

# Agentes Lógicos III

Profa. Mercedes Gonzales  
Márquez

# Verificação eficiente de modelos proposicionais

## Satisfatibilidade Booleana (SAT)

O problema de determinar se uma sentença proposicional é satisfazível (se existe um modelo que a torna verdadeira) é o famoso problema **SAT**. Foi o primeiro problema provado como NP-completo.

## Davis-Putnam-Logemann-Loveland (DPLL)

O algoritmo DPLL é um método **completo** para resolver o problema SAT. Ele determina se uma sentença na Forma Normal Conjuntiva (FNC) é satisfazível.

Essencialmente, é uma busca em profundidade com retrocesso (backtracking) no espaço de modelos possíveis.

Ele atribui valores (V ou F) às variáveis uma por uma.

O que o torna especial são as heurísticas que permitem podar a árvore de busca drasticamente.

# Verificação eficiente de modelos proposicionais

Ele incorpora três técnicas para fazer mais eficiente esta enumeração recursiva de profundidade.



## 1. Encerramento Antecipado

Detecta falsidade ou verdade de cláusulas mesmo com modelos parciais, evitando avaliar todas as variáveis.



## 2. Símbolo Puro

Identifica variáveis cujo sinal é sempre o mesmo nas cláusulas restantes, fixando seu valor imediatamente.



## 3. Cláusula Unitária

Quando uma cláusula tem apenas um literal não atribuído, seu valor é forçado, criando um efeito cascata.

# HEURÍSTICA 1: ENCERRAMENTO ANTECIPADO

- O algoritmo pode avaliar cláusulas mesmo sem ter atribuído valores a todas as variáveis. Como a FNC é uma conjunção (AND) de cláusulas, se **uma única cláusula for avaliada como Falsa**, a sentença inteira será Falsa.
- Exemplo:  $(A \vee B) \wedge (A \vee C) \wedge (B \vee C)$   
Se atribuirmos **A = Falso** e **C = Falso**, a cláusula do meio se torna Falsa.  
O DPLL retrocede imediatamente, sem precisar testar os valores de B!

# HEURÍSTICA 2: SÍMBOLO PURO

- **Variáveis com Sinal Único**
- Um símbolo é **puro** se aparece com o mesmo sinal (sempre positivo ou sempre negativo) em todas as cláusulas que ainda não foram avaliadas como verdadeiras. Podemos fixar o valor desse símbolo para tornar essas cláusulas verdadeiras.
- Exemplo:  $(\mathbf{A} \vee \neg B) \wedge (\neg B \vee \neg C) \wedge (\mathbf{A} \vee C)$   
O símbolo **A** é puro (aparece apenas positivamente).  
Atribuímos **A = Verdadeiro** imediatamente, satisfazendo a primeira e a terceira cláusulas.

# HEURÍSTICA 3: CLÁUSULA UNITÁRIA

## Propagação Unitária (Efeito Cascata)

Uma cláusula é **unitária** se todos os seus literais, exceto um, já foram avaliados como Falsos no modelo parcial. Para que a cláusula seja Verdadeira, o literal restante **tem seu valor forçado**.

Exemplo: Se temos a cláusula  $(A \vee B)$  e já atribuímos **A = Falso**.

A cláusula se reduz a **(B)**. O valor de **B é forçado a Verdadeiro**.

Isso pode simplificar outras cláusulas, criando uma "propagação unitária" em cadeia.

# RESUMO DA EXECUÇÃO DO DPLL

## 1. Inicialização

Recebe a sentença em FNC e um modelo vazio.

## 2. Aplicação de Heurísticas (Loop)

Verifica encerramento antecipado. Procura símbolos puros e atribui valores. Procura cláusulas unitárias e força valores.

## 3. Busca (Branching)

Se as heurísticas não resolverem tudo, escolhe uma variável não atribuída, tenta V (e chama recursivamente). Se falhar, tenta F (backtracking).

## 4. Conclusão

Retorna o Modelo Satisfazível ou declara que a sentença é Insatisfazível (Completo).

# EXECUÇÃO DO DPLL

Exemplo: Fórmula em FNC

$$(A \vee B \vee \neg C) \wedge (\neg A \vee C) \wedge (B \vee \neg D) \wedge (\neg B \vee D) \wedge (\neg A \vee \neg C \vee D)$$

**C1:**  $(A \vee B \vee \neg C)$

**C2:**  $(\neg A \vee C)$

**C3:**  $(B \vee \neg D)$

**C4:**  $(\neg B \vee D)$

**C5:**  $(\neg A \vee \neg C \vee D)$

# EXECUÇÃO DO DPLL

## 1. Branching (Escolha)

Escolhemos  $A = \text{Verdadeiro (V)}$ . Modelo Parcial:  $\{A=V\}$

## 2. Encerramento Antecipado & Simplificação

$C1(V \vee B \vee \neg C) \rightarrow \text{Satisfeita!}$  |  $C2(\neg V \vee C) \rightarrow \text{Reduz para (C)}$  |  $C5(\neg V \vee \neg C \vee D) \rightarrow \text{Reduz para } (\neg C \vee D)$

## 3. Cláusula Unitária (Cascata 1)

$C2$  agora é unitária:  $(C)$ . Força  $C = V$ . Modelo Parcial:  $\{A=V, C=V\}$

## 4. Propagação Unitária (Cascata 2)

Em  $C5(\neg C \vee D)$ , como  $C=V$ , reduz para  $(D)$ . Força  $D = V$ . Modelo Parcial:  $\{A=V, C=V, D=V\}$

## 5. Propagação Unitária (Cascata 3)

Em  $C3(B \vee \neg D)$ , como  $D=V$ , reduz para  $(B)$ . Força  $B = V$ . Modelo Completo:  $\{A=V, B=V, C=V, D=V\}$

# EXECUÇÃO DO DPLL

## 1. Branching (Escolha)

Escolhemos  $A = \text{Verdadeiro (V)}$ . Modelo Parcial:  $\{A=V\}$

## 2. Encerramento Antecipado & Simplificação

$C1(V \vee B \vee \neg C) \rightarrow \text{Satisfeita!}$  |  $C2(\neg V \vee C) \rightarrow \text{Reduz para (C)}$  |  $C5(\neg V \vee \neg C \vee D) \rightarrow \text{Reduz para } (\neg C \vee D)$

## 3. Cláusula Unitária (Cascata 1)

$C2$  agora é unitária:  $(C)$ . Força  $C = V$ . Modelo Parcial:  $\{A=V, C=V\}$

## 4. Propagação Unitária (Cascata 2)

Em  $C5(\neg C \vee D)$ , como  $C=V$ , reduz para  $(D)$ . Força  $D = V$ . Modelo Parcial:  $\{A=V, C=V, D=V\}$

## 5. Propagação Unitária (Cascata 3)

Em  $C3(B \vee \neg D)$ , como  $D=V$ , reduz para  $(B)$ . Força  $B = V$ . Modelo Completo:  $\{A=V, B=V, C=V, D=V\}$

# Verificação eficiente de modelos proposicionais

1: Execução Direta (Sem Back-tracking) Use este exemplo primeiro para mostrar como as regras de simplificação resolvem o problema sem precisar adivinhar valores.

Fórmula Inicial:  $\Phi = (A \vee B) \wedge (\neg A \vee C) \wedge (\neg C) \wedge (A \vee \neg B \vee D)$

Passo 1 (Propagação Unitária PU): A terceira cláusula ( $\neg C$ ) é unitária. Para torná-la verdadeira, precisamos assumir  $C = \text{Falso}$ .

Passo 2 (Simplificação): Substituindo  $C = \text{Falso}$  na fórmula, a segunda cláusula ( $\neg A \vee C$ ) vira ( $\neg A \vee \text{Falso}$ ), simplificando para ( $\neg A$ ). A nova fórmula é:  $\Phi' = (A \vee B) \wedge (\neg A) \wedge (A \vee \neg B \vee D)$

Passo 3 (PU): Agora a cláusula ( $\neg A$ ) é unitária. Para torná-la verdadeira, assumimos  $A = \text{Falso}$ .

# Verificação eficiente de modelos proposicionais

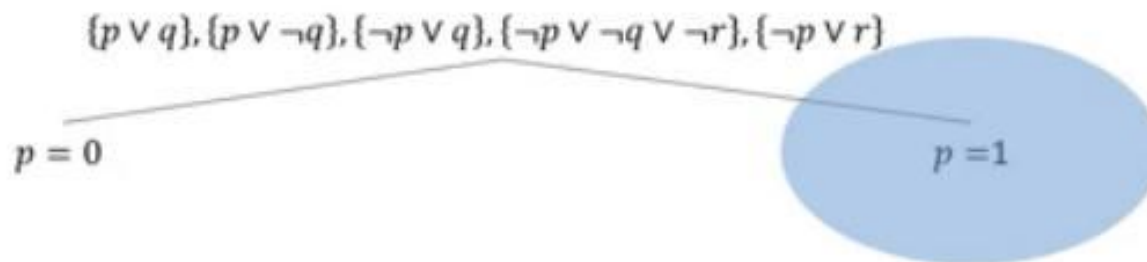
Passo 4 (Simplificação): Substituindo  $A = \text{Falso}$  na fórmula: A primeira cláusula  $(A \vee B)$  vira  $(\text{Falso} \vee B)$ , que simplifica para  $(B)$ . A quarta cláusula  $(A \vee \neg B \vee D)$  vira  $(\text{Falso} \vee \neg B \vee D)$ , que simplifica para  $(\neg B \vee D)$ . A nova fórmula é:  $\Phi'' = (B) \wedge (\neg B \vee D)$

Passo 5 (PU): A cláusula  $(B)$  é unitária. Assumimos  $B = \text{Verdadeiro}$ .

Passo 6 (Simplificação): Substituindo  $B = \text{Verdadeiro}$  na última cláusula  $(\neg B \vee D)$ , temos  $(\text{Falso} \vee D)$ , que simplifica para  $(D)$ .

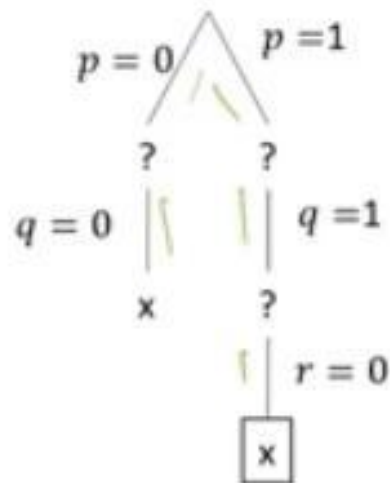
Passo 7 (PU): A cláusula restante  $(D)$  força a atribuição  $D = \text{Verdadeiro}$ . Resultado 1: A fórmula é Satisfatível. A atribuição que a resolve é:  $\{A = \text{Falso}, B = \text{Verdadeiro}, C = \text{Falso}, D = \text{Verdadeiro}\}$

# Verificação eficiente de modelos proposicionais



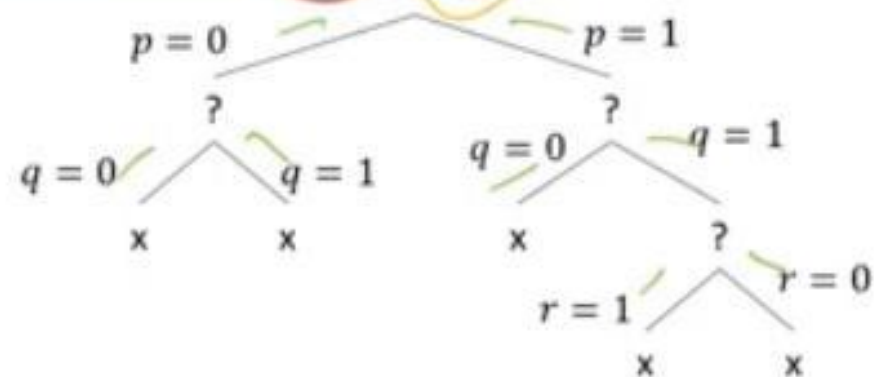
| Original                         | Simplificado         |
|----------------------------------|----------------------|
| $p \vee q$                       | —                    |
| $p \vee \neg q$                  | —                    |
| $\neg p \vee q$                  | $q$                  |
| $\neg p \vee \neg q \vee \neg r$ | $\neg q \vee \neg r$ |
| $\neg p \vee r$                  | $r$                  |

# Verificação eficiente de modelos proposicionais



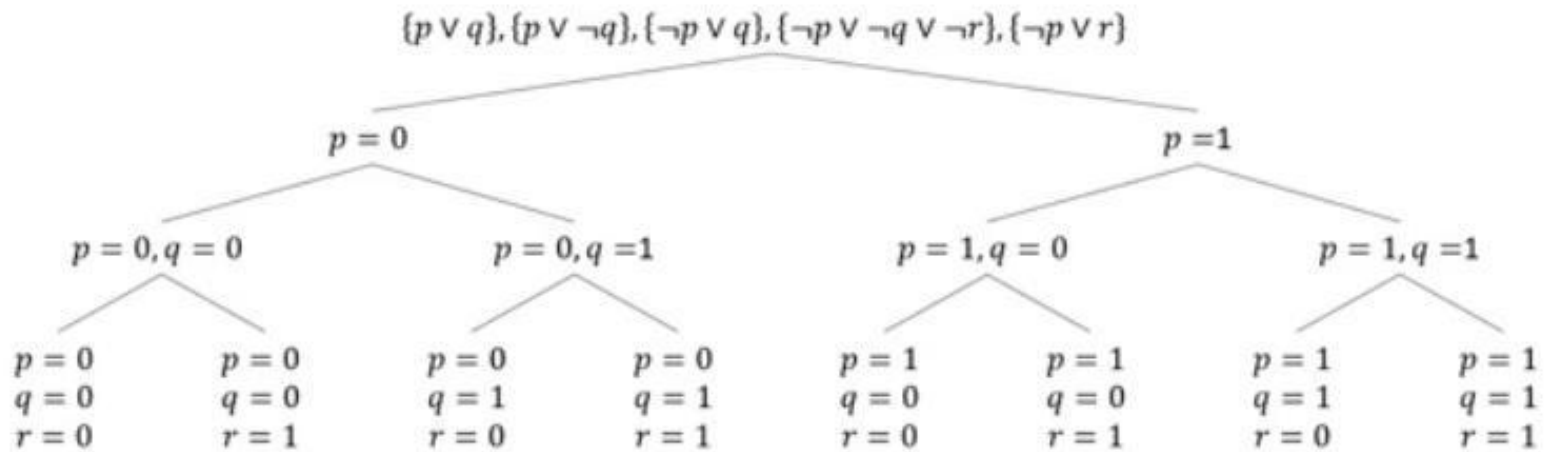
| Original                         | Simplificado |
|----------------------------------|--------------|
| $p \vee q$                       | -            |
| $p \vee \neg q$                  | -            |
| $\neg p \vee q$                  | -            |
| $\neg p \vee \neg q \vee \neg r$ | -            |
| $\neg p \vee r$                  | FALSE        |

$R = T$



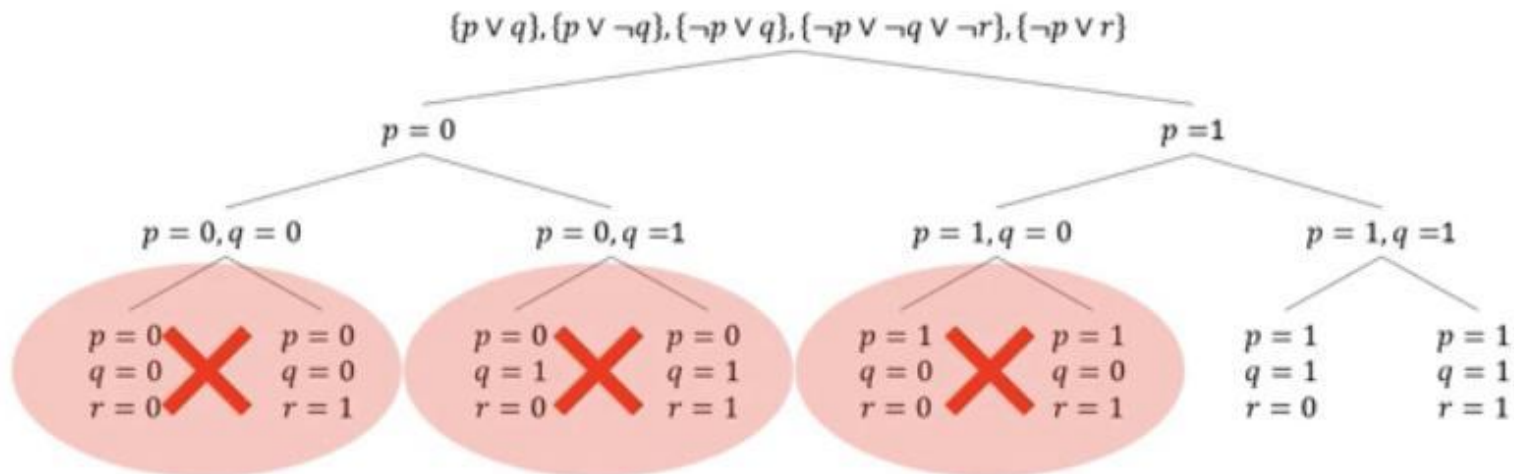
# Verificação eficiente de modelos proposicionais

- Verificação de modelos sem back-tracking



# Verificação eficiente de modelos proposicionais

- Verificação de modelos com back-tracking



Deixou de fazer 6 testes (6 de 8), não precisa processar tudo

# Verificação eficiente de modelos proposicionais

- Verificação de modelos com back-tracking

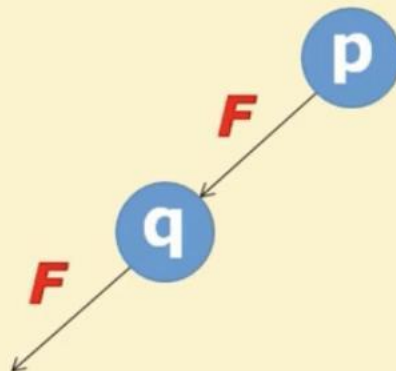
$\{\bar{p}qr, prs, pr\bar{s}, p\bar{r}s, \bar{p}\bar{r}s, \bar{q}rs, \bar{p}q\bar{r}, \bar{p}\bar{q}\bar{r}\}$

# Verificação eficiente de modelos proposicionais

- Verificação de modelos com back-tracking

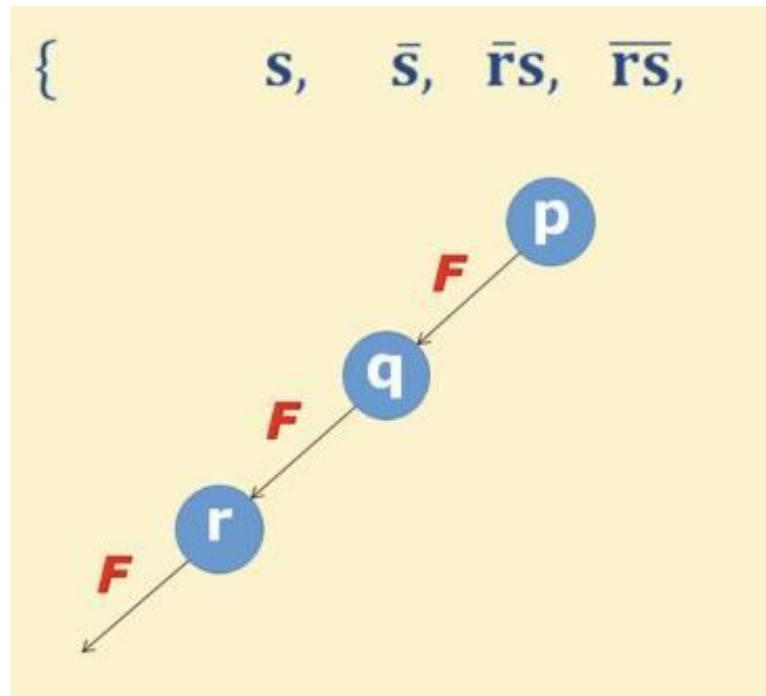
$\{\bar{p}qr, prs, pr\bar{s}, p\bar{r}s, p\bar{r}\bar{s}, \bar{q}rs, \bar{p}q\bar{r}, \bar{p}q\bar{r}\}$

{         $rs, r\bar{s}, \bar{r}s, \bar{r}\bar{s}, \boxed{\bar{q}rs}$         }



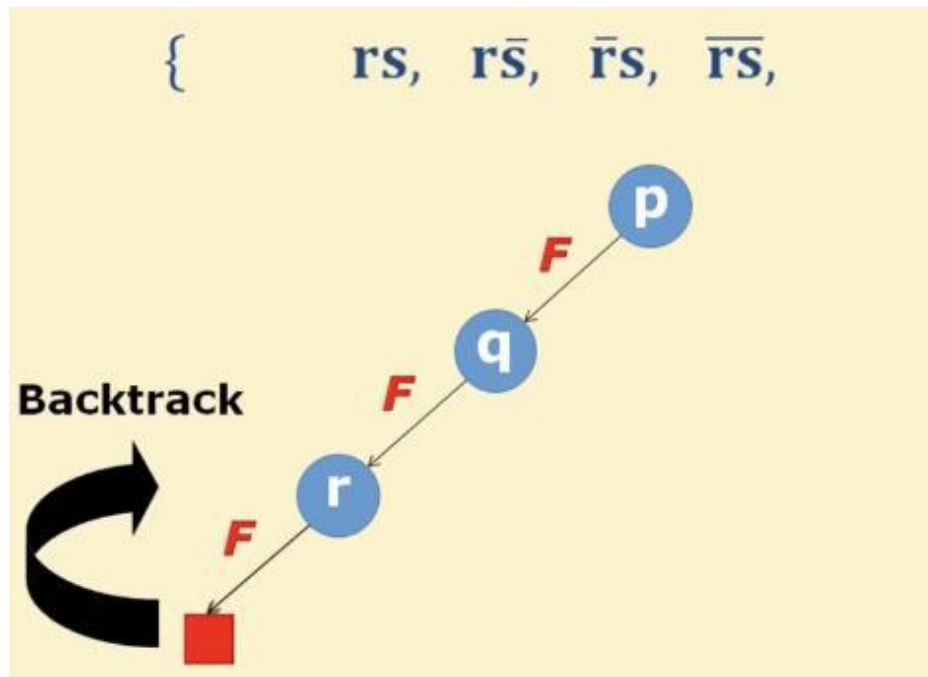
# Verificação eficiente de modelos proposicionais

- Verificação de modelos com back-tracking



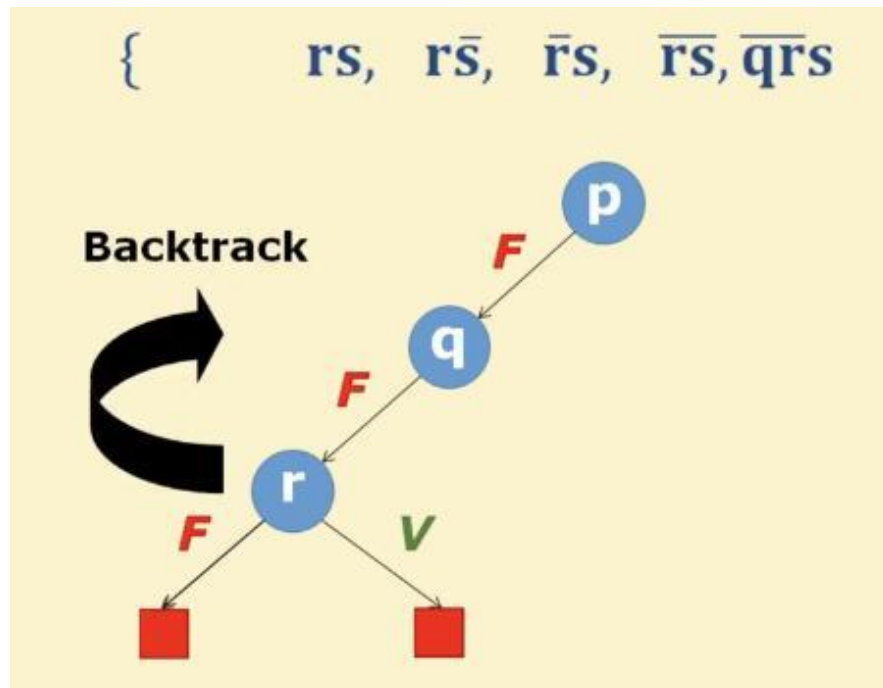
# Verificação eficiente de modelos proposicionais

- Verificação de modelos com back-tracking



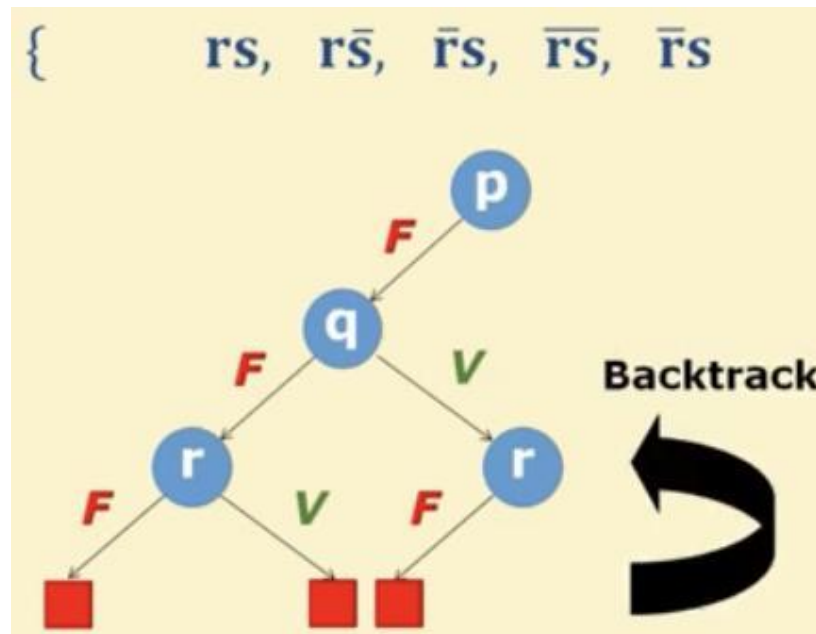
# Verificação eficiente de modelos proposicionais

- Verificação de modelos com back-tracking



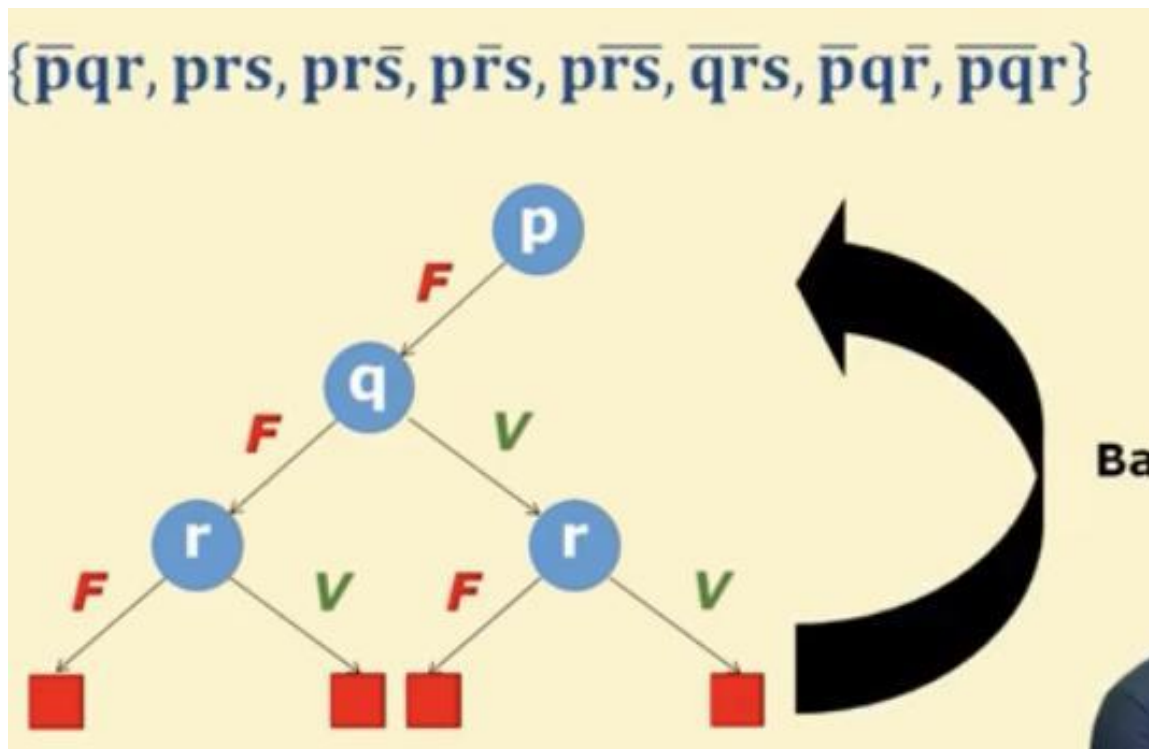
# Verificação eficiente de modelos proposicionais

- Verificação de modelos com back-tracking



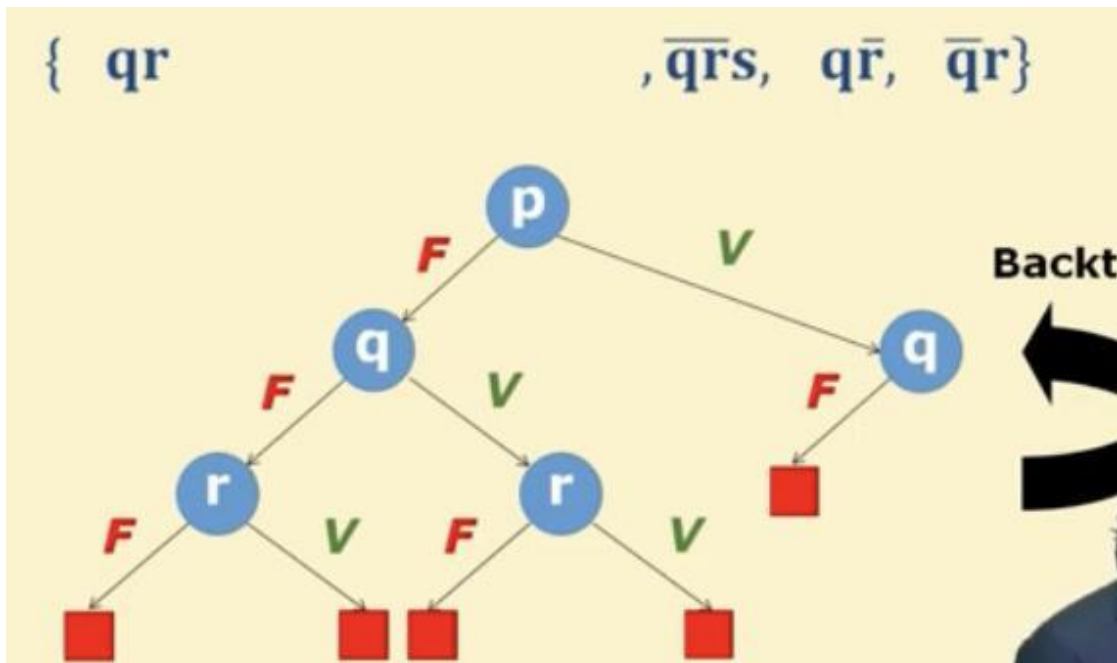
# Verificação eficiente de modelos proposicionais

- Verificação de modelos com back-tracking



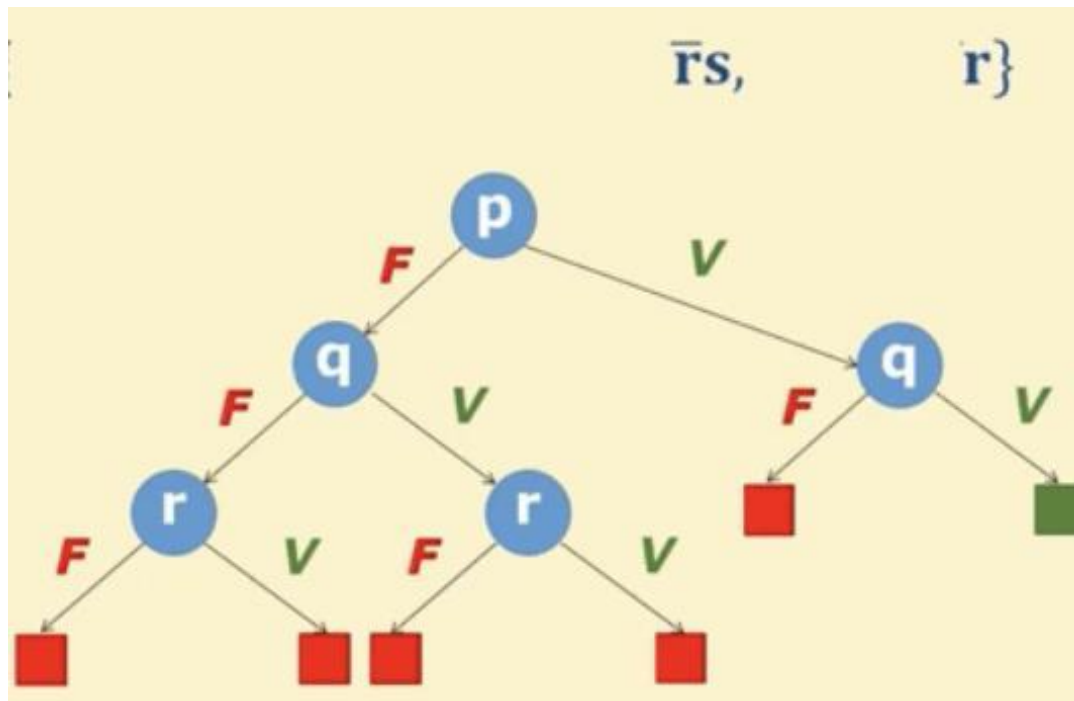
# Verificação eficiente de modelos proposicionais

- Verificação de modelos com back-tracking



# Verificação eficiente de modelos proposicionais

- Verificação de modelos com back-tracking



# Verificação eficiente de modelos proposicionais

- Verificação de modelos com back-tracking

