Projection, Consistency, and George Boole

John Hooker
Carnegie Mellon University

CP 2015, Cork, Ireland

Projection as a Unifying Concept

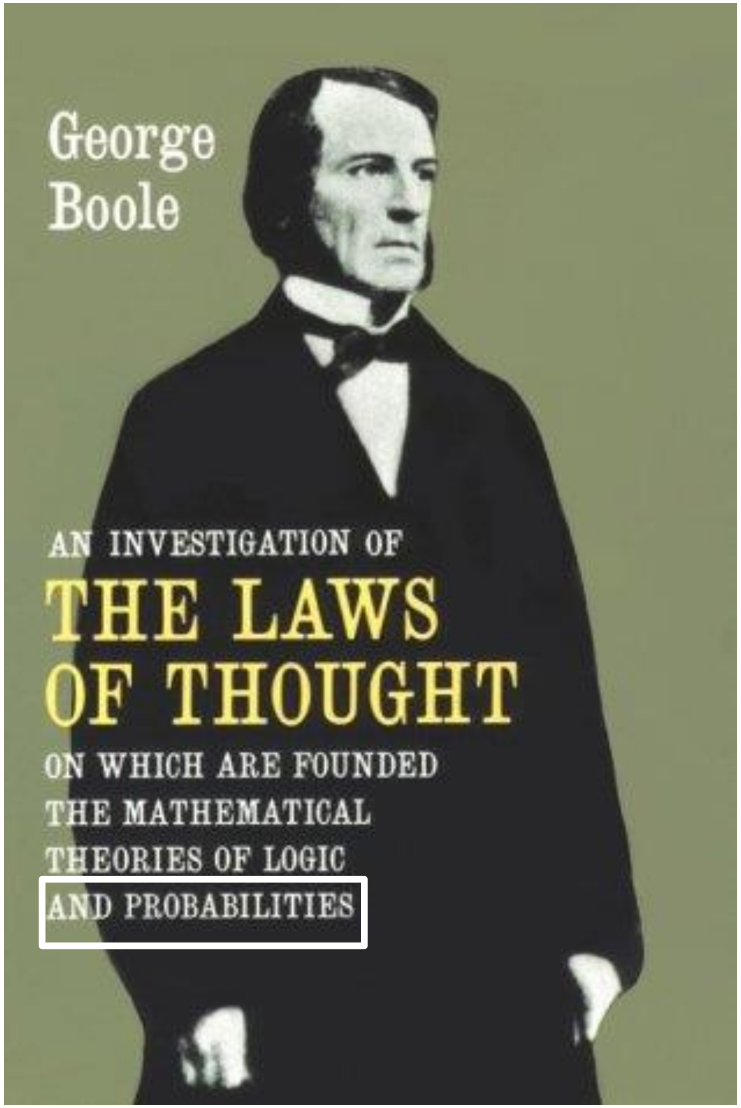
- **Projection** underlies both **optimization** and **logical inference**.
  - **Optimization** is **projection** onto a cost variable.
  - **Logical inference** is **projection** onto a subset of variables.
  - These 3 concepts are linked in **George Boole**'s work on probability logic.

Projection as a Unifying Concept

- **Projection** underlies both **optimization** and **logical inference**.
  - **Optimization** is **projection** onto a cost variable.
  - **Logical inference** is **projection** onto a subset of variables.
  - These 3 concepts are linked in **George Boole**'s work on probability logic.
- **Consistency maintenance** can likewise be seen as **projection**.
  - Leads to a simple type of consistency based explicitly on projection.

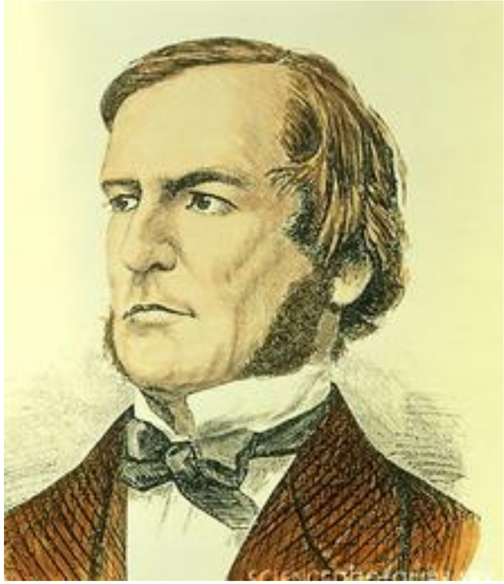
Probability Logic

- George Boole is best known for propositional logic and Boolean algebra.
  - But he proposed a highly original approach to **probability logic**.



Logical Inference

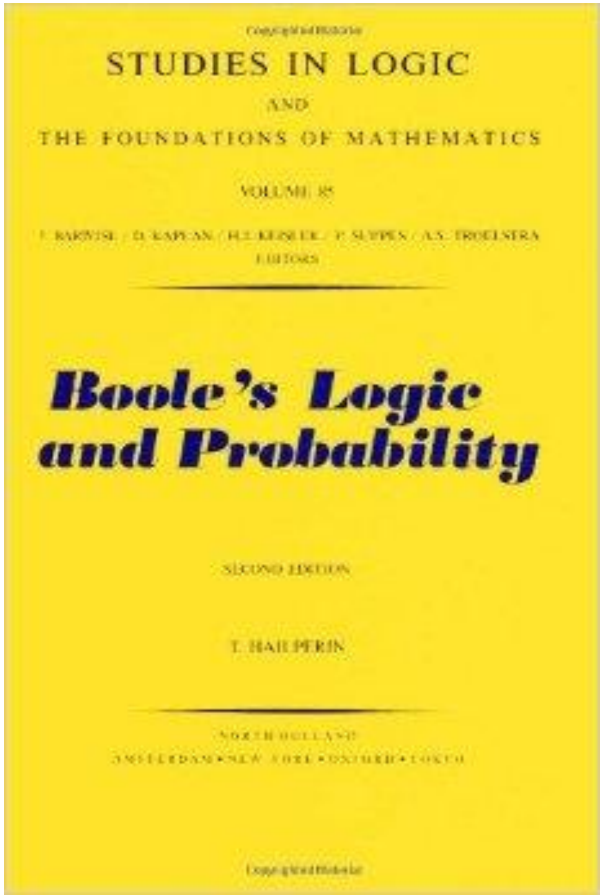
- The problem:
  - Given a set *S* of propositions
    - Each with a given probability.
  - And a proposition P that can be deduced from *S*…
  - What probability can be assigned to *P*?



Boole’s Probability Logic

- In 1970s, Theodore Hailperin offered an interpretation of Boole’s probability logic
  - …based on modern concept of **linear programming**.
  - LP first clearly formulated in 1930s by Kantorovich.

Hailperin (1976)



Boole's Probability Logic

Example

Clause	Probability
x_1	0.9
if x_1 then x_2	0.8
if x_2 then x_3	0.4

Deduce probability range for x_3

Boole's Probability Logic

Example

Clause	Probability
x_1	0.9
if x_1 then x_2	0.8
if x_2 then x_3	0.4

Interpret if-then statements as material conditionals

Deduce probability range for x_3

Boole's Probability Logic

Example

Clause	Probability
x_1	0.9
$\bar{x}_1 \vee x_2$	0.8
$\bar{x}_2 \vee x_3$	0.4

Interpret if-then statements as material conditionals

Deduce probability range for x_3

Boole's Probability Logic

Example

Clause	Probability
x_1	0.9
$\bar{x}_1 \vee x_2$	0.8
$\bar{x}_2 \vee x_3$	0.4

Deduce probability range for x_3

Linear programming model

$$\text{min/ max} \quad \pi_0$$

$$\begin{bmatrix} 01010101 \\ 00001111 \\ 11110011 \\ 11011101 \\ 11111111 \end{bmatrix} \begin{bmatrix} p_{000} \\ p_{001} \\ p_{010} \\ \vdots \\ p_{111} \end{bmatrix} = \begin{bmatrix} \pi_0 \\ 0.9 \\ 0.8 \\ 0.4 \\ 1 \end{bmatrix}$$

p_{000} = probability that $(x_1,x_2,x_3) = (0,0,0)$

Boole's Probability Logic

Example

Clause	Probability
x_1	0.9
$\bar{x}_1 \vee x_2$	0.8
$\bar{x}_2 \vee x_3$	0.4

Deduce probability range for x_3

Linear programming model

$$\min/\max \ \pi_0$$

$$\begin{bmatrix} 01010101 \\ 00001111 \\ 11110011 \\ 11011101 \\ 11111111 \end{bmatrix} \begin{bmatrix} p_{000} \\ p_{001} \\ p_{010} \\ \vdots \\ p_{111} \end{bmatrix} = \begin{bmatrix} \pi_0 \\ 0.9 \\ 0.8 \\ 0.4 \\ 1 \end{bmatrix}$$

p_{000} = probability that $(x_1,x_2,x_3) = (0,0,0)$

Solution: $\pi_0 \in [0.1, 0.4]$

Boole's Probability Logic

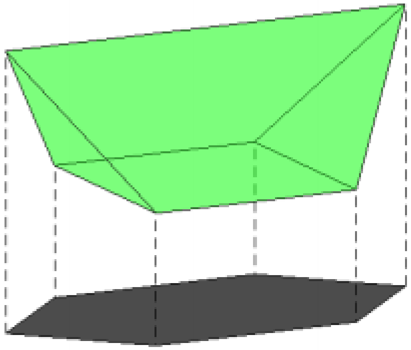
- This LP model was re-invented in AI community.

Nilsson (1986)

- Column generation methods are now available.
  - To deal with exponential number of variables.

Projection

- An LP can be solved by **Fourier elimination**
  - The only known method in Boole's day
  - This is a **projection** method.



Fourier (1827)

Projection

- Eliminate variables we want to project out.
  - To solve $$\min/\max\{y_0 \mid y_0 = ay, Ay \geq b\}$$
    project out all variables y_j except y_0.

Projection

- Eliminate variables we want to project out.
  - To solve $$\min/\max\{y_0 \mid y_0 = ay,\, Ay \geq b\}$$ project out all variables y_j except y_0.
  - To project out y_j, eliminate it from pairs of inequalities:

$$c\bar{y} + c_0 y_j \geq \gamma \qquad\qquad d\bar{y} - d_0 y_j \geq \delta$$

where $c_0, d_0 \geq 0$

Projection

- Eliminate variables we want to project out.
  - To solve $$\min/\max\left\{y_0 \mid y_0 = ay, Ay \ge b\right\}$$
    project out all variables y_j except y_0.
  - To project out y_j, eliminate it from pairs of inequalities:

$$c\bar{y} + c_0 y_j \ge \gamma \qquad\qquad d\bar{y} - d_0 y_j \ge \delta$$

$$-\frac{c}{c_0}\bar{y} + \frac{\gamma}{c_0} \le y_j \le \frac{d}{d_0}\bar{y} - \frac{\delta}{d_0}$$

Projection

- Eliminate variables we want to project out.
  - To solve $$\min/\max\left\{y_0 \mid y_0 = ay,\, Ay \ge b\right\}$$
    project out all variables y_j except y_0.
  - To project out y_j, eliminate it from pairs of inequalities:

$$c\bar{y} + c_0 y_j \ge \gamma \qquad\qquad d\bar{y} - d_0 y_j \ge \delta$$

$$-\frac{c}{c_0}\bar{y} + \frac{\gamma}{c_0} \le \cancel{y_j} \le \frac{d}{d_0}\bar{y} - \frac{\delta}{d_0}$$

Projection

- Eliminate variables we want to project out.
  - To solve
    $$\min/\max\{y_0 \mid y_0 = ay,\ Ay \geq b\}$$
    project out all variables y_j except y_0.
  - To project out y_j, eliminate it from pairs of inequalities:

$$c\bar{y} + c_0 y_j \geq \gamma \qquad\qquad d\bar{y} - d_0 y_j \geq \delta$$

$$-\frac{c}{c_0}\bar{y} + \frac{\gamma}{c_0} \leq \cancel{y_j} \leq \frac{d}{d_0}\bar{y} - \frac{\delta}{d_0}$$

$$\left(\frac{c}{c_0} + \frac{d}{d_0}\right)\bar{y} \geq \frac{\gamma}{c_0} + \frac{\delta}{d_0}$$

Projection

- Projection as a common theme:
  - **Optimization:** Project onto objective function variable
  - **Logical inference:** Project onto propositional variables of interest
  - **Consistency maintenance:** ???

- Look at logical inference next…

Inference as Projection

- Project onto propositional variables of interest
  - Suppose we wish to infer from these clauses everything we can about propositions x_1, x_2, x_3

$$
\begin{array}{llllll}
x_1 & & & \vee\, x_4 & \vee\, x_5 & \\
x_1 & & & \vee\, x_4 & \vee\, \bar{x}_5 & \\
x_1 & & & & \vee\, x_5 & \vee\, x_6 \\
x_1 & & & & \vee\, x_5 & \vee\, \bar{x}_6 \\
 & x_2 & & & \vee\, \bar{x}_5 & \vee\, x_6 \\
 & x_2 & & & \vee\, \bar{x}_5 & \vee\, \bar{x}_6 \\
 & & x_3 & \vee\, \bar{x}_4 & \vee\, x_5 & \\
 & & x_3 & \vee\, \bar{x}_4 & \vee\, \bar{x}_5 &
\end{array}
$$

Inference as Projection

- Project onto propositional variables of interest
  - Suppose we wish to infer from these clauses everything we can about propositions x_1, x_2, x_3

$$
\begin{array}{lllllll}
x_1 & & & \vee\, x_4 & \vee\, x_5 & & \\
x_1 & & & \vee\, x_4 & \vee\, \bar{x}_5 & & \\
x_1 & & & & \vee\, x_5 & \vee\, x_6 & \\
x_1 & & & & \vee\, x_5 & \vee\, \bar{x}_6 & \\
 & x_2 & & & \vee\, \bar{x}_5 & \vee\, x_6 & \\
 & x_2 & & & \vee\, \bar{x}_5 & \vee\, \bar{x}_6 & \\
 & & x_3 & \vee\, \bar{x}_4 & \vee\, x_5 & & \\
 & & x_3 & \vee\, \bar{x}_4 & \vee\, \bar{x}_5 & &
\end{array}
$$

We can deduce

$$x_1 \vee x_2$$

$$x_1 \vee x_3$$

This is a projection onto x_1, x_2, x_3

Inference as Projection

- Resolution as a projection method
  - Similar to Fourier elimination
    - Actually, identical to Fourier elimination + rounding
  - To project out x_j, eliminate it from pairs of clauses:

$$
\begin{tikzcd}
C \vee x_j \arrow[dr] & & D \vee \bar{x}_j \arrow[dl] \\
& C \vee D &
\end{tikzcd}
$$

Quine (1952,1955)
JH (1992,2012)

Inference as Projection

- Resolution as a projection method
  - Similar to Fourier elimination
    - Actually, identical to Fourier elimination + rounding
  - To project out x_j, eliminate it from pairs of clauses:

$$\begin{tikzcd}
C \vee x_j \arrow[dr] & & D \vee \bar{x}_j \arrow[dl] \\
& C \vee D &
\end{tikzcd}$$

  - This is **too slow**.
  - Another approach is logic-based Benders decomposition…

Inference as Projection

- Benders decomposition computes a projection
  - Benders cuts describe projection onto master problem variables.

Current
Master problem

$$\boxed{x_1 \vee x_2}$$

Benders cut
from previous
iteration

Inference as Projection

- Benders decomposition computes a projection
  - Benders cuts describe projection onto master problem variables.

Current Master problem

$x_1 \vee x_2$

solution of master
$(x_1, x_2, x_3) = (0,1,0)$

Resulting subproblem

$$x_4 \vee x_5$$
$$x_4 \vee \bar{x}_5$$
$$x_5 \vee x_6$$
$$x_5 \vee \bar{x}_6$$
$$\bar{x}_4 \vee x_5$$
$$\bar{x}_4 \vee \bar{x}_5$$

Inference as Projection

- Benders decomposition computes a projection
  - Benders cuts describe projection onto master problem variables.

Current Master problem

$x_1 \vee x_2$

solution of master

$(x_1,x_2,x_3) = (0,1,0)$

Resulting subproblem

$$x_4 \vee x_5$$
$$x_4 \vee \bar{x}_5$$
$$x_5 \vee x_6$$
$$x_5 \vee \bar{x}_6$$
$$\bar{x}_4 \vee x_5$$
$$\bar{x}_4 \vee \bar{x}_5$$

Subproblem is **infeasible**.
$(x_1,x_3)=(0,0)$
creates infeasibility

Inference as Projection

- Benders decomposition computes a projection
  - Benders cuts describe projection onto master problem variables.

Current Master problem

$x_1 \vee x_2$

$x_1 \vee x_3$

solution of master $(x_1, x_2, x_3) = (0,1,0)$ →

← Benders cut (nogood)

Resulting subproblem

$$
\begin{array}{l}
x_4 \vee x_5 \\
x_4 \vee \bar{x}_5 \\
x_5 \vee x_6 \\
x_5 \vee \bar{x}_6 \\
\bar{x}_4 \vee x_5 \\
\bar{x}_4 \vee \bar{x}_5
\end{array}
$$

Subproblem is **infeasible**.
$(x_1, x_3) = (0,0)$
creates infeasibility

Inference as Projection

- Benders decomposition computes a projection
  - Benders cuts describe projection onto master problem variables.

Current Master problem

$x_1 \vee x_2$

$x_1 \vee x_3$

→ solution of master $(x_1,x_2,x_3) = (0,1,1)$ →

Resulting subproblem

$x_4 \vee x_5$

$x_4 \vee \bar{x}_5$

$x_5 \vee x_6$

$x_5 \vee \bar{x}_6$

Inference as Projection

- Benders decomposition computes a projection
  - Benders cuts describe projection onto master problem variables.

Current Master problem

$x_1 \vee x_2$

$x_1 \vee x_3$

solution of master
$(x_1,x_2,x_3) = (0,1,1)$

Resulting subproblem

$x_4 \vee x_5$

$x_4 \vee \bar{x}_5$

$x_5 \vee x_6$

$x_5 \vee \bar{x}_6$

Subproblem is **feasible**

Inference as Projection

- Benders decomposition computes a projection
  - Benders cuts describe projection onto master problem variables.

Current
Master problem

$$x_1 \vee x_2$$
$$x_1 \vee x_3$$
$$x_1 \vee \bar{x}_2 \vee \bar{x}_3$$

solution of master
$(x_1, x_2, x_3) = (0,1,1)$

Enumerative
Benders cut

Resulting
subproblem

$$x_4 \vee x_5$$
$$x_4 \vee \bar{x}_5$$
$$x_5 \vee x_6$$
$$x_5 \vee \bar{x}_6$$

Subproblem is
feasible

Inference as Projection

- Benders decomposition computes a projection
  - Logic-based Benders cuts describe projection onto master problem variables.

Current Master problem

$x_1 \vee x_2$

$x_1 \vee x_3$

$x_1 \vee \bar{x}_2 \vee \bar{x}_3$

solution of master $(x_1,x_2,x_3) = (0,1,1)$

Enumerative Benders cut

Resulting subproblem

$x_4 \vee x_5$

$x_4 \vee \bar{x}_5$

$x_5 \vee x_6$

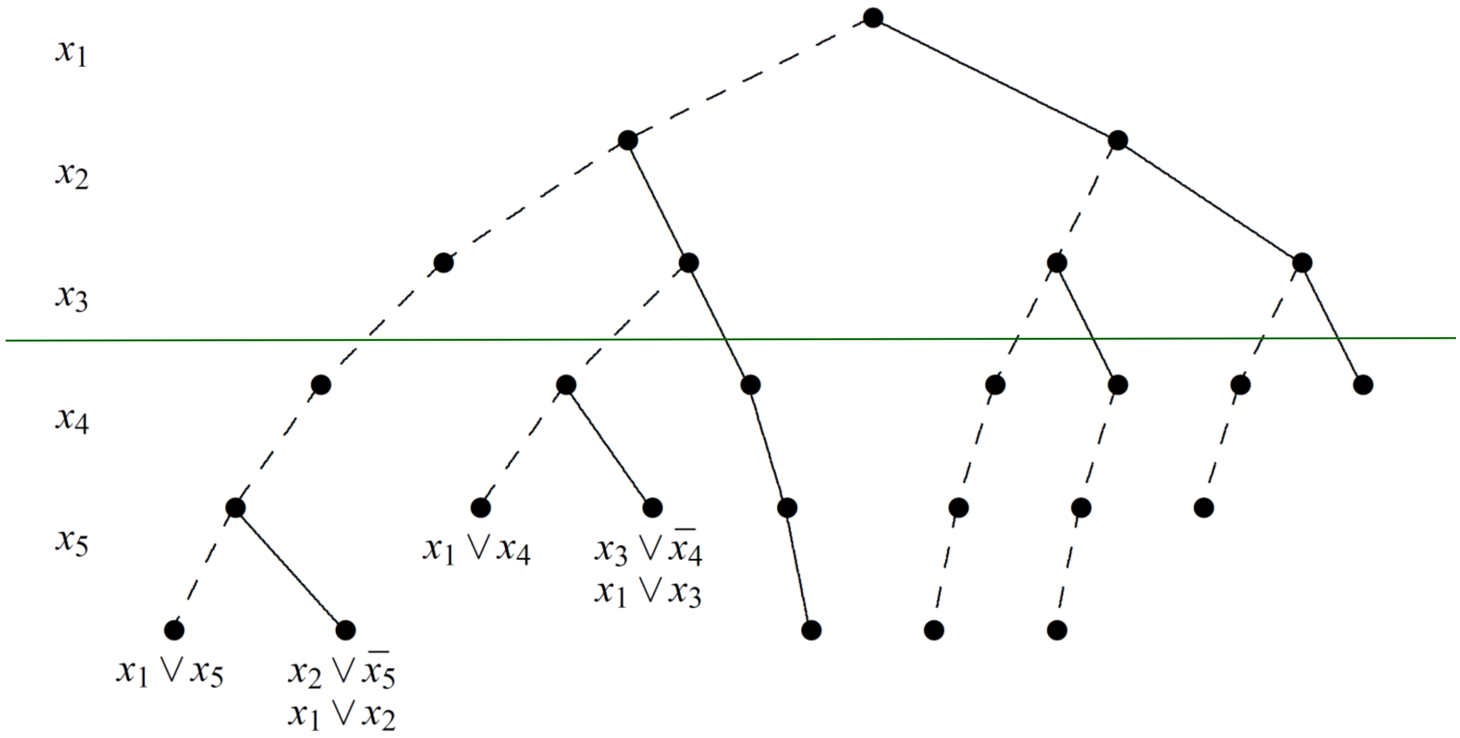
$x_5 \vee \bar{x}_6$

Continue until **master** is infeasible. Black Benders cuts describe projection.

JH and Yan (1995)
JH (2012)

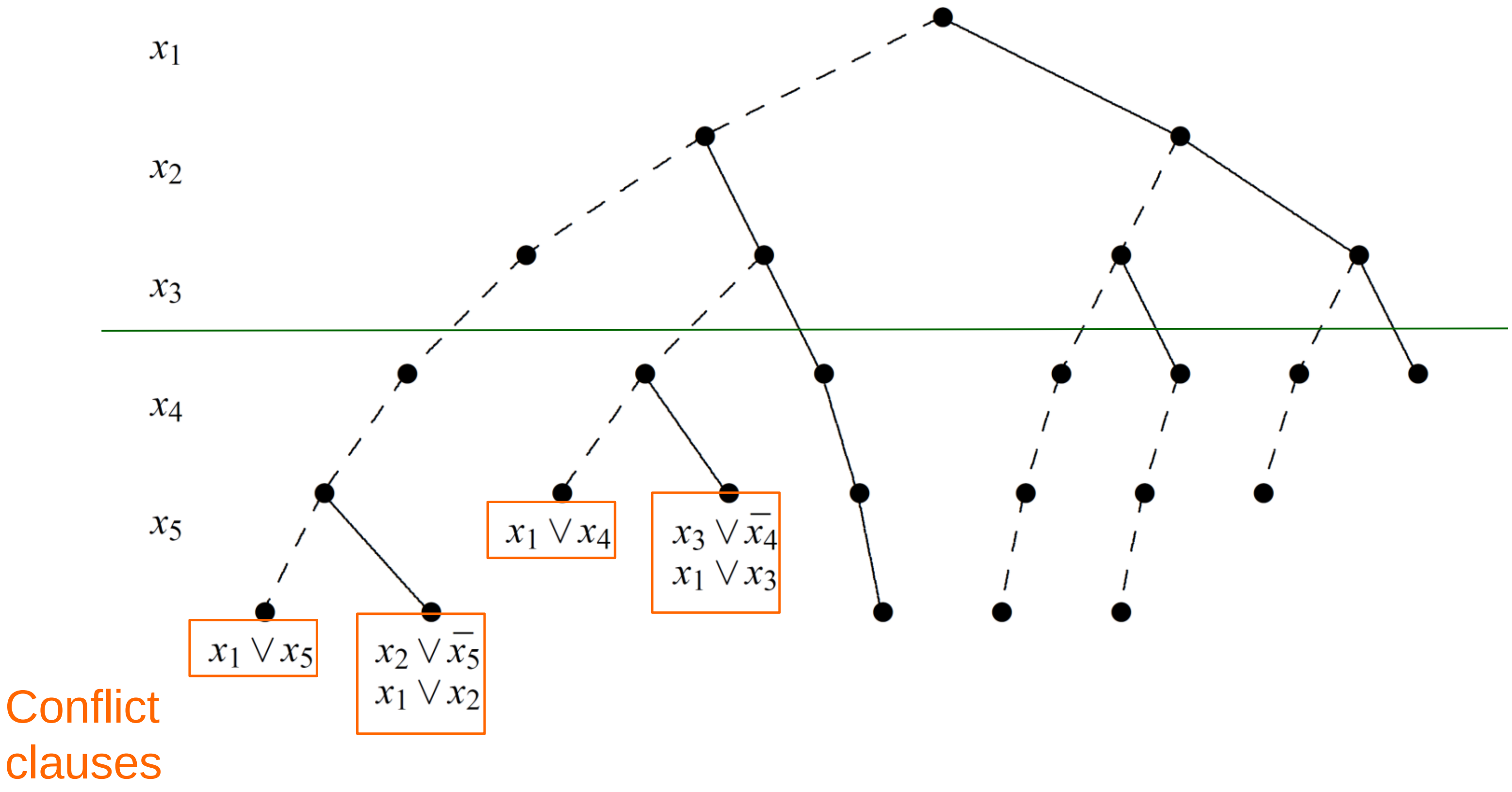
Inference as Projection

- Benders cuts = conflict clauses in a SAT algorithm
  - Branch on x_1, x_2 first.



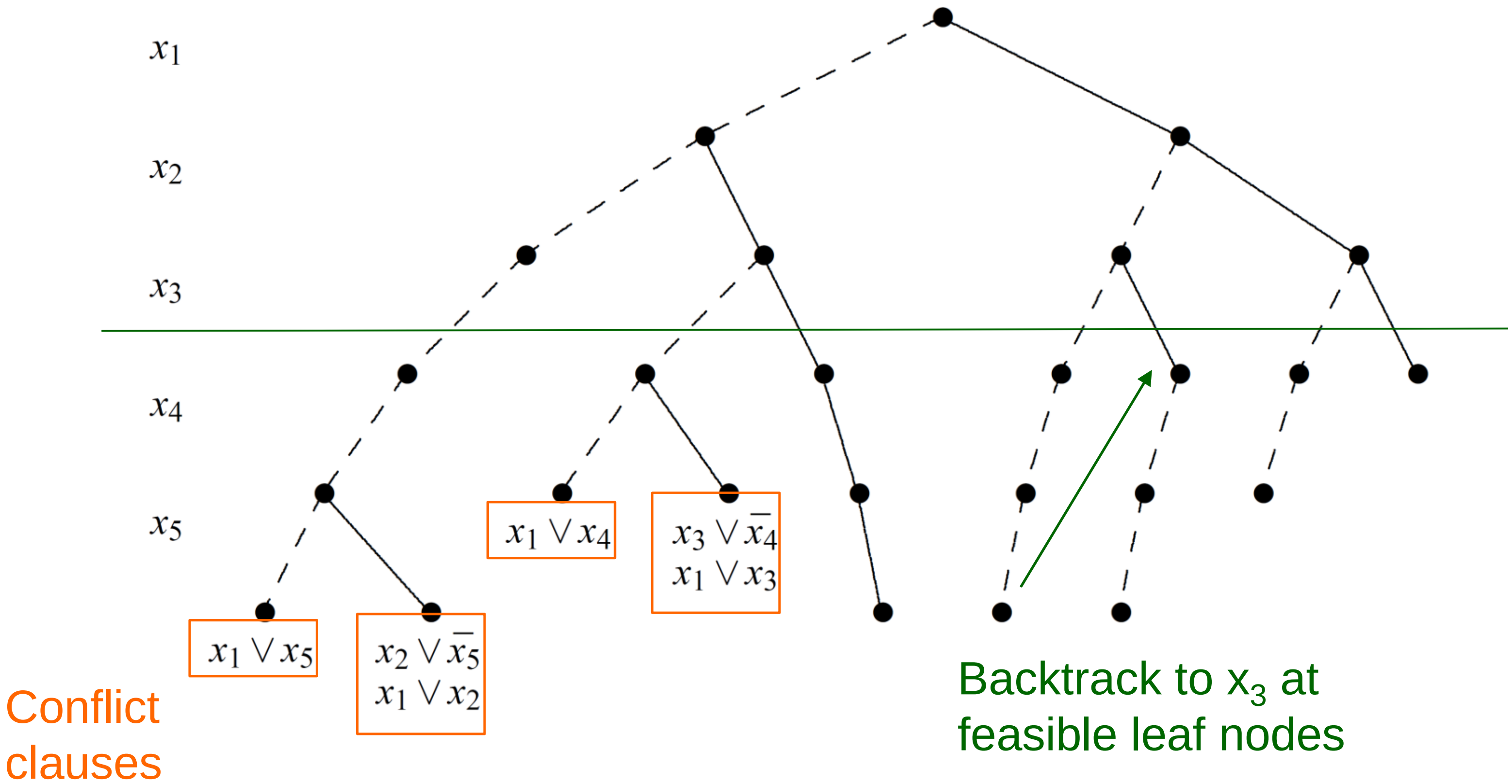
Inference as Projection

- Benders cuts = conflict clauses in a SAT algorithm
  - Branch on x_1, x_2 first.



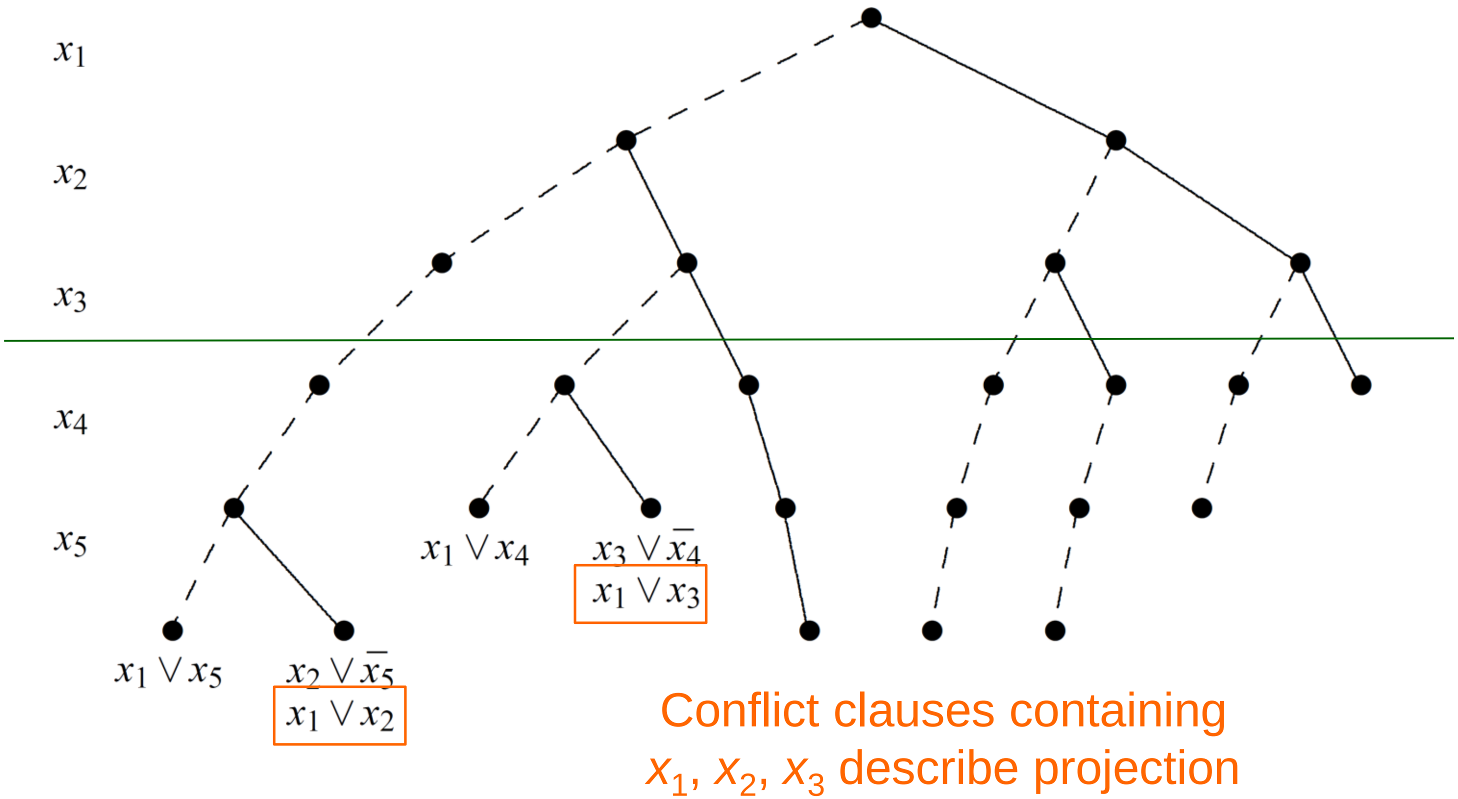
Inference as Projection

- Benders cuts = conflict clauses in a SAT algorithm
  - Branch on x_1, x_2 first.



Inference as Projection

- Benders cuts = conflict clauses in a SAT algorithm
  - Branch on x_1, x_2 first.



Inference as Projection

- Projection methods similar to Fourier elimination
  - For logical clauses Quine (1952,1955)
  - For cardinality clauses JH (1988)
  - For 0-1 linear inequalities JH (1992)
  - For general integer linear inequalities Williams & JH (2015)

Consistency as Projection

- Domain consistency
  - Project onto each individual variable x_j.

Consistency as Projection

- Domain consistency
  - Project onto each individual variable x_j.

Example:

Constraint set

$$\text{alldiff}\left(x_1, x_2, x_3\right)$$
$$x_1 \in \{a, b\}$$
$$x_2 \in \{a, b\}$$
$$x_3 \in \{b, c\}$$

Consistency as Projection

- Domain consistency
  - Project onto each individual variable x_j.

Example:

Constraint set	Solutions
$\text{alldiff}(x_1, x_2, x_3)$	(x_1, x_2, x_3)
$x_1 \in \{a, b\}$	(a, b, c)
$x_2 \in \{a, b\}$	(b, a, c)
$x_3 \in \{b, c\}$	

Consistency as Projection

- Domain consistency
  - Project onto each individual variable x_j.

Example:

Constraint set

$$\text{alldiff}(x_1, x_2, x_3)$$
$$x_1 \in \{a, b\}$$
$$x_2 \in \{a, b\}$$
$$x_3 \in \{b, c\}$$

Solutions

$$(x_1, x_2, x_3)$$
$$(a, b, c)$$
$$(b, a, c)$$

Projection onto x_1

$$x_1 \in \{a, b\}$$

Projection onto x_2

$$x_2 \in \{a, b\}$$

Projection onto x_3

$$x_3 \in \{c\}$$

Consistency as Projection

- Domain consistency
  - Project onto each individual variable x_j.

Example:

Constraint set	Solutions	Projection onto x_1
$\text{alldiff}(x_1, x_2, x_3)$	(x_1, x_2, x_3)	$x_1 \in \{a, b\}$
$x_1 \in \{a, b\}$	(a, b, c)	Projection onto x_2
$x_2 \in \{a, b\}$	(b, a, c)	$x_2 \in \{a, b\}$
$x_3 \in \{b, c\}$		Projection onto x_3
		$x_3 \in \{c\}$

This achieves domain consistency.

Consistency as Projection

- Domain consistency
  - Project onto each individual variable x_j.

Example:

Constraint set	Solutions	Projection onto x_1
$\text{alldiff}(x_1, x_2, x_3)$	(x_1, x_2, x_3)	$x_1 \in \{a,b\}$
$x_1 \in \{a,b\}$	(a,b,c)	Projection onto x_2
$x_2 \in \{a,b\}$	(b,a,c)	$x_2 \in \{a,b\}$
$x_3 \in \{b,c\}$		Projection onto x_3
		$x_3 \in \{c\}$

This achieves domain consistency.

We will regard a projection as a **constraint set**.

Consistency as Projection

$x_J = (x_j \mid j \in J)$

- *k*-consistency
  - Can be defined:
    - A constraint set S is k-consistent if:
      - for every $J \subseteq \{1, \ldots, n\}$ with $|J| = k - 1$,
      - every assignment $x_J = v_J \in D_j$ for which (x_J, x_j) does not violate S,
      - and every variable $x_j \notin x_J$,

    there is an assignment $x_j = v_j \in D_j$ for which $(x_J, x_j) = (v_J, v_j)$ does not violate S.

Consistency as Projection

$x_J = (x_j \mid j \in J)$

- *k*-consistency
  - Can be defined:
    - A constraint set S is k-consistent if:
      - for every $J \subseteq \{1, \ldots, n\}$ with $|J| = k - 1$,
      - every assignment $x_J = v_J \in D_j$ for which (x_J, x_j) does not violate S,
      - and every variable $x_j \notin x_J$,

    there is an assignment $x_j = v_j \in D_j$ for which $(x_J, x_j) = (v_J, v_j)$ does not violate S.
  - To achieve k-consistency:
    - Project the constraints containing each set of k variables onto subsets of $k - 1$ variables.

Consistency as Projection

- Consistency and backtracking:
  - Strong *k*-consistency for entire constraint set avoids backtracking…
    - if the primal graph has width < *k* with respect to branching order.

Freuder (1982)

  - No point in achieving strong *k*-consistency for **individual constraints** if we propagate through **domain store**.
    - Domain consistency has same effect.

J-Consistency

- A type of consistency more directly related to projection.
  - Constraint set S is ***J*-consistent** if it contains the **projection** of S onto x_J.
    - S is domain consistent if it is $\{ j \}$-consistent for each j.
    - Resolution and logic-based Benders achieve J-consistency for SAT.

$$\boxed{x_J = (x_j \mid j \in J)}$$

J-Consistency

- *J*-consistency and backtracking:
  - If we branch on variables x_1, x_2, …, a natural strategy is to project out x_n, x_{n-1}, …
    - until computational burden is excessive.

J-Consistency

- *J*-consistency and backtracking:
  - If we branch on variables x_1, x_2, …, a natural strategy is to project out x_n, x_{n-1}, …
    - until computational burden is excessive.
  - No point in achieving *J*-consistency for **individual constraints** if we propagate through a **domain store**.
    - However, *J*-consistency can be useful if we propagate through a richer data structure
    - …such as **decision diagrams**
    - …which can be more effective as a propagation medium.

Andersen, Hadžić JH, Tiedemann (2007)
Bergman, Ciré, van Hoeve, JH (2014)

Propagating *J*-Consistency

Example:

$$\text{among}\left((x_1,x_2),\{c,d\},1,2\right)$$
$$(x_1 = c) \Rightarrow (x_2 = d)$$
$$\text{alldiff}\left(x_1,x_2,x_3,x_4\right)$$
$$x_1,x_2 \in \{a,b,c,d\}$$
$$x_3 \in \{a,b\}$$
$$x_4 \in \{c,d\}$$

Already domain consistent for individual constraints.

If we branch on x_1 first, must consider all 4 branches $x_1 = a,b,c,d$

Propagating *J*-Consistency

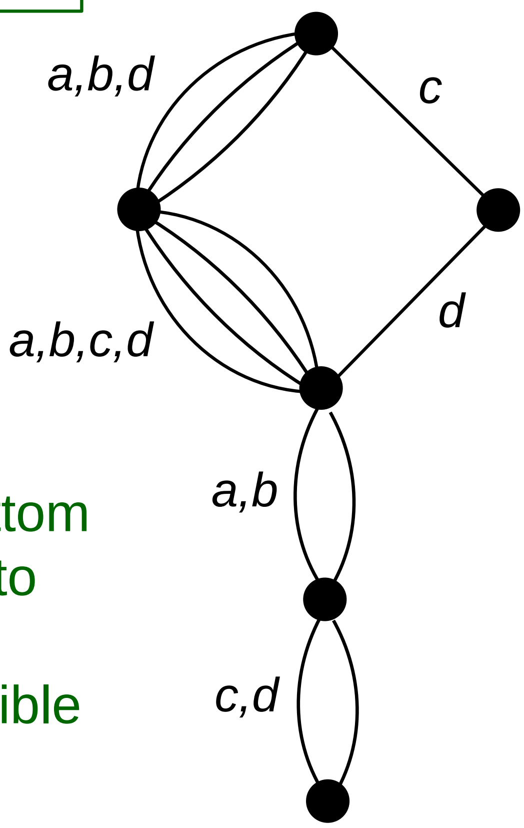
Example:

$$\text{among}\big((x_1,x_2),\{c,d\},1,2\big)$$
$$(x_1 = c) \Rightarrow (x_2 = d)$$

$$\text{alldiff}\,(x_1,x_2,x_3,x_4)$$

$$x_1,x_2 \in \{a,b,c,d\}$$
$$x_3 \in \{a,b\}$$
$$x_4 \in \{c,d\}$$

Suppose we propagate through a relaxed decision diagram of width 2 for these constraints



52 paths from top to bottom represent assignments to x_1, x_2, x_3, x_4

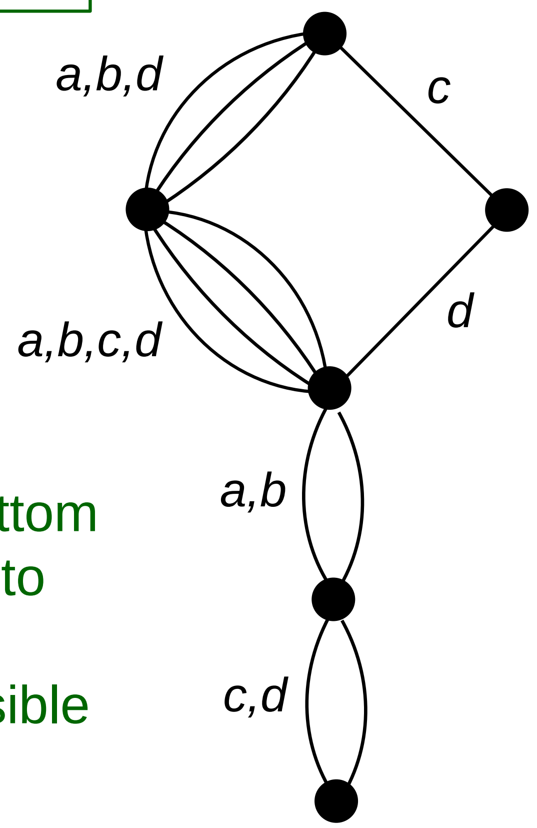
36 of these are the feasible assignments.

Propagating *J*-Consistency

Example:

$$\text{among}\big((x_1,x_2),\{c,d\},1,2\big)$$
$$(x_1 = c) \Rightarrow (x_2 = d)$$
$$\text{alldiff}\,(x_1,x_2,x_3,x_4)$$
$$x_1,x_2 \in \{a,b,c,d\}$$
$$x_3 \in \{a,b\}$$
$$x_4 \in \{c,d\}$$

Suppose we propagate through a relaxed decision diagram of width 2 for these constraints



52 paths from top to bottom represent assignments to x_1, x_2, x_3, x_4
36 of these are the feasible assignments.

Projection of alldiff onto x_1, x_2 is

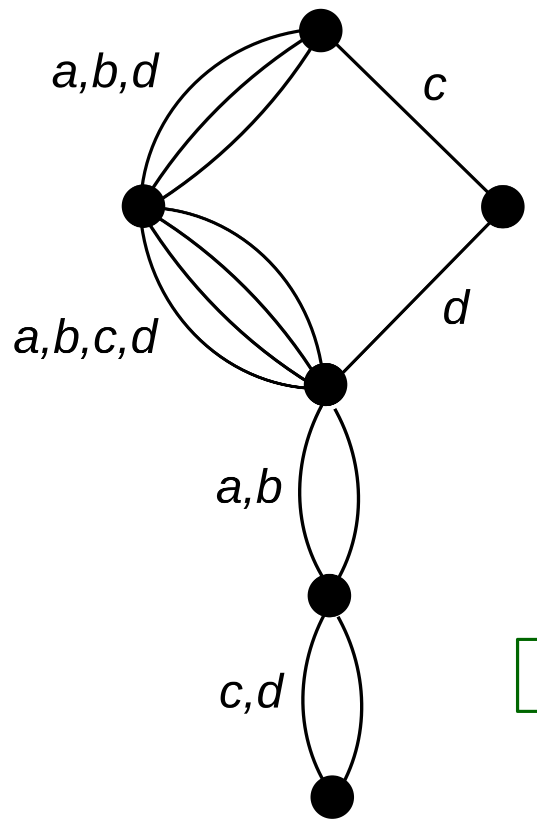
$$\text{alldiff}\,(x_1,x_2)$$
$$\text{atmost}\big((x_1,x_2),\{a,b\},1\big)$$
$$\text{atmost}\big((x_1,x_2),\{c,d\},1\big)$$

Propagating *J*-Consistency

Let's propagate the 2nd atmost constraint in the projected alldiff through the relaxed decision diagram.

Let the length of a path be number of arcs with labels in {*c*,*d*}.

For each arc, indicate length of shortest path from top to that arc.



Projection of alldiff onto x_1, x_2 is

$$\text{alldiff}(x_1, x_2)$$

$$\text{atmost}((x_1, x_2), \{a, b\}, 1)$$

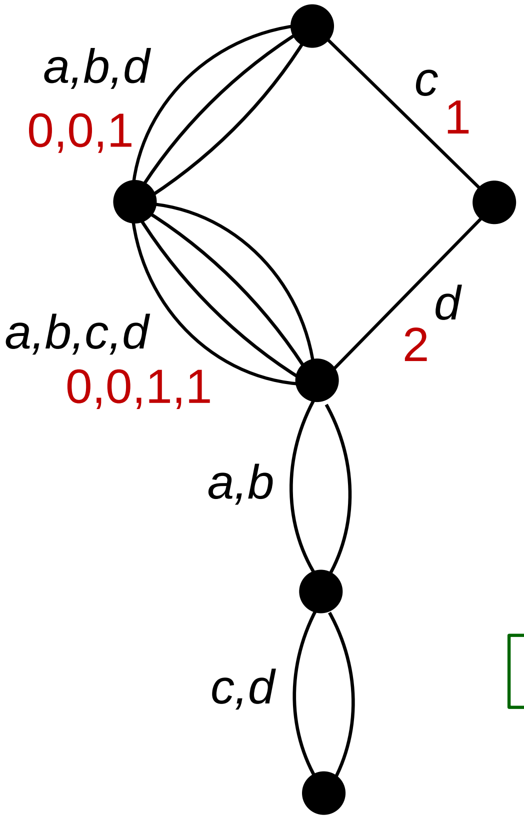
$$\boxed{\text{atmost}((x_1, x_2), \{\text{c}, \text{d}\}, 1)}$$

Propagating *J*-Consistency

Let's propagate the 2nd atmost constraint in the projected alldiff through the relaxed decision diagram.

Let the length of a path be number of arcs with labels in $\{c,d\}$.

For each arc, indicate length of shortest path from top to that arc.



Projection of alldiff onto x_1, x_2 is

$$\text{alldiff}\left(x_1, x_2\right)$$

$$\text{atmost}\left((x_1, x_2), \{a, b\}, 1\right)$$

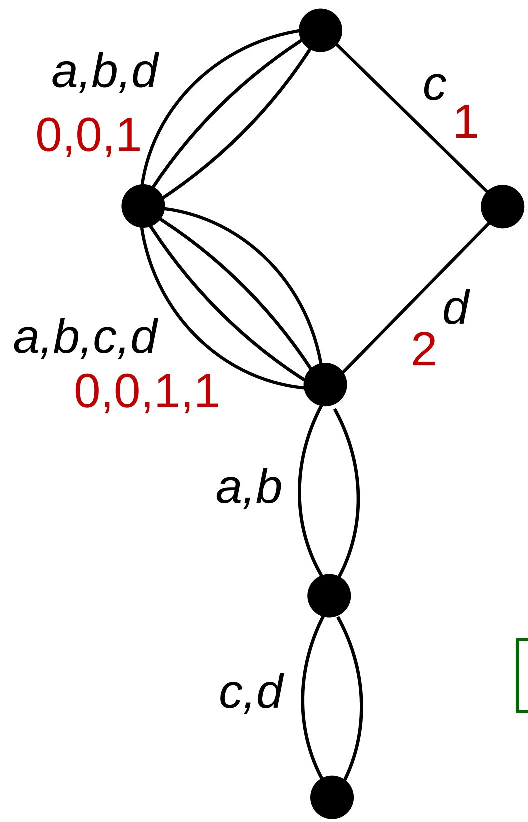
$$\boxed{\text{atmost}\left((x_1, x_2), \{\text{c}, \text{d}\}, 1\right)}$$

Propagating *J*-Consistency

Let's propagate the 2nd atmost constraint in the projected alldiff through the relaxed decision diagram.

Let the length of a path be number of arcs with labels in $\{c,d\}$.

For each arc, indicate length of shortest path from top to that arc.



Remove arcs with label > 1

Projection of alldiff onto x_1, x_2 is

$$\text{alldiff}\left(x_1, x_2\right)$$

$$\text{atmost}\left((x_1, x_2), \{a, b\}, 1\right)$$

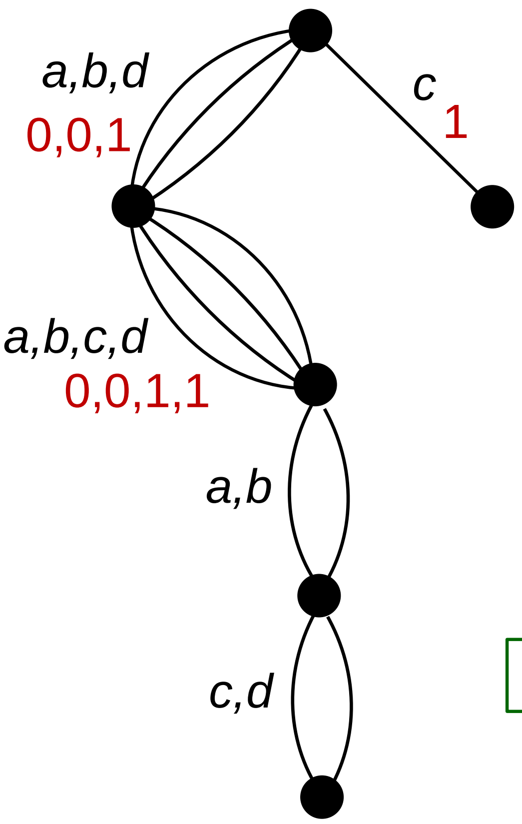
$$\boxed{\text{atmost}\left((x_1, x_2), \{\text{c}, \text{d}\}, 1\right)}$$

Propagating *J*-Consistency

Let's propagate the 2nd atmost constraint in the projected alldiff through the relaxed decision diagram.

Let the length of a path be number of arcs with labels in $\{c,d\}$.

For each arc, indicate length of shortest path from top to that arc.



Remove arcs with label > 1

Projection of alldiff onto x_1, x_2 is

$$\text{alldiff}\left(x_1, x_2\right)$$

$$\text{atmost}\left((x_1, x_2), \{a, b\}, 1\right)$$

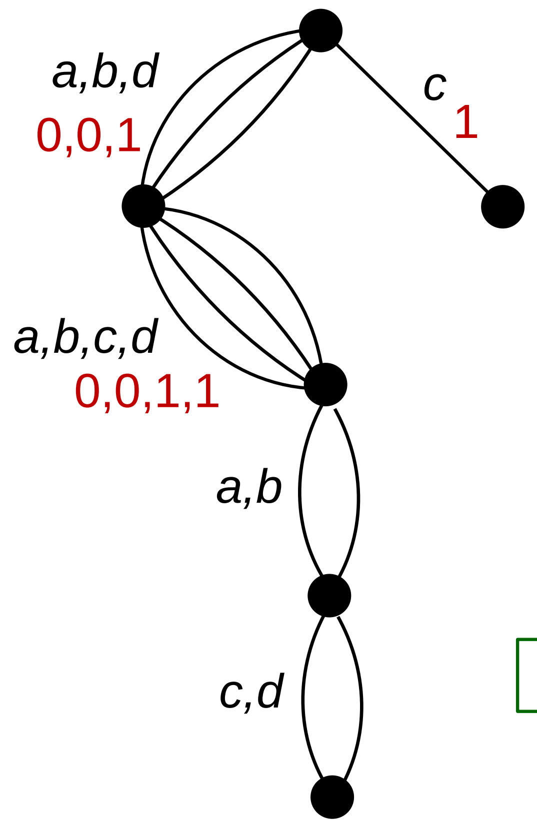
$$\boxed{\text{atmost}\left((x_1, x_2), \{\text{c}, \text{d}\}, 1\right)}$$

Propagating *J*-Consistency

Let's propagate the 2nd atmost constraint in the projected alldiff through the relaxed decision diagram.

Let the length of a path be number of arcs with labels in {*c*,*d*}.

For each arc, indicate length of shortest path from top to that arc.



Remove arcs with label > 1

Clean up.

Projection of alldiff onto x_1, x_2 is

$$\text{alldiff}(x_1, x_2)$$

$$\text{atmost}((x_1, x_2), \{a, b\}, 1)$$

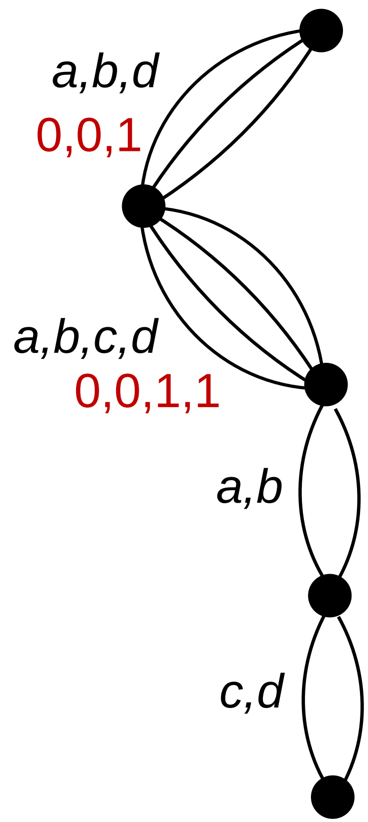
$$\boxed{\text{atmost}((x_1, x_2), \{\text{c}, \text{d}\}, 1)}$$

Propagating *J*-Consistency

Let's propagate the 2nd atmost constraint in the projected alldiff through the relaxed decision diagram.

Let the length of a path be number of arcs with labels in $\{c,d\}$.

For each arc, indicate length of shortest path from top to that arc.



Remove arcs with label > 1

Clean up.

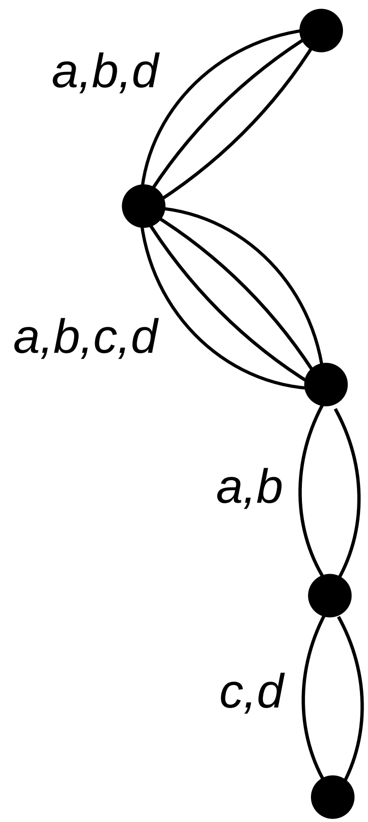
Projection of alldiff onto x_1, x_2 is

$$\text{alldiff}(x_1, x_2)$$
$$\text{atmost}((x_1, x_2), \{a, b\}, 1)$$
$$\boxed{\text{atmost}((x_1, x_2), \{\text{c}, \text{d}\}, 1)}$$

Propagating *J*-Consistency

Let's propagate the 2nd atmost constraint in the projected alldiff through the relaxed decision diagram.

We need only branch on *a,b,d* rather than *a,b,c,d*



Remove arcs with label > 1

Clean up.

Projection of alldiff onto x_1, x_2 is

$$\text{alldiff}(x_1, x_2)$$

$$\text{atmost}\left((x_1, x_2), \{a, b\}, 1\right)$$

$$\boxed{\text{atmost}\left((x_1, x_2), \{\text{c}, \text{d}\}, 1\right)}$$

Achieving *J*-consistency

Constraint	**How hard to project?**
among	Easy and fast.
sequence	More complicated but fast.
regular	Easy and basically same labor as domain consistency.
alldiff	Quite complicated but practical for small domains.

J-consistency for Among

Projection of $\mathrm{among}\big((x_1,\ldots,x_n),V,t,u\big)$ onto $x_1,\ldots,x_{n-1}$ is

$$\mathrm{among}\big((x_1,\ldots,x_{n-1}),V,t',u'\big)$$

where

$$(t',u') = \begin{cases} \big((t-1)^+, u-1\big) & \text{if } D_n \subseteq V \\ \big(t, \min\{u, n-1\}\big) & \text{if } D_n \cap V = \emptyset \\ \big((t-1)^+, \min\{u, n-1\}\big) & \text{otherwise} \end{cases}$$

J-consistency for Among

Projection of $\text{among}\big((x_1,\ldots,x_n),V,t,u\big)$ onto $x_1,\ldots,x_{n-1}$ is

$$\text{among}\big((x_1,\ldots,x_{n-1}),V,t',u'\big)$$

where

$$(t',u') = \begin{Bmatrix} \big((t-1)^+, u-1\big) & \text{if } D_n \subseteq V \\ \big(t, \min\{u, n-1\}\big) & \text{if } D_n \cap V = \varnothing \\ \big((t-1)^+, \min\{u, n-1\}\big) & \text{otherwise} \end{Bmatrix}$$

Example

$$\text{among}\big((x_1,x_2,x_3,x_4,x_5),\{c,d\},t,u\big)$$

$$D_1 = \{a,b\}$$
$$D_2 = \{a,b,c\}$$
$$D_3 = \{a,\mathrm{d}\}$$
$$D_4 = \{c,d\}$$
$$D_5 = \{d\}$$

J-consistency for Among

Projection of $\text{among}\big((x_1,\ldots,x_n),V,t,u\big)$ onto $x_1,\ldots,x_{n-1}$ is

$$\text{among}\big((x_1,\ldots,x_{n-1}),V,t',u'\big)$$

where

$$(t',u')=\begin{cases}\big((t-1)^+,u-1\big) & \text{if } D_n\subseteq V\\ \big(t,\min\{u,n-1\}\big) & \text{if } D_n\cap V=\varnothing\\ \big((t-1)^+,\min\{u,n-1\}\big) & \text{otherwise}\end{cases}$$

Example

$$\text{among}\big((x_1,x_2,x_3,x_4,x_5),\{c,d\},t,u\big)$$

$$\text{among}\big((x_1,x_2,x_3,x_4),\{c,d\},(t-1)^+,u-1\big)$$

$$D_1=\{a,b\}$$
$$D_2=\{a,b,c\}$$
$$D_3=\{a,\text{d}\}$$
$$D_4=\{c,d\}$$
$$D_5=\{d\}$$

J-consistency for Among

Projection of $\text{among}\big((x_1,\ldots,x_n),V,t,u\big)$ onto $x_1,\ldots,x_{n-1}$ is

$$\text{among}\big((x_1,\ldots,x_{n-1}),V,t',u'\big)$$

where

$$(t',u') = \begin{cases} \big((t-1)^+, u-1\big) & \text{if } D_n \subseteq V \\ \big(t, \min\{u, n-1\}\big) & \text{if } D_n \cap V = \varnothing \\ \big((t-1)^+, \min\{u, n-1\}\big) & \text{otherwise} \end{cases}$$

Example

$$D_1 = \{a,b\}$$
$$D_2 = \{a,b,c\}$$
$$D_3 = \{a,\text{d}\}$$
$$D_4 = \{c,d\}$$
$$D_5 = \{d\}$$

$$\text{among}\big((x_1,x_2,x_3,x_4,x_5),\{c,d\},t,u\big)$$
$$\text{among}\big((x_1,x_2,x_3,x_4),\{c,d\},(t-1)^+,u-1\big)$$
$$\text{among}\big((x_1,x_2,x_3),\{c,d\},(t-2)^+,u-2\big)$$

J-consistency for Among

Projection of $\text{among}\big((x_1,\ldots,x_n),V,t,u\big)$ onto $x_1,\ldots,x_{n-1}$ is

$$\text{among}\big((x_1,\ldots,x_{n-1}),V,t',u'\big)$$

where

$$(t',u') = \begin{cases} \big((t-1)^+, u-1\big) & \text{if } D_n \subseteq V \\ \big(t, \min\{u, n-1\}\big) & \text{if } D_n \cap V = \varnothing \\ \big((t-1)^+, \min\{u, n-1\}\big) & \text{otherwise} \end{cases}$$

Example

$$D_1 = \{a,b\}$$
$$D_2 = \{a,b,c\}$$
$$D_3 = \{a,\mathrm{d}\}$$
$$D_4 = \{c,d\}$$
$$D_5 = \{d\}$$

$$\text{among}\big((x_1,x_2,x_3,x_4,x_5),\{c,d\},t,u\big)$$
$$\text{among}\big((x_1,x_2,x_3,x_4),\{c,d\},(t-1)^+,u-1\big)$$
$$\text{among}\big((x_1,x_2,x_3),\{c,d\},(t-2)^+,u-2\big)$$
$$\text{among}\big((x_1,x_2),\{c,d\},(t-3)^+,\min\{u-2,2\}\big)$$

J-consistency for Among

Projection of $\text{among}\big((x_1,\ldots,x_n),V,t,u\big)$ onto $x_1,\ldots,x_{n-1}$ is

$$\text{among}\big((x_1,\ldots,x_{n-1}),V,t',u'\big)$$

where

$$(t',u') = \left\{\begin{array}{ll} \big((t-1)^+,u-1\big) & \text{if } D_n \subseteq V \\ \big(t,\min\{u,n-1\}\big) & \text{if } D_n \cap V = \varnothing \\ \big((t-1)^+,\min\{u,n-1\}\big) & \text{otherwise} \end{array}\right\}$$

Example

$$\text{among}\big((x_1,x_2,x_3,x_4,x_5),\{c,d\},t,u\big)$$

$$D_1 = \{a,b\} \qquad \text{among}\big((x_1,x_2,x_3,x_4),\{c,d\},(t-1)^+,u-1\big)$$

$$D_2 = \{a,b,c\} \qquad \text{among}\big((x_1,x_2,x_3),\{c,d\},(t-2)^+,u-2\big)$$

$$D_3 = \{a,\mathrm{d}\} \qquad \text{among}\big((x_1,x_2),\{c,d\},(t-3)^+,\min\{u-2,2\}\big)$$

$$D_4 = \{c,d\} \qquad \text{among}\big((x_1),\{c,d\},(t-4)^+,\min\{u-2,1\}\big)$$

$$D_5 = \{d\}$$

J-consistency for Among

Projection of $\text{among}\big((x_1,\ldots,x_n),V,t,u\big)$ onto $x_1,\ldots,x_{n-1}$ is

$$\text{among}\big((x_1,\ldots,x_{n-1}),V,t',u'\big)$$

where

$$(t',u') = \begin{cases} \big((t-1)^+, u-1\big) & \text{if } D_n \subseteq V \\ \big(t, \min\{u, n-1\}\big) & \text{if } D_n \cap V = \varnothing \\ \big((t-1)^+, \min\{u, n-1\}\big) & \text{otherwise} \end{cases}$$

Example

$$D_1 = \{a,b\}$$
$$D_2 = \{a,b,c\}$$
$$D_3 = \{a,\mathrm{d}\}$$
$$D_4 = \{c,d\}$$
$$D_5 = \{d\}$$

$$\text{among}\big((x_1,x_2,x_3,x_4,x_5),\{c,d\},t,u\big)$$
$$\text{among}\big((x_1,x_2,x_3,x_4),\{c,d\},(t-1)^+,u-1\big)$$
$$\text{among}\big((x_1,x_2,x_3),\{c,d\},(t-2)^+,u-2\big)$$
$$\text{among}\big((x_1,x_2),\{c,d\},(t-3)^+,\min\{u-2,2\}\big)$$
$$\text{among}\big((x_1),\{c,d\},(t-4)^+,\min\{u-2,1\}\big)$$
$$\text{among}\big((),\{c,d\},(t-4)^+,\min\{u-2,0\}\big)$$

J-consistency for Among

Projection of $\text{among}\left((x_1,\ldots,x_n),V,t,u\right)$ onto $x_1,\ldots,x_{n-1}$ is

$$\text{among}\left((x_1,\ldots,x_{n-1}),V,t',u'\right)$$

where

$$(t',u') = \begin{cases} \left((t-1)^+, u-1\right) & \text{if } D_n \subseteq V \\ \left(t, \min\{u, n-1\}\right) & \text{if } D_n \cap V = \emptyset \\ \left((t-1)^+, \min\{u, n-1\}\right) & \text{otherwise} \end{cases}$$

Example

$$D_1 = \{a,b\}$$
$$D_2 = \{a,b,c\}$$
$$D_3 = \{a,\mathrm{d}\}$$
$$D_4 = \{c,d\}$$
$$D_5 = \{d\}$$

$$\text{among}\left((x_1,x_2,x_3,x_4,x_5),\{c,d\},t,u\right)$$
$$\text{among}\left((x_1,x_2,x_3,x_4),\{c,d\},(t-1)^+,u-1\right)$$
$$\text{among}\left((x_1,x_2,x_3),\{c,d\},(t-2)^+,u-2\right)$$
$$\text{among}\left((x_1,x_2),\{c,d\},(t-3)^+,\min\{u-2,2\}\right)$$
$$\text{among}\left((x_1),\{c,d\},(t-4)^+,\min\{u-2,1\}\right)$$
$$\text{among}\left((),\{c,d\},(t-4)^+,\min\{u-2,0\}\right)$$

Feasible if and only if $(t-4)^+ \le \min\{u-2,0\}$

J-Consistency for Sequence

- Projection is based on an integrality property.
  - The coefficient matrix of the inequality formulation has consecutive ones property.
  - So projection of the convex hull of the feasible set is an **integral polyhedron**.
    - **Polyhedral projection** therefore suffices.
    - Straightforward (but tedious) application of **Fourier elimination** yields the projection.

J-Consistency for Sequence

- Projection is based on an integrality property.
  - The coefficient matrix of the inequality formulation has consecutive ones property.
  - So projection of the convex hull of the feasible set is an **integral polyhedron**.
    - **Polyhedral projection** therefore suffices.
    - Straightforward (but tedious) application of **Fourier elimination** yields the projection.

- Projection onto any subset of variables is a **generalized sequence constraint**.
  - Complexity of projecting out x_k is $O(kq)$, where q = length of the overlapping sequences.

Following standard convention, we assume without loss of generality that the `sequence` constraint applies to 0-1 variables $x_1, \ldots, x_n$ [23, 46]. It enforces overlapping constraints of the form

$$\texttt{among}((x_{\ell-q+1}, \ldots, x_\ell), \{1\}, L_\ell, U_\ell) \tag{5}$$

Theorem 4. *Given any $k \in \{0, \ldots, n\}$, the projection of the* `sequence` *constraint defined by (5) onto $(x_1, \ldots, x_k)$ is described by a generalized* `sequence` *constraint that enforces constraints of the form*

$$\textit{among}\Big((x_i, \ldots, x_\ell), \{1\}, L^\ell_{\ell-i+1}, U^\ell_{\ell-i+1}\Big) \tag{6}$$

where $i = \ell - q + 1, \ldots, \ell$ for $\ell = q, \ldots, k$ and $i = 1, \ldots, \ell$ for $\ell = 1, \ldots, q - 1$. The projection of the `sequence` *constraint onto $(x_1, \ldots, x_{k-1})$ is given by (6) with $L^\ell_{\ell-i+1}$ replaced by $\hat{L}^\ell_{\ell-i+1}$ and $U^\ell_{\ell-i+1}$ by $\hat{U}^\ell_{\ell-i+1}$, where*

$$\begin{aligned} \hat{L}^\ell_i &= \begin{cases} \max\{L^\ell_i, L^k_{i+k-\ell} - U^k_{k-\ell}\}, \ \textit{for}\ i = 1, \ldots, q - k + \ell, \\ L^\ell_i, \ \textit{for}\ i = q - k + \ell + 1, \ldots, q \end{cases} \\ \hat{U}^\ell_i &= \begin{cases} \min\{U^\ell_i, U^k_{i+k-\ell} - L^k_{k-\ell}\}, \ \textit{for}\ i = 1, \ldots, q - k + \ell, \\ U^\ell_i, \ \textit{for}\ i = q - k + \ell + 1, \ldots, q \end{cases} \end{aligned} \tag{7}$$

J-Consistency for Sequence

Example $\text{among}\big((x_{t-3},\ldots,x_t),\{1\},2,2\big), \quad t=4,5,6$

$$x_1,x_3,x_4,x_6 \in \{0,1\}, \quad x_2,x_5 \in \{1\}$$

To project out x_6, add constraint

$$\text{among}\big((x_3,x_4,x_5),\{1\},1,1\big)$$

J-Consistency for Sequence

Example $\quad \text{among}\big((x_{t-3},\ldots,x_t),\{1\},2,2\big), \quad t = 4,5,6$

$$x_1,x_3,x_4,x_6 \in \{0,1\}, \quad x_2,x_5 \in \{1\}$$

To project out x_6, add constraint

$$\text{among}\big((x_3,x_4,x_5),\{1\},1,1\big)$$

To project out x_5, add constraints

$$\text{among}\big((x_2,x_3,x_4),\{1\},1,1\big) \quad \text{among}\big((x_3,x_4),\{1\},0,0\big)$$

J-Consistency for Sequence

Example $\text{among}\big((x_{t-3},\ldots,x_t),\{1\},2,2\big),\quad t=4,5,6$

$$x_1,x_3,x_4,x_6 \in \{0,1\},\quad x_2,x_5 \in \{1\}$$

To project out x_6, add constraint

$$\text{among}\big((x_3,x_4,x_5),\{1\},1,1\big)$$

To project out x_5, add constraints

$$\text{among}\big((x_2,x_3,x_4),\{1\},1,1\big)\quad \text{among}\big((x_3,x_4),\{1\},0,0\big)$$

To project out x_4, add constraints

$$\text{among}\big((x_1),\{1\},1,1\big)\qquad \text{among}\big((x_1,x_2,x_3),\{1\},1,2\big)$$

$$\text{among}\big((x_2,x_3),\{1\},0,1\big)\quad \text{among}\big((x_3),\{1\},0,0\big)$$

J-Consistency for Sequence

Example $\quad \text{among}\big((x_{t-3},\ldots,x_t),\{1\},2,2\big),\quad t=4,5,6$

$$x_1,x_3,x_4,x_6 \in \{0,1\},\quad x_2,x_5 \in \{1\}$$

To project out x_6, add constraint

$$\text{among}\big((x_3,x_4,x_5),\{1\},1,1\big)$$

To project out x_5, add constraints

$$\text{among}\big((x_2,x_3,x_4),\{1\},1,1\big) \quad \text{among}\big((x_3,x_4),\{1\},0,0\big)$$

To project out x_4, add constraints

$$\text{among}\big((x_1),\{1\},1,1\big) \qquad \text{among}\big((x_1,x_2,x_3),\{1\},1,2\big)$$

$$\text{among}\big((x_2,x_3),\{1\},0,1\big) \quad \text{among}\big((x_3),\{1\},0,0\big)$$

To project out x_3, fix $(x_1,x_2) = (1,1)$

J-Consistency for Regular Constraint

- Projection can be read from state transition graph.
  - Complexity of projecting onto $x_1, \ldots, x_k$ for all k is $O(nm^2)$, where n = number of variables, m = max number of states per stage.

J-Consistency for Regular Constraint

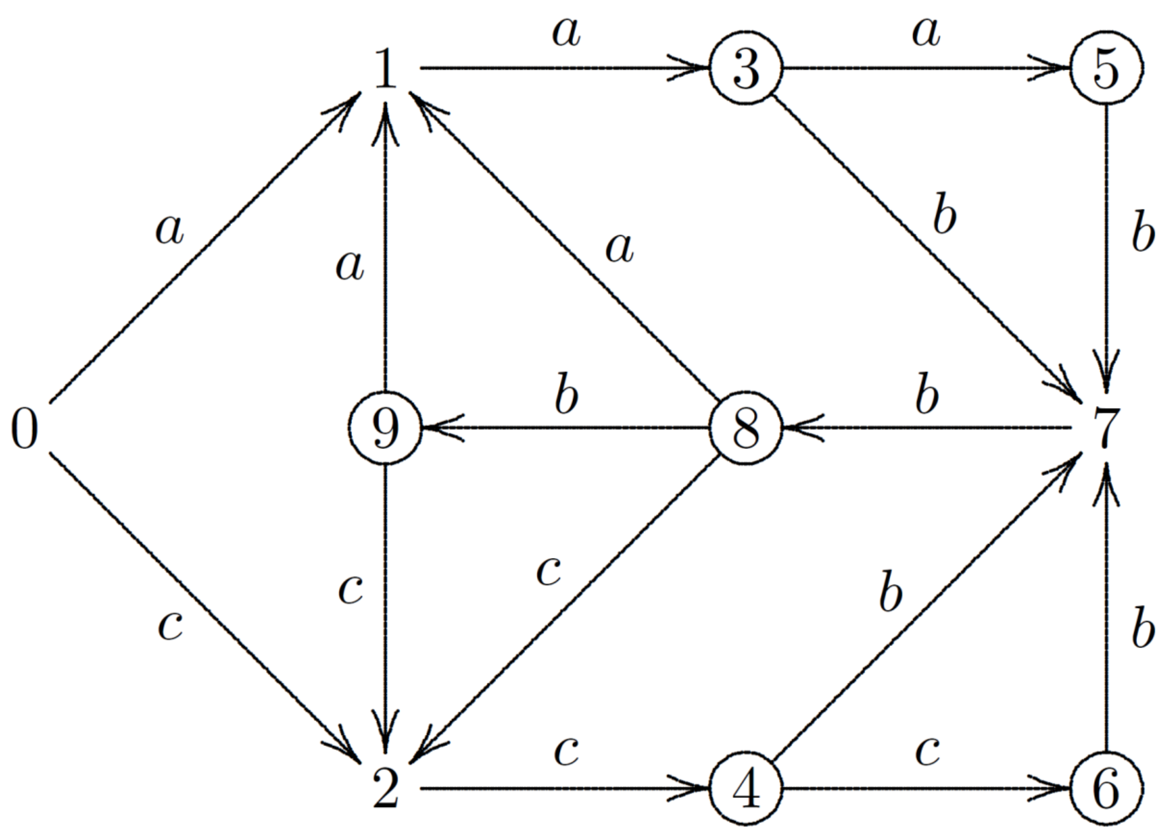
- Projection can be read from state transition graph.
  - Complexity of projecting onto $x_1, \ldots, x_k$ for all k is $O(nm^2)$, where n = number of variables, m = max number of states per stage.

- Shift scheduling example
  - Assign each worker to shift $x_i \in \{a,b,c\}$ on each day $i = 1,\ldots,7$.
  - Must work any given shift 2 or 3 days in a row.
  - No direct transition between shifts *a* and *c*.
  - Variable domains: $D_1 = D_5 = \{a,c\}$, $D_2 = \{a,b,c\}$,
    $D_3 = D_6 = D_7 = \{a,b\}$, $\mathrm{D}_4 = \{b,c\}$

J-Consistency for Regular Constraint

Deterministic finite automaton for this problem instance:

= absorbing state

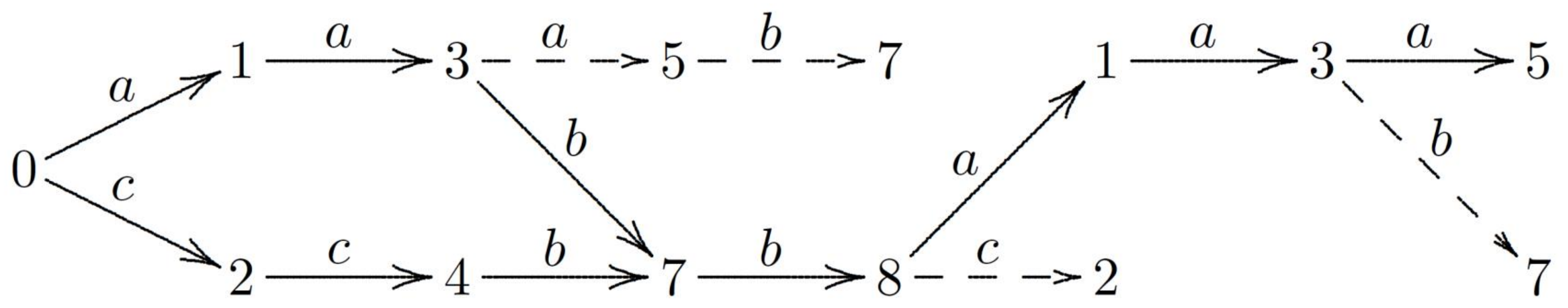


Regular language expression:

(((aa|aaa)(bb|bbb))*|((cc|ccc)(bb|bbb))*)*(c|(aa|aaa)|(cc|ccc))

J-Consistency for Regular Constraint

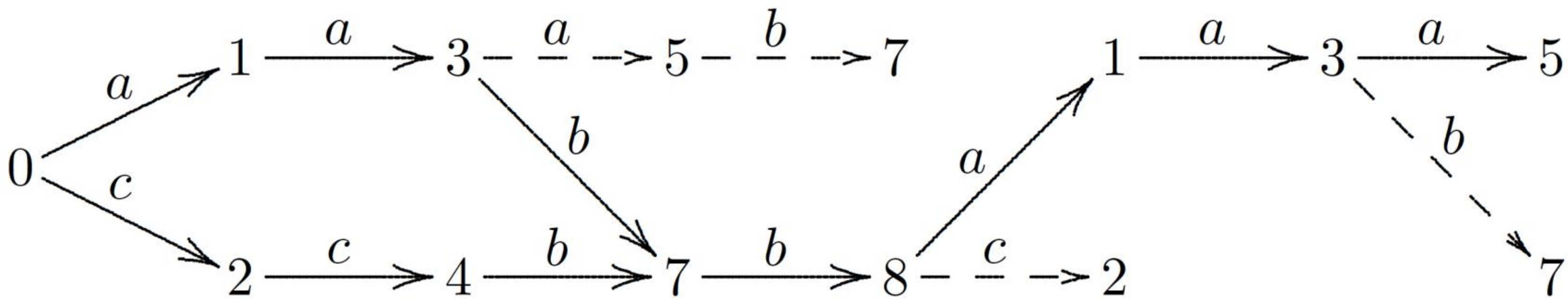
State transition graph for 7 stages
Dashed lines lead to unreachable states.



$j =$	1	2	3	4	5	6	7	8
$D_j =$	$\{a, c\}$	$\{a, b, c\}$	$\{a, b\}$	$\{b, c\}$	$\{a, c\}$	$\{a, b\}$	$\{a, b\}$	

J-Consistency for Regular Constraint

State transition graph for 7 stages
Dashed lines lead to unreachable states.

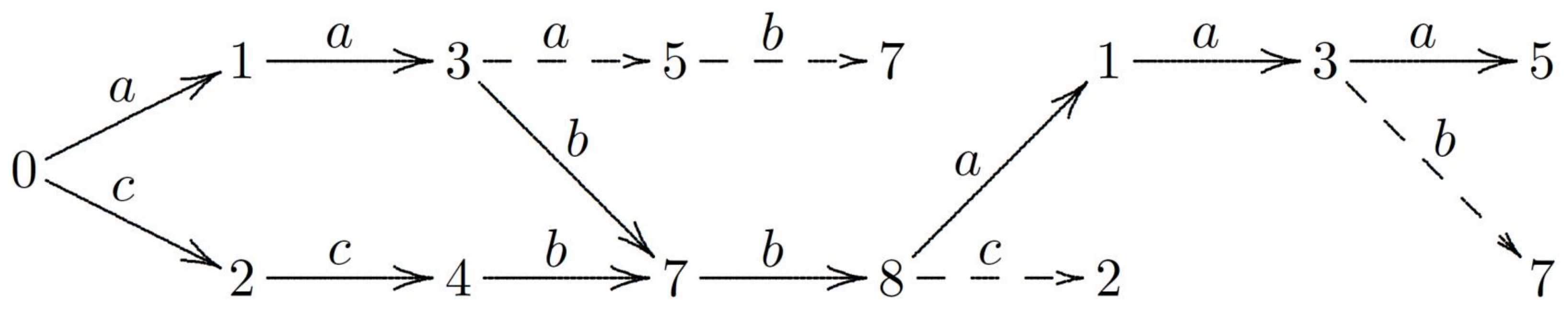


$j =$	1	2	3	4	5	6	7	8
$D_j =$	$\{a, c\}$	$\{a, b, c\}$	$\{a, b\}$	$\{b, c\}$	$\{a, c\}$	$\{a, b\}$	$\{a, b\}$	
$D'_j =$	$\{a, c\}$	$\{a, c\}$	$\{b\}$	$\{b\}$	$\{a\}$	$\{a\}$	$\{a\}$	

Filtered domains

J-Consistency for Regular Constraint

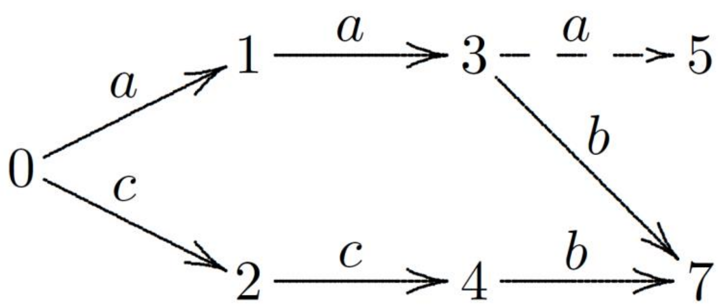
To project onto x_1, x_2, x_3, truncate the graph at stage 4.



$j =$	1	2	3	4	5	6	7	8
$D_j =$	$\{a, c\}$	$\{a, b, c\}$	$\{a, b\}$	$\{b, c\}$	$\{a, c\}$	$\{a, b\}$	$\{a, b\}$	
$D'_j =$	$\{a, c\}$	$\{a, c\}$	$\{b\}$	$\{b\}$	$\{a\}$	$\{a\}$	$\{a\}$	

J-Consistency for Regular Constraint

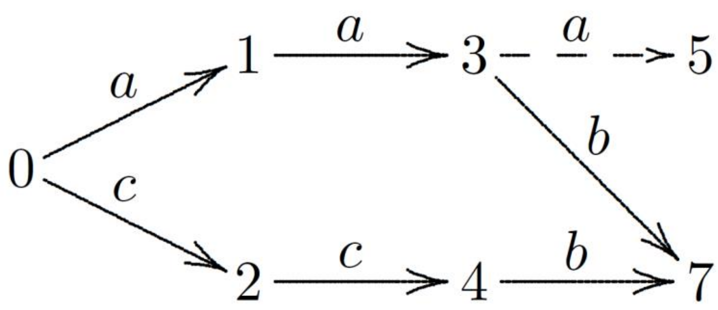
To project onto x_1, x_2, x_3, truncate the graph at stage 4.



$j =$	1	2	3	4
$D_j =$	$\{a, c\}$	$\{a, b, c\}$	$\{a, b\}$	
$D'_j =$	$\{a, c\}$	$\{a, c\}$	$\{b\}$	

J-Consistency for Regular Constraint

To project onto x_1, x_2, x_3, truncate the graph at stage 4.



$$\begin{array}{lcccc} j = & 1 & 2 & 3 & 4 \\ D_j = & \{a, c\} & \{a, b, c\} & \{a, b\} & \\ D'_j = & \{a, c\} & \{a, c\} & \{b\} & \end{array}$$

Resulting graph can be viewed as a constraint that describes the projection.

Constraint is easily propagated through a relaxed decision diagram.

J-Consistency for Alldiff Constraint

- Projection is inherently complicated.
  - But it can simplify for small domains.

- The result is a **disjunction** of constraint sets,
  - …each of which contains an alldiff constraint and some atmost constraints.

J-Consistency for Alldiff Constraint

Example $\text{alldiff}\left(x_1, x_2, x_3, x_4, x_5\right)$

$$D_1 = \{a, b, c\}, D_2 = \{c, d, e\}, D_3 = \{d, e, f\}, D_4 = \{e, f, g\}, D_5 = \{a, f, g\}$$

J-Consistency for Alldiff Constraint

Example $\text{alldiff}(x_1, x_2, x_3, x_4, x_5)$

$$D_1 = \{a, b, c\}, D_2 = \{c, d, e\}, D_3 = \{d, e, f\}, D_4 = \{e, f, g\}, \boxed{D_5 = \{a, f, g\}}$$

Projecting out x_5, we get

$$\text{alldiff}(x_1, x_2, x_3, x_4), \quad \text{atmost}((x_1, x_2, x_3, x_4), \{a, f, g\}, 2)$$

because x_5 must take one of the values in $\{a, f, g\}$, …

J-Consistency for Alldiff Constraint

Example $\text{alldiff}\left(x_1, x_2, x_3, x_4, x_5\right)$

$$D_1 = \{a,b,c\},\ D_2 = \{c,d,e\},\ D_3 = \{d,e,f\},\ D_4 = \{e,f,g\},\ \boxed{D_5 = \{a,f,g\}}$$

Projecting out x_5, we get

$$\text{alldiff}\left(x_1, x_2, x_3, x_4\right), \quad \text{atmost}\left((x_1, x_2, x_3, x_4), \{a,f,g\}, \boxed{2}\right)$$

because x_5 must take one of the values in $\{a,f,g\}$, leaving 2 for other x_i s.

J-Consistency for Alldiff Constraint

Example $\text{alldiff}(x_1, x_2, x_3, x_4, x_5)$

$$D_1 = \{a,b,c\},\ D_2 = \{c,d,e\},\ D_3 = \{d,e,f\},\ D_4 = \{e,f,g\},\ D_5 = \{a,f,g\}$$

Projecting out x_5, we get

$$\text{alldiff}(x_1, x_2, x_3, x_4), \quad \text{atmost}((x_1, x_2, x_3, x_4), \boxed{\{a,f,g\}}, 2)$$

because x_5 must take one of the values in $\{a,f,g\}$, leaving 2 for other x_i s.

Projecting out x_4, we note that $x_4 \in \{a,f,g\}$ or $x_4 \notin \{a,f,g\}$.

J-Consistency for Alldiff Constraint

Example $\text{alldiff}\left(x_1, x_2, x_3, x_4, x_5\right)$

$$D_1 = \{a, b, c\},\ D_2 = \{c, d, e\},\ D_3 = \{d, e, f\},\ D_4 = \{e, f, g\},\ D_5 = \{a, f, g\}$$

Projecting out x_5, we get

$$\text{alldiff}\left(x_1, x_2, x_3, x_4\right), \quad \text{atmost}\left((x_1, x_2, x_3, x_4), \{a, f, g\}, 2\right)$$

because x_5 must take one of the values in $\{a,f,g\}$, leaving 2 for other x_i s.

Projecting out x_4, we note that $x_4 \in \{a,f,g\}$ or $x_4 \notin \{a,f,g\}$.

If $x_4 \in \{a,f,g\}$, we get

$$\text{alldiff}\left(x_1, x_2, x_3\right), \quad \text{atmost}\left((x_1, x_2, x_3), \{a, f, g\}, 1\right)$$

J-Consistency for Alldiff Constraint

Example $\text{alldiff}(x_1, x_2, x_3, x_4, x_5)$

$$D_1 = \{a,b,c\},\ D_2 = \{c,d,e\},\ D_3 = \{d,e,f\},\ \boxed{D_4 = \{e,f,g\}},\ D_5 = \{a,f,g\}$$

Projecting out x_5, we get

$$\text{alldiff}(x_1,x_2,x_3,x_4), \quad \text{atmost}((x_1,x_2,x_3,x_4),\{a,f,g\},2)$$

because x_5 must take one of the values in $\{a,f,g\}$, leaving 2 for other x_i s.

Projecting out x_4, we note that $x_4 \in \{a,f,g\}$ or $x_4 \notin \{a,f,g\}$.

If $x_4 \in \{a,f,g\}$, we get

$$\text{alldiff}(x_1,x_2,x_3), \quad \text{atmost}((x_1,x_2,x_3),\{a,f,g\},1)$$

If $x_4 \notin \{a,f,g\}$, we get $x_4 = e$, and we remove e from other domains.

J-Consistency for Alldiff Constraint

Example $\text{alldiff}(x_1, x_2, x_3, x_4, x_5)$

$$D_1 = \{a,b,c\}, D_2 = \{c,d,e\}, D_3 = \{d,e,f\}, D_4 = \{e,f,g\}, D_5 = \{a,f,g\}$$

Projecting out x_5, we get

$$\text{alldiff}(x_1, x_2, x_3, x_4), \quad \text{atmost}((x_1, x_2, x_3, x_4), \{a,f,g\}, 2)$$

because x_5 must take one of the values in $\{a,f,g\}$, leaving 2 for other x_i s.

Projecting out x_4, we note that $x_4 \in \{a,f,g\}$ or $x_4 \notin \{a,f,g\}$.

If $x_4 \in \{a,f,g\}$, we get

$$\text{alldiff}(x_1, x_2, x_3), \quad \text{atmost}((x_1, x_2, x_3), \{a,f,g\}, 1)$$

If $x_4 \notin \{a,f,g\}$, we get $x_4 = e$, and we remove e from other domains.

So the projection is

$$\begin{bmatrix} \text{alldiff}(x_1, x_2, x_3) \\ \text{atmost}((x_1, x_2, x_3), \{a,f,g\}, 1) \end{bmatrix} \vee \begin{bmatrix} D_1 = \{a,b,c\} \\ D_2 = \{c,d\} \\ D_3 = \{d,f\} \end{bmatrix}$$

J-Consistency for Alldiff Constraint

Example $$\text{alldiff}(x_1, x_2, x_3, x_4, x_5)$$

$$D_1 = \{a,b,c\}, D_2 = \{c,d,e\}, D_3 = \{d,e,f\}, D_4 = \{e,f,g\}, D_5 = \{a,f,g\}$$

Projecting out x_3, we get simply

$$\text{alldiff}(x_1, x_2)$$

Projecting out x_2, we get the original domain for x_1

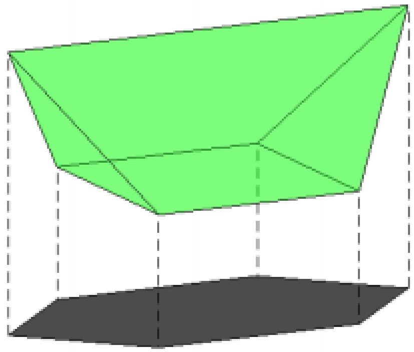
$$D_1 = \{a,b,c\}$$

Algorithm 2 Given a projection of $\texttt{alldiff}(x^n)$ onto x^k, this algorithm computes a projection onto x^{k-1}. The projection onto x^k is assumed to be a disjunction of constraint sets, each of which has the form (10). The above algorithm is applied to each disjunct, after which the disjunction of all created constraint sets forms the projection onto x^{k-1}.

For all $i \in I$: if $\texttt{atmost}(x^k, V_i, b_i)$ is redundant then remove i from I.
For all $i \in I$:
    If $D_k \cap V_i \neq \emptyset$ then
        If $b_i > 1$ then
            Create a constraint set consisting of $\texttt{alldiff}(x^{k-1})$,
            $\texttt{atmost}(x^{k-1}, V_{i'}, b_{i'})$ for $i' \in I \setminus \{i\}$, and $\texttt{atmost}\ (x^{k-1}, V_i, b_i - 1)$.
Let $R = D_k \setminus \bigcup_{i \in I} V_i$.
If $|R| > 1$ then
    Create a constraint set consisting of $\texttt{alldiff}(x^{k-1})$,
    $\texttt{atmost}(x^{k-1}, V_{i'}, b_{i'})$ for $i' \in I$, and $\texttt{atmost}\ (x^{k-1}, R, |R| - 1)$.
Else if $|R| = 1$ then
    Let $R = \{v\}$ and remove v from D_j for $j = 1, \ldots, k-1$ and from V_i for $i \in I$.
    If D_j is nonempty for $j = 1, \ldots, k-1$ then
        For all $i' \in I$: if $\texttt{atmost}(x^{k-1}, V_{i'}, b_{i'})$ is redundant then remove i' from i.
        Create a constraint set consisting of $\texttt{alldiff}(x^{k-1})$ and
        $\texttt{atmost}(x^{k-1}, V_{i'}, b_{i'})$ for $i' \in I$.

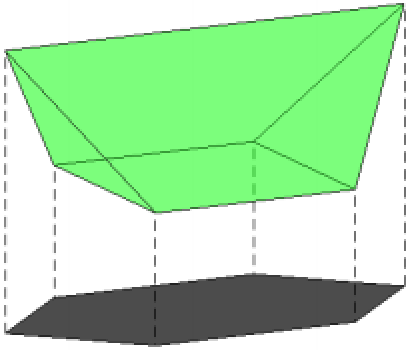
Bounds Consistency as Projection

- Bounds consistency
  - Most naturally defined when domains can be embedded in the real numbers.
  - Then we achieve bounds consistency by **projecting** the convex hull of the feasible set onto each x_j.
  - **Continuous *J*-consistency** is achieved by **projecting** the convex hull onto x_J.



Bounds Consistency as Projection

- Cutting planes
  - Projection onto x_J is defined by cutting planes that contain variables in x_J.
    - Close relationship to integer programming..



Bounds Consistency as Projection

- Usefulness of cutting planes
  - They can be propagated through LP relaxation.
    - This can help bound the objective function as well as achieve consistency.
  - They can reduce backtracking
    - Even when LP relaxation is not used.
    - This has never been studied in integer programming!