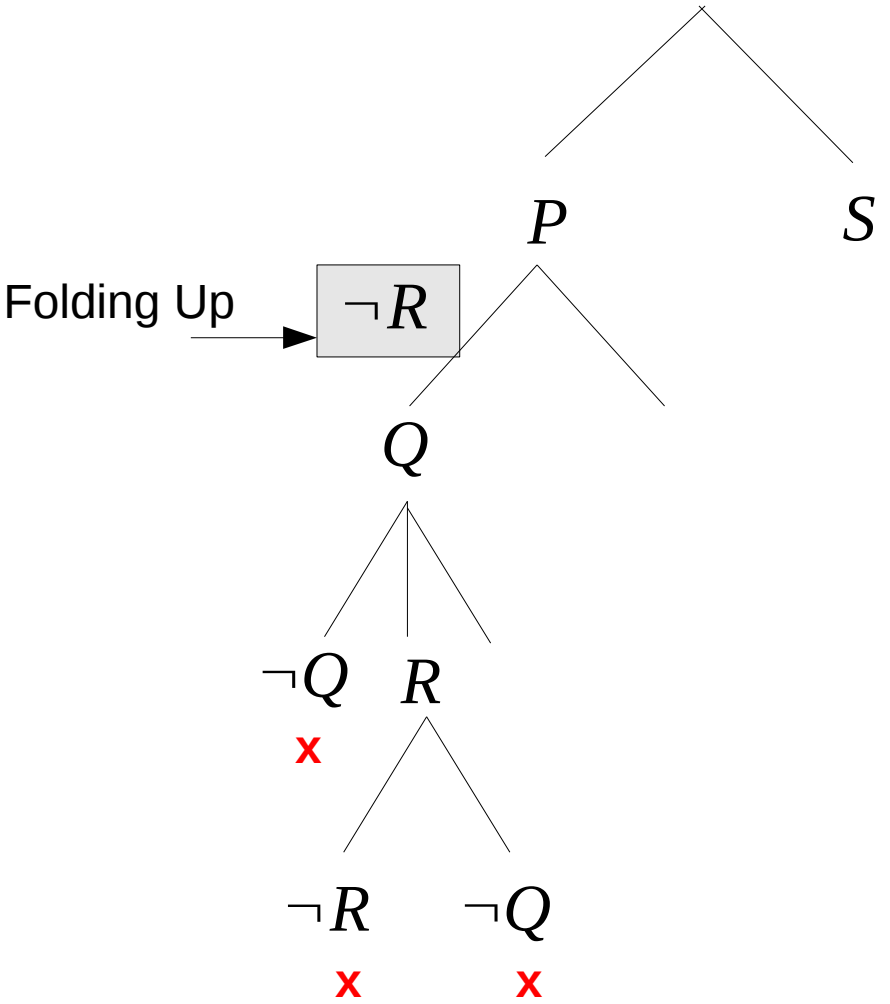
Folding Up - Exemplo



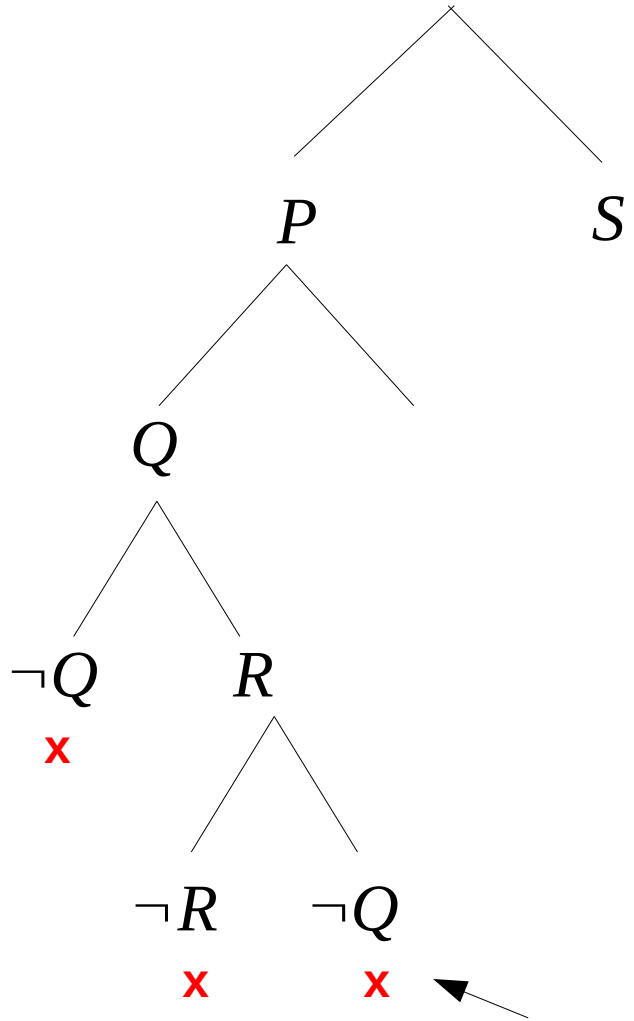


Folding Up – Exemplo 2





Folding Up – Exemplo 3

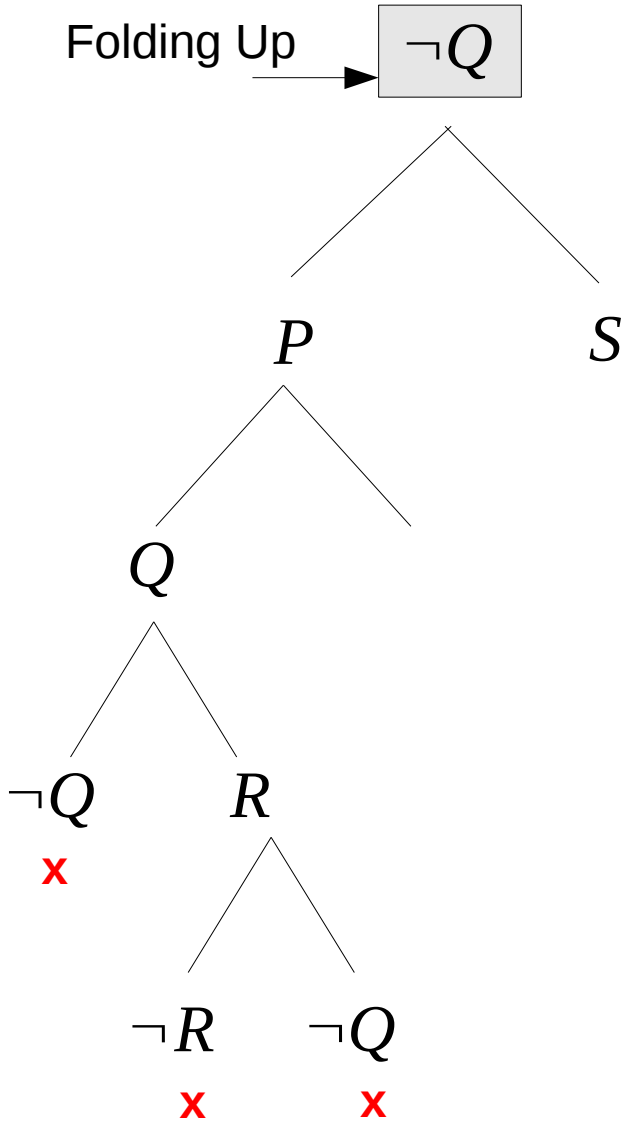


Q fechado com $\neg Q$

Sem dependência de P

Sobe $\neg Q$ até a raíz

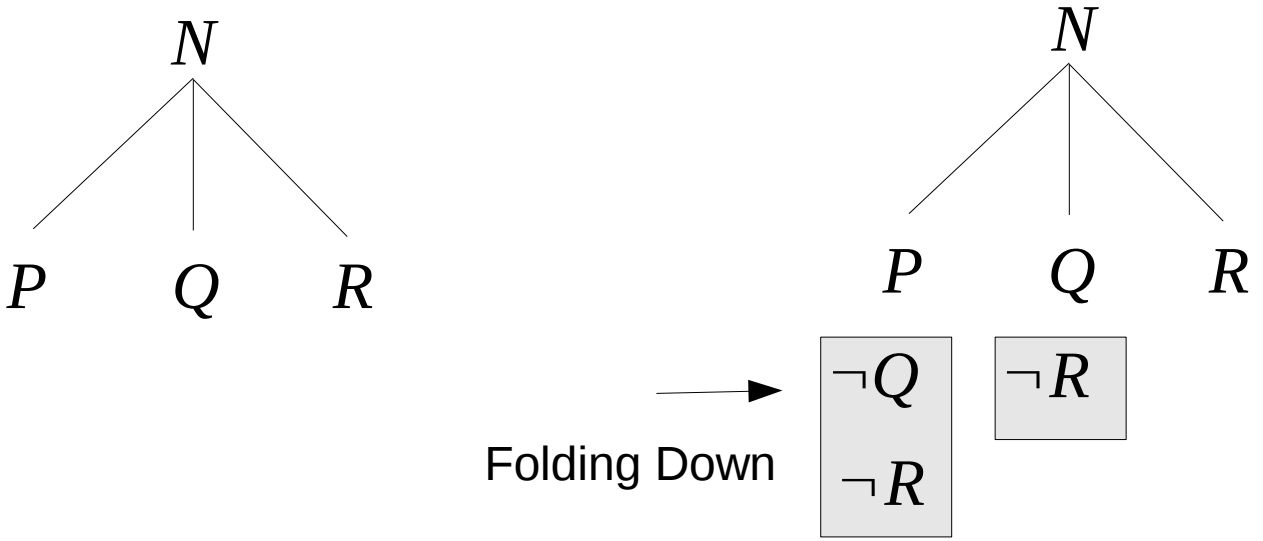
Folding Up



Folding Down

- Regra que funciona de forma semelhante a uma Fatorização Otimista. Considera-se a negação de irmãos não explorados.
- Para qualquer nó N, pode-se utilizar a negação de todos os descendentes diretos não explorados para tentar provar um outro descendente direto.
- Depende da ordem em que os nós serão explorados.

Folding Down - Exemplo



Pois

$$P \vee Q \vee R \equiv (P \wedge \neg Q \wedge \neg R) \vee (Q \wedge \neg R) \vee R$$

Folding Down – Tabela Verdade

P	Q	R	¬Q	¬R	F_1	F_2
0	0	0	1	1	**0**	**0**
0	0	1	1	0	**1**	**1**
0	1	0	0	1	**1**	**1**
0	1	1	0	0	**1**	**1**
1	0	0	1	1	**1**	**1**
1	0	1	1	0	**1**	**1**
1	1	0	0	1	**1**	**1**
1	1	1	0	0	**1**	**1**

$$F_1 = P \vee Q \vee R$$
$$F_2 = \left(P \wedge \neg Q \wedge \neg R\right) \vee \left(Q \wedge \neg R\right) \vee R$$

Restrição de Combinação

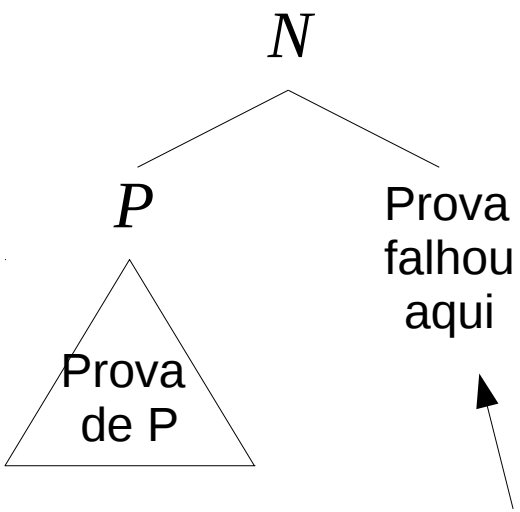
- Não é possível combinar o folding up com o folding down de qualquer forma.
- A combinação pode gerar ciclos de dependência, assim como na Fatorização.

Anti-Lemmata

- Para provar N, temos que tentar provar todos os sub-objetivos de N.
- Quando se prova um sub-objetivo de N, a prova é armazenada em N.
- Se a tentativa de provar N falhar e for necessário fazer backtracking, a prova armazenada vira um anti-lemmata.
- Se em uma nova tentativa de provar N, após o backtracking encontrar um termo que com substituição seja igual ao anti-lemmata, é feito backtracking imediato.

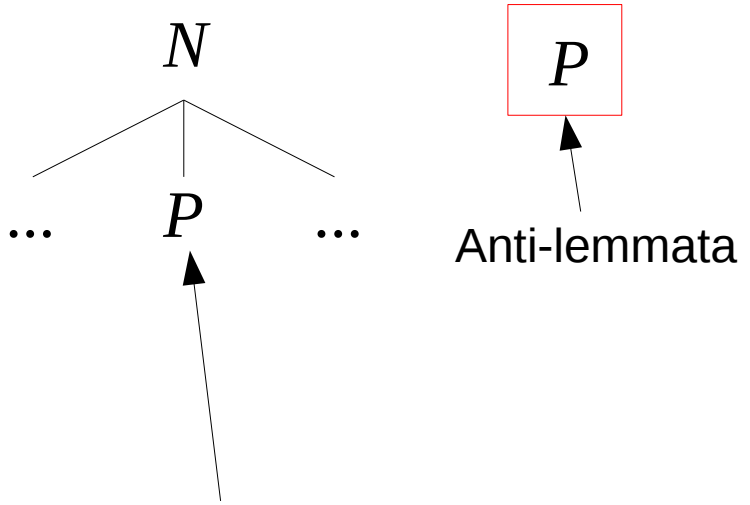
Anti-Lemmata - Exemplo

Primeira Tentativa



P vira um
Anti-Lemmata
E faz backtrack
Para tentar provar
de outra forma

Segunda Tentativa



Encontrou um anti-lemmata
Faz backtracking imediatamente.

Experimentos

- Experimentos em Letz et al. (1994) mostram um ganho de desempenho ao utilizar Cálculo de Conexões com Folding Up e Folding Down.
- O ganho é ainda maior com a utilização de Folding Up e Anti-lemmata em conjunto.

Conclusão

- As técnicas de redução de Redundância derivadas da fatorização se mostram eficientes no cálculo de conexões.
- O uso de anti-lemmata reduz a redundância entre tentativas de prova diferentes, após backtracking.

Referências

- Bibel, W. (1981). Mating in Matrices. In: Proceedings of the German Workshop on Artificial Intelligence (GWAI). pp 171-187.
- Letz et al. (1994). Controlled Integration of the Cut Rule into Connection Tableau Calculi. In: Journal of Automated Reasoning. Vol 13. Issue 3. pp 297-337.
- Otten, J. (2010). Restricting Backtracking in Connection Calculi. In: AI Communications. 23. pp 159-182.
- Robinson, J. & Voronkov A. (2001). Handbook of Automated Reasoning. MIT Press.