

命题逻辑





命题逻辑

语法





字母表

定义1.1（字母表）. 命题逻辑的字母表含三类符号：

(1) 命题符号：

$p \quad q \quad r \quad \dots$

(2) 联结符号（联结词）：

$\neg \quad \wedge \quad \vee \quad \rightarrow$

(3) 辅助符号（标点符号）：

$(\quad)$



命题公式

- 命题逻辑的所有原子公式和公式的集分别记为 **PS** （命题符集合）和 **$PROP$** （命题集）。
 - 公式由表达式定义
 - 公式相当于自然语言中符合语法规则的语句



命题公式

- 命题逻辑的所有原子公式和公式的集分别记为 **PS** （命题符集合）和 **$PROP$** （命题集）。
 - 公式由表达式定义
 - 公式相当于自然语言中符合语法规则的语句
- 表达式不一定是公式
 - p
 - pq
 - (r)
 - $p \wedge \rightarrow q$
 - $(p \vee q)$



命题的定义

定义1.2 ($PROP$) . $A \in PROP$ 当且仅当它能有限次地由以下

(i)~(iii)生成:

(i) $PS \subseteq PROP$;

(ii) 如果 $A \in PROP$, 则 $(\neg A) \in PROP$;

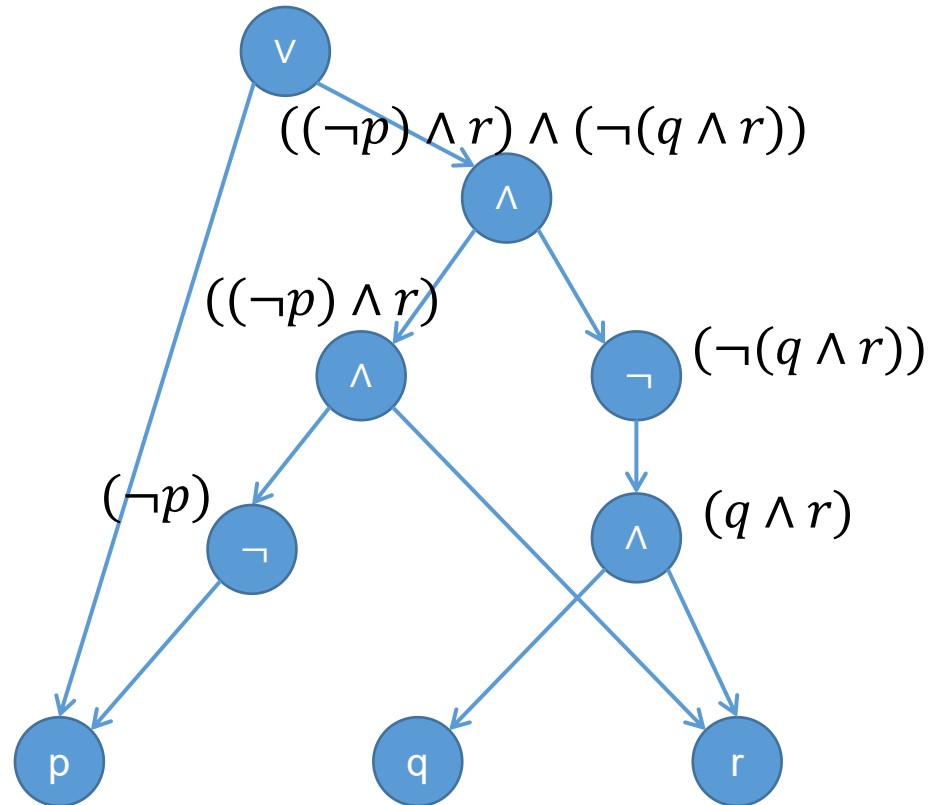
(iii) 如果 $A, B \in PROP$, 则 $(A * B) \in PROP$ 。

其中 $* \in \{\wedge, \vee, \rightarrow\}$ 。

- 定义中的(i)~(iii)称为命题公式的**形成规则**。

命题的定义

- 例, $(p \vee (((\neg p) \wedge r) \wedge (\neg(q \wedge r))))$ 。





命题逻辑

语义





语法与语义

- 语法：符号表达式的形式结构
- 语义：符号和符号表达式的涵义（给符号以某种解释）



命题的语义

- 什么是命题逻辑的语义？
- 对于任意的**赋值** $v: PS \rightarrow \{T, F\}$, 定义一个**解释**

$$\hat{v}: PROP \rightarrow \{T, F\}$$

联结词定义的布尔函数

定义1.3. 令真值集 $B = \{T, F\}$,

- 联结词 $\neg$ 被解释为一元函数 $H_{\neg}: B \rightarrow B$;
- 联结词 $*$ 被解释为二元函数 $H_*: B^2 \rightarrow B$,
其中 $*$ $\in \{\wedge, \vee, \rightarrow\}$;
- $H_{\neg}, H_{\wedge}, H_{\vee}, H_{\rightarrow}$ 定义如下:

p	q	$H_{\neg}(p)$	$H_{\wedge}(p, q)$	$H_{\vee}(p, q)$	$H_{\rightarrow}(p, q)$
T	T	F	T	T	T
T	F	F	F	T	F
F	T	T	F	T	T
F	F	T	F	F	T



命题的语义

定义1.4（命题的语义）.

- v 为一个**赋值**指它是函数 $v: PS \rightarrow B$,
从而对任何命题符 P_i , $v(P_i)$ 为T或F;



命题的语义

定义1.4（命题的语义）.

- v 为一个**赋值**指它是函数 $v: PS \rightarrow B$,
从而对任何命题符 P_i , $v(P_i)$ 为T或F;
 - 对于任何赋值 v , 定义 $\hat{v}: PROP \rightarrow B$ 如下:
$$\hat{v}(P_n) = v(P_n), n \in N;$$
$$\hat{v}(\neg A) = H_{\neg}(\hat{v}(A));$$
$$\hat{v}(A * B) = H_*(\hat{v}(A), \hat{v}(B)), \text{ 其中 } * \in \{\wedge, \vee, \rightarrow\}.$$
- 对于命题A, 它在赋值 v 下的**解释** $\hat{v}(A)$ 为T或F。



命题的语义

例, $A = (p \wedge q) \rightarrow (\neg q \vee r)$, 设 v 是一个赋值, 使得
$$v(p) = v(q) = v(r) = 1.$$



命题的语义

例, $A = (p \wedge q) \rightarrow (\neg q \vee r)$, 设 v 是一个赋值, 使得
$$v(p) = v(q) = v(r) = 1.$$

那么, 我们有

$$\begin{aligned}\hat{v}(p \wedge q) &= H_{\wedge}(p, q) = 1, \\ \hat{v}(\neg q \vee r) &= H_{\vee}(H_{\neg}(q), r) = 1, \\ \hat{v}(A) &= H_{\rightarrow}(H_{\wedge}(p, r), H_{\vee}(H_{\neg}(q), r)) = 1.\end{aligned}$$



命题的语义

例, $A = (p \wedge q) \rightarrow (\neg q \vee r)$, 设 v 是一个赋值, 使得
$$v(p) = v(q) = v(r) = 0.$$



命题的语义

例, $A = (p \wedge q) \rightarrow (\neg q \vee r)$, 设 v 是一个赋值, 使得
$$v(p) = v(q) = v(r) = 0.$$

我们有

$$\begin{aligned}\hat{v}(p \wedge q) &= H_{\wedge}(p, q) = 0, \\ \hat{v}(\neg q \vee r) &= H_{\vee}(H_{\neg}(q), r) = 1, \\ \hat{v}(A) &= H_{\rightarrow}(H_{\wedge}(p, q), H_{\vee}(H_{\neg}(q), r)) = 1.\end{aligned}$$



可满足性

定义1.5. 设 $A \in PROP$, v 为赋值, $\Gamma \subseteq PROP$.

1. v 满足 A , 记为 $v \models A$, 指 $\hat{v}(A) = T$;

A 是可满足的, 指 $\exists v$ 使得 $v \models A$;

2. v 满足 Γ , 记为 $v \models \Gamma$, 指对于 $\forall B \in \Gamma$, $v \models B$;

Γ 是可满足的, 指 $\exists v$ 使得 $v \models \Gamma$.

注: 若 $v \not\models A$, 则 $v \models \neg A$.

Γ 的可满足性蕴含 Γ 中所有公式的可满足性。

但反之不一定成立。



永真式

定义1.6. 设 A 为命题， v 为赋值。

1. A 为永真式（也称重言式），记为 $\models A$ ，
指对于 $\forall v$ 都有 $\hat{v}(A) = T$ ；
2. A 为矛盾式，指对于 $\forall v$ 都有 $\hat{v}(A) = F$ ；

例， $A \rightarrow A$ ，

$\neg\neg A \rightarrow A$ ，

$(A \wedge B) \rightarrow (B \wedge A)$.



语义结论

定义1.7. 设 $A \in PROP$, v 为赋值, $\Gamma \subseteq PROP$.

A 是 Γ 的语义结论 (也称逻辑推论), 记为 $\Gamma \models A$,

指对所有 v , 若 $v \models \Gamma$, 则 $v \models A$ 。

注: 此处 $\models$ 也是元语言中的符号,

$\Gamma \models A$ 也可以读作“ Γ 逻辑地蕴含 A ”,

$\Gamma \models A$ 不是形式语言中的公式, 是元语言中的命题。



逻辑等价

定义1.8. 设 A, B 为命题, A 与 B **逻辑等价** (也称**逻辑等值**), 记为 $A \simeq B$, 指对于任意赋值 v , $v \models A$ 当且仅当 $v \models B$ 。

注: 有如下等价的定义:

$A \simeq B$, 当且仅当 $A \models B$ 且 $B \models A$ 。

任何赋值 v , $\hat{v}(A) = \hat{v}(B)$ 。



等值替换

$$(1) A \rightarrow B \simeq \neg A \vee B;$$

$$(2) A \leftrightarrow B \simeq (\neg A \vee B) \wedge (A \vee \neg B);$$

(1)~(4): 消去 $\rightarrow$, $\leftrightarrow$, $\oplus$

$$(3) A \leftrightarrow B \simeq (A \wedge B) \vee (\neg A \wedge \neg B);$$

$$(4) A \oplus B \simeq (A \wedge \neg B) \vee (\neg A \wedge B) \simeq \neg(A \leftrightarrow B);$$

可以说, $\rightarrow$, $\leftrightarrow$, $\oplus$ 可以由 $\neg$, $\wedge$, $\vee$ 定义。



等值替换

$$(1) A \rightarrow B \simeq \neg A \vee B;$$

$$(2) A \leftrightarrow B \simeq (\neg A \vee B) \wedge (A \vee \neg B);$$

(1)~(4): 消去 $\rightarrow$, $\leftrightarrow$, $\oplus$

$$(3) A \leftrightarrow B \simeq (A \wedge B) \vee (\neg A \wedge \neg B);$$

$$(4) A \oplus B \simeq (A \wedge \neg B) \vee (\neg A \wedge B) \simeq \neg(A \leftrightarrow B);$$

$$(5) \neg\neg A \simeq A;$$

(5)~(7): 消去 $\neg$ 的辖域中的 $\neg$, $\wedge$, $\vee$

$$(6) \neg(A_1 \wedge \dots \wedge A_n) \simeq \neg A_1 \vee \dots \vee \neg A_n;$$

$$(7) \neg(A_1 \vee \dots \vee A_n) \simeq \neg A_1 \wedge \dots \wedge \neg A_n;$$



等值替换

$$(1) A \rightarrow B \simeq \neg A \vee B;$$

$$(2) A \leftrightarrow B \simeq (\neg A \vee B) \wedge (A \vee \neg B);$$

(1)~(4): 消去 $\rightarrow$, $\leftrightarrow$, $\oplus$

$$(3) A \leftrightarrow B \simeq (A \wedge B) \vee (\neg A \wedge \neg B);$$

$$(4) A \oplus B \simeq (A \wedge \neg B) \vee (\neg A \wedge B) \simeq \neg(A \leftrightarrow B);$$

$$(5) \neg\neg A \simeq A;$$

(5)~(7): 消去 $\neg$ 的辖域中的 $\neg$, $\wedge$, $\vee$

$$(6) \neg(A_1 \wedge \dots \wedge A_n) \simeq \neg A_1 \vee \dots \vee \neg A_n;$$

$$(7) \neg(A_1 \vee \dots \vee A_n) \simeq \neg A_1 \wedge \dots \wedge \neg A_n;$$

(8): 消去 $\wedge$ 的辖域中的 $\vee$
(9): 消去 $\vee$ 的辖域中的 $\wedge$

$$(8) A \wedge (B_1 \vee \dots \vee B_n) \simeq (A \wedge B_1) \vee \dots \vee (A \wedge B_n);$$

$$(9) A \vee (B_1 \wedge \dots \wedge B_n) \simeq (A \vee B_1) \wedge \dots \wedge (A \vee B_n).$$



等值替换

$$(10) A \vee A \simeq A$$

$$(11) A \wedge A \simeq A$$

$$(12) A \vee (A \wedge B) \simeq A$$

$$(13) A \wedge (A \vee B) \simeq A$$

$$(14) A \vee (B \wedge \neg B \wedge C) \simeq A$$

$$(15) A \wedge (B \vee \neg B \vee C) \simeq A$$

(10)(11): 重复项

(12)(13): 一个子句的所有文字出现在另一个子句中

(14)(15): 删去含互补文字的子句



析取范式-合取范式

定义1.9（文字，子句）.

- （1）命题符和命题符的否定式称为**文字**（Literal）；
- （2）以文字为析（合）取项的析（合）取式称为**析（合）取子式**，简称**子式**，也称**子句**（Clause）。



析取范式-合取范式

定义1.10（范式 Normal Form）.

- (1) 命题A为**析取范式** ($\vee\wedge$ -nf, DNF), 指A为 m 个合取子式的析取式, 呈形 $\bigvee_{i=1}^m (\bigwedge_{k=1}^{n_i} P_{i,k})$ 。
- (2) 命题A为**合取范式** ($\wedge\vee$ -nf, CNF), 指A为 l 个析取子式的合取式, 呈形 $\bigwedge_{j=1}^l (\bigvee_{k=1}^{n_j} Q_{j,k})$ 。

以上

- $\bigwedge_{k=1}^n B_k$ 为 $(\dots(((B_1 \wedge B_2) \wedge B_3)\dots \wedge B_n)\dots)$ 的简写;
- $\bigvee_{k=1}^n B_k$ 为 $(\dots(((B_1 \vee B_2) \vee B_3)\dots \vee B_n)\dots)$ 的简写。



析取范式-合取范式

析取范式 $\bigvee_{i=1}^m (\bigwedge_{k=1}^n P_{i,k})$ 为如下形式：

$$(P_{11} \wedge \dots \wedge P_{1n_1}) \vee \dots \vee (P_{m1} \wedge \dots \wedge P_{mn_m}),$$

文字



析取范式-合取范式

析取范式 $\bigvee_{i=1}^m (\bigwedge_{k=1}^n P_{i,k})$ 为如下形式:

$$\underline{(P_{11} \wedge \dots \wedge P_{1n_1})} \vee \dots \vee \underline{(P_{m1} \wedge \dots \wedge P_{mn_m})},$$

子句



析取范式-合取范式

析取范式 $\bigvee_{i=1}^m (\bigwedge_{k=1}^n P_{i,k})$ 为如下形式：

$$(P_{11} \wedge \dots \wedge P_{1n_1}) \vee \dots \vee (P_{m1} \wedge \dots \wedge P_{mn_m}),$$

一个文字为假，则子句为假；一个子句为真，则公式为真

合取范式 $\bigwedge_{j=1}^l (\bigvee_{k=1}^n Q_{j,k})$ 为如下形式：

$$(Q_{11} \vee \dots \vee Q_{1n_1}) \wedge \dots \wedge (Q_{l1} \vee \dots \vee Q_{ln_l}).$$

一个文字为假，则子句为假；一个子句为真，则公式为真



析取范式-合取范式

例,

$$(1) \ p$$

$$(2) \ \neg p \vee q$$

$$(3) \ \neg p \wedge q \wedge \neg r$$

$$(4) \ \neg p \vee (q \wedge \neg r)$$

$$(5) \ \neg p \wedge (q \vee \neg r) \wedge (\neg q \vee r)$$



命题逻辑

CS与AI中的应用





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：Alice $\leftrightarrow$ Bob, Cecile $\leftrightarrow$ David
 - 2、如果Alice则Cecile：Alice $\rightarrow$ Cecile
 - 3、要么David要么Eva： $\neg$ (Eva $\leftrightarrow$ David)
 - 4、邀请Bob和Fred：Bob $\wedge$ 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 $v \models A$ ？
 - 此赋值 v 也被称为问题的一个解

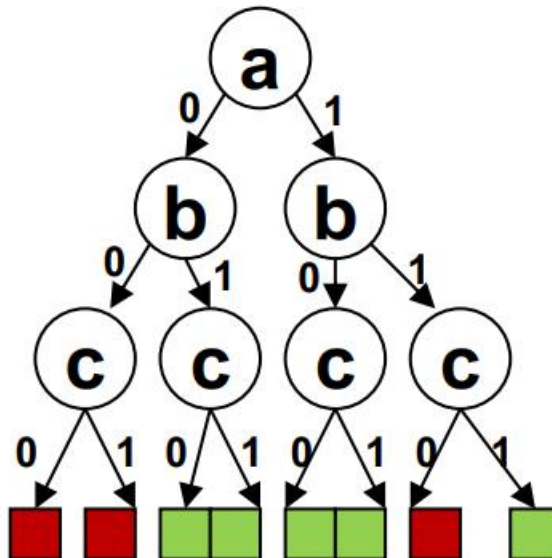
可满足性问题

- 给定一个命题公式 A ，问是否存在一个赋值 v ，使得 $v \models A$ ？

➤ 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

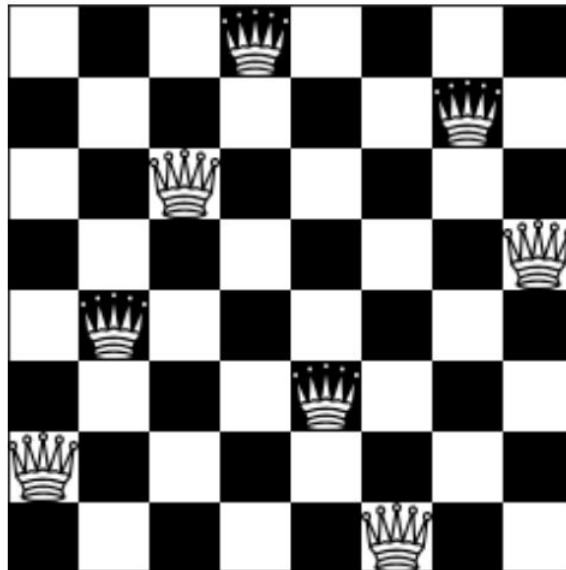




可满足性问题

- 给定一个命题公式 A ，问是否存在一个赋值 v ，使得 $v \models A$ ？
 - 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；
 - $P \subseteq NP$ ✓ $P=NP$?（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

不同行:

第 i 行只有一个皇后

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

不同行:

第 i 行只有一个皇后

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

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

不同行:

第 i 行只有一个皇后

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

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

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

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

$\wedge \dots$

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

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}



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

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

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

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

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

$\wedge \dots$

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

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 $\{1, \dots, n\}$ 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 $\{1, \dots, n\}$ 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 $n = 4k + 2$ 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

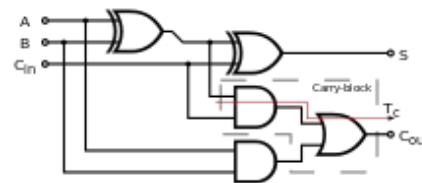
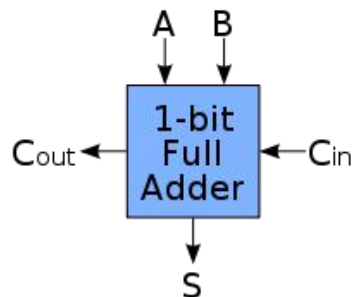
- 加法电路的形式化 (1-bit)

➤ $A + B + C_{in} = C_{out}S \Leftrightarrow$

➤ $C_{out} = (A \text{ and } B) \text{ or } (C_{in} \text{ and } (A \text{ or } B))$

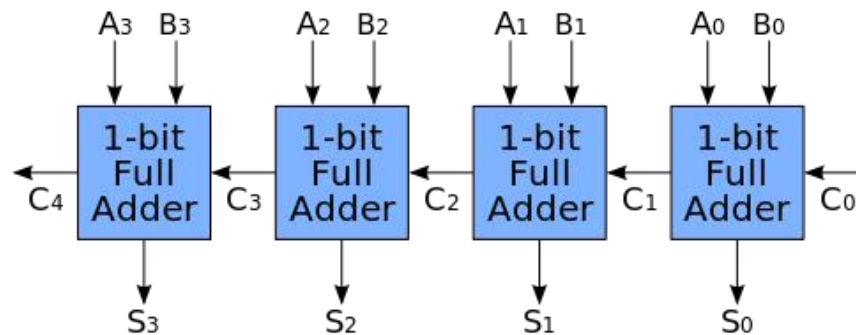
➤ $S = A \text{ xor } B \text{ xor } C_{in}$

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)





Circuit to SAT

- 乘法 $\Leftrightarrow$ 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

- 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
```

Circuit to SAT



AND-Gate

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

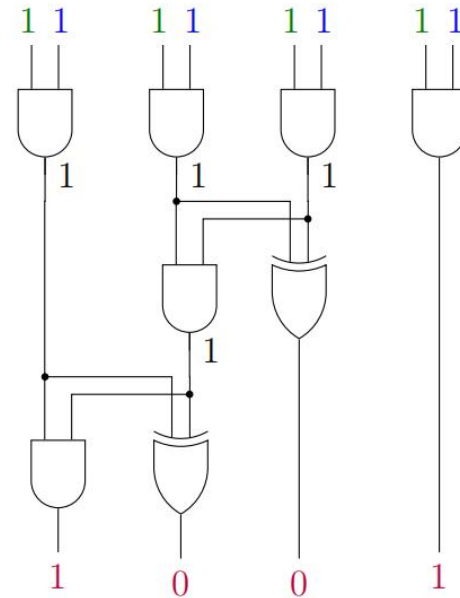


XOR-Gate

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

$$\begin{array}{r}
 1 \quad 1 \quad \cdot \quad 1 \quad 1 \\
 \hline
 1 \quad 1 \\
 1 \quad 1 \quad 0 \\
 \hline
 1 \quad 0 \quad 0 \quad 1
 \end{array}$$

$$3 \cdot 3 = 9$$



Circuit to SAT

- $s_3 \leftrightarrow g_1 \wedge g_4$
- $s_2 \leftrightarrow g_1 \oplus g_4$
- $s_1 \leftrightarrow g_2 \oplus g_3$
- $s_0 \leftrightarrow a_0 \wedge b_0$
- $g_1 \leftrightarrow a_1 \wedge b_1$
- $g_2 \leftrightarrow a_0 \wedge b_1$
- $g_3 \leftrightarrow a_1 \wedge b_0$
- $g_4 \leftrightarrow g_2 \wedge g_3$



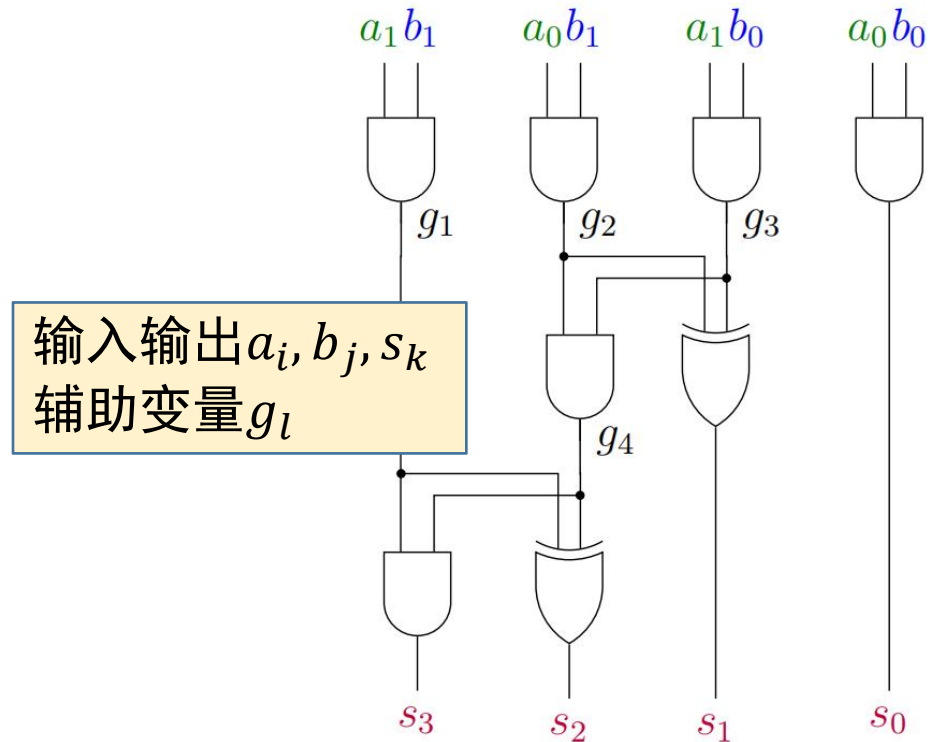
AND-Gate

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



Circuit to SAT

$$\begin{aligned}
 F = & (s_3 \leftrightarrow g_1 \wedge g_4) \wedge \\
 & (s_2 \leftrightarrow g_1 \oplus g_4) \wedge \\
 & (s_1 \leftrightarrow g_2 \oplus g_3) \wedge \\
 & (s_0 \leftrightarrow a_0 \wedge b_0) \wedge \\
 & (g_1 \leftrightarrow a_1 \wedge b_1) \wedge \\
 & (g_2 \leftrightarrow a_0 \wedge b_1) \wedge \\
 & (g_3 \leftrightarrow a_1 \wedge b_0) \wedge \\
 & (g_4 \leftrightarrow g_2 \wedge g_3)
 \end{aligned}$$



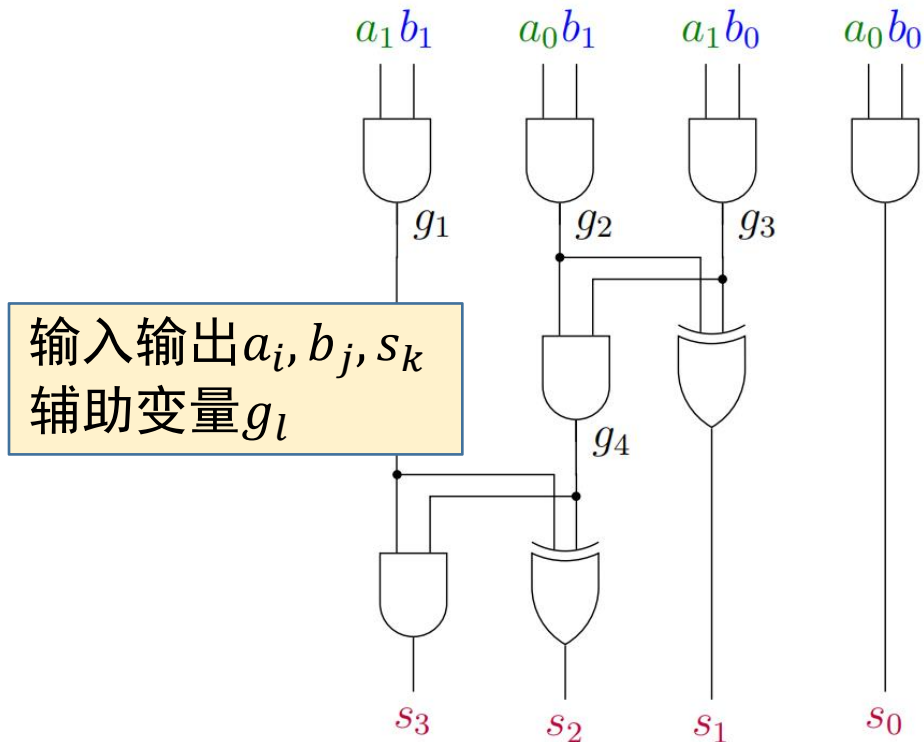
AND-Gate

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

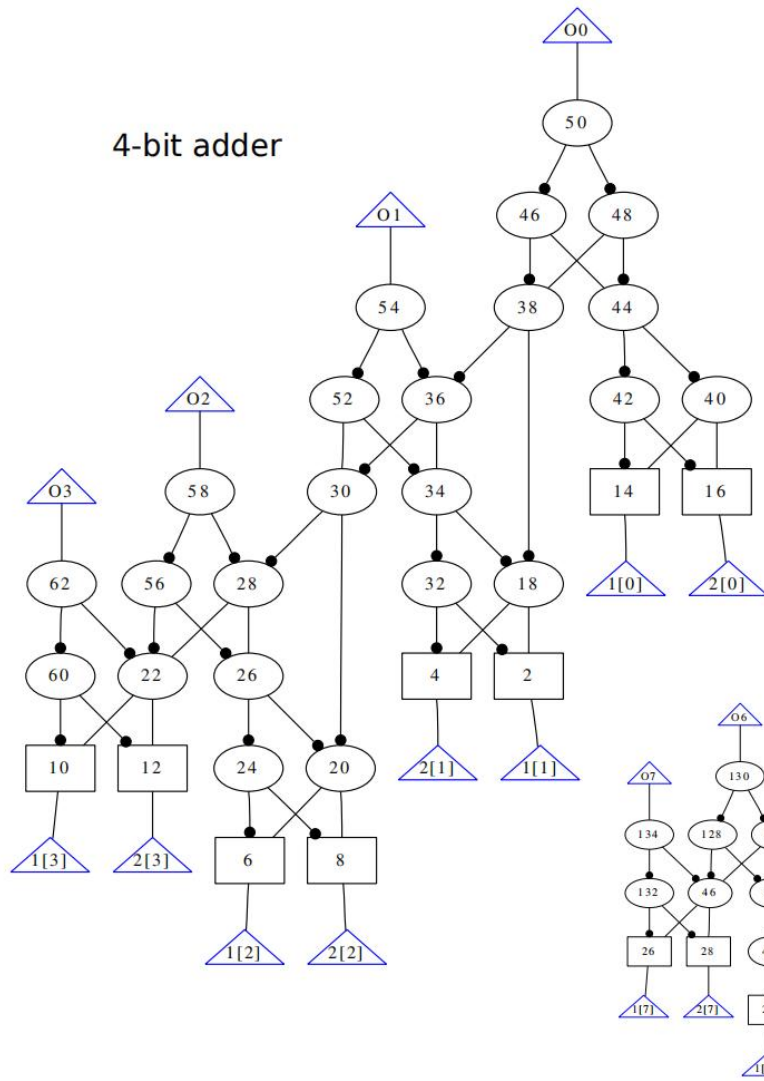


XOR-Gate

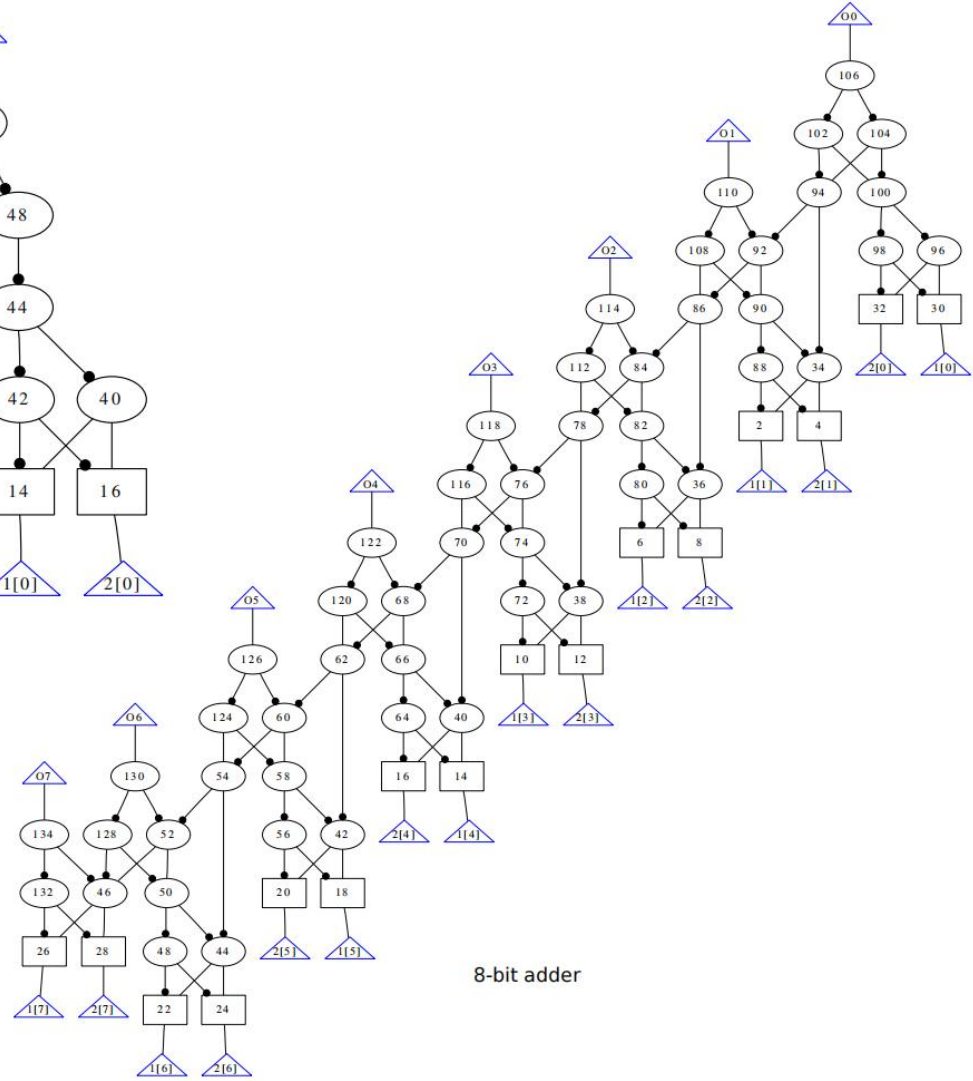
f	g	y
0	0	0
0	1	1
1	0	1
1	1	0



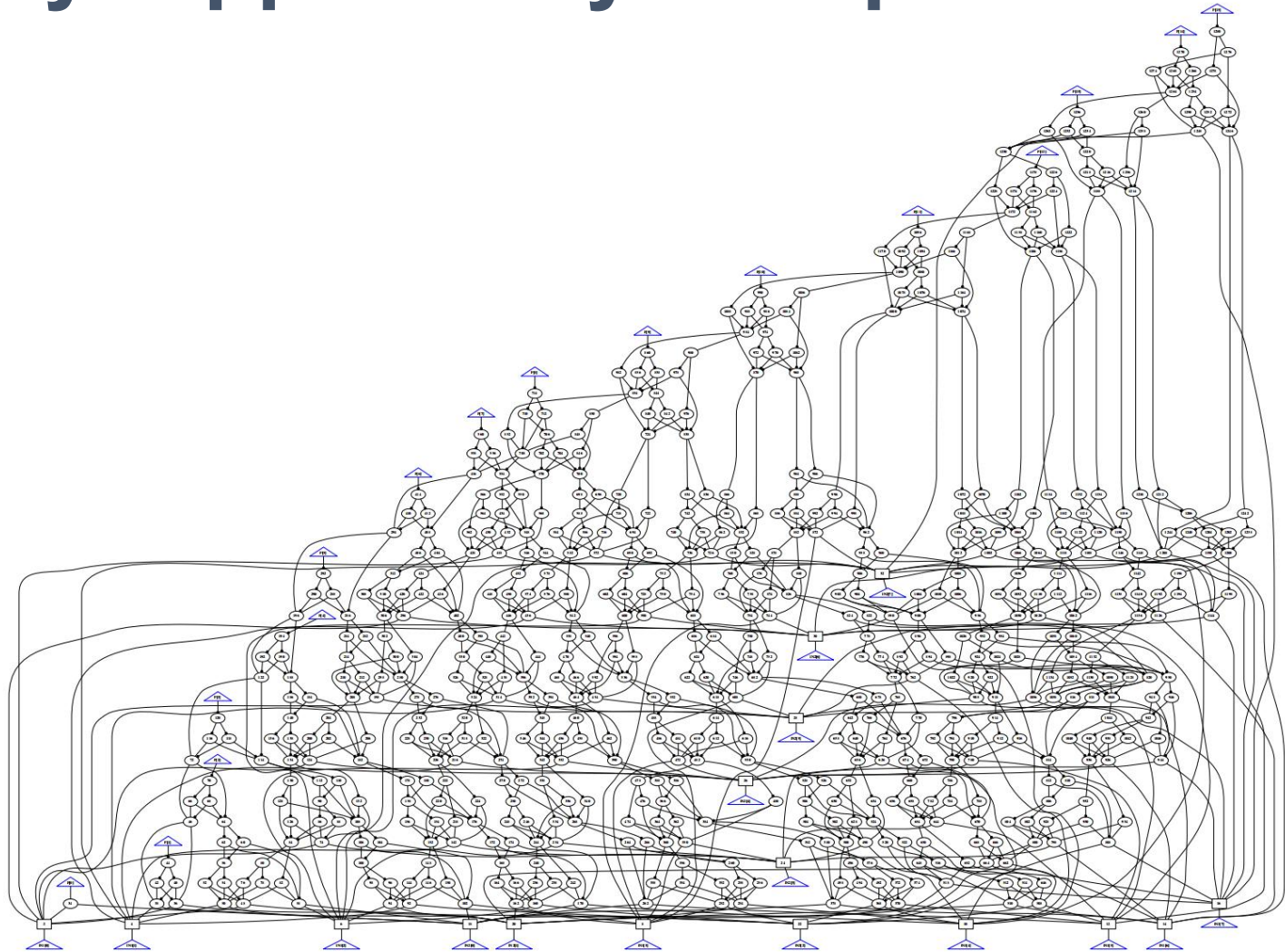
4-bit adder



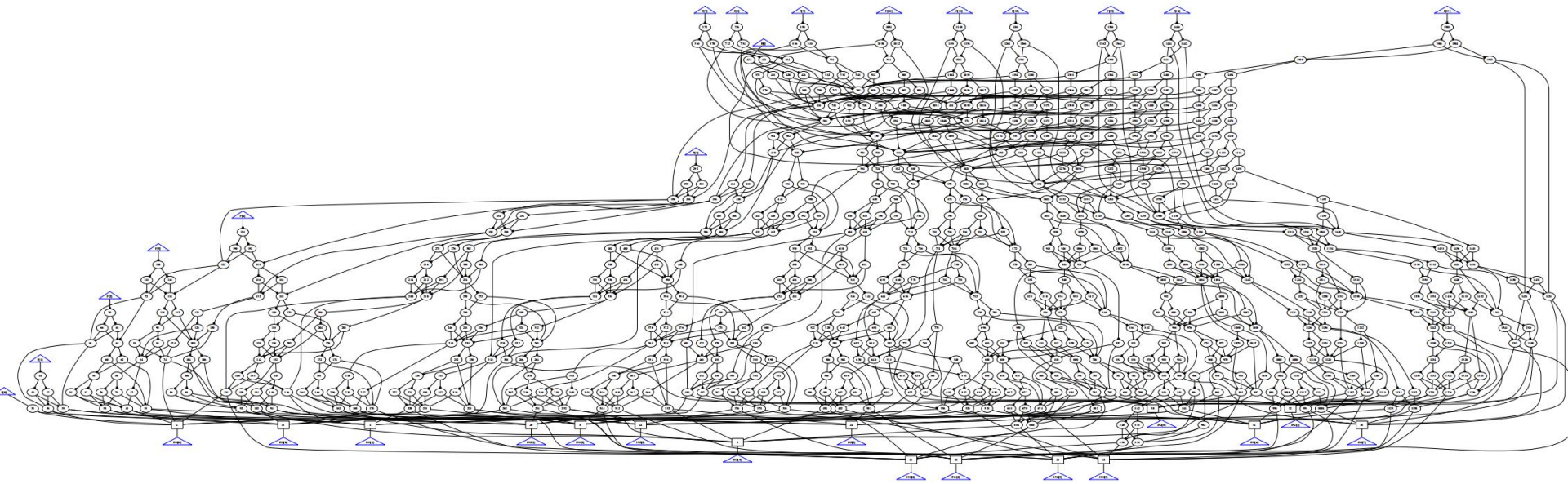
8-bit adder



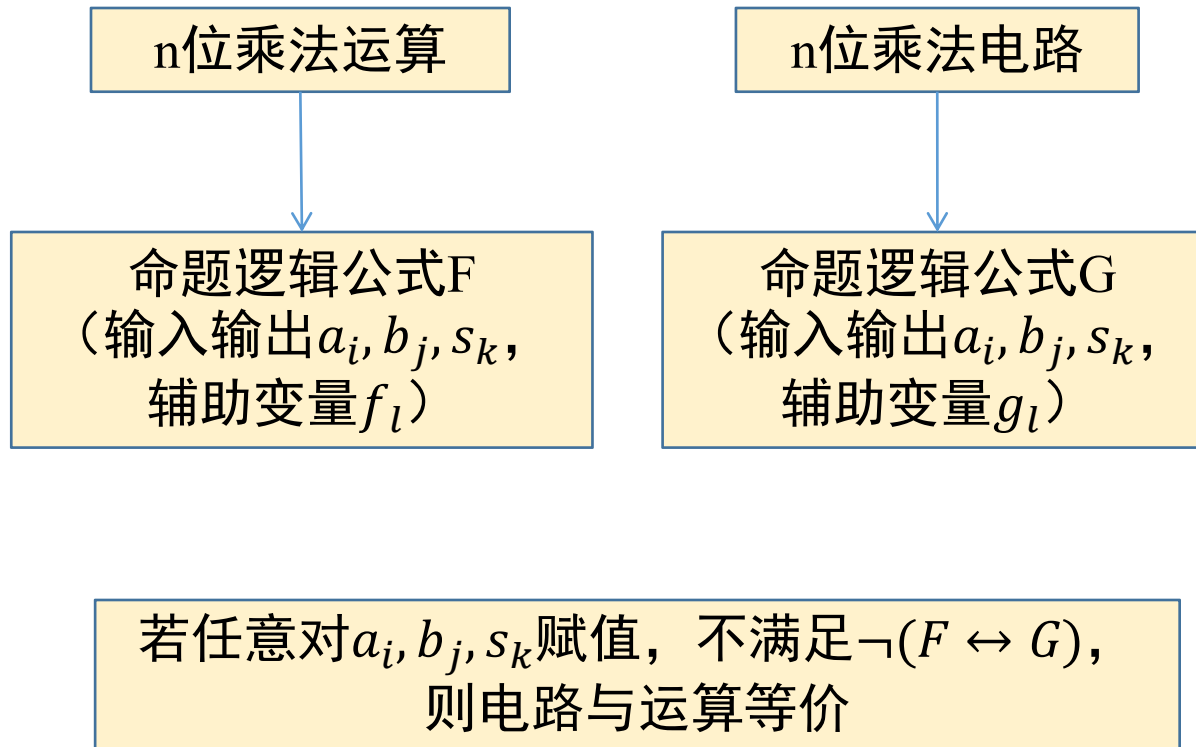
Array Ripple Carry Multiplier



Wallace-Tree Carry-Lookahead Multiplier



等价性验证



等价性验证



original C code

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



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



optimized C code

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



```
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();
```



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

$\Rightarrow$

optimized C code

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



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

等价性验证



$$\begin{aligned} \text{original} &\equiv \text{if } \neg a \wedge \neg b \text{ then } h \text{ else if } \neg a \text{ then } g \text{ else } f \\ &\equiv (\neg a \wedge \neg b) \wedge h \vee \neg(\neg a \wedge \neg b) \wedge \text{if } \neg a \text{ then } g \text{ else } f \\ &\equiv (\neg a \wedge \neg b) \wedge h \vee \neg(\neg a \wedge \neg b) \wedge (\neg a \wedge g \vee a \wedge f) \end{aligned}$$

$$\begin{aligned} \text{optimized} &\equiv \text{if } a \text{ then } f \text{ else if } b \text{ then } g \text{ else } h \\ &\equiv a \wedge f \vee \neg a \wedge \text{if } b \text{ then } g \text{ else } h \\ &\equiv a \wedge f \vee \neg a \wedge (b \wedge g \vee \neg b \wedge h) \end{aligned}$$



SAT问题应用

- 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
- 规划问题
- 密码学自动化分析