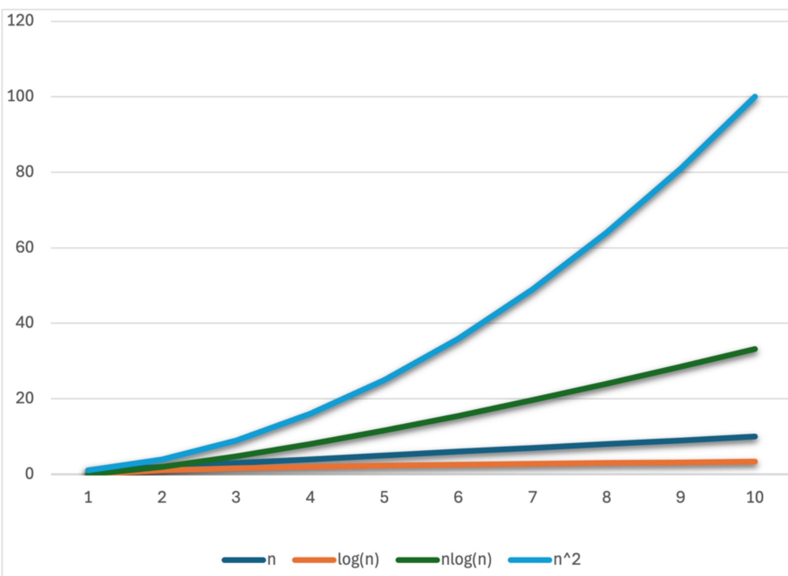
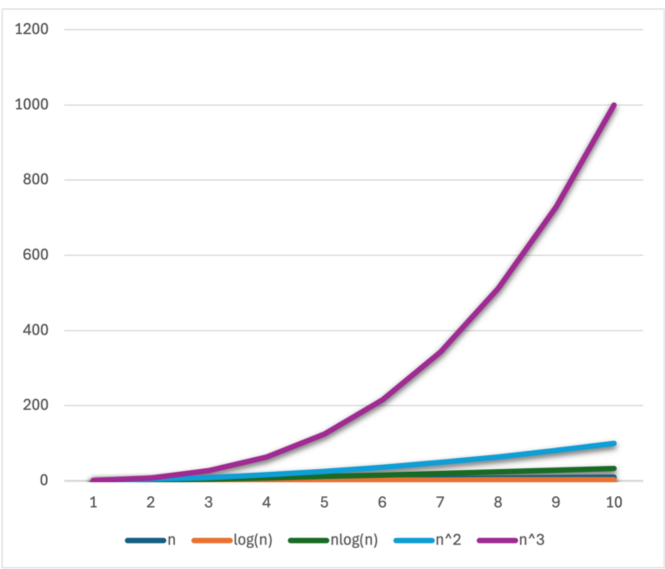
SAT Encodings

SAT Encodings

- Encoding in SAT can be challenging.
- Need to be careful with the encoding size.
  - Otherwise, SAT solving efficiency may significantly degrade.





Cardinality Constraints

- $\sum_{1 \le i \le n} x_i \bowtie k$ where $k \in \{<, \le, =, \neq, >, \ge\}$
  - A special case is $\sum_{1 \le i \le n} x_i = 1$ $(\sum_{1 \le i \le n} x_i \le 1 \wedge \sum_{1 \le i \le n} x_i \ge 1)$
  - Let
    - AtmostOne($[x_1, x_2, \ldots, x_n]$) refer to $\sum_{1 \le i \le n} x_i \le 1$
    - AtleastOne($[x_1, x_2, \ldots, x_n]$) refer to $\sum_{1 \le i \le n} x_i \ge 1$
    - ExactlyOne($[x_1, x_2, \ldots, x_n]$) refer to AtmostOne($[x_1, x_2, \ldots, x_n]$) $\wedge$ AtleastOne($[x_1, x_2, \ldots, x_n]$)

Cardinality Constraints

- Frequently occur in SAT models. E.g.:
  - N-queens
    - Exactly one queen on each row and column.
    - At most one queen on each diagonal.
  - Sudoku
    - Exactly one presence of each number in each row, column and 3x3 grid.
    - Exactly one number in each cell.

Cardinality Constraints

- How do we encode such constraints?
- AtleastOne$([x_1, x_2, \ldots, x_n\ ])$
  - $x_1 \vee x_2 \vee \ldots \vee x_n$
- AtmostOne$([x_1, x_2, \ldots, x_n\ ])$
  - There exist many possibilities.

AtmostOne: Pairwise Encoding

- Any combination of 2 variables cannot be true at the same time.

$$\bigwedge_{1 \le i < j \le n} \neg(x_i \wedge x_j)$$

  - Simple, no additional variables.
  - $O(n^2)$ clauses.

AtmostOne: Sequential Encoding

- Introduce n new variables s_i to indicate that the sum has reached 1 by i.

$$
\begin{aligned}
& x_1 \rightarrow s_1 \wedge \\
\bigwedge_{1<i<n} \Big[\; & ((x_i \vee s_{i-1}) \rightarrow s_i) \wedge (s_{i-1} \rightarrow \overline{x_i}) \; \Big] \\
& \wedge (s_{n-1} \rightarrow \overline{x_n})
\end{aligned}
$$

AtmostOne: Sequential Encoding

- Introduce n new variables s_i to indicate that the sum has reached 1 by i.

$$(\overline{x_1} \vee s_1) \wedge$$

$$\bigwedge_{1<i<n} [\ (\overline{x_i} \vee s_i) \wedge (\overline{s_{i-1}} \vee s_i) \wedge (\overline{s_{i-1}} \vee \overline{x_i})\ ]$$

$$\wedge\ (\overline{s_{n-1}} \vee \overline{x_n})$$

  - $O(n)$ clauses and $O(n)$ new variables.

AtmostOne: Bitwise Encoding

- Introduce m new variables r_i where $m = \log_2 n$.
- For $1 \leq i \leq n$, let $b_{i,1}, \ldots, b_{i,m}$ be the binary encoding of $i - 1$.

$$\bigwedge_{1 \leq i \leq n} x_i \rightarrow (r_1 = b_{i,1} \wedge r_2 = b_{i,2} \wedge \ldots \wedge r_m = b_{i,m})$$

- Example: $x_1 + x_2 + x_3 \leq 1$

$m = 2$

$$x_1 \rightarrow (r_1 = 0 \wedge r_2 = 0) \wedge$$
$$x_2 \rightarrow (r_1 = 0 \wedge r_2 = 1) \wedge$$
$$x_3 \rightarrow (r_1 = 1 \wedge r_2 = 0)$$

AtmostOne: Bitwise Encoding

- Introduce m new variables r_i where $m = \log_2 n$.
- For $1 \le i \le n$, let $b_{i,1}, \ldots, b_{i,m}$ be the binary encoding of $i - 1$.

$$\bigwedge_{1 \le i \le n} x_i \rightarrow (r_1 = b_{i,1} \land r_2 = b_{i,2} \land \ldots \land r_m = b_{i,m})$$

- Example: $x_1 + x_2 + x_3 \le 1$

$m = 2$

$$(\overline{x_1} \lor \overline{r_1}) \land (\overline{x_1} \lor \overline{r_2}) \land$$
$$(\overline{x_2} \lor \overline{r_1}) \land (\overline{x_2} \lor r_2) \land$$
$$(\overline{x_3} \lor r_1) \land (\overline{x_3} \lor \overline{r_2})$$

AtmostOne: Bitwise Encoding

- Introduce m new variables r_i where $m = \log_2 n$.
- For $1 \leq i \leq n$, let $b_{i,1}, \ldots, b_{i,m}$ be the binary encoding of $i-1$.

$$\bigwedge_{1 \leq i \leq n} \bigwedge_{1 \leq j \leq m} \overline{x}_i \vee r_j \;[\vee\, \overline{r}_j\,]$$

if bit j of the binary encoding of $i-1$ is 1 [or 0].

  - $O(n \log_2 n)$ clauses and $O(\log_2 n)$ new variables.

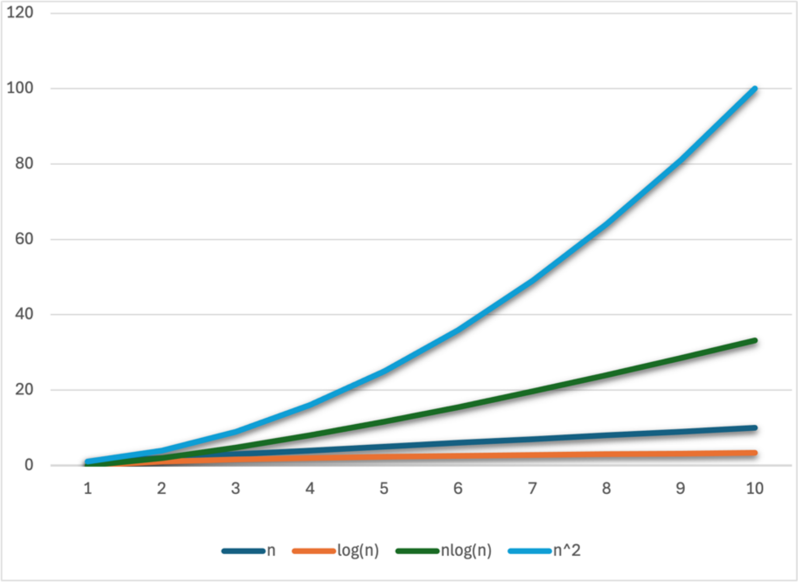
AtmostOne: Heule Encoding

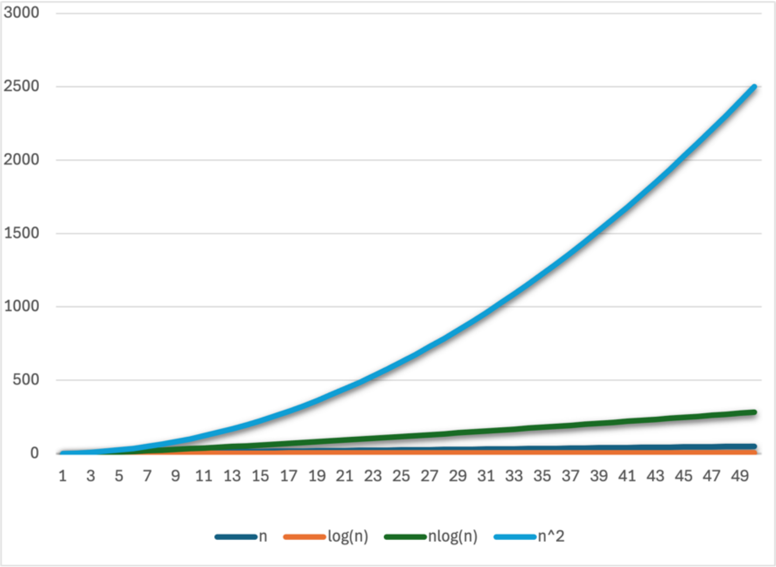
- Split the constraint using additional variables.
- When $n \leq 4$, apply pairwise encoding, using at most 6 clauses.

$$\bigwedge_{1 \leq i < j \leq n} \neg(x_i \wedge x_j)$$

- When $n > 4$:
  - introduce a new Boolean variable y.
  - AtmostOne$([x_1, x_2, x_3, y]) \wedge$ AtmostOne$([\bar{y}, x_4, \dots, x_n])$
    - Encode the second one recursively.
  - $O(n)$ clauses and $O(n)$ new variables.

AtmostOne: Encoding Size Differences





Cardinality Constraints

- $\sum_{1 \le i \le n} x_i \ \bowtie k$ where $k \in \{<, \le, =, \neq, >, \ge\}$
  - $\sum_{1 \le i \le n} x_i = k$ iff $(\sum_{1 \le i \le n} x_i \le k) \wedge (\sum_{1 \le i \le n} x_i \ge k)$
  - $\sum_{1 \le i \le n} x_i \neq k$ iff $(\sum_{1 \le i \le n} x_i > k) \vee (\sum_{1 \le i \le n} x_i < k)$
  - $\sum_{1 \le i \le n} x_i \ge k$ iff $\sum_{1 \le i \le n} \overline{x_i} \le n - k$
  - $\sum_{1 \le i \le n} x_i > k$ iff $\sum_{1 \le i \le n} \overline{x_i} \le n - k - 1$
  - $\sum_{1 \le i \le n} x_i < k$ iff $\sum_{1 \le i \le n} x_i \le k - 1$

Cardinality Constraints

- $\sum_{1 \le i \le n} x_i \ \bowtie k$ where $k \in \{<, \le, =, \ne, >, \ge\}$
  - Another special case is $\sum_{1 \le i \le n} x_i = k \ \ (\sum_{1 \le i \le n} x_i \le k \land \sum_{1 \le i \le n} x_i \ge k)$
  - Let
    - AtmostK($[x_1, x_2, \dots, x_n]$) refer to $\sum_{1 \le i \le n} x_i \le k$
    - AtleastK($[x_1, x_2, \dots, x_n]$) refer to $\sum_{1 \le i \le n} x_i \ge k$
    - ExactlyK($[x_1, x_2, \dots, x_n]$) iff AtmostK($[x_1, x_2, \dots, x_n]$) $\land$ AtleastK($[x_1, x_2, \dots, x_n]$)
  - Frequently occur in SAT models, e.g., nurse scheduling.
    - For fairness, across all days, each nurse is assigned to a number of shifts between a minimum and a maximum value.

Cardinality Constraints

- $\sum_{1 \le i \le n} x_i \bowtie k$ where $k \in \{<, \le, =, \ne, >, \ge\}$
  - $\sum_{1 \le i \le n} x_i = k$ iff $(\sum_{1 \le i \le n} x_i \le k) \wedge (\sum_{1 \le i \le n} x_i \ge k)$
  - $\sum_{1 \le i \le n} x_i \ne k$ iff $(\sum_{1 \le i \le n} x_i > k) \vee (\sum_{1 \le i \le n} x_i < k)$
  - $\sum_{1 \le i \le n} x_i \ge k$ iff $\sum_{1 \le i \le n} \overline{x_i} \le n - k$
  - $\sum_{1 \le i \le n} x_i > k$ iff $\sum_{1 \le i \le n} \overline{x_i} \le n - k - 1$
  - $\sum_{1 \le i \le n} x_i < k$ iff $\sum_{1 \le i \le n} x_i \le k - 1$

Cardinality Constraints

- $\sum_{1 \le i \le n} x_i \;\; \bowtie k$ where $k \in \{<, \le, =, \neq, >, \ge\}$
  - $\sum_{1 \le i \le n} x_i = k$ iff $(\sum_{1 \le i \le n} x_i \le k) \wedge (\sum_{1 \le i \le n} x_i \ge k)$
  - $\sum_{1 \le i \le n} x_i \neq k$ iff $(\sum_{1 \le i \le n} x_i > k) \vee (\sum_{1 \le i \le n} x_i < k)$
  - $\sum_{1 \le i \le n} x_i \ge k$ iff $\sum_{1 \le i \le n} \overline{x_i} \le n - k$
  - $\sum_{1 \le i \le n} x_i > k$ iff $\sum_{1 \le i \le n} \overline{x_i} \le n - k - 1$
  - $\sum_{1 \le i \le n} x_i < k$ iff $\sum_{1 \le i \le n} x_i \le k - 1$

AtmostK: Generalized Pairwise Encoding

- AtmostOne: any combination of 2 variables cannot be true at the same time.

$$\bigwedge_{1 \le i < j \le n} \neg(x_i \wedge x_j) \equiv \bigwedge_{1 \le i < j \le n} \overline{x_i} \vee \overline{x_j}$$

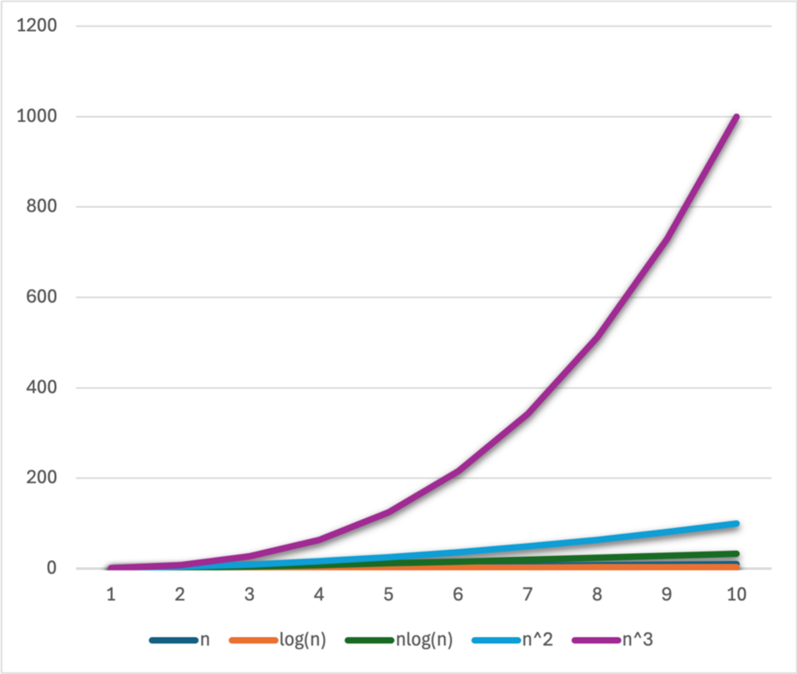
  - $O(n^2)$ clauses

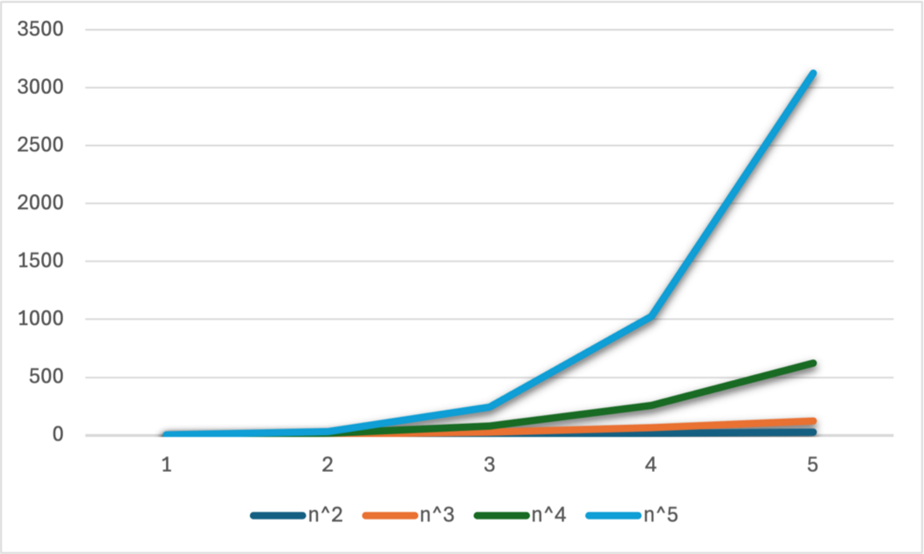
- AtmostK: any combination of $k+1$ variables cannot be true at the same time.

$$\bigwedge_{\substack{M \subseteq \{1..n\} \\ |M| = k+1}} \bigvee_{i \in M} \overline{x_i}$$

  - $O(n^{k+1})$ clauses

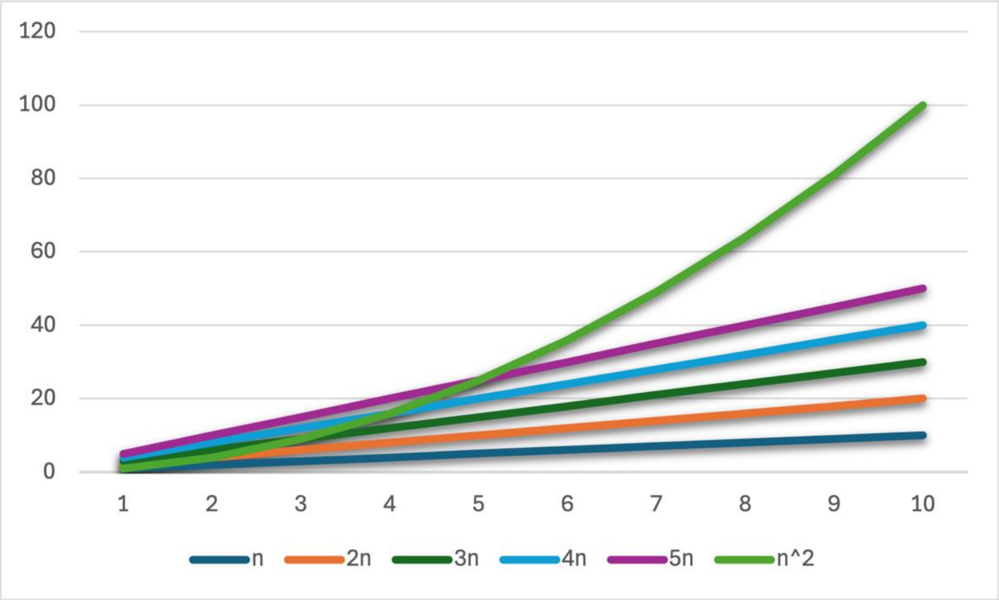
AtmostK: Generalized Pairwise Encoding Size





AtmostK: Sequential Encoding

- AtmostOne: introduce n new variables s_i to indicate that the sum has reached 1 by i.
  - $O(n)$ clauses and $O(n)$ new variables.
- AtmostK: introduce $n * k$ new variables s_{ij} to indicate that the sum has reached j by i.
  - $O(kn)$ clauses and $O(kn)$ new variables.



AtmostK: Sequential Encoding

- AtmostK: introduce $n * k$ new variables $s_{i,j}$ to indicate that the sum has reached to j by i.

$$(x_1 \rightarrow s_{1,1}) \wedge \bigwedge_{2 \le j \le k} \overline{s_{1,j}}$$

$$\bigwedge_{1<i<n} \left[\, ((x_i \vee s_{i-1,1}) \rightarrow s_{i,1}) \wedge \bigwedge_{2 \le j \le k} (((x_i \wedge s_{i-1,j-1}) \vee s_{i-1,j}) \rightarrow s_{i,j}) \wedge \left(s_{i-1,k} \rightarrow \overline{x_i}\right) \right]$$

$$\wedge \left(s_{n-1,k} \rightarrow \overline{x_n}\right)$$

AtmostK: Sequential Encoding

- AtmostOne: introduce $n * 1$ new variables s_i to indicate that the sum has reached to 1 by i.

$$(x_1 \to s_1) \wedge \cancel{\bigwedge_{2\le j\le k} \overline{s_{1,j}}}$$

$$\bigwedge_{1<i<n} \Big[\ ((x_i \vee s_{i-1}) \to s_i)\ \wedge \cancel{\bigwedge_{2\le j\le k} (((x_i \wedge s_{i-1,j-1}) \vee s_{i-1,j}) \to s_{i,j})}$$

$$\wedge\, (s_{i-1} \to \overline{x_i})\ \Big]$$

$$\wedge\, (s_{n-1} \to \overline{x_n})$$

AtmostK: Sequential Encoding

- AtmostK: introduce $n * k$ new variables $s_{i,j}$ to indicate that the sum has reached to j by i.

$$
\begin{array}{ll}
(\neg x_1 \vee s_{1,1}) & \\
(\neg s_{1,j}) & \text{for } 1 < j \leq k \\
\left.\begin{array}{l}
(\neg x_i \vee s_{i,1}) \\
(\neg s_{i-1,1} \vee s_{i,1}) \\
\left.\begin{array}{l}
(\neg x_i \vee \neg s_{i-1,j-1} \vee s_{i,j}) \\
(\neg s_{i-1,j} \vee s_{i,j})
\end{array}\right\} \quad \text{for } 1 < j \leq k \\
(\neg x_i \vee \neg s_{i-1,k})
\end{array}\right\} & \text{for } 1 < i < n \\
(\neg x_n \vee \neg s_{n-1,k}) &
\end{array}
$$

  - $O(nk)$ clauses and $O(nk)$ new variables.

SAT Encodings

- What properties should SAT encodings have?
  - Number of variables.
  - Number of clauses.
  - Other?

Arc-Consistency

- Let us consider an encoding E of a constraint C such that there is a correspondence between the assignments of the variables in C with Boolean assignments of the variables in E.
- E is arc-consistent if:
  - whenever a partial assignment is inconsistent wrt C (i.e., cannot be extended to a solution of C), unit propagation in E causes conflict;
  - Otherwise, unit propagation in E discards arc-inconsistent values (values that cannot be assigned).

Arc-Consistency

- E.g., AtmostOne$([x_1, x_2, \ldots, x_n])$
  - If there are two variables x_i and x_j assigned to T then unit propagation should give a conflict.
  - If there is one x_i assigned to T then all other x_j should be assigned to F by unit propagation.

Arc-Consistency

- Are the AtmostOne encodings we have seen so far ac-consistent encodings?
  - Let’s revisit them.

Arc-Consistency

- Are the AtmostOne encodings we have seen so far ac-consistent encodings?
- Let’s consider another AtmostOne encoding which is not arc-consistent.

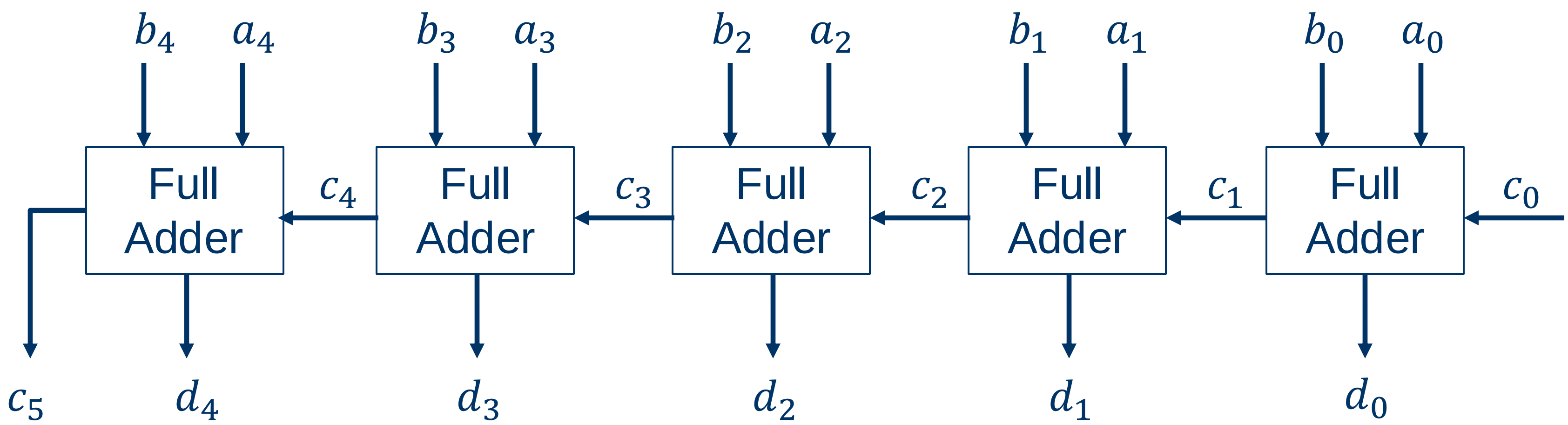
Addition in Propositional Logic

- Decision problem
  - Given a and b (represented in binary), find d (represented in binary) satisfying $a + b = d$.
- Variables
  - $a_{n-1} \dots a_0$, $b_{n-1} \dots b_0$, $d_{n-1} \dots d_0$
  - Carries $c_n c_{n-1} \dots c_0$
- Constraints
  - Compute d_i from right to left starting from $c_0 = 0$.

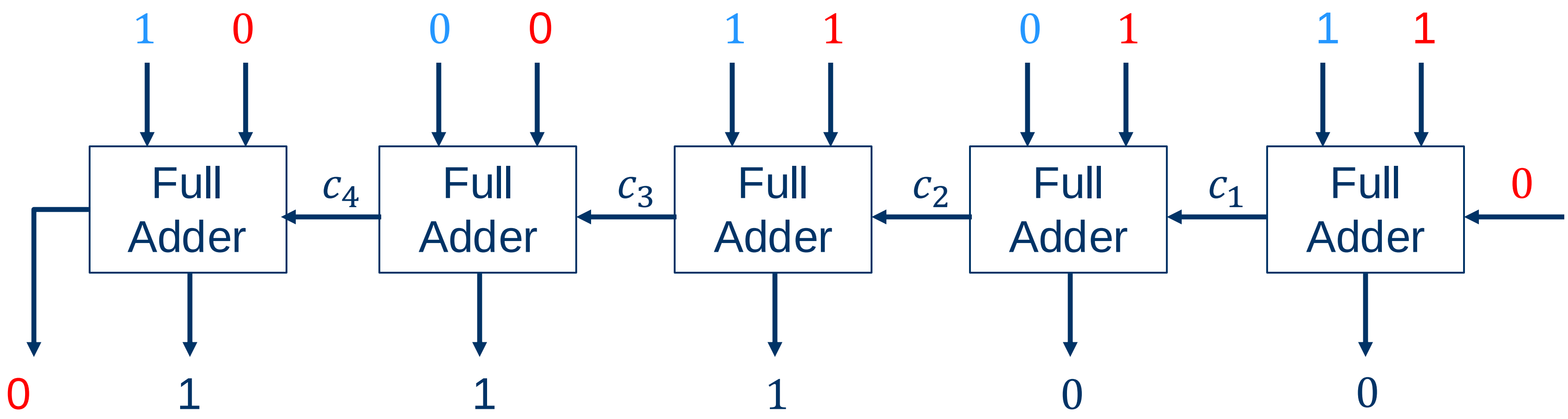
Example

$$
\begin{array}{lcccccc}
c \rightarrow & 0 & 0 & 1 & 1 & 1 & 0 \\
a = 7 \rightarrow & & 0 & 0 & 1 & 1 & 1 \\
b = 21 \rightarrow & & 1 & 0 & 1 & 0 & 1 \\
\hline
d = 28 \rightarrow & & 1 & 1 & 1 & 0 & 0
\end{array}
$$

5-bit Binary Adder using Full Adder



7+21 with Full Adders



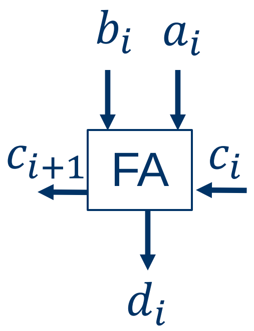
$c \rightarrow$	0	0	1	1	1	0
$a = 7 \rightarrow$		0	0	1	1	1
$b = 21 \rightarrow$		1	0	1	0	1
$d = 28 \rightarrow$		1	1	1	0	0

Full Adder Encoding

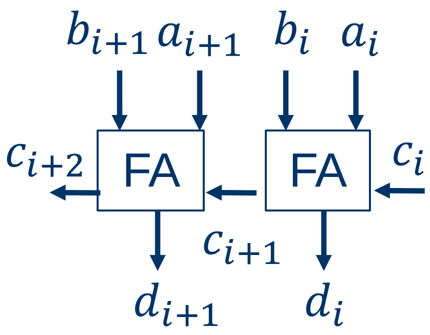
- $d_i = a_i + b_i + c_i \; mod \; 2, \qquad i = 0, \ldots, n-1$
  - $a_i \leftrightarrow b_i \leftrightarrow c_i \leftrightarrow d_i$
  - $d_i \leftrightarrow (a_i \wedge \overline{b_i} \wedge \overline{c_i}) \vee (\overline{a_i} \wedge b_i \wedge \overline{c_i}) \vee (\overline{a_i} \wedge \overline{b_i} \wedge c_i) \vee (a_i \wedge b_i \wedge c_i)$
- $c_{i+1} = 1 \leftrightarrow a_i + b_i + c_i > 1, \qquad i = 0, \ldots, n-1$
  - $c_{i+1} \leftrightarrow (a_i \wedge b_i) \vee (a_i \wedge c_i) \vee (b_i \wedge c_i)$
- $c_n = 0$ (to fit in n bits) and $c_0 = 0$ (initial carry)
  - $\overline{c_o} \wedge \overline{c_n}$

AtmostOne via Full Adder Encoding

- AtmostOne($[x_1, x_2]$) using one FA
  - a_i, b_i take the values of x_1, x_2
  - $\overline{c_i} \land \overline{c_{i+1}}$
- AtmostOne($[x_1, x_2, x_3]$) using one FA
  - a_i, b_i, c_i take the values of x_1, x_2, x_3
  - $\overline{c_{i+1}}$

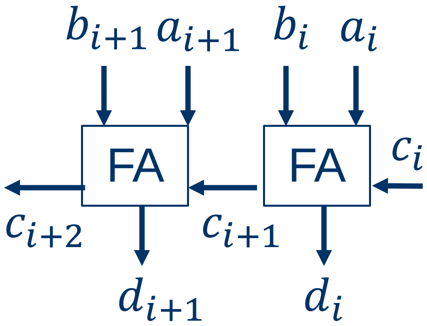


- AtmostOne($[x_1, x_2, x_3, x_4]$) using two FAs
  - $a_i, b_i, a_{i+1}, b_{i+1}$ take the values of x_1, x_2, x_3, x_4
  - $\overline{c_i} \land \overline{c_{i+1}} \land \overline{c_{i+2}} \land (\overline{d_{i+1}} \lor \overline{d_i})$
- AtmostOne($[x_1, x_2, x_3, x_4, x_5]$) using two FAs
  - $a_i, b_i, a_{i+1}, b_{i+1}, c_i$ take the values of x_1, x_2, x_3, x_4, x_5
  - $\overline{c_{i+1}} \land \overline{c_{i+2}} \land (\overline{d_{i+1}} \lor \overline{d_i})$



AtmostK via Full Adder Encoding

- AtmostK$([x_1, x_2, x_3, x_4], 2)$ using two FAs
  - $a_i, b_i, a_{i+1}, b_{i+1}$ take the values of x_1, x_2, x_3, x_4
  - $\overline{c_i} \wedge (c_{i+2} \rightarrow (\overline{d_{i+1}} \wedge \overline{d_i} \wedge \overline{c_{i+1}}))$



AtmostK via Full Adder Encoding

- Consider $x_1 + x_1 + \ x_3 \ \leq 0$.
  - Unit propagation should set $\overline{x_1}$, $\overline{x_2}$, and $\overline{x_3}$.
- Adder encoding using one FA:
  - a_i, b_i, c_i take the values of $\ x_1, x_2, x_3$
  - $\overline{d_i} \ \wedge \overline{c_{i+1}}$
  - $d_i \leftrightarrow (x_1 \wedge \overline{x_2} \wedge \overline{x_3}) \vee (\overline{x_1} \wedge x_2 \wedge \overline{x_3}) \ \vee (\overline{x_1} \wedge \overline{x_2} \wedge x_3) \vee (x_1 \wedge x_2 \wedge x_3)$
  - $c_{i+1} \ \leftrightarrow (x_1 \ \wedge x_2) \vee (x_1 \ \wedge x_3) \ \vee (x_2 \ \wedge x_3)$
- Note that:
  - $\overline{d_i} \ \rightarrow (\overline{x_1} \ \vee x_2 \vee x_3) \wedge (x_1 \vee \overline{x_2} \ \vee x_3) \wedge (x_1 \vee x_2 \vee \overline{x_3}) \wedge (\overline{x_1} \ \vee \overline{x_2} \vee \overline{x_3})$
  - $\overline{c_{i+1}} \rightarrow (\overline{x_1} \ \vee \overline{x_2}) \wedge (\overline{x_1} \vee \overline{x_3}) \ \wedge \ (\overline{x_2} \vee \overline{x_3})$
  - Unit propagation cannot propagate anything!
- Adder encoding is not an arc-consistent encoding!