Forma Normal Clausal e SAT Solvers

Lógica Propositional Clássica

Marcelo Finger

Departamento de Ciência da Computação
Instituto de Matemática e Estatística
Universidade de São Paulo

2022

Tópicos

1. Forma Normal Conjuntiva ou Clausal
2. SAT

Próximo Tópico

1. Forma Normal Conjuntiva ou Clausal

2. SAT

Forma Normal Conjuntiva (FNC/CNF)

Literal: um átomo ou sua negação. $L ::= p|\neg p$

Forma Normal Conjuntiva (FNC/CNF)

LITERAL: um átomo ou sua negação. $L ::= p|\neg p$

CLÁUSULA: disjunção de literais. $D ::= L|L \vee D$

Forma Normal Conjuntiva (FNC/CNF)

Literal:	um átomo ou sua negação.	$L ::= p \mid \neg p$
Cláusula:	disjunção de literais.	$D ::= L \mid L \vee D$
CNF:	conjunção de cláusulas.	$C ::= D \mid D \wedge C$

Forma Normal Conjuntiva (FNC/CNF)

Literal:	um átomo ou sua negação.	$L ::= p \mid \neg p$
Cláusula:	disjunção de literais.	$D ::= L \mid L \vee D$
CNF:	conjunção de cláusulas.	$C ::= D \mid D \wedge C$

Exemplos:

- $(\neg q \vee p \vee r) \wedge (\neg p \vee r) \wedge q$

Forma Normal Conjuntiva (FNC/CNF)

Literal:	um átomo ou sua negação.	$L ::= p \mid \neg p$
Cláusula:	disjunção de literais.	$D ::= L \mid L \vee D$
CNF:	conjunção de cláusulas.	$C ::= D \mid D \wedge C$

Exemplos:

- $(\neg q \vee p \vee r) \wedge (\neg p \vee r) \wedge q$
- $(p \vee r) \wedge (\neg p \vee r) \wedge (p \vee \neg r)$

Forma Normal Conjuntiva (FNC/CNF)

Literal: um átomo ou sua negação. $L ::= p|\neg p$

Cláusula: disjunção de literais. $D ::= L|L \vee D$

CNF: conjunção de cláusulas. $C ::= D|D \wedge C$

Exemplos:

- $(\neg q \vee p \vee r) \wedge (\neg p \vee r) \wedge q$
- $(p \vee r) \wedge (\neg p \vee r) \wedge (p \vee \neg r)$
- $(\neg(p \vee q) \vee r) \wedge (q \vee r)$

Forma Normal Conjuntiva (FNC/CNF)

Literal: um átomo ou sua negação. $L ::= p|\neg p$

Cláusula: disjunção de literais. $D ::= L|L \vee D$

CNF: conjunção de cláusulas. $C ::= D|D \wedge C$

Exemplos:

- $(\neg q \vee p \vee r) \wedge (\neg p \vee r) \wedge q$
- $(p \vee r) \wedge (\neg p \vee r) \wedge (p \vee \neg r)$
- $\cancel{(\neg(p \vee q) \vee r) \wedge (q \vee r)}$

Por que CNF?

1. Uma fórmula em CNF é válida sse todas as suas cláusulas são válidas.

Por que CNF?

1. Uma fórmula em CNF é válida sse todas as suas cláusulas são válidas.
2. Uma cláusula é válida sse ela contém um átomo e sua negação.

Por que CNF?

1. Uma fórmula em CNF é válida sse todas as suas cláusulas são válidas.
2. Uma cláusula é válida sse ela contém um átomo e sua negação.

(φ é satisfatível sse $\neg\varphi$ não é válida)

Transformar em CNF - Tabela Verdade

p	q	$\neg p$	$\neg q$	$p \to \neg q$	$q \vee \neg p$	$(p \to \neg q) \to (q \vee \neg p)$
T	T	F	F	F	T	T
T	F	F	T	T	F	F
F	T	T	F	T	T	T
F	F	T	T	T	T	T

Transformar em CNF - Tabela Verdade

p	q	$\neg p$	$\neg q$	$p \to \neg q$	$q \vee \neg p$	$(p \to \neg q) \to (q \vee \neg p)$
T	T	F	F	F	T	T
T	F	F	T	T	F	F
F	T	T	F	T	T	T
F	F	T	T	T	T	T

$v((p \to \neg q) \to (q \vee \neg p) = F$ apenas quando $v(p) = T$ e $v(q) = F$

Transformar em CNF - Tabela Verdade

p	q	$\neg p$	$\neg q$	$p \to \neg q$	$q \vee \neg p$	$(p \to \neg q) \to (q \vee \neg p)$
T	T	F	F	F	T	T
T	F	F	T	T	F	F
F	T	T	F	T	T	T
F	F	T	T	T	T	T

$v((p \to \neg q) \to (q \vee \neg p) = F$ apenas quando $v(p) = T$ e $v(q) = F$

$(p \to \neg q) \to (q \vee \neg p) \equiv \neg p \vee q$

Exemplo

$(p \vee \neg q) \to r$

p	q	r	$\neg q$	$p \vee \neg q$	$(p \vee \neg q) \to r$
F	F	F	T	T	F
F	F	T	T	T	T
F	T	F	F	F	T
F	T	T	F	F	T
T	F	F	T	T	F
T	F	T	T	T	T
T	T	F	F	T	F
T	T	T	F	T	T

Exemplo

$(p \vee \neg q) \rightarrow r$

p	q	r	$\neg q$	$p \vee \neg q$	$(p \vee \neg q) \rightarrow r$
F	F	F	T	T	F
F	F	T	T	T	T
F	T	F	F	T	F
F	T	T	F	F	T
T	F	F	T	T	F
T	F	T	T	T	T
T	T	F	F	T	F
T	T	T	F	T	T

Transformar em CNF - usando equivalências

1. Eliminar $\to$: $\varphi \to \psi \equiv \neg\varphi \vee \psi$

Transformar em CNF - usando equivalências

1. Eliminar $\to$: $\varphi \to \psi \equiv \neg\varphi \vee \psi$
2. Mover $\neg$ para dentro: $\neg(\varphi \wedge \psi) \equiv \neg\varphi \vee \neg\psi$ e $\neg(\varphi \vee \psi) \equiv \neg\varphi \wedge \neg\psi$

Transformar em CNF - usando equivalências

1. Eliminar $\to$: $\varphi \to \psi \equiv \neg\varphi \vee \psi$
2. Mover $\neg$ para dentro: $\neg(\varphi \wedge \psi) \equiv \neg\varphi \vee \neg\psi$ e $\neg(\varphi \vee \psi) \equiv \neg\varphi \wedge \neg\psi$
3. Eliminar dupla negação: $\neg\neg\varphi \equiv \varphi$

Transformar em CNF - usando equivalências

1. Eliminar $\to$: $\varphi \to \psi \equiv \neg\varphi \vee \psi$
2. Mover $\neg$ para dentro: $\neg(\varphi \wedge \psi) \equiv \neg\varphi \vee \neg\psi$ e $\neg(\varphi \vee \psi) \equiv \neg\varphi \wedge \neg\psi$
3. Eliminar dupla negação: $\neg\neg\varphi \equiv \varphi$
4. Distribuir $\vee$ e $\wedge$: $\varphi \vee (\psi \wedge \chi) \equiv (\varphi \vee \psi) \wedge (\varphi \vee \chi)$

Transformar em CNF - usando equivalências

1. Eliminar $\to$: $\varphi \to \psi \equiv \neg\varphi \vee \psi$
2. Mover $\neg$ para dentro: $\neg(\varphi \wedge \psi) \equiv \neg\varphi \vee \neg\psi$ e $\neg(\varphi \vee \psi) \equiv \neg\varphi \wedge \neg\psi$
3. Eliminar dupla negação: $\neg\neg\varphi \equiv \varphi$
4. ~~Distribuir $\vee$ e $\wedge$: $\varphi \vee (\psi \wedge \chi) \equiv (\varphi \vee \psi) \wedge (\varphi \vee \chi)$~~

Transformar em CNF - usando equivalências

1. Eliminar $\rightarrow$: $\varphi \rightarrow \psi \equiv \neg\varphi \vee \psi$
2. Mover $\neg$ para dentro: $\neg(\varphi \wedge \psi) \equiv \neg\varphi \vee \neg\psi$ e $\neg(\varphi \vee \psi) \equiv \neg\varphi \wedge \neg\psi$
3. Eliminar dupla negação: $\neg\neg\varphi \equiv \varphi$
4. ~~Distribuir $\vee$ e $\wedge$: $\varphi \vee (\psi \wedge \chi) \equiv (\varphi \vee \psi) \wedge (\varphi \vee \chi)$~~
5. Substituir conjunções não-clausais por átomos novos. Seja p_{novo} tal que $p_{novo} \leftrightarrow (\psi \wedge \theta)$. Substituir $\varphi \vee (\psi \wedge \theta)$ por:
   - $\varphi \vee p_{novo}$
   - $\neg p_{novo} \vee \psi$ $\qquad (p_{novo} \rightarrow \psi)$
   - $\neg p_{novo} \vee \theta$ $\qquad (p_{novo} \rightarrow \theta)$
   - $\neg\psi \vee \neg\theta \vee p_{novo}$ $\qquad (\psi \wedge \theta \rightarrow p_{novo})$

Próximo Tópico

1. Forma Normal Conjuntiva ou Clausal

2. SAT

O problema SAT

"Dada uma fórmula, decidir se ela é satisfatível."

O problema SAT

"Dada uma fórmula, decidir se ela é satisfatível."

- SAT é NP-completo [Cook 1971]

O problema SAT

"Dada uma fórmula, decidir se ela é satisfatível."

- SAT é NP-completo [Cook 1971]
- Competição desde 2002:
  http://www.satcompetition.org/

NaiveSAT

Entrada: φ, em CNF
Saída: v, se $v(\varphi) = T$; "não", caso contrário

```
Para toda valoração v sobre os átomos de φ faça:
     se v(C) = T então devolva v
Devolva ``não''
```

Resolução Clausal

Única regra:

$$\frac{\varphi \vee p \qquad \psi \vee \neg p}{\varphi \vee \psi}$$

Robinson, J. Alan (1965). "A Machine-Oriented Logic Based on the Resolution Principle".

Resolução Clausal

Única regra:

$$\frac{\varphi \lor p \qquad \psi \lor \neg p}{\varphi \lor \psi}$$

Robinson, J. Alan (1965). "A Machine-Oriented Logic Based on the Resolution Principle".

Completa para *refutação*!

Refutação por resolução

Raciocínio por contradição:

1. Para provar $\varphi_1, \ldots, \varphi_n \vdash \psi$, transforme $\chi = \varphi_1 \land \ldots \land \varphi_n \land \neg\psi$ em CNF.

Refutação por resolução

Raciocínio por contradição:

1. Para provar $\varphi_1, \ldots, \varphi_n \vdash \psi$, transforme $\chi = \varphi_1 \wedge \ldots \wedge \varphi_n \wedge \neg\psi$ em CNF.
2. Seja C o conjunto de cláusulas obtido.

Refutação por resolução

Raciocínio por contradição:

1. Para provar $\varphi_1, \ldots, \varphi_n \vdash \psi$, transforme $\chi = \varphi_1 \wedge \ldots \wedge \varphi_n \wedge \neg\psi$ em CNF.
2. Seja C o conjunto de cláusulas obtido.
3. Aplique resolução ao conjunto quantas vezes for possível.

Refutação por resolução

Raciocínio por contradição:

1. Para provar $\varphi_1, \ldots, \varphi_n \vdash \psi$, transforme $\chi = \varphi_1 \wedge \ldots \wedge \varphi_n \wedge \neg\psi$ em CNF.
2. Seja C o conjunto de cláusulas obtido.
3. Aplique resolução ao conjunto quantas vezes for possível.
4. Se em algum momento gerar uma cláusula vazia, devolva T (χ não é SAT).

Refutação por resolução

Raciocínio por contradição:

1. Para provar $\varphi_1, \ldots, \varphi_n \vdash \psi$, transforme $\chi = \varphi_1 \wedge \ldots \wedge \varphi_n \wedge \neg\psi$ em CNF.
2. Seja C o conjunto de cláusulas obtido.
3. Aplique resolução ao conjunto quantas vezes for possível.
4. Se em algum momento gerar uma cláusula vazia, devolva T (χ não é SAT).
5. Senão, devolva F (χ é SAT).

Refutação por resolução

Raciocínio por contradição:

1. Para provar $\varphi_1, \ldots, \varphi_n \vdash \psi$, transforme $\chi = \varphi_1 \wedge \ldots \wedge \varphi_n \wedge \neg\psi$ em CNF.
2. Seja C o conjunto de cláusulas obtido.
3. Aplique resolução ao conjunto quantas vezes for possível.
4. Se em algum momento gerar uma cláusula vazia, devolva T (χ não é SAT).
5. Senão, devolva F (χ é SAT).

Refutação por resolução

Raciocínio por contradição:

1. Para provar $\varphi_1, \ldots, \varphi_n \vdash \psi$, transforme $\chi = \varphi_1 \wedge \ldots \wedge \varphi_n \wedge \neg\psi$ em CNF.
2. Seja C o conjunto de cláusulas obtido.
3. Aplique resolução ao conjunto quantas vezes for possível.
4. Se em algum momento gerar uma cláusula vazia, devolva T (χ não é SAT).
5. Senão, devolva F (χ é SAT).

Exemplo:

- $p \to q, r \to s \vdash (p \vee r) \to (q \vee s)$

Exemplo passo a passo

$$p \to q, r \to s \vdash (p \vee r) \to (q \vee s)$$

1. Transformar $\chi = p \to q \wedge r \to s \wedge \neg((p \vee r) \to (q \vee s))$ em CNF:

$p \to q \wedge r \to s \wedge \neg((p \vee r) \to (q \vee s)) \equiv$
(eliminando implicações)
$(\neg p \vee q) \wedge (\neg r \vee s) \wedge \neg(\neg(p \vee r) \vee (q \vee s)) \equiv$
(movendo negações para dentro)
$(\neg p \vee q) \wedge (\neg r \vee s) \wedge (\neg\neg(p \vee r) \wedge \neg(q \vee s)) \equiv$
$(\neg p \vee q) \wedge (\neg r \vee s) \wedge \neg\neg(p \vee r) \wedge (\neg q \wedge \neg s) \equiv$
(eliminando a dupla negação)
$(\neg p \vee q) \wedge (\neg r \vee s) \wedge (p \vee r) \wedge \neg q \wedge \neg s$

Exemplo passo a passo

2. $C = \{\neg p \vee q, \neg r \vee s, p \vee r, \neg q, \neg s\}$

3. Aplicar resolução:

1. $\neg p \vee q$
2. $\neg r \vee s$
3. $p \vee r$
4. $\neg q$
5. $\neg s$
6. $\neg p$ (1,4)
7. r (3,6)
8. $\neg r$ (2,5)
9. [] (7,8)

Exemplo passo a passo

4. Como geramos a cláusula vazia, a fórmula χ não é satisfatível.

Isso quer dizer que o sequente

$$p \to q, r \to s \vdash (p \vee r) \to (q \vee s)$$

é válido.

O Algoritmo DPLL

Davis & Putnam, 1960; Davis, Longemann & Loveland, 1962

- Evita construir valorações completas.

O Algoritmo DPLL

Davis & Putnam, 1960; Davis, Longemann & Loveland, 1962

- Evita construir valorações completas.
- *Símbolo puro:* Aparece só positivo ou só negativo.

O Algoritmo DPLL

Davis & Putnam, 1960; Davis, Longemann & Loveland, 1962

- Evita construir valorações completas.
- *Símbolo puro:* Aparece só positivo ou só negativo.
- *Propagação Unitária:* Preferência por cláusulas com um só literal.

O Algoritmo DPLL

1. Começa com modelo vazio.

O Algoritmo DPLL

1. Começa com modelo vazio.
2. Se alguma cláusula é F, devolve F.

O Algoritmo DPLL

1. Começa com modelo vazio.
2. Se alguma cláusula é F, devolve F.
3. Se todas as cláusulas são T, devolve T.

O Algoritmo DPLL

1. Começa com modelo vazio.
2. Se alguma cláusula é F, devolve F.
3. Se todas as cláusulas são T, devolve T.
4. Remove todas as cláusulas com símbolo puro.

O Algoritmo DPLL

1. Começa com modelo vazio.
2. Se alguma cláusula é F, devolve F.
3. Se todas as cláusulas são T, devolve T.
4. Remove todas as cláusulas com símbolo puro.
5. Quando tem cláusulas unitárias, acrescenta ao modelo.

O Algoritmo DPLL

1. Começa com modelo vazio.
2. Se alguma cláusula é F, devolve F.
3. Se todas as cláusulas são T, devolve T.
4. Remove todas as cláusulas com símbolo puro.
5. Quando tem cláusulas unitárias, acrescenta ao modelo.
6. Se não, **escolhe** literal l e acrescenta.

O Algoritmo DPLL

1. Começa com modelo vazio.
2. Se alguma cláusula é F, devolve F.
3. Se todas as cláusulas são T, devolve T.
4. Remove todas as cláusulas com símbolo puro.
5. Quando tem cláusulas unitárias, acrescenta ao modelo.
6. Se não, **escolhe** literal l e acrescenta.
7. Apaga as cláusulas contendo l.

O Algoritmo DPLL

1. Começa com modelo vazio.
2. Se alguma cláusula é F, devolve F.
3. Se todas as cláusulas são T, devolve T.
4. Remove todas as cláusulas com símbolo puro.
5. Quando tem cláusulas unitárias, acrescenta ao modelo.
6. Se não, **escolhe** literal l e acrescenta.
7. Apaga as cláusulas contendo l.
8. Apaga o oposto de l das outras cláusulas

O Algoritmo DPLL

1. Começa com modelo vazio.
2. Se alguma cláusula é F, devolve F.
3. Se todas as cláusulas são T, devolve T.
4. Remove todas as cláusulas com símbolo puro.
5. Quando tem cláusulas unitárias, acrescenta ao modelo.
6. Se não, **escolhe** literal l e acrescenta.
7. Apaga as cláusulas contendo l.
8. Apaga o oposto de l das outras cláusulas
9. Repete até achar contradição, neste caso, refaz a escolha, ou não ter mais o que aplicar.

O Algoritmo DPLL

1. Começa com modelo vazio.
2. Se alguma cláusula é F, devolve F.
3. Se todas as cláusulas são T, devolve T.
4. Remove todas as cláusulas com símbolo puro.
5. Quando tem cláusulas unitárias, acrescenta ao modelo.
6. Se não, **escolhe** literal l e acrescenta.
7. Apaga as cláusulas contendo l.
8. Apaga o oposto de l das outras cláusulas
9. Repete até achar contradição, neste caso, refaz a escolha, ou não ter mais o que aplicar.
10. Se o modelo tem contradição, a fórmula não é SAT. Se não, é uma valoração.