命题逻辑（四）

如何表示推理问题？

- 若我们想组织一个聚会，邀请客人有以下的规则：
  - 1、如果两人是夫妻，则我们要么同时邀请两个人要么都不邀请。Alice和Bob是夫妻，Cecile和David是夫妻。
  - 2、如果我们邀请了Alice那么我们也需要邀请Cecile。
  - 3、David和Eva不会同时出席，所以不能同时邀请。
  - 4、我们想同时邀请Bob和Fred。
- 问：我们如何确定一个邀请名单？

如何表示推理问题？

- 命题变元：Alice、Bob、Cecile、David、Eva、Fred；
- 命题逻辑约束：
  - 1、邀请夫妻：$\text{Alice} \leftrightarrow \text{Bob}, \text{Cecile} \leftrightarrow \text{David}$
  - 2、如果Alice则Cecile：$\text{Alice} \rightarrow \text{Cecile}$
  - 3、要么David要么Eva：$\neg(\text{Eva} \leftrightarrow \text{David})$
  - 4、邀请Bob和Fred：$\text{Bob} \wedge \text{Fred}$

如何表示推理问题？

- 写成命题逻辑公式：
  - $(\text{Alice} \leftrightarrow \text{Bob}) \wedge (\text{Cecile} \leftrightarrow \text{David}) \wedge (\text{Alice} \rightarrow \text{Cecile}) \wedge \neg(\text{Eva} \leftrightarrow \text{David}) \wedge \text{Bob} \wedge \text{Fred}$
- 符合规则的邀请名单，即使得上述公式的为真的一组赋值
  - 例如，$\text{Alice} = \text{Bob} = \text{Cecile} = \text{David} = \text{Fred} = \text{T}, \text{Eva} = \text{F}$

可满足性问题

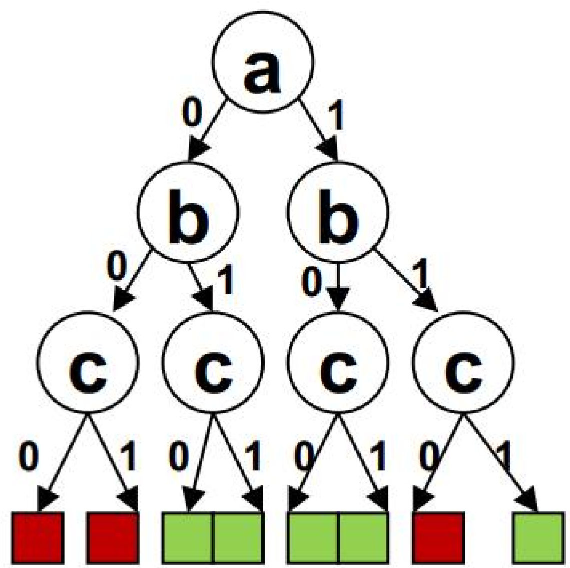
- 给定一个命题公式A，问是否存在一个赋值v，使得$v \vDash A$?
  - 此赋值v也被称为问题的一个解

可满足性问题

- 给定一个命题公式A，问是否存在一个赋值v，使得$v \vDash A$?
  - 此赋值v也被称为问题的一个解

$$F = (a \vee b) \wedge (\neg a \vee \neg b \vee c)$$

- 对n个变量的问题，一共有2^n组可能的赋值

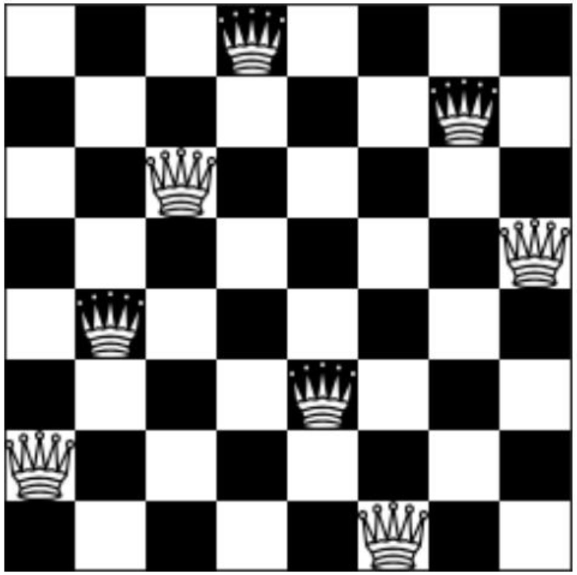


可满足性问题

- 给定一个命题公式A，问是否存在一个赋值v，使得$v \vDash A$？
  - 此赋值v也被称为问题的一个解

- 命题逻辑公式的可满足性问题（也称布尔可满足性问题，或SAT问题）
  - 第一个被证明的NP完全问题（NP-Complete, NPC）（它是NP问题且所有NP问题可以多项式时间归约到它）；
  - 非确定性算法：将问题分解为猜测和验证两个部分；
  - 验证一个赋值是公式的一个解很容易（多项式时间，即NP）；
  - 找到一个解很困难；
  - P ⊆ NP✓ P=NP？（七个千禧年难题）

n-皇后问题



n-皇后问题

x_{11}	x_{12}	x_{13}	x_{14}	x_{15}	x_{16}	x_{17}	x_{18}
x_{21}	x_{22}	x_{23}	x_{24}	x_{25}	x_{26}	x_{27}	x_{28}
x_{31}	x_{32}	x_{33}	x_{34}	x_{35}	x_{36}	x_{37}	x_{38}
x_{41}	x_{42}	x_{43}	x_{44}	x_{45}	x_{46}	x_{47}	x_{48}
x_{51}	x_{52}	x_{53}	x_{54}	x_{55}	x_{56}	x_{57}	x_{58}
x_{61}	x_{62}	x_{63}	x_{64}	x_{65}	x_{66}	x_{67}	x_{68}
x_{71}	x_{72}	x_{73}	x_{74}	x_{75}	x_{76}	x_{77}	x_{78}
x_{81}	x_{82}	x_{83}	x_{84}	x_{85}	x_{86}	x_{87}	x_{88}

n-皇后问题

$x_{ij} = T$ 表示 (i,j) 处有皇后

x_{11}	x_{12}	x_{13}	x_{14}	x_{15}	x_{16}	x_{17}	x_{18}
x_{21}	x_{22}	x_{23}	x_{24}	x_{25}	x_{26}	x_{27}	x_{28}
x_{31}	x_{32}	x_{33}	x_{34}	x_{35}	x_{36}	x_{37}	x_{38}
x_{41}	x_{42}	x_{43}	x_{44}	x_{45}	x_{46}	x_{47}	x_{48}
x_{51}	x_{52}	x_{53}	x_{54}	x_{55}	x_{56}	x_{57}	x_{58}
x_{61}	x_{62}	x_{63}	x_{64}	x_{65}	x_{66}	x_{67}	x_{68}
x_{71}	x_{72}	x_{73}	x_{74}	x_{75}	x_{76}	x_{77}	x_{78}
x_{81}	x_{82}	x_{83}	x_{84}	x_{85}	x_{86}	x_{87}	x_{88}

n-皇后问题

$x_{ij} = T$ 表示 (i,j) 处有皇后

x_{11}	x_{12}	x_{13}	x_{14}	x_{15}	x_{16}	x_{17}	x_{18}
x_{21}	x_{22}	x_{23}	x_{24}	x_{25}	x_{26}	x_{27}	x_{28}
x_{31}	x_{32}	x_{33}	x_{34}	x_{35}	x_{36}	x_{37}	x_{38}
x_{41}	x_{42}	x_{43}	x_{44}	x_{45}	x_{46}	x_{47}	x_{48}
x_{51}	x_{52}	x_{53}	x_{54}	x_{55}	x_{56}	x_{57}	x_{58}
x_{61}	x_{62}	x_{63}	x_{64}	x_{65}	x_{66}	x_{67}	x_{68}
x_{71}	x_{72}	x_{73}	x_{74}	x_{75}	x_{76}	x_{77}	x_{78}
x_{81}	x_{82}	x_{83}	x_{84}	x_{85}	x_{86}	x_{87}	x_{88}

不同行:

第 i 行只有一个皇后

n-皇后问题

$x_{ij} = T$ 表示 (i,j) 处有皇后

x_{11}	x_{12}	x_{13}	x_{14}	x_{15}	x_{16}	x_{17}	x_{18}
x_{21}	x_{22}	x_{23}	x_{24}	x_{25}	x_{26}	x_{27}	x_{28}
x_{31}	x_{32}	x_{33}	x_{34}	x_{35}	x_{36}	x_{37}	x_{38}
x_{41}	x_{42}	x_{43}	x_{44}	x_{45}	x_{46}	x_{47}	x_{48}
x_{51}	x_{52}	x_{53}	x_{54}	x_{55}	x_{56}	x_{57}	x_{58}
x_{61}	x_{62}	x_{63}	x_{64}	x_{65}	x_{66}	x_{67}	x_{68}
x_{71}	x_{72}	x_{73}	x_{74}	x_{75}	x_{76}	x_{77}	x_{78}
x_{81}	x_{82}	x_{83}	x_{84}	x_{85}	x_{86}	x_{87}	x_{88}

不同行:

第 i 行只有一个皇后

$\Rightarrow x_{i1}, .., x_{i8}$中只有一个为真

n-皇后问题

$x_{ij} = T$ 表示 (i,j) 处有皇后

x_{11}	x_{12}	x_{13}	x_{14}	x_{15}	x_{16}	x_{17}	x_{18}
x_{21}	x_{22}	x_{23}	x_{24}	x_{25}	x_{26}	x_{27}	x_{28}
x_{31}	x_{32}	x_{33}	x_{34}	x_{35}	x_{36}	x_{37}	x_{38}
x_{41}	x_{42}	x_{43}	x_{44}	x_{45}	x_{46}	x_{47}	x_{48}
x_{51}	x_{52}	x_{53}	x_{54}	x_{55}	x_{56}	x_{57}	x_{58}
x_{61}	x_{62}	x_{63}	x_{64}	x_{65}	x_{66}	x_{67}	x_{68}
x_{71}	x_{72}	x_{73}	x_{74}	x_{75}	x_{76}	x_{77}	x_{78}
x_{81}	x_{82}	x_{83}	x_{84}	x_{85}	x_{86}	x_{87}	x_{88}

不同行:

第 i 行只有一个皇后

$\Rightarrow x_{i1}, .., x_{i8}$中只有一个为真

$\Rightarrow (x_{i1} \vee x_{i2} \vee ... \vee x_{i8})$

$\wedge (\neg x_{i1} \vee \neg x_{i2}) \wedge ... \wedge (\neg x_{i1} \vee \neg x_{i8})$

$\wedge (\neg x_{i2} \vee \neg x_{i3}) \wedge ... \wedge (\neg x_{i2} \vee \neg x_{i8})$

$\wedge ...$

$\wedge (\neg x_{i7} \vee \neg x_{i8})$

n-皇后问题

$x_{ij} = T$ 表示 (i,j) 处有皇后

不同行:

第 i 行只有一个皇后

$\Rightarrow x_{i1}, .., x_{i8}$中只有一个为真

$\Rightarrow (x_{i1} \vee x_{i2} \vee \ldots \vee x_{i8})$

$\wedge (\neg x_{i1} \vee \neg x_{i2}) \wedge \ldots \wedge (\neg x_{i1} \vee \neg x_{i8})$

$\wedge (\neg x_{i2} \vee \neg x_{i3}) \wedge \ldots \wedge (\neg x_{i2} \vee \neg x_{i8})$

$\wedge \ldots$

$\wedge (\neg x_{i7} \vee \neg x_{i8})$

x_{11}	x_{12}	x_{13}	x_{14}	x_{15}	x_{16}	x_{17}	x_{18}
x_{21}	x_{22}	x_{23}	x_{24}	x_{25}	x_{26}	x_{27}	x_{28}
x_{31}	x_{32}	x_{33}	x_{34}	x_{35}	x_{36}	x_{37}	x_{38}
x_{41}	x_{42}	x_{43}	x_{44}	x_{45}	x_{46}	x_{47}	x_{48}
x_{51}	x_{52}	x_{53}	x_{54}	x_{55}	x_{56}	x_{57}	x_{58}
x_{61}	x_{62}	x_{63}	x_{64}	x_{65}	x_{66}	x_{67}	x_{68}
x_{71}	x_{72}	x_{73}	x_{74}	x_{75}	x_{76}	x_{77}	x_{78}
x_{81}	x_{82}	x_{83}	x_{84}	x_{85}	x_{86}	x_{87}	x_{88}

不同列

不同对角线

拉丁方

- n阶拉丁方：$n \times n$矩阵，每行每列$\{1, ..., \mathrm{n}\}$各仅出现一次

1	2	3
2	3	1
3	1	2

1	3	2
2	1	3
3	2	1

- 引入命题变元$a_{i,j,k}$表示$a_{i,j}$位置是否取值k
- $a_{i,j}$位置仅取一个值：$a_{i,j,1}, a_{i,j,2}, a_{i,j,3}$中一个为真
- 第一行各不相同：$a_{1,1,1}, a_{1,2,1}, a_{1,3,1}$中一个为真，$a_{1,1,2}$, $a_{1,2,2}, a_{1,3,2}$中一个为真，且$a_{1,1,3}, a_{1,2,3}, a_{1,3,3}$中一个为真

拉丁方

- n阶拉丁方：$n \times n$矩阵，每行每列$\{1, \ldots, \mathrm{n}\}$各仅出现一次

1	2	3
2	3	1
3	1	2

1	3	2
2	1	3
3	2	1

- 正交拉丁方问题

(1,1)	(2,3)	(3,2)
(2,2)	(3,1)	(1,3)
(3,3)	(1,2)	(2,1)

- 数独问题
- 其它拉丁方问题

不存在 $\mathrm{n} = 4\mathrm{k} + 2$ 阶的正交拉丁方？（欧拉猜想）

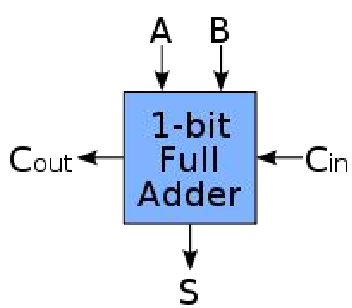
- 四色问题、七桥问题、……
- 0-1整数规划、集合覆盖问题、背包问题、……

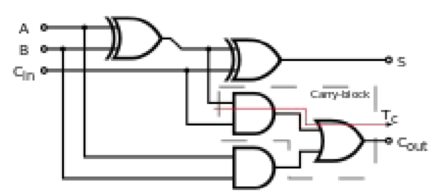
- 任何NP问题都可以在多项式时间规约为SAT问题

Circuit to SAT

- 加法电路的形式化（1-bit）
  - $A + B + C_{in} = C_{out}S \Leftrightarrow$
  - C_{out} = (A and B) or (C_{in} and (A or B))
  - S = A xor B xor C_{in}

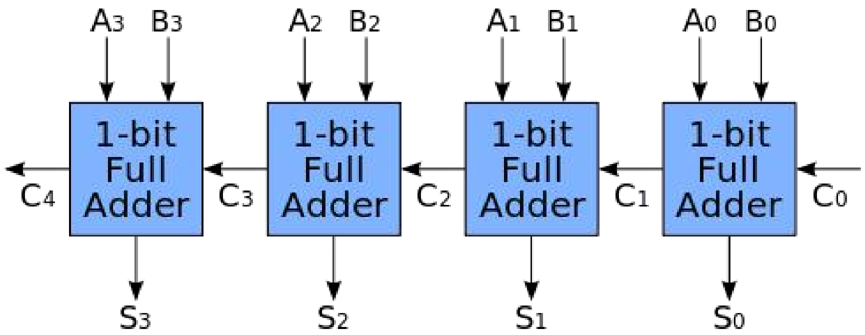
Inputs			Outputs	
A	B	C_{in}	C_{out}	S
0	0	0	0	0
1	0	0	0	1
0	1	0	0	1
1	1	0	1	0
0	0	1	0	1
1	0	1	1	0
0	1	1	1	0
1	1	1	1	1



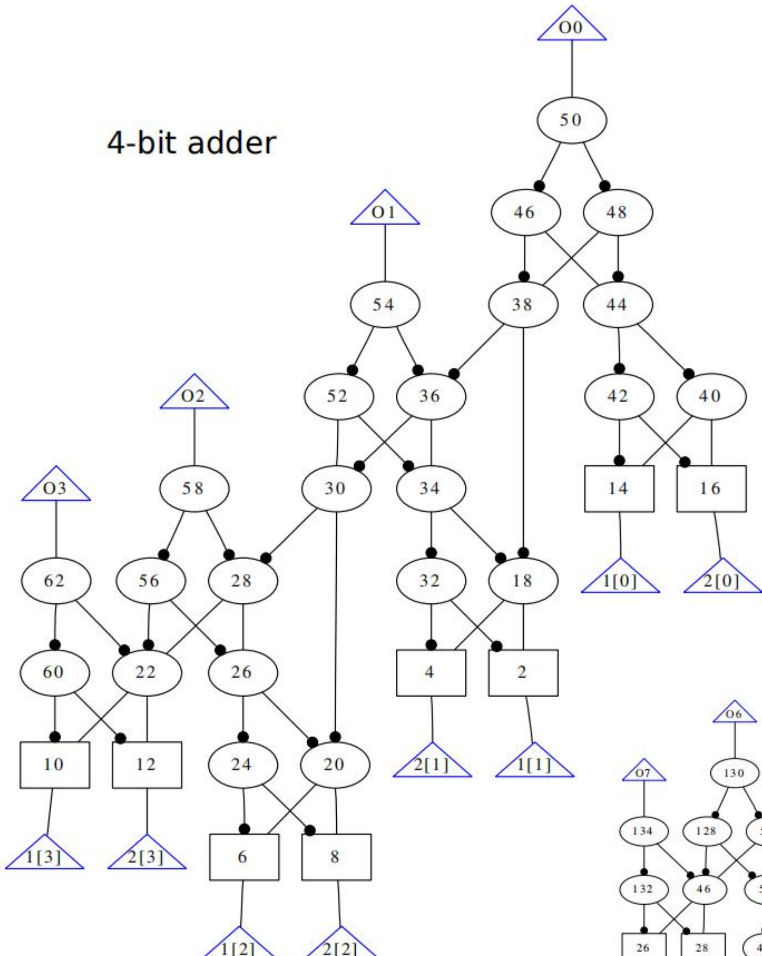


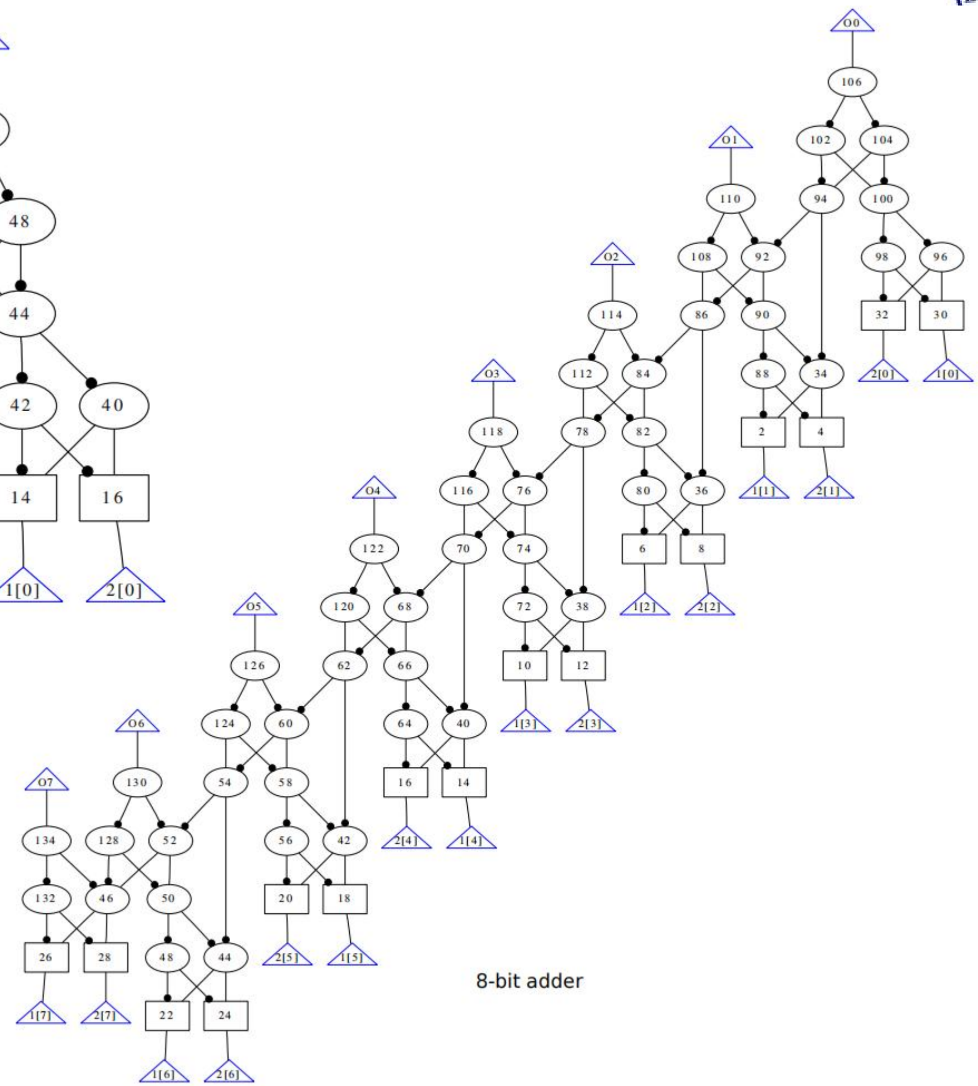
Circuit to SAT

- 加法电路的形式化（n-bit）



4-bit adder





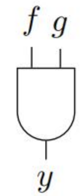
8-bit adder

Circuit to SAT

- 乘法 ⇔ 移位+加法

```
     1011   (this is 11 in decimal)
   x 1110   (this is 14 in decimal)
   ======
     0000   (this is 1011 x 0)
    1011    (this is 1011 x 1, shifted one position to the left)
   1011     (this is 1011 x 1, shifted two positions to the left)
+ 1011      (this is 1011 x 1, shifted three positions to the left)
=========
 10011010   (this is 154 in decimal)
```

Circuit to SAT



AND-Gate

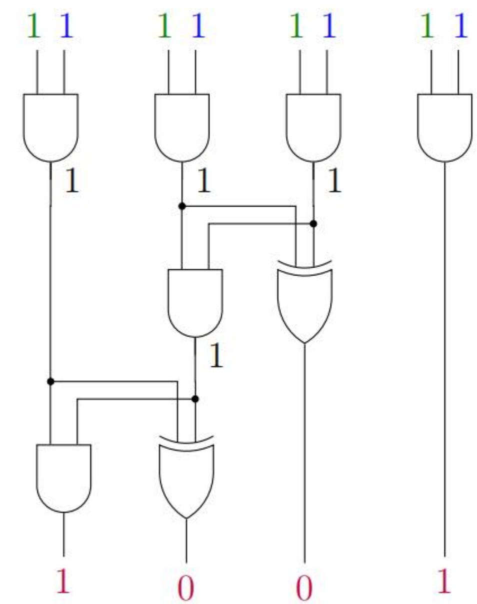
f	g	y
0	0	0
0	1	0
1	0	0
1	1	1

XOR-Gate

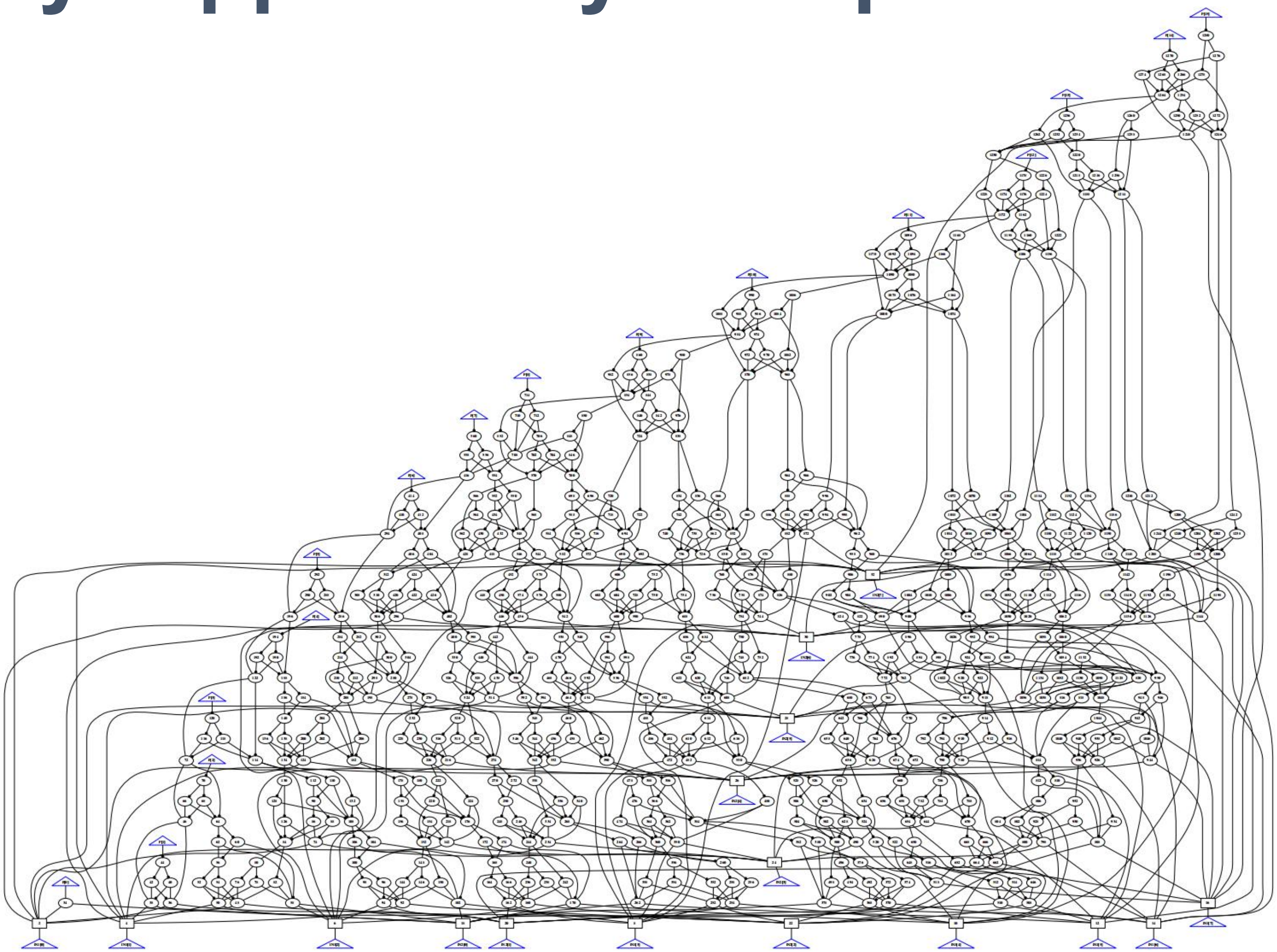
f	g	y
0	0	0
0	1	1
1	0	1
1	1	0

$$\begin{array}{cccc} 1\;1 & \cdot & 1 & 1 \\ \hline & & 1 & 1 \\ {}_1 & 1 & {}_1\;1 & {}_0 \\ \hline 1 & 0 & 0 & 1 \end{array}$$

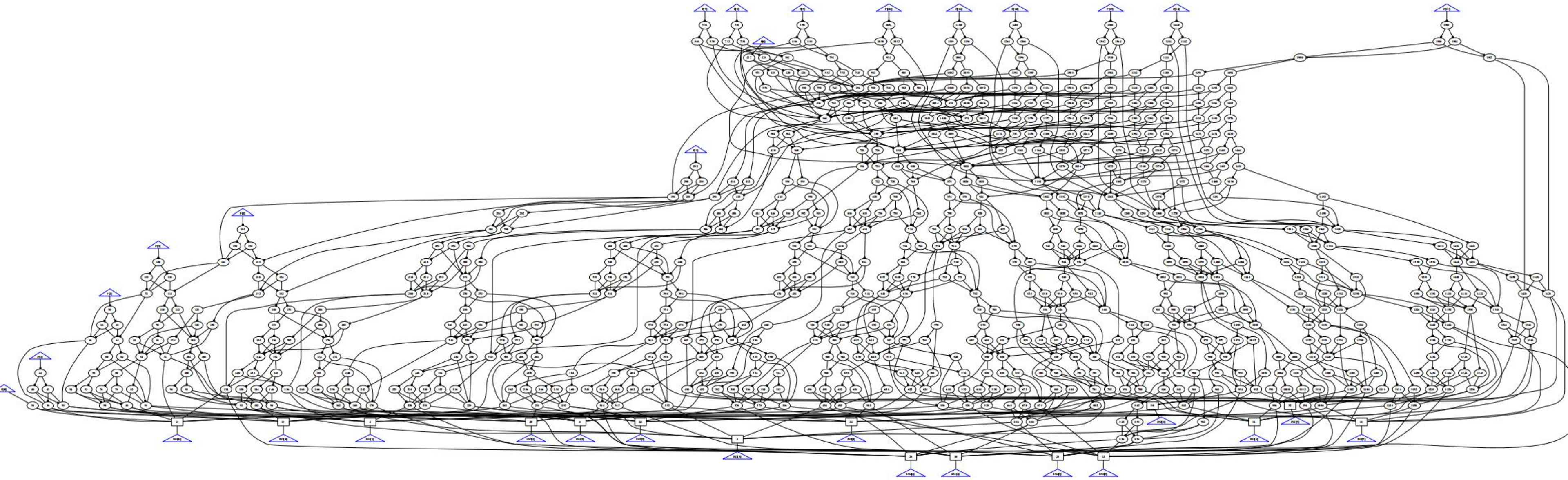
$3 \cdot 3 = 9$



Array Ripple Carry Multiplier



Wallace-Tree Carry-Lookahead Multiplier



Circuit to SAT

- 整数除法
  - 有余数，引入辅助变量表示余数

```
      11   R=10
11 )1011
    -11
     101
     -11
      10  <-- remainder, R
```

```
          0011
x         abcd
==========
            ef
            11  x d
           11   x c
          11    x b
         11     x a
=========
  00001011
```

等价性验证

original C code

```
if(!a && !b) h();
else if(!a) g();
else f();
```

$\Downarrow$

```
if(!a) {
  if(!b) h();
  else g();
} else f();
```

$\Rightarrow$

optimized C code

```
if(a) f();
else if(b) g();
else h();
```

$\Uparrow$

```
if(a) f();
else {
  if(!b) h();
  else g(); }
```

等价性验证

左右是否等效?

original C code

```
if(!a && !b) h();
else if(!a) g();
else f();
```

$\Downarrow$

```
if(!a) {
  if(!b) h();
  else g();
} else f();
```

$\Rightarrow$

optimized C code

```
if(a) f();
else if(b) g();
else h();
```

$\Uparrow$

```
if(a) f();
else {
  if(!b) h();
  else g(); }
```

等价性验证

$$
\begin{aligned}
\mathit{original} &\equiv \textbf{if } \neg a \land \neg b \textbf{ then } h \textbf{ else } \textbf{ if } \neg a \textbf{ then } g \textbf{ else } f \\
&\equiv (\neg a \land \neg b) \land h \lor \neg(\neg a \land \neg b) \land \textbf{ if } \neg a \textbf{ then } g \textbf{ else } f \\
&\equiv (\neg a \land \neg b) \land h \lor \neg(\neg a \land \neg b) \land (\neg a \land g \lor a \land f)
\end{aligned}
$$

$$
\begin{aligned}
\mathit{optimized} &\equiv \textbf{if } a \textbf{ then } f \textbf{ else } \textbf{ if } b \textbf{ then } g \textbf{ else } h \\
&\equiv a \land f \lor \neg a \land \textbf{ if } b \textbf{ then } g \textbf{ else } h \\
&\equiv a \land f \lor \neg a \land (b \land g \lor \neg b \land h)
\end{aligned}
$$

SAT问题求解的应用

- 有界模型检验 (BMC, 2007 Turing Award)
- 芯片自动化设计 (EDA)
- 程序分析、软件验证
- 自动定理证明
  - Boolean Pythagorean Triples (200TB), Schur Number Five (2PB), Certification: Coq, ACL2, Isabelle
- AI与规划问题
- 密码学自动化方法

合取范式

- SAT求解器的输入，为DIMACS CNF格式

$$(x \lor y \lor \neg z) \land (\neg y \lor z)$$

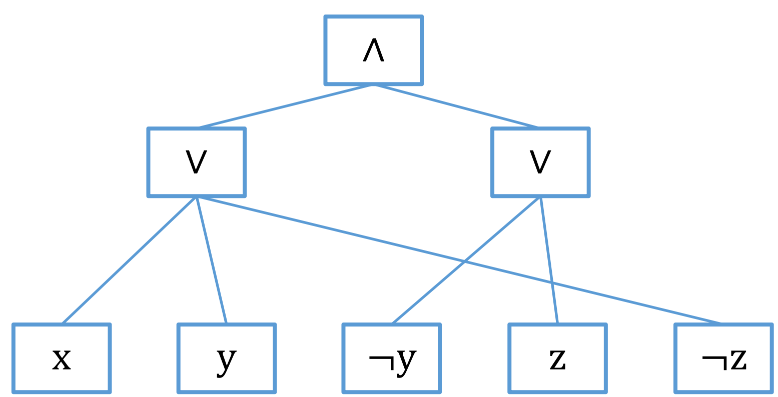
1 2 -3 -2 3

```
p cnf 3 2
1 2 -3 0
-2 3 0
```

- 合取范式（Conjunctive Normal Form, CNF）
  - 公式 = 若干子句 (Clause) 的合取

    子句 = 若干文字 (Literal) 的析取

    文字 = 一个命题符或它的否定



合取范式

给定合取范式F，$v \vDash F$当且仅当v满足F的每个子句

- SAT求解器的输入，为DIMACS CNF格式

(x ∨ y ∨ ¬z) ∧ (¬y ∨ z)

1 2 -3 -2 3

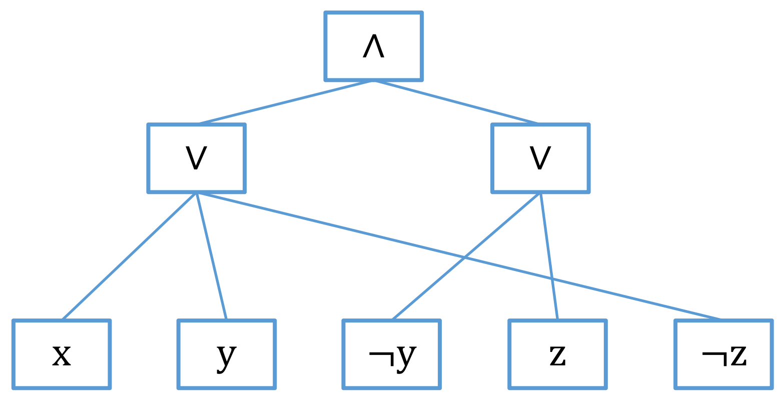
```
p cnf 3 2
1 2 -3 0
-2 3 0
```

形式简单✔
求解高效✔

- 合取范式（Conjunctive Normal Form, CNF）
  - 公式 = 若干子句 (Clause) 的合取

    子句 = 若干文字 (Literal) 的析取

    文字 = 一个命题符或它的否定



如何将一般公式写成合取范式？

- 列真值表生成合取范式
  - 真值表行数为2^n，只能处理很小规模

如何将一般公式写成合取范式？

- 等价替换变形成CNF：
  - $A \to B \simeq \neg A \vee B$
  - $A \leftrightarrow B \simeq (\neg A \vee B) \wedge (A \vee \neg B)$
  - $A \oplus B \simeq (A \wedge \neg B) \vee (\neg A \wedge B) \simeq \neg(A \leftrightarrow B)$
  - 德摩根律：$\neg(a \vee b) \simeq \neg a \wedge \neg b$

    $\neg(a \wedge b) \simeq \neg a \vee \neg b$
  - 分配律：$a \vee (b \wedge c) \simeq (a \vee b) \wedge (a \vee c)$

    $a \wedge (b \vee c) \simeq (a \wedge b) \vee (a \wedge c)$

如何将一般公式写成合取范式？

$$(\text{Alice} \leftrightarrow \text{Bob}) \wedge (\text{Cecile} \leftrightarrow \text{David}) \wedge (\text{Alice} \rightarrow \text{Cecile}) \wedge$$
$$\neg(\text{Eva} \leftrightarrow \text{David}) \wedge \text{Bob} \wedge \text{Fred}$$

$$(\mathrm{A} \leftrightarrow \mathrm{B}) \wedge (\mathrm{C} \leftrightarrow \mathrm{D}) \wedge (\mathrm{A} \rightarrow \mathrm{C}) \wedge \neg(\mathrm{E} \leftrightarrow \mathrm{D}) \wedge \mathrm{B} \wedge \mathrm{F}$$

$$\simeq (\neg A \vee B) \wedge (A \vee \neg B) \wedge (\neg C \vee D) \wedge (C \vee \neg D) \wedge (\neg \mathrm{A} \vee \mathrm{C}) \wedge$$
$$\neg((\neg E \vee D) \wedge (E \vee \neg D)) \wedge \mathrm{B} \wedge \mathrm{F}$$

如何将一般公式写成合取范式？

$$(\mathrm{A} \leftrightarrow \mathrm{B}) \wedge (\mathrm{C} \leftrightarrow \mathrm{D}) \wedge (\mathrm{A} \to \mathrm{C}) \wedge \neg(\mathrm{E} \leftrightarrow \mathrm{D}) \wedge \mathrm{B} \wedge \mathrm{F}$$

$$\simeq (\neg A \vee B) \wedge (A \vee \neg B) \wedge (\neg C \vee D) \wedge (C \vee \neg D) \wedge (\neg \mathrm{A} \vee \mathrm{C}) \wedge$$

$$\boxed{\neg((\neg E \vee D) \wedge (E \vee \neg D))} \wedge \mathrm{B} \wedge \mathrm{F}$$

$$\neg((\neg E \vee D) \wedge (E \vee \neg D) \simeq (E \wedge \neg D) \vee (\neg E \wedge D)$$

$$\simeq ((E \wedge \neg D) \vee \neg E) \wedge ((E \wedge \neg D) \vee D)$$

$$\simeq (E \vee \neg E) \wedge (\neg D \vee \neg E) \wedge (E \vee D) \wedge (\neg D \vee D)$$

$$\simeq (\neg D \vee \neg E) \wedge (E \vee D)$$

如何将一般公式写成合取范式？

$(\mathrm{A} \leftrightarrow \mathrm{B}) \wedge (\mathrm{C} \leftrightarrow \mathrm{D}) \wedge (\mathrm{A} \rightarrow \mathrm{C}) \wedge \neg(\mathrm{E} \leftrightarrow \mathrm{D}) \wedge \mathrm{B} \wedge \mathrm{F}$

$\simeq (\neg A \vee B) \wedge (A \vee \neg B) \wedge (\neg C \vee D) \wedge (C \vee \neg D) \wedge (\neg \mathrm{A} \vee \mathrm{C}) \wedge$

$\boxed{\neg((\neg E \vee D) \wedge (E \vee \neg D))} \wedge \mathrm{B} \wedge \mathrm{F}$

$\Downarrow$

$\neg((\neg E \vee D) \wedge (E \vee \neg D) \simeq (E \wedge \neg D) \vee (\neg E \wedge D)$

$\simeq ((E \wedge \neg D) \vee \neg E) \wedge ((E \wedge \neg D) \vee D)$

$\simeq (E \vee \neg E) \wedge (\neg D \vee \neg E) \wedge (E \vee D) \wedge (\neg D \vee D)$

$\simeq (\neg D \vee \neg E) \wedge (E \vee D)$

公式长度指数增长

例，$(p_1 \wedge p_2 \wedge p_3) \vee$
$(q_1 \wedge q_2 \wedge q_3)$

Tseitin变换

- 引入新的命题变元代替重复的子式

原子式	替换	替换后的合取范式子式
$\neg x$	$z \leftrightarrow \neg x$	$(x \vee z) \wedge (\neg x \vee \neg z)$
$x \wedge y$	$z \leftrightarrow x \wedge y$	$(x \vee \neg z) \wedge (y \vee \neg z) \wedge (\neg x \vee \neg y \vee z)$
$x \vee y$	$z \leftrightarrow x \vee y$	$(\neg x \vee z) \wedge (\neg y \vee z) \wedge (x \vee y \vee \neg z)$

例，$(x1 \wedge x2) \vee (\neg(x3 \wedge \neg x4))$

- 引入新变量：$y1, y2, y3, y4, y5$
- 子式替换：

  $y1 \leftrightarrow x1 \wedge x2$

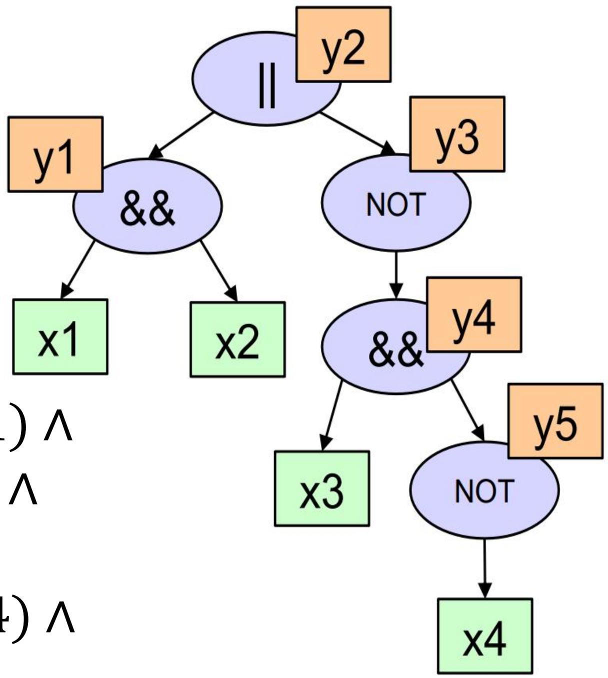
  $y2 \leftrightarrow y1 \vee y3$

  $y3 \leftrightarrow \neg y4$

  $y4 \leftrightarrow x3 \wedge y5$

  $y5 \leftrightarrow \neg x4$

替换	替换后的合取范式子式
$z \leftrightarrow \neg x$	$(x \vee z) \wedge (\neg x \vee \neg z)$
$z \leftrightarrow x \wedge y$	$(x \vee \neg z) \wedge (y \vee \neg z) \wedge (\neg x \vee \neg y \vee z)$
$z \leftrightarrow x \vee y$	$(\neg x \vee z) \wedge (\neg y \vee z) \wedge (x \vee y \vee \neg z)$



- 合取范式：

$$
\begin{aligned}
&(x1 \vee \neg y1) \wedge (x2 \vee \neg y1) \wedge (\neg x1 \vee \neg x2 \vee y1) \wedge \\
&(\neg y1 \vee y2) \wedge (\neg y3 \vee y2) \wedge (y1 \vee y3 \vee \neg y2) \wedge \\
&(y3 \vee y4) \wedge (\neg y3 \vee \neg y4) \wedge \\
&(x3 \vee \neg y4) \wedge (y5 \vee \neg y4) \wedge (\neg x3 \vee \neg y5 \vee y4) \wedge \\
&(x4 \vee y5) \wedge (\neg x4 \vee \neg y5) \wedge \\
&y2
\end{aligned}
$$

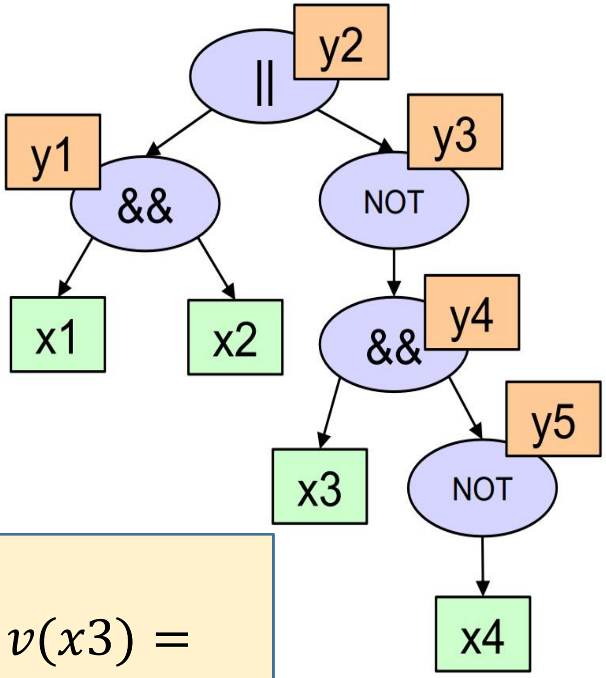
Tseitin变换

- 给定命题公式A，经过Tseitin变换后的合取范式为B，则
  - $B \vDash A$，但$A \nvDash B$
  - 若赋值v，$v \vDash A$，则可构造v'使得$v' \vDash B$

例，$(x1 \wedge x2) \vee (\neg(x3 \wedge \neg x4))$

- 新变量：y1, y2, y3, y4, y5
- 合取范式：

$$(x1 \vee \neg y1) \wedge (x2 \vee \neg y1) \wedge (\neg x1 \vee \neg x2 \vee y1) \wedge$$
$$(\neg y1 \vee y2) \wedge (\neg y3 \vee y2) \wedge (y1 \vee y3 \vee \neg y2) \wedge$$
$$(y3 \vee y4) \wedge (\neg y3 \vee \neg y4) \wedge$$
$$(x3 \vee \neg y4) \wedge (y5 \vee \neg y4) \wedge (\neg x3 \vee \neg y5 \vee y4) \wedge$$
$$(x4 \vee y5) \wedge (\neg x4 \vee \neg y5) \wedge$$
$$y2$$



赋值v：
$v(x1) = v(x2) = v(x3) = v(x4) = v(y1) = v(y2) = v(y3) = v(y4) = v(y5) = 1$

3-SAT问题

- 合取范式中每个子句长度小于等于3，这类公式的可满足性问题被称为3-SAT问题

- 3-SAT问题也是NP完全问题，即一般SAT问题可以多项式时间归约到3-SAT问题

SAT求解算法

- 早期回溯搜索

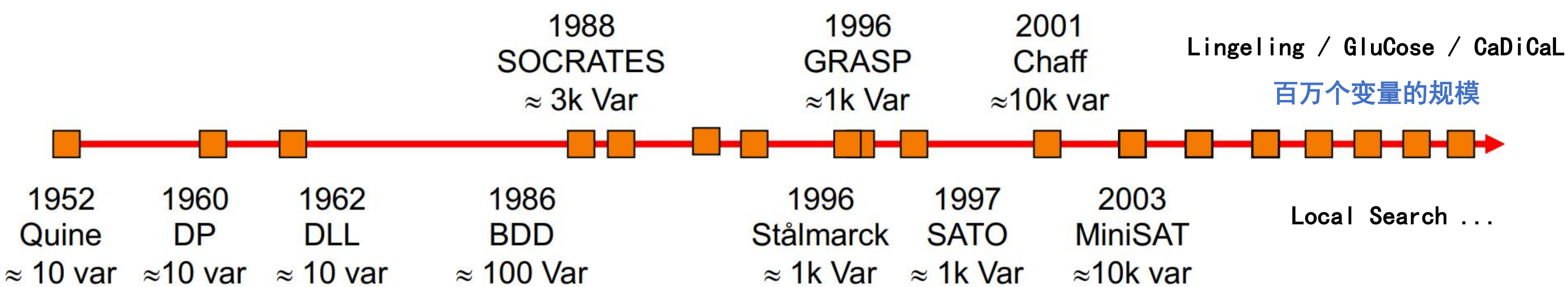
  Davis, Putnam, Logemann and Loveland 1962 [DLL, DPLL]

- 冲突驱动的子句学习

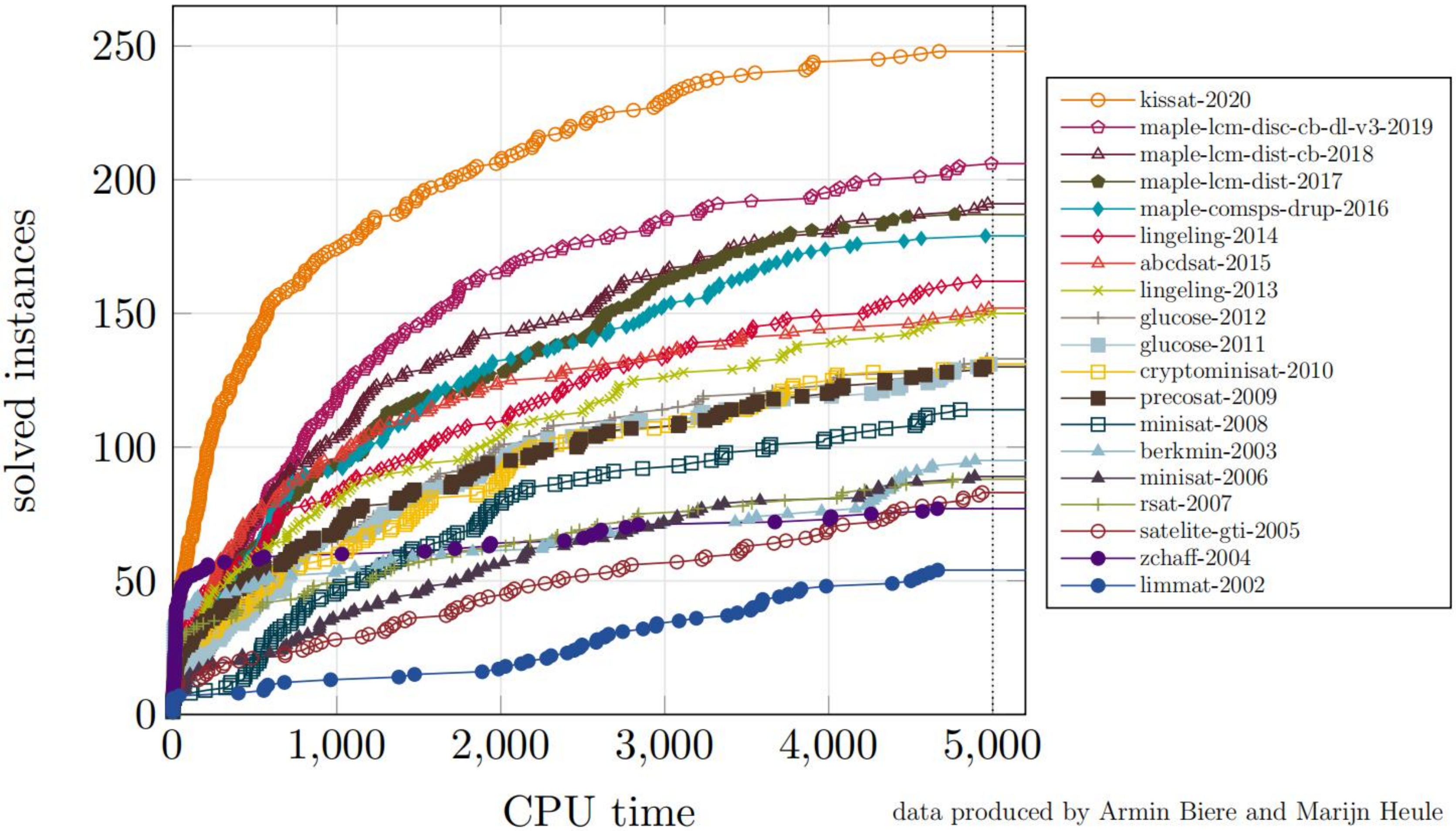
  Conflict Driven Clause Learning [CDCL]

- ……

SAT问题的求解



1988
SOCRATES
≈ 3k Var
1996
GRASP
≈1k Var
2001
Chaff
≈10k var
Lingeling / GluCose / CaDiCaL
百万个变量的规模
1952
Quine
≈ 10 var
1960
DP
≈10 var
1962
DLL
≈ 10 var
1986
BDD
≈ 100 Var
1996
Stålmarck
≈ 1k Var
1997
SATO
≈ 1k Var
2003
MiniSAT
≈10k var
Local Search ...



data produced by Armin Biere and Marijn Heule