LOCAL SEARCH FOR CSPs

R&N: Chap. 6

LOCAL SEARCH FOR CSPs

- In the CSP formulation as a search problem, path is irrelevant, so we can use complete-state formulation
- State
  - An assignment of values to variables
- Successors(s)
  - All states resulted from by choosing a new value for a variable
- Cost-function(s)
  - Number of violated constraints
- Global minimum
  - h(s) = 0

MIN-CONFLICTS

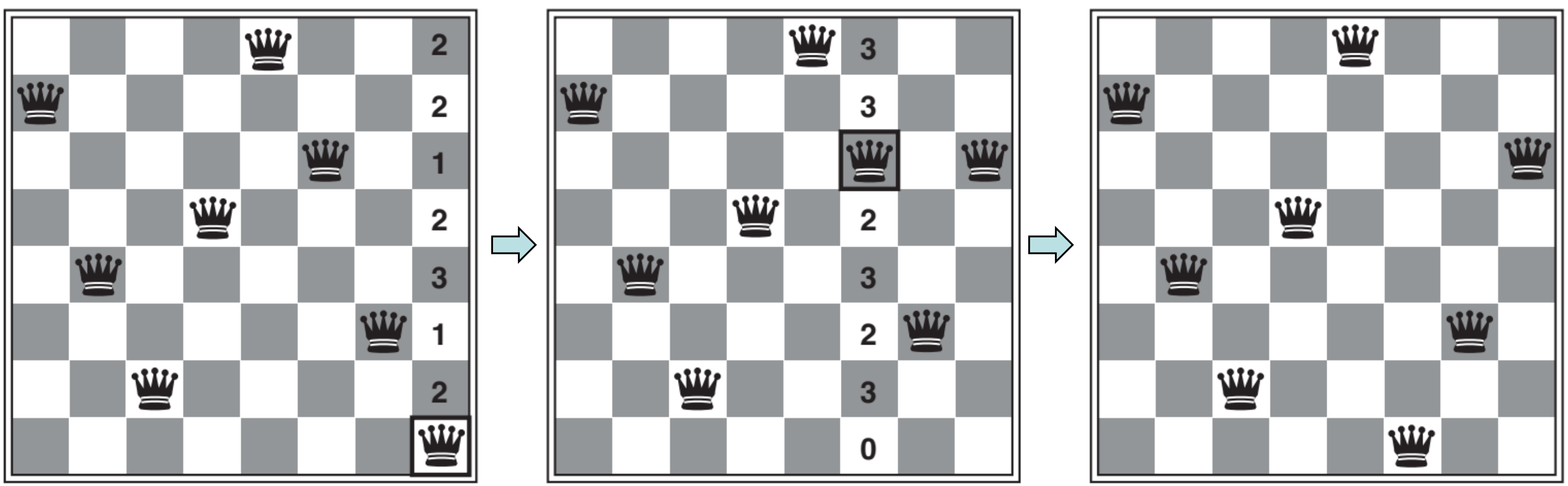
- Min-conflicts heuristic
  - Select the value that results in the minimum number of conflicts with other variables

```
function MIN-CONFLICTS(csp, max steps) returns a solution or failure
      inputs: csp, a constraint satisfaction problem
              max steps, the number of steps allowed before giving up

      current ← an initial complete assignment for csp
      for i = 1 to max steps do
         if current is a solution for csp then return current
         var←a randomly chosen conflicted variable from csp.VARIABLES
         value←the value v for var that minimizes CONFLICTS(var,v, current, csp)
         set var = value in current
      return failure
```

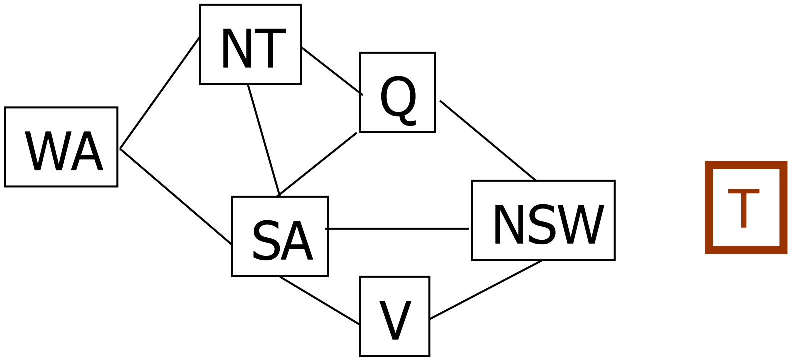
8-Queens example

- For the n-queens problem, if you don't count the initial placement of queens, the run time of min-conflicts is roughly independent of problem size.
  - It solves even the million-queens problem in an average of 50 steps (after the initial assignment)
  - N-queens is easy for local search because solutions are densely distributed throughout the state space.



Exploiting the Structure of CSP

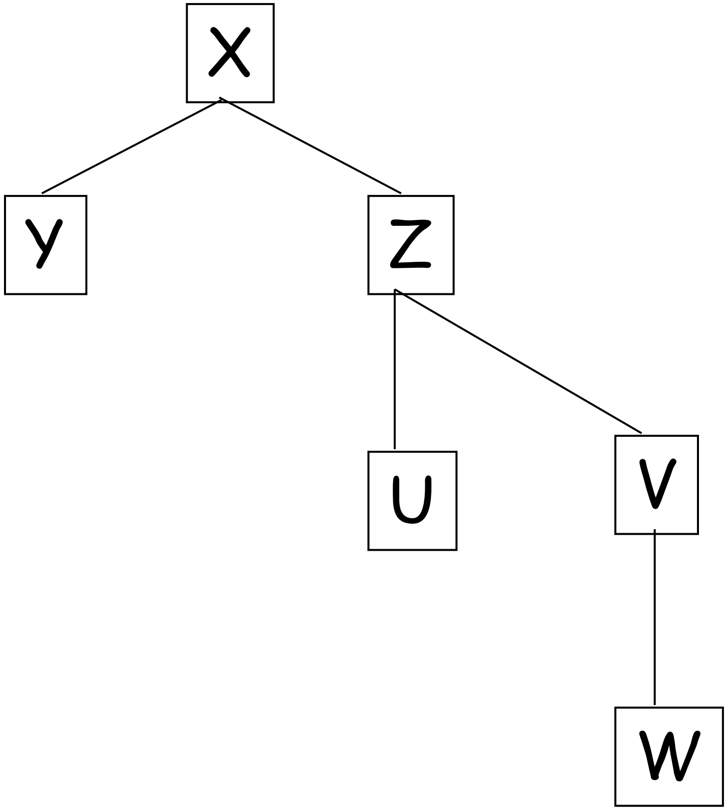
If the constraint graph contains several components, then solve one independent CSP per component



- Suppose each sub-problem has c variables out of n
  - Worst-case solution cost is $O((n/c) . d^c)$
- Example: n = 80, d = 2, c =20
  - 2^{80} = 4 billion years at 10 million nodes/sec
  - 4.2^{20} = 0.4 seconds at 10 million nodes/sec

Exploiting the Structure of CSP

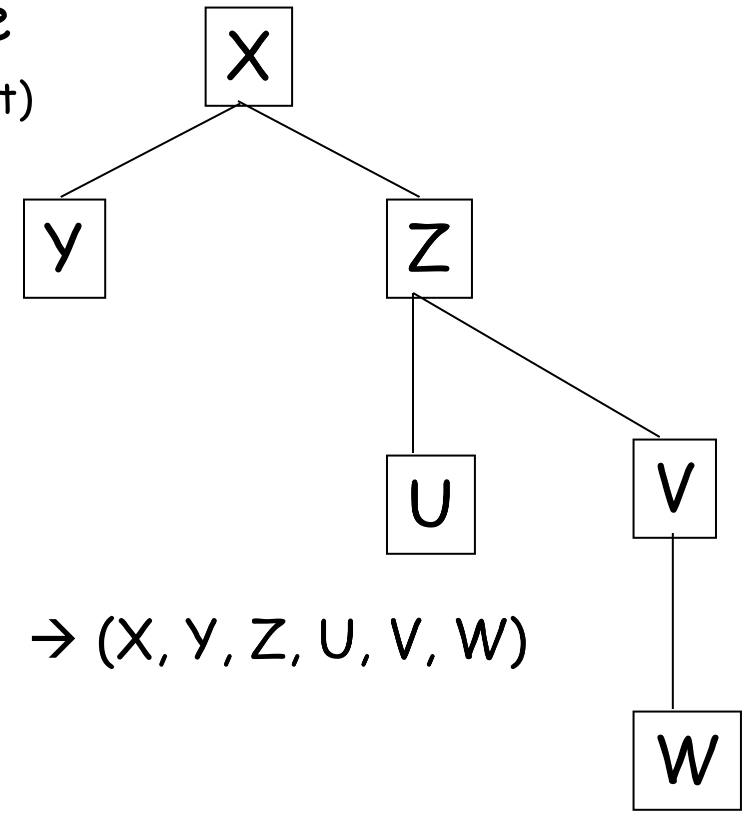
If the constraint graph is a tree, then:



Exploiting the Structure of CSP

If the constraint graph is a tree, then:

1. Order the variables from the root to the leaves (Topological sort)
   → $(X_1, X_2, \ldots, X_n)$

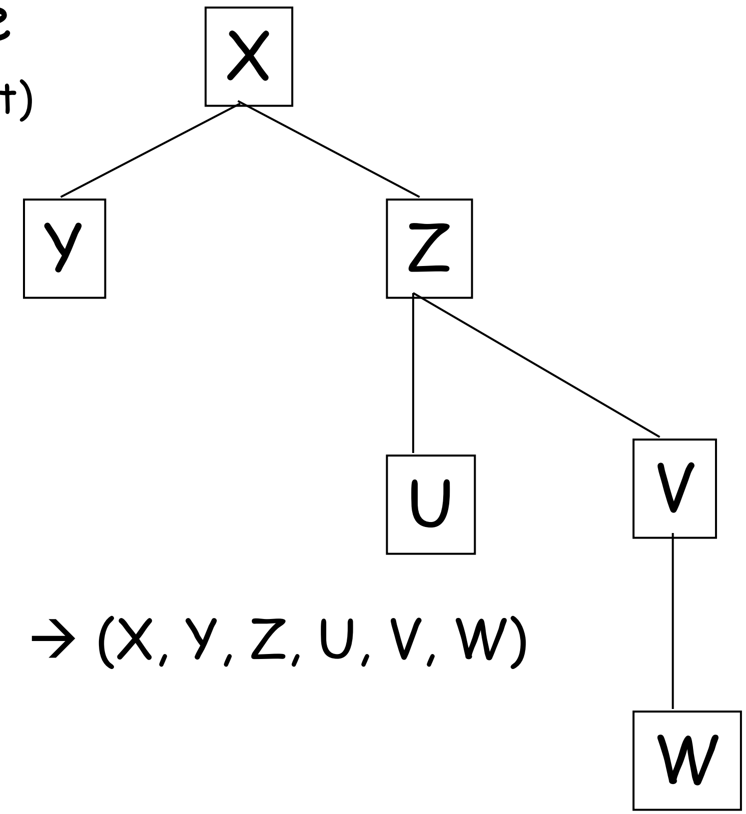


→ (X, Y, Z, U, V, W)

Exploiting the Structure of CSP

If the constraint graph is a tree, then:

1. Order the variables from the root to the leaves (Topological sort)
   → $(X_1, X_2, \ldots, X_n)$
2. For $j = n, n-1, \ldots, 2$ call REMOVE-VALUES(X_j, X_i) where X_i is the parent of X_j

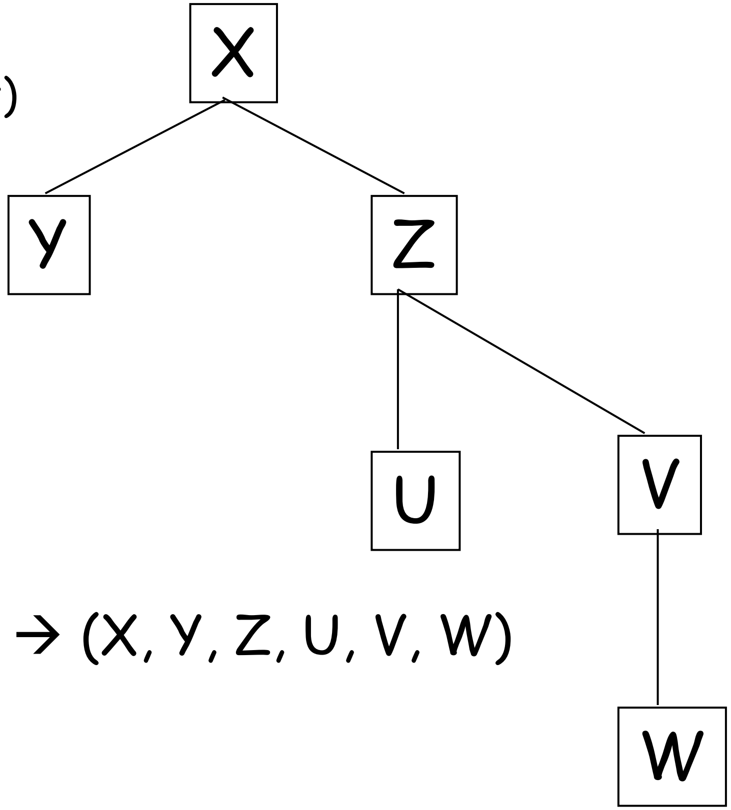


→ (X, Y, Z, U, V, W)

Exploiting the Structure of CSP

If the constraint graph is a tree, then:

1. Order the variables from the root to the leaves (Topological sort) → $(X_1, X_2, \ldots, X_n)$
2. For $j = n, n-1, \ldots, 2$ call REMOVE-VALUES(X_j, X_i) where X_i is the parent of X_j
3. Assign any valid value to X_1

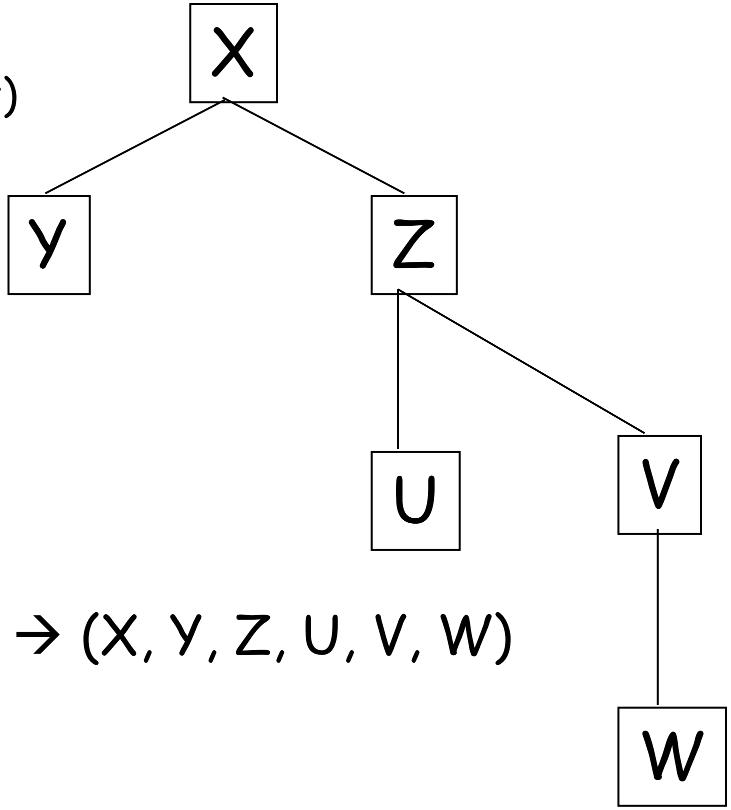


→ (X, Y, Z, U, V, W)

Exploiting the Structure of CSP

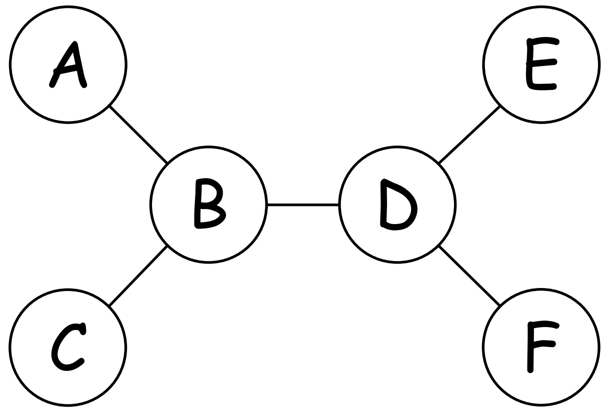
If the constraint graph is a tree, then:

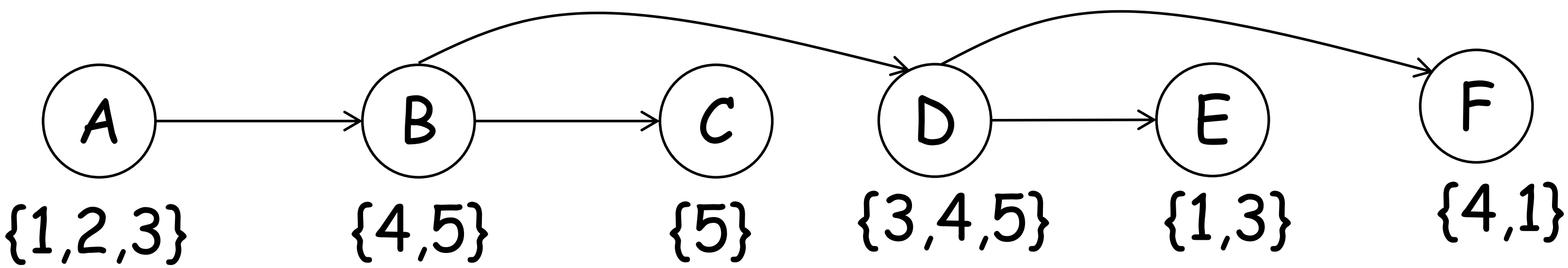
1. Order the variables from the root to the leaves (Topological sort)
   → $(X_1, X_2, \ldots, X_n)$
2. For j = n, n-1, …, 2 call REMOVE-VALUES(X_j, X_i) where X_i is the parent of X_j
3. Assign any valid value to X_1
4. For j = 2, …, n do
   Assign any value to X_j consistent with the value assigned to X_i, where X_i is the parent of X_j



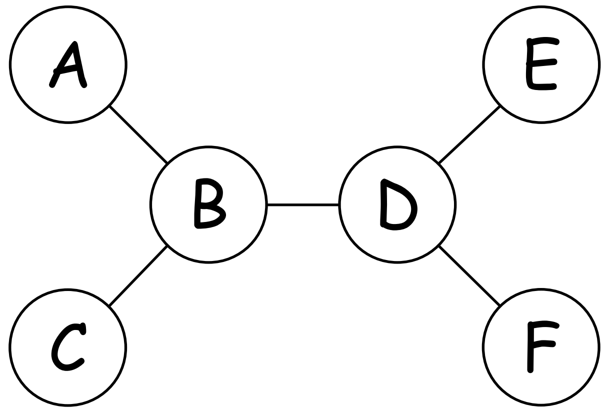
→ (X, Y, Z, U, V, W)

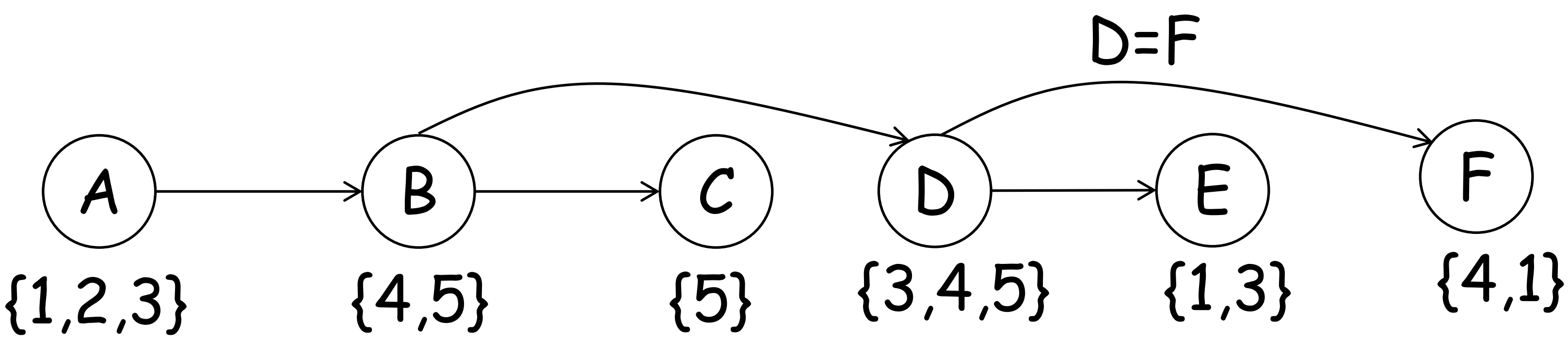
Example



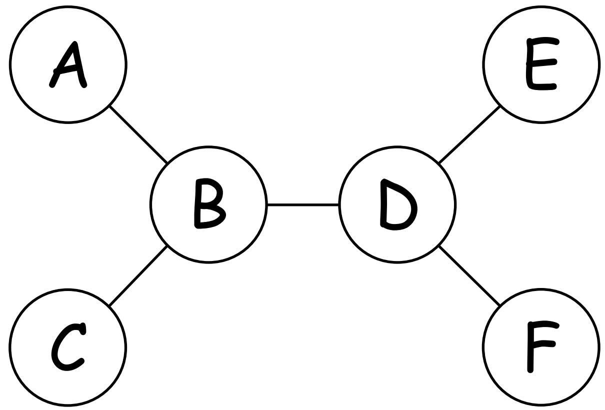


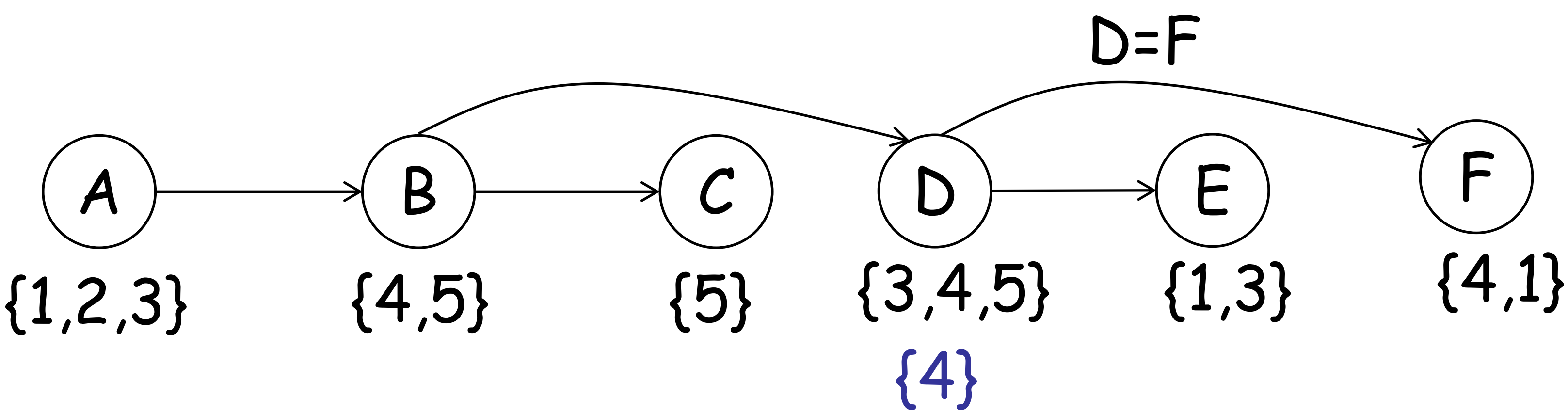
Example



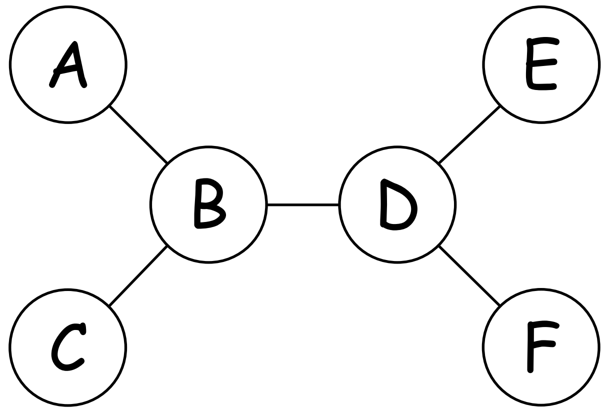


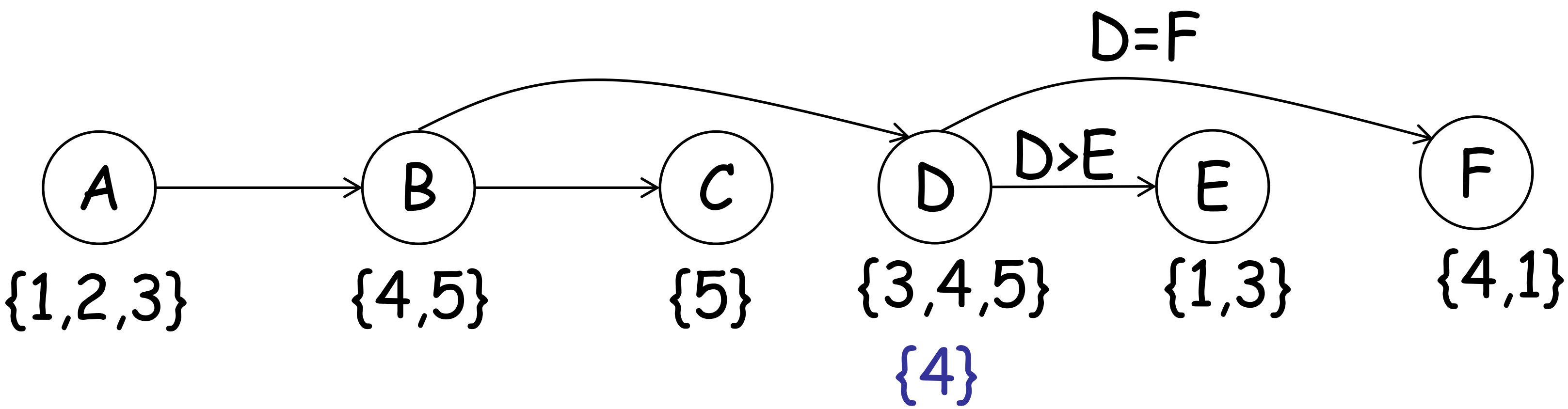
Example



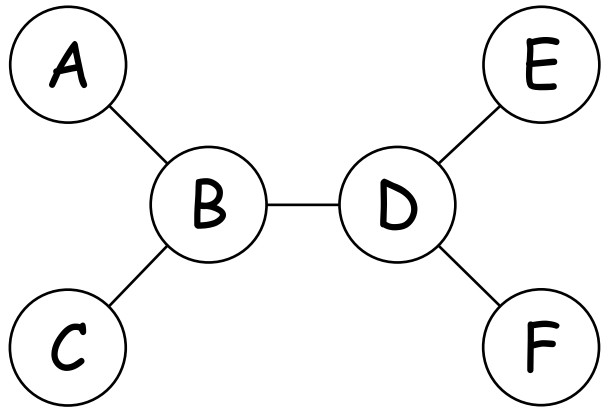


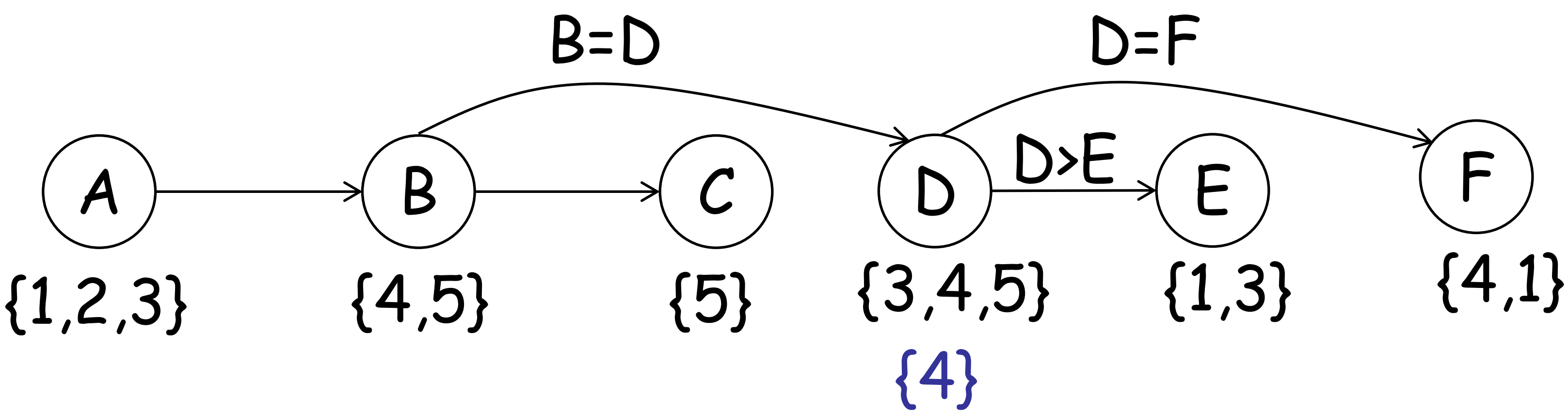
Example



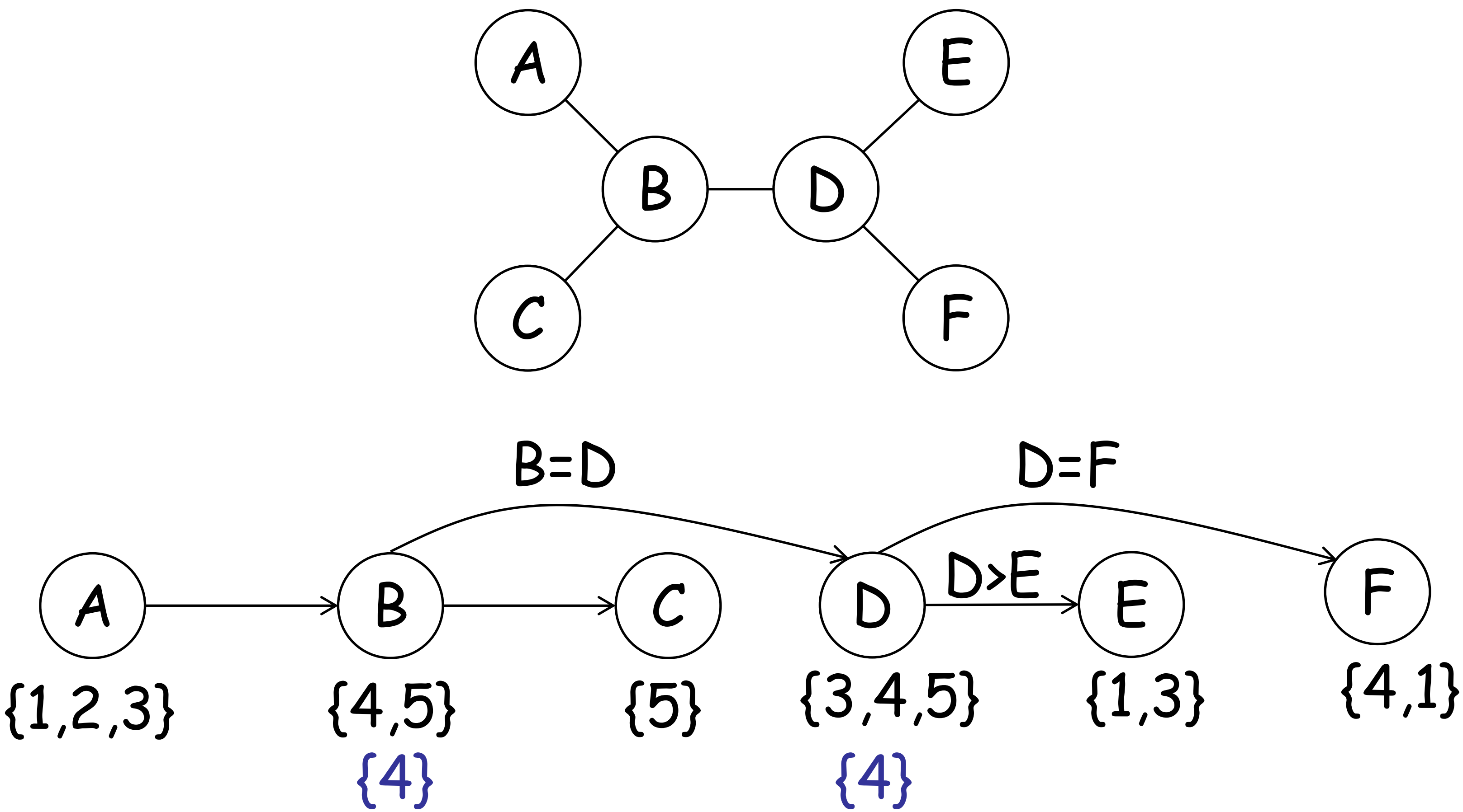


Example

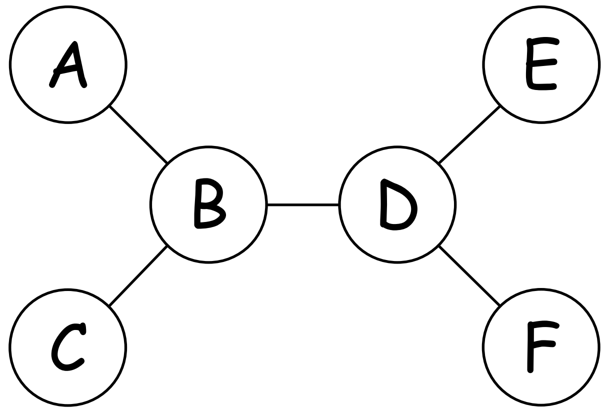


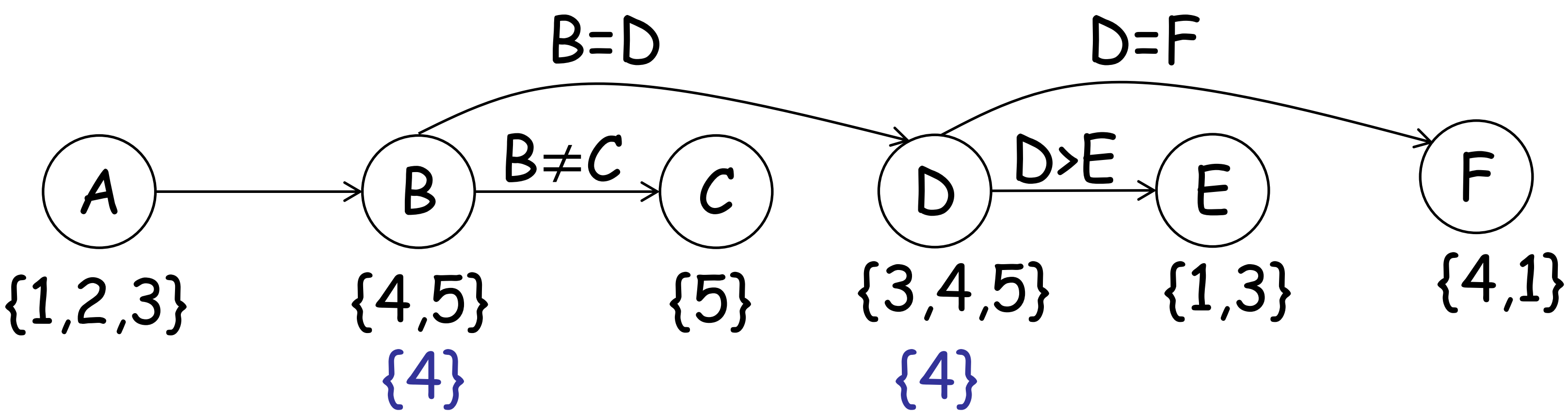


Example

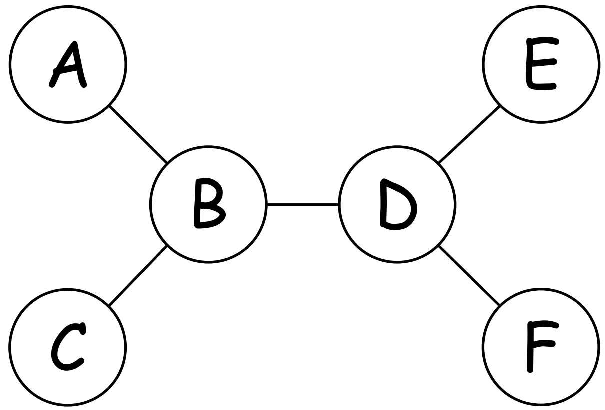


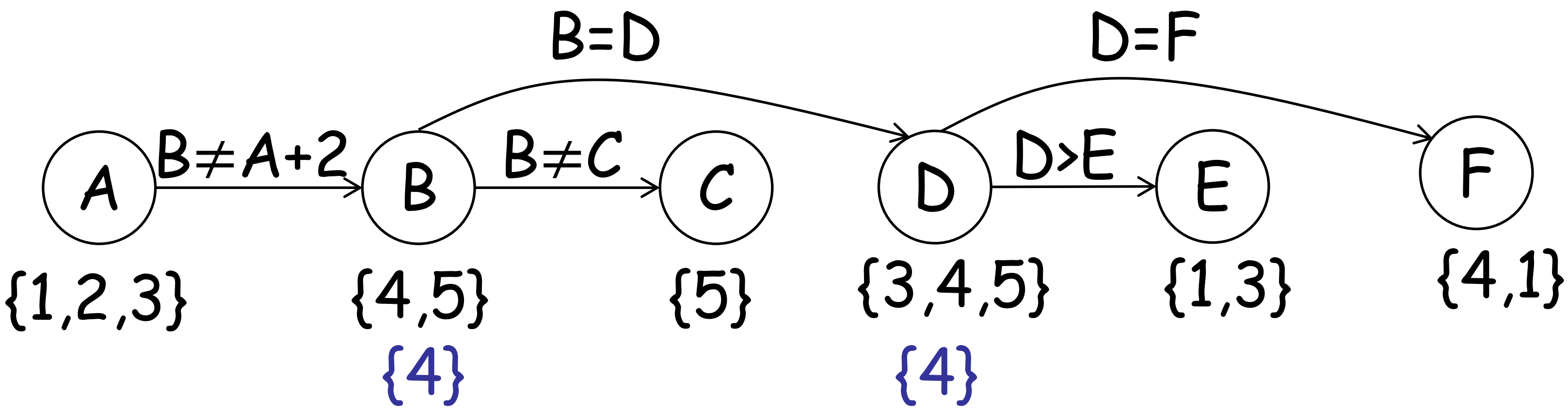
Example



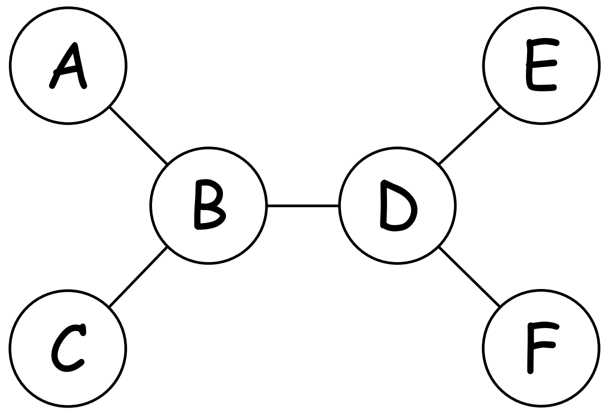


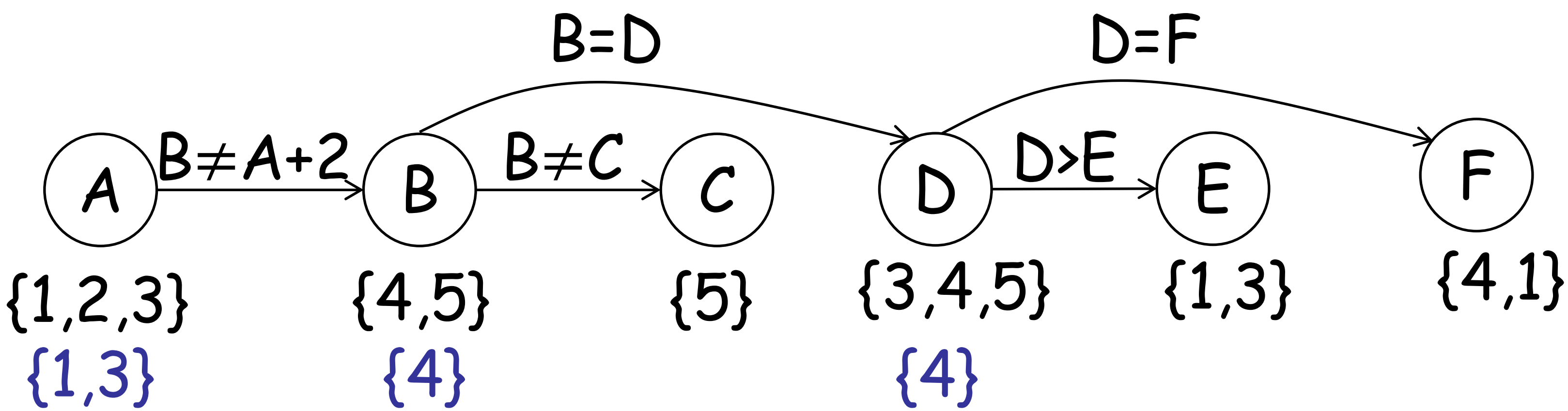
Example



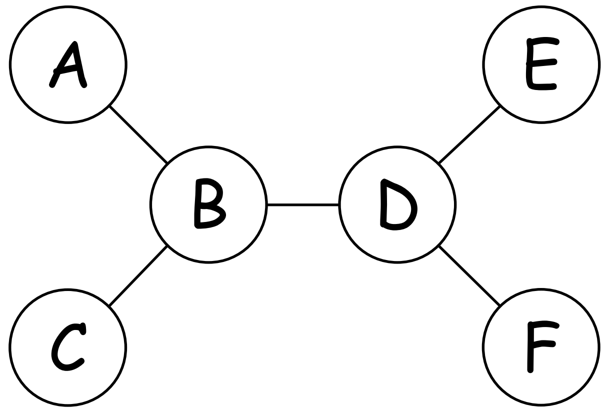


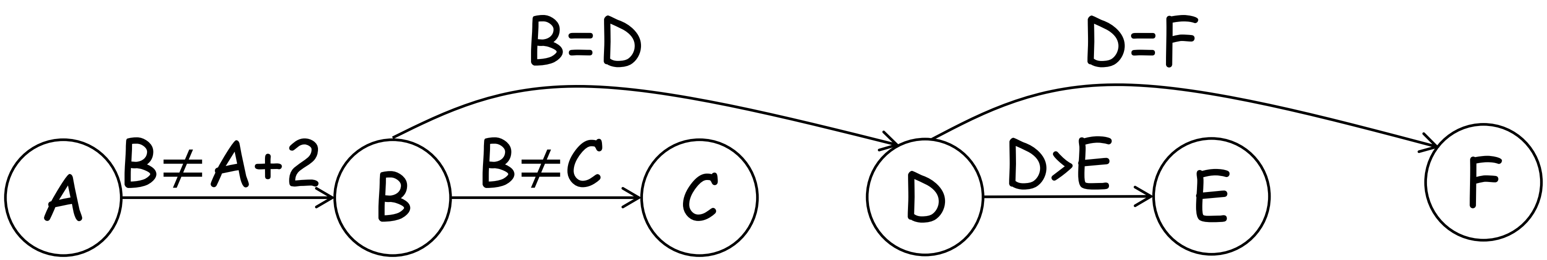
Example





Example

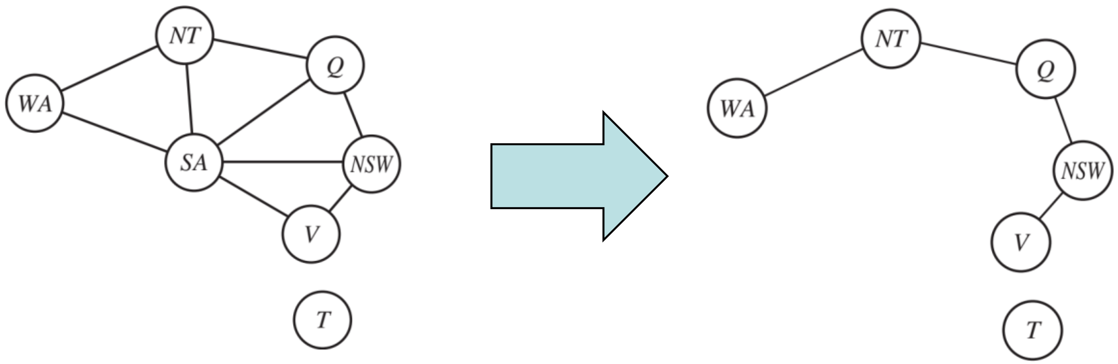




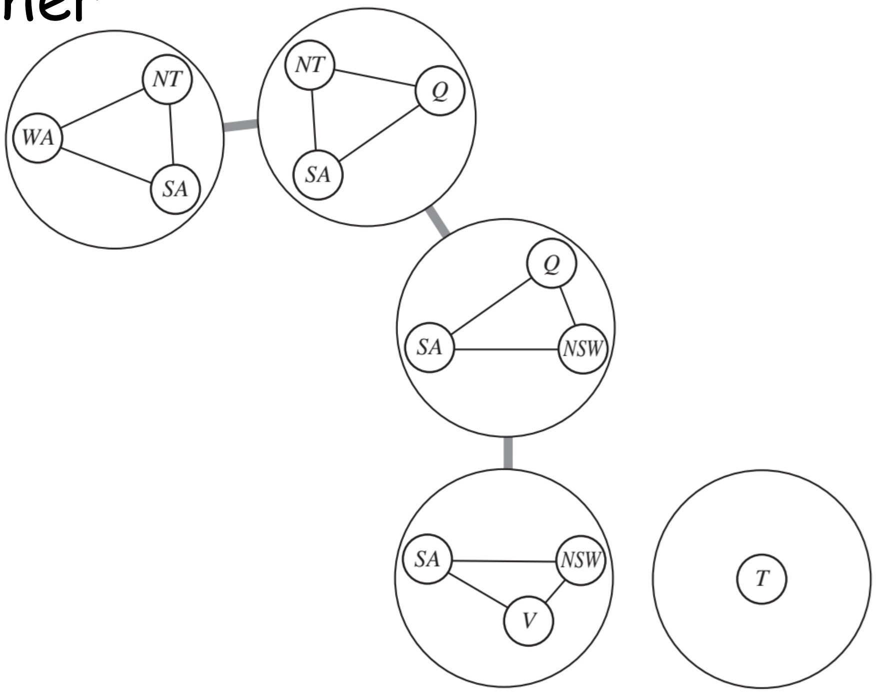
A	B	C	D	E	F
{1,2,3}	{4,5}	{5}	{3,4,5}	{1,3}	{4,1}
{1,3}	{4}		{4}		
A=3,	B=4,	C=5,	D=4,	E=1,	F=4

Reduction of general graphs into trees

- Removing nodes



- Collapsing nodes together



Cut-set conditioning

Find a subset S such that the remaining graph becomes a tree
For each possible consistent assignment to S
  remove inconsistent values from domains of remaining variables
  solve the remaining CSP which has a tree structure

- Cut-set size c gives runtime

$$d^c . (n-c)d^2$$

Combinations of values for the variables in S

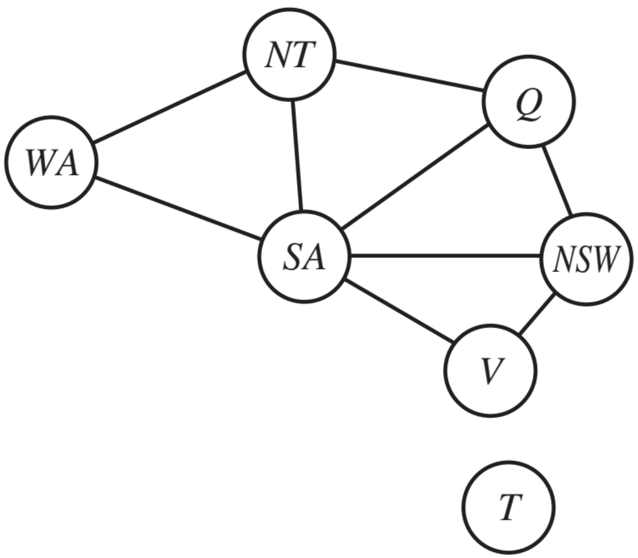
For each combination we must solve a tree problem of size $n - c$

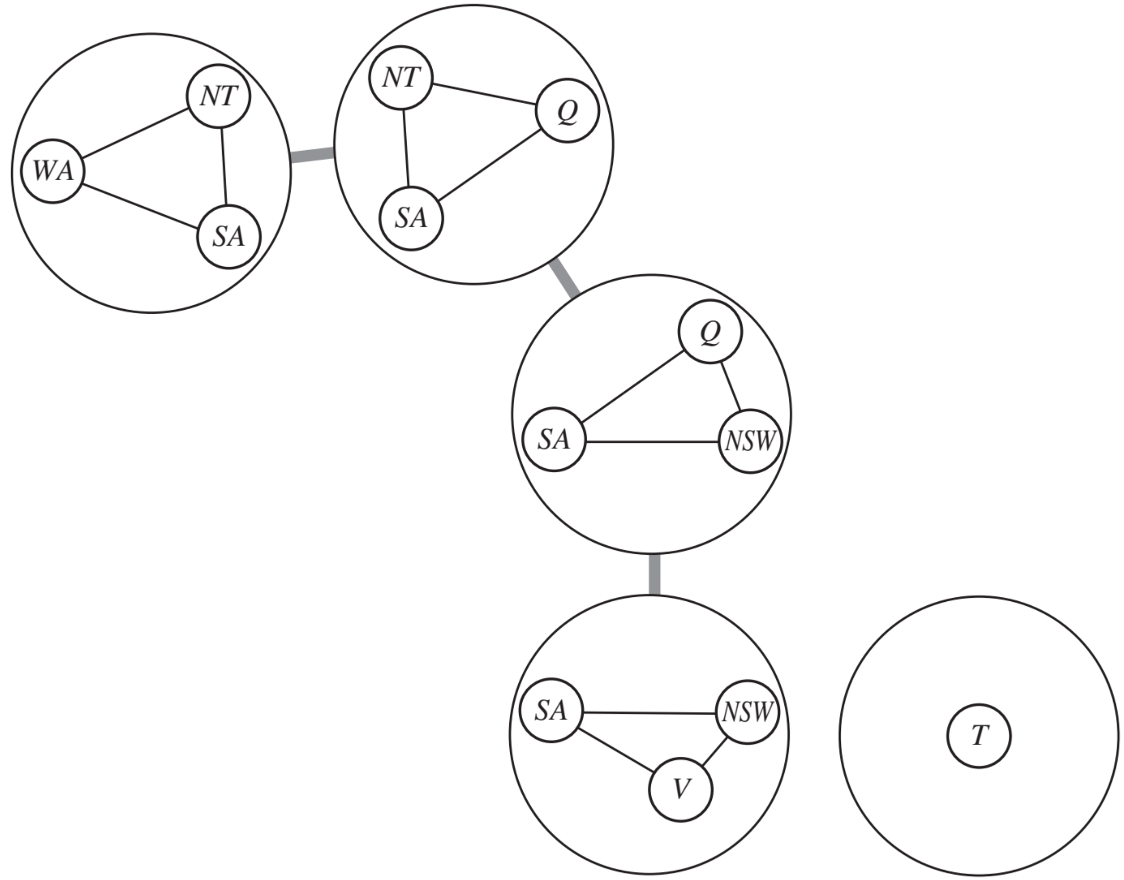
  - Very fast for small c
  - c can be as large as $n - 2$

Tree Decomposition

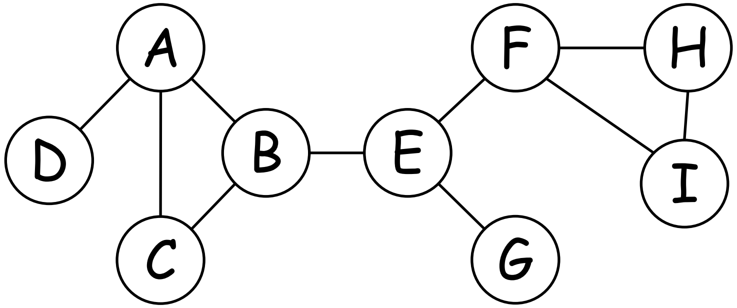
- Create a tree-structured graph of overlapping subproblems (each sub-problem as a mega-variable)
  - Every variable in the original problem appears in at least one of the subproblems
  - If two variables are connected by a constraint in the original problem, they must appear together (along with the constraint) in at least one of the subproblems
  - If a variable appears in two subproblems in the tree, it must appear in every subproblem along the path connecting those subproblems
- Solve each sub-problem (enforcing local constraints)
- Solve the tree-structured CSP over mega-variables

Tree Decomposition

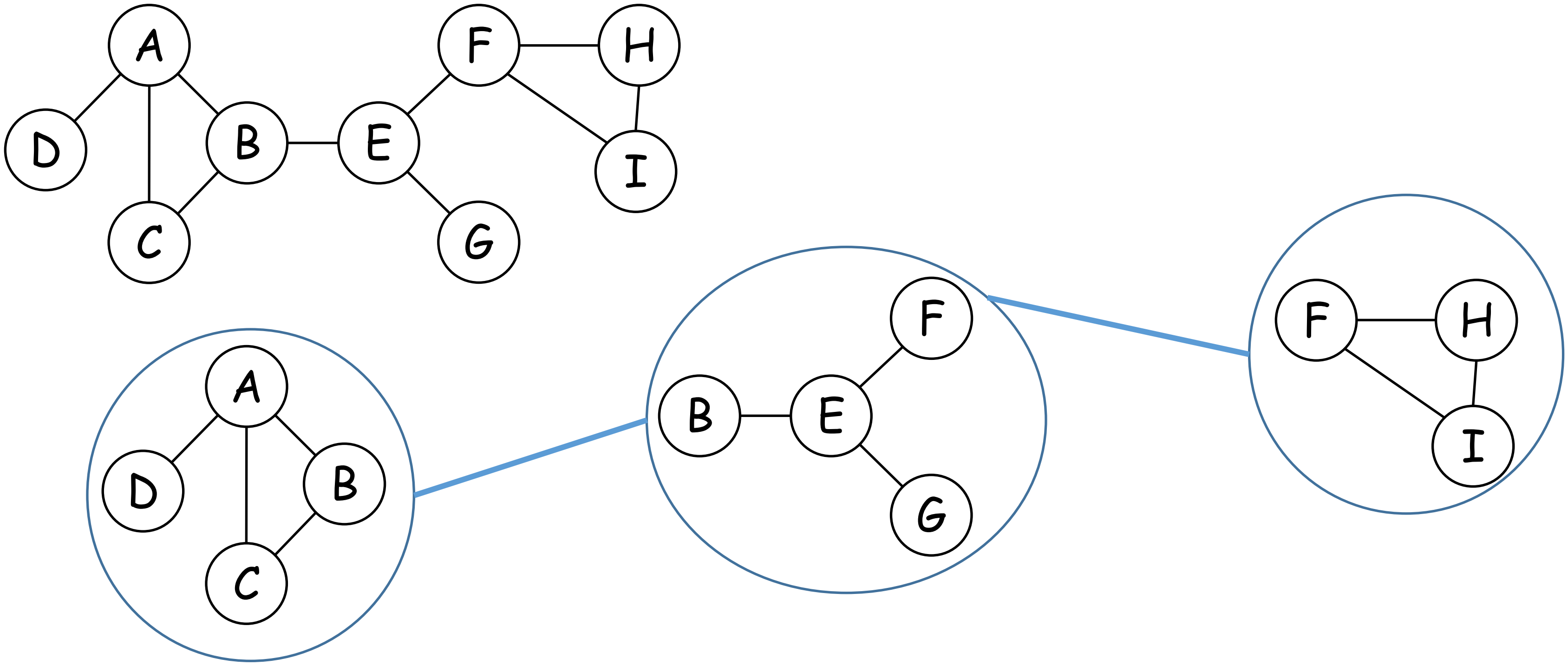




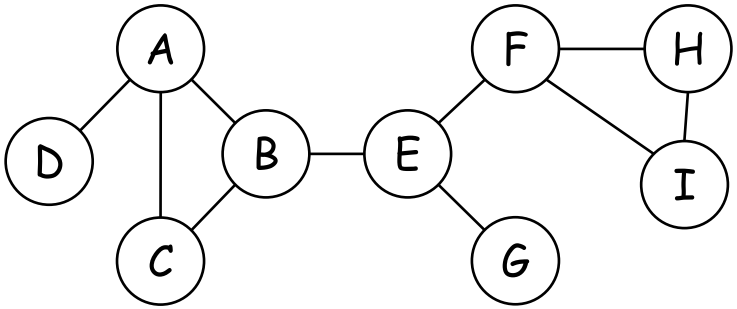
Tree Decomposition: Example

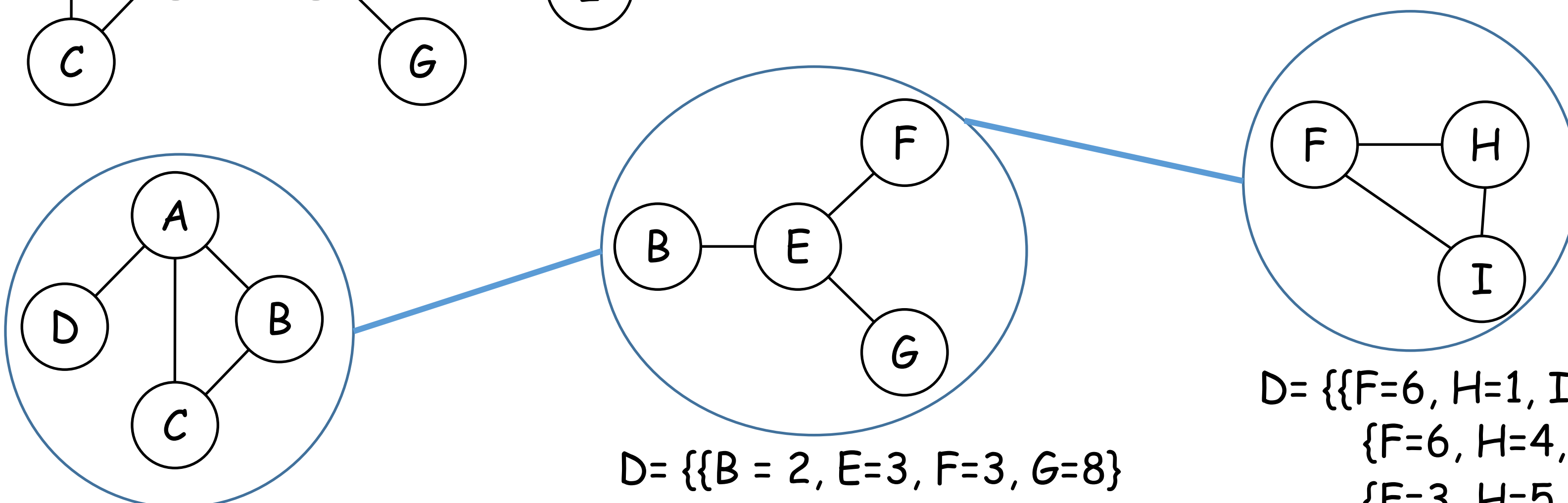


Tree Decomposition: Example



Tree Decomposition: Example



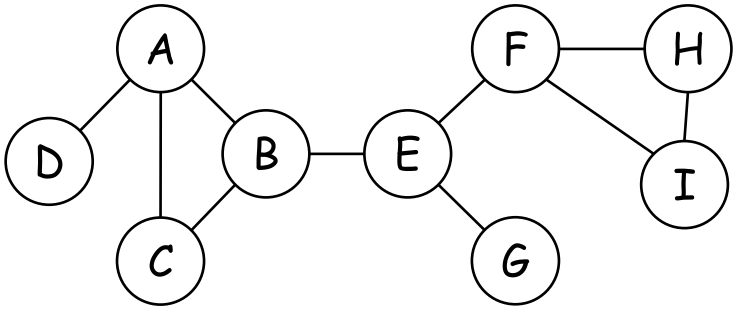


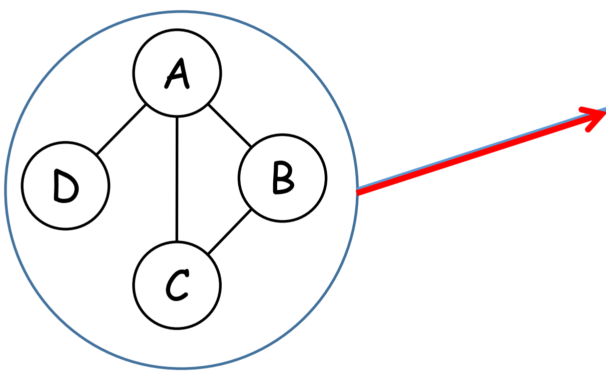
D= {{A = 2, B=1, C=3, D=1}
{A = 3, B=2, C=5, D=4}}

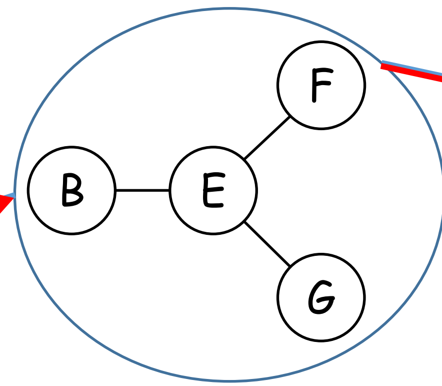
D= {{B = 2, E=3, F=3, G=8}
{B = 3, E=3, F=9, G=1}
{B = 2, E=4, F=6, G=7}}

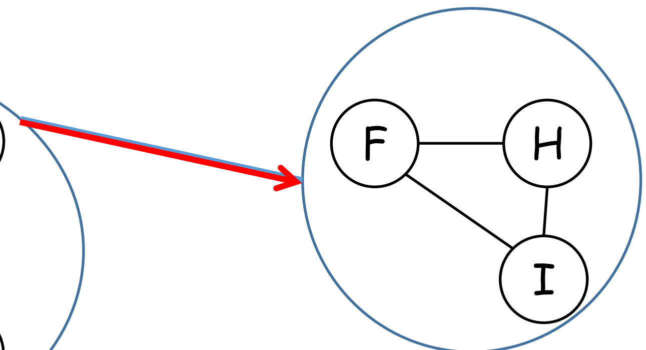
D= {{F=6, H=1, I=2}
{F=6, H=4, I=1}
{F=3, H=5, I=7}}
{F=7, H=6, I=4}}

Tree Decomposition: Example







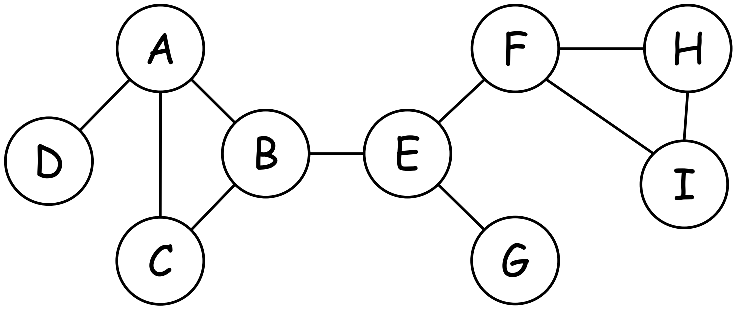


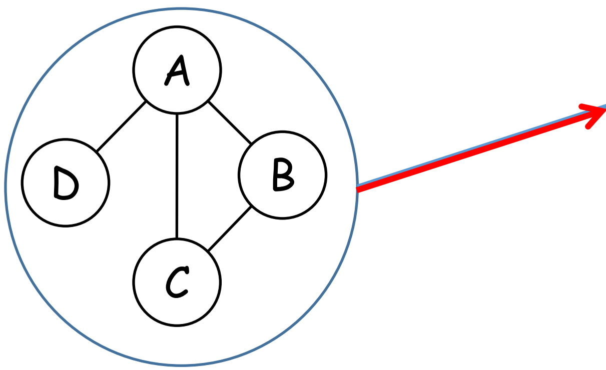
D= {{A = 2, B=1, C=3, D=1}
{A = 3, B=2, C=5, D=4}}

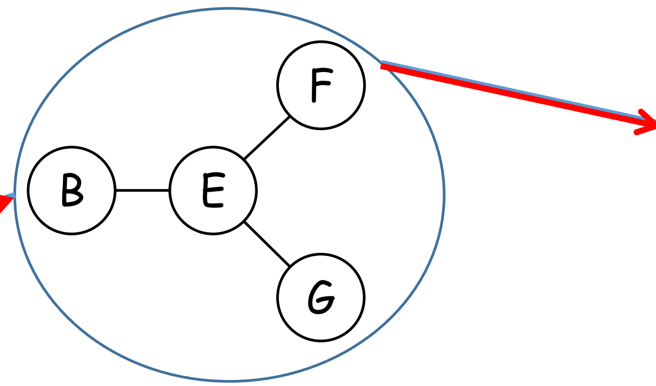
D= {{B = 2, E=3, F=3, G=8}
{B = 3, E=3, F=9, G=1}
{B = 2, E=4, F=6, G=7}}

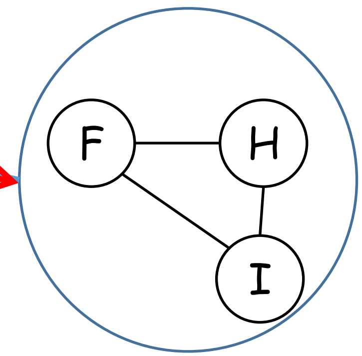
D= {{F=6, H=1, I=2}
{F=6, H=4, I=1}
{F=3, H=5, I=7}}
{F=7, H=6, I=4}}

Tree Decomposition: Example







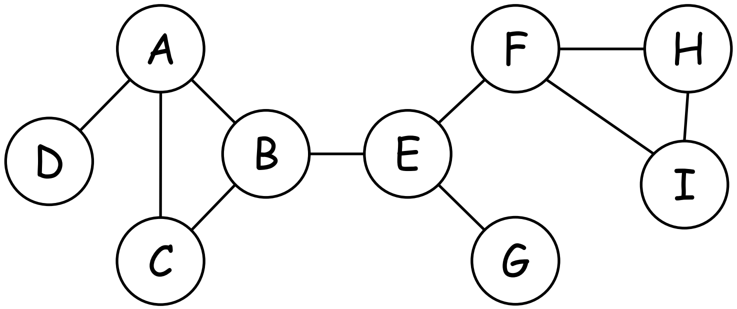


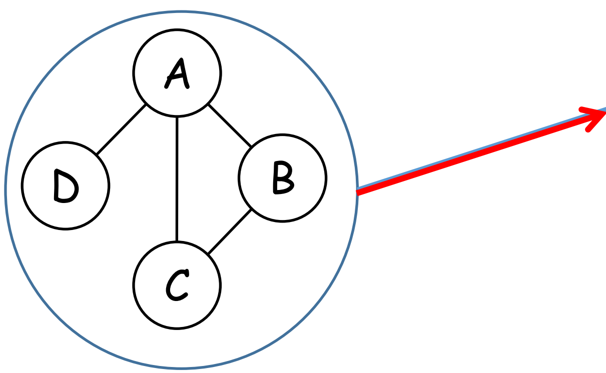
D= {{A = 2, B=1, C=3, D=1}
{A = 3, B=2, C=5, D=4}}

D= {{B = 2, E=3, F=3, G=8}
~~{B = 3, E=3, F=9, G=1}~~
{B = 2, E=4, F=6, G=7}}

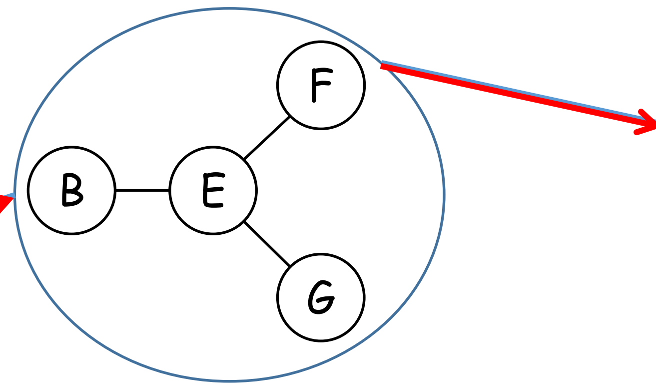
D= {{F=6, H=1, I=2}
{F=6, H=4, I=1}
{F=3, H=5, I=7}}
{F=7, H=6, I=4}}

Tree Decomposition: Example

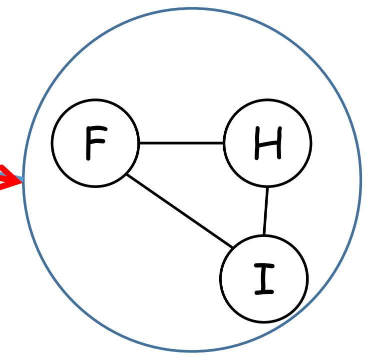




D= {~~{A = 2, B=1, C=3, D=1}~~
{A = 3, B=2, C=5, D=4}}

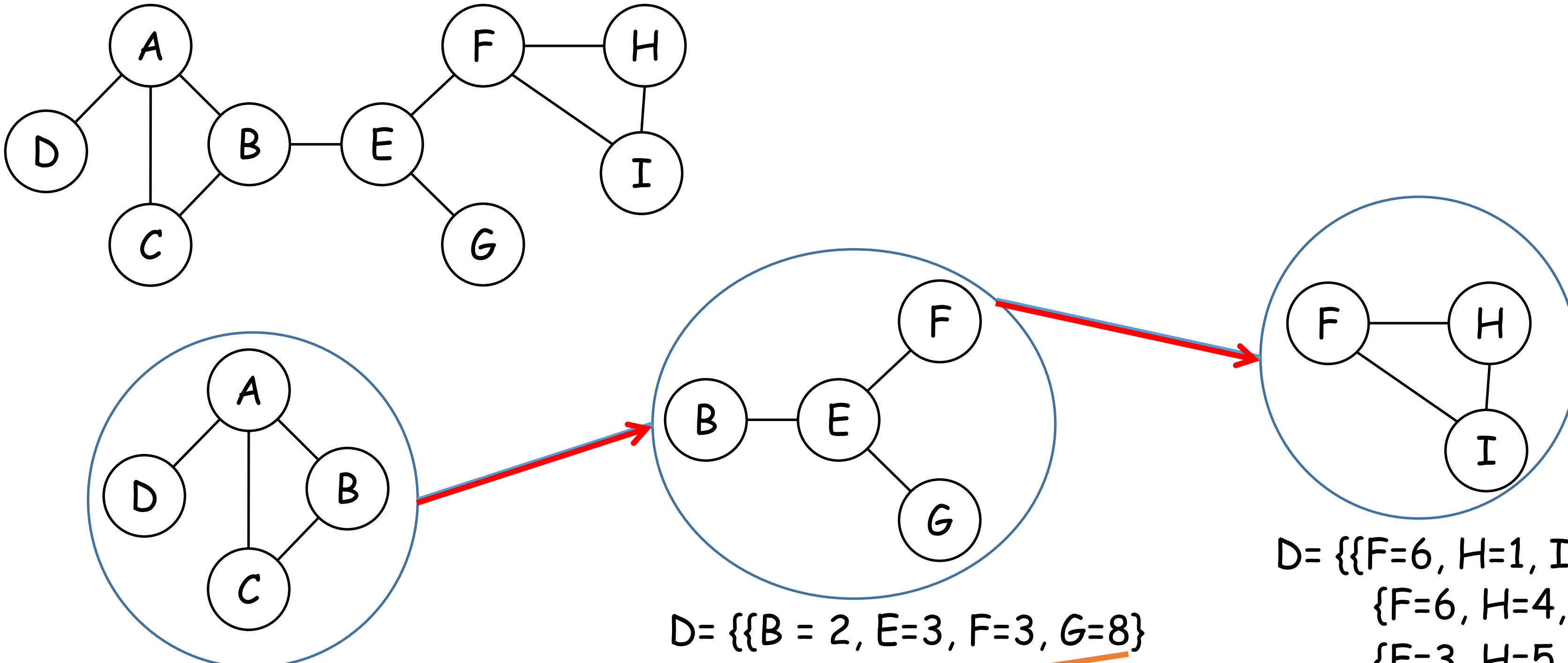


D= {{B = 2, E=3, F=3, G=8}
~~{B = 3, E=3, F=9, G=1}~~
{B = 2, E=4, F=6, G=7}}



D= {{F=6, H=1, I=2}
{F=6, H=4, I=1}
{F=3, H=5, I=7}}
{F=7, H=6, I=4}}

Tree Decomposition: Example



D= {{~~A = 2, B=1, C=3, D=1}~~
{A = 3, B=2, C=5, D=4}}

D= {{B = 2, E=3, F=3, G=8}
~~{B = 3, E=3, F=9, G=1}~~
{B = 2, E=4, F=6, G=7}}

D= {{F=6, H=1, I=2}
{F=6, H=4, I=1}
{F=3, H=5, I=7}}
{F=7, H=6, I=4}}

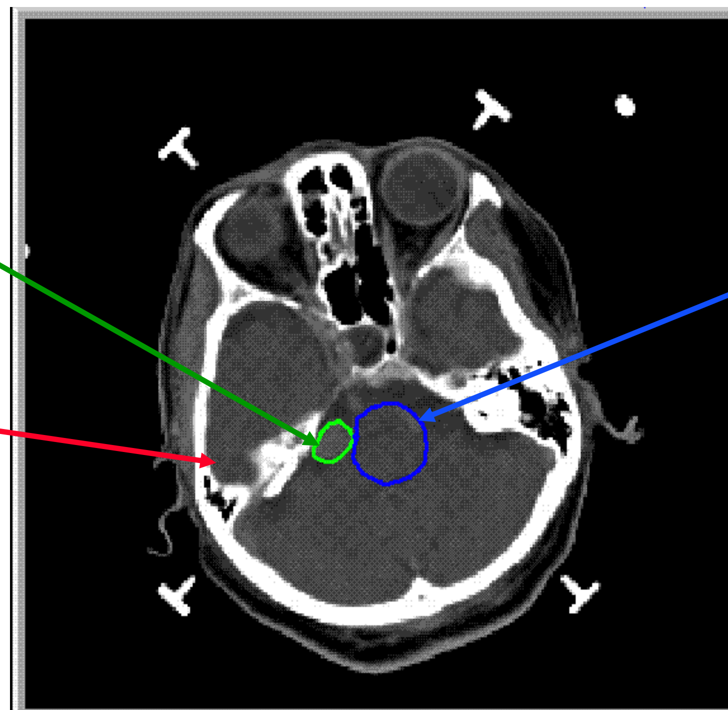
{A = 3, B=2, C=5, D=4, B = 2, E=4, F=6, G=7, F=6, H=4, I=1}

Applications of CSP

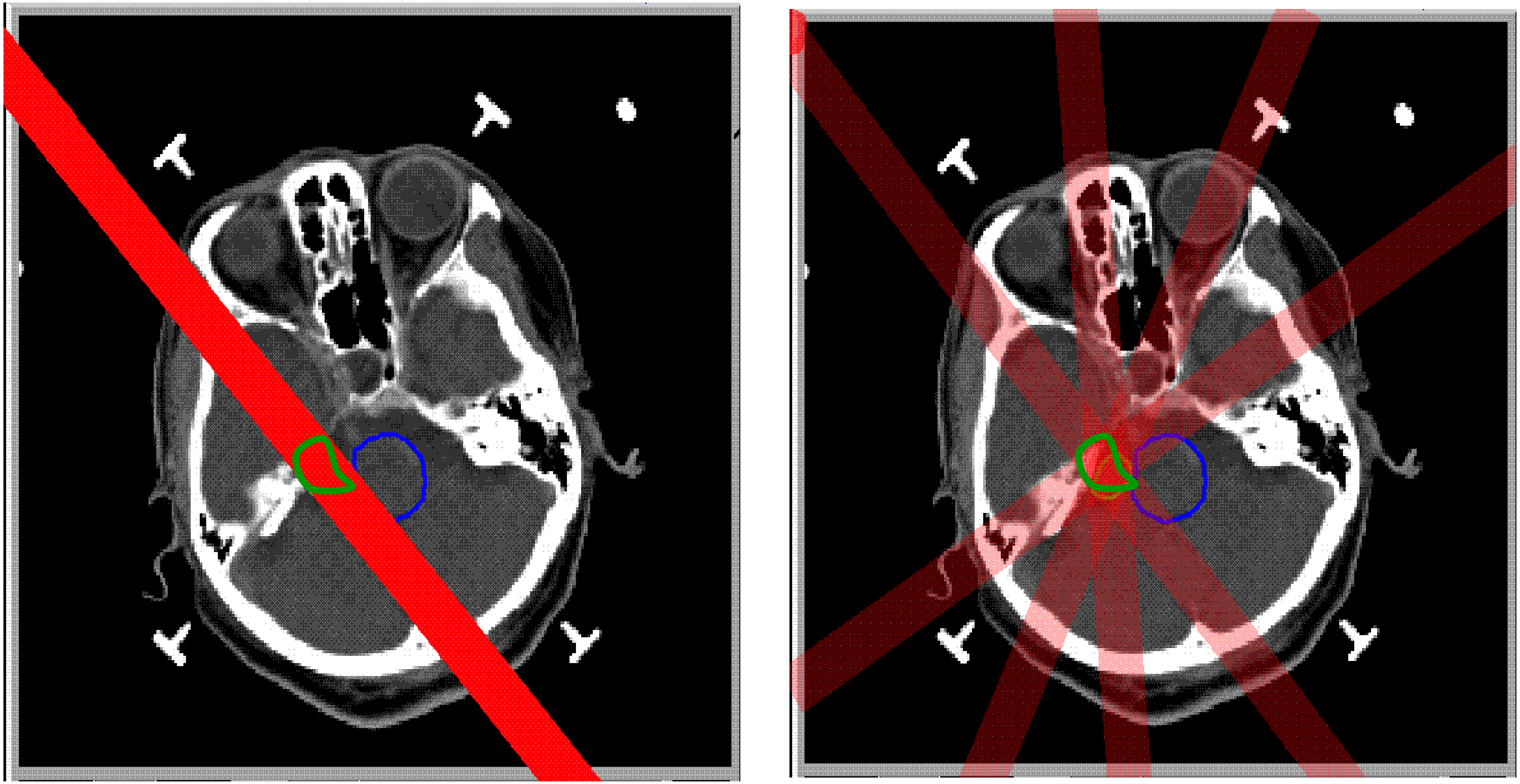
- CSP techniques are widely used
- Applications include:
  - Crew assignments to flights
  - Management of transportation fleet
  - Flight/rail schedules
  - Job shop scheduling
  - Task scheduling in port operations
  - Design, including spatial layout design
  - Radiosurgical procedures

Radiosurgery

Minimally invasive procedure that uses a beam of radiation as an ablative surgical instrument to destroy tumors



Problem



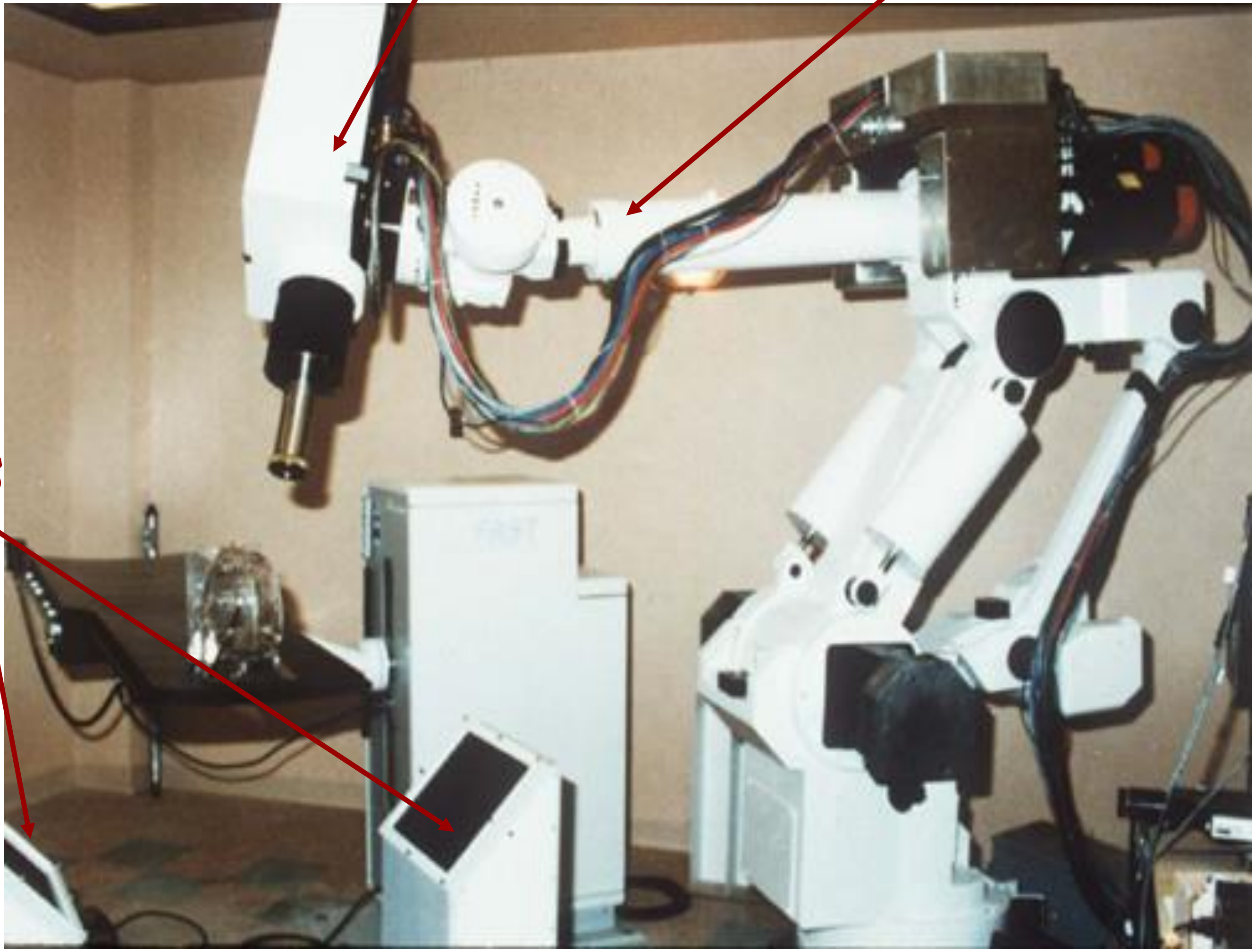
Burn tumor without damaging healthy tissue

The CyberKnife

linear accelerator

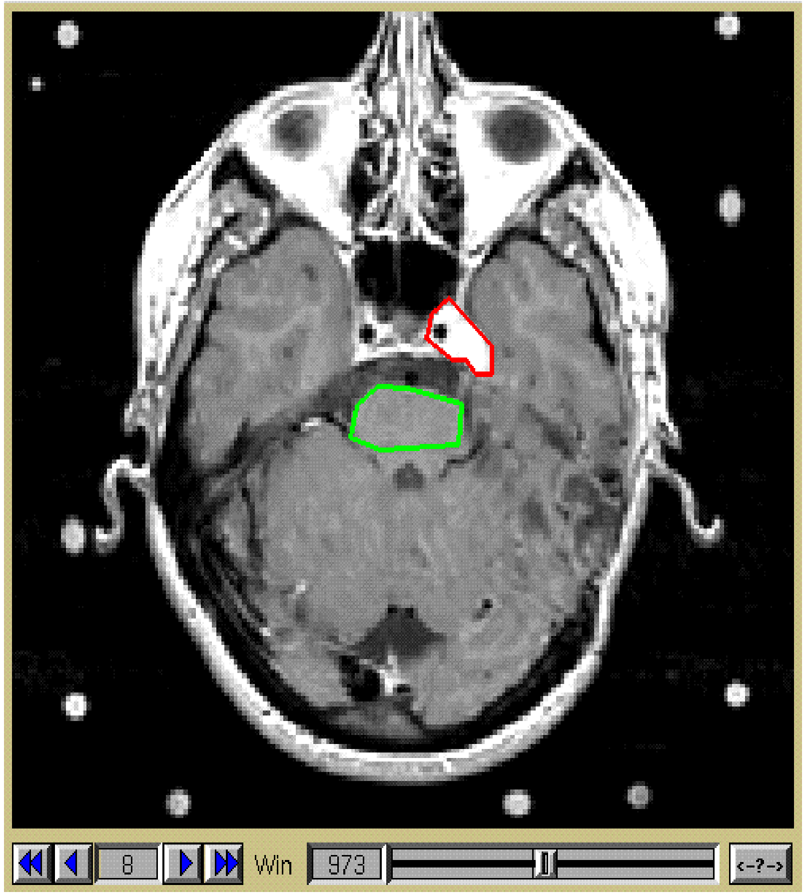
robot arm

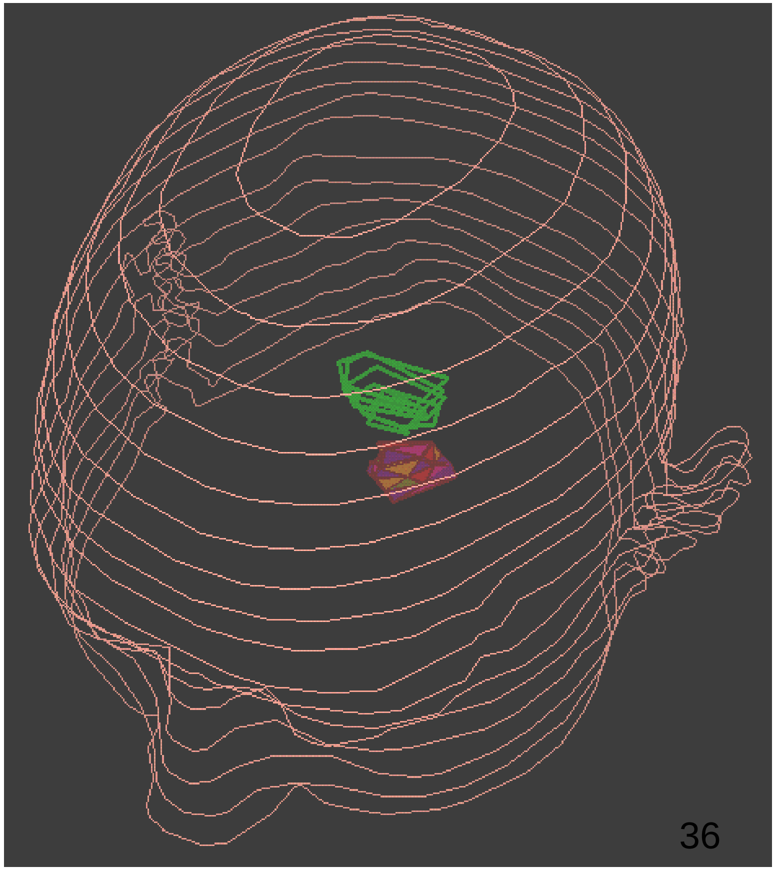
X-Ray
cameras



Inputs

1) Regions of interest

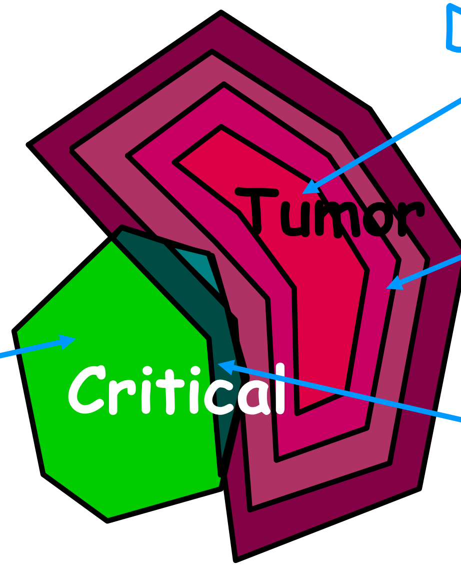




Inputs

2) Dose constraints

Dose to tumor

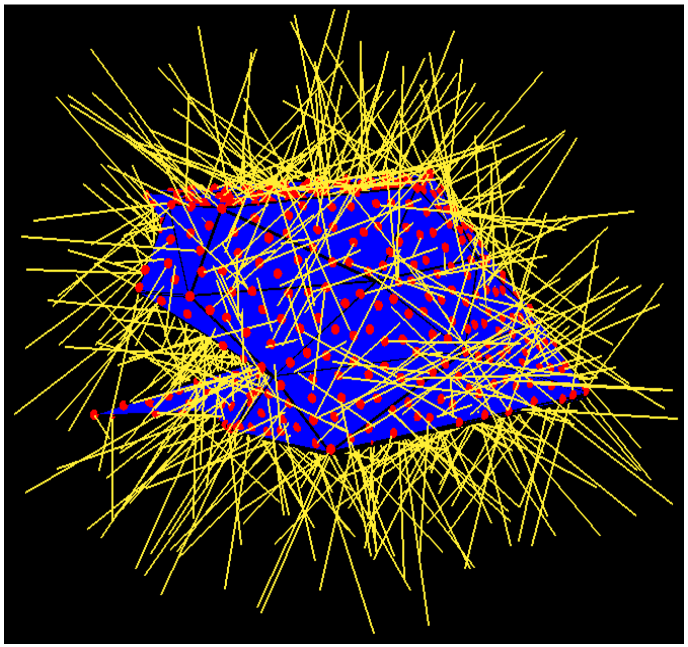


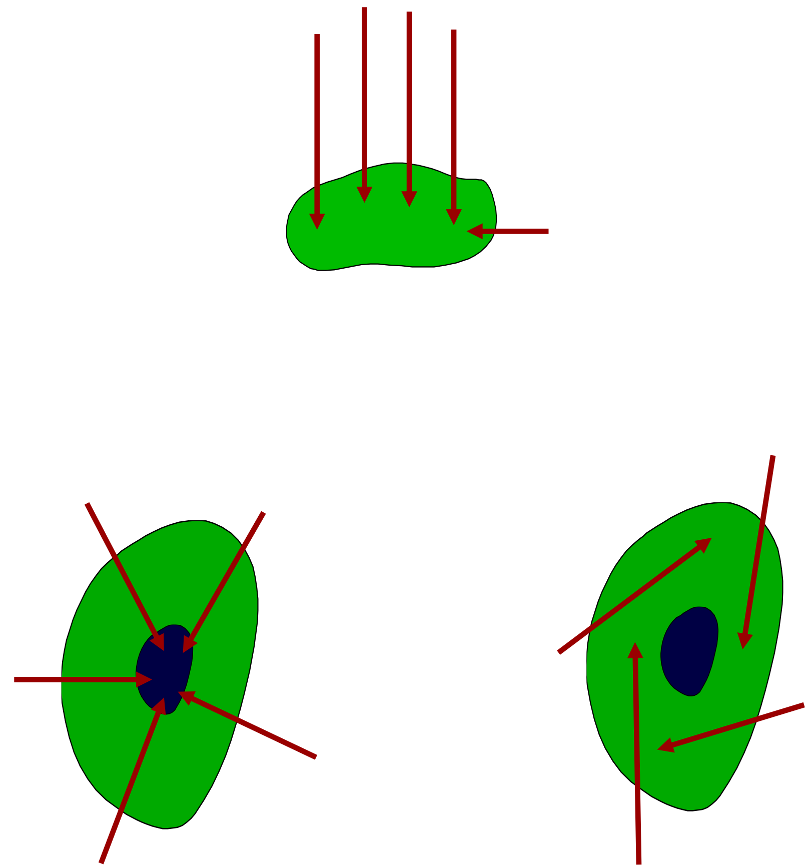
Falloff of dose around tumor

Dose to critical structure

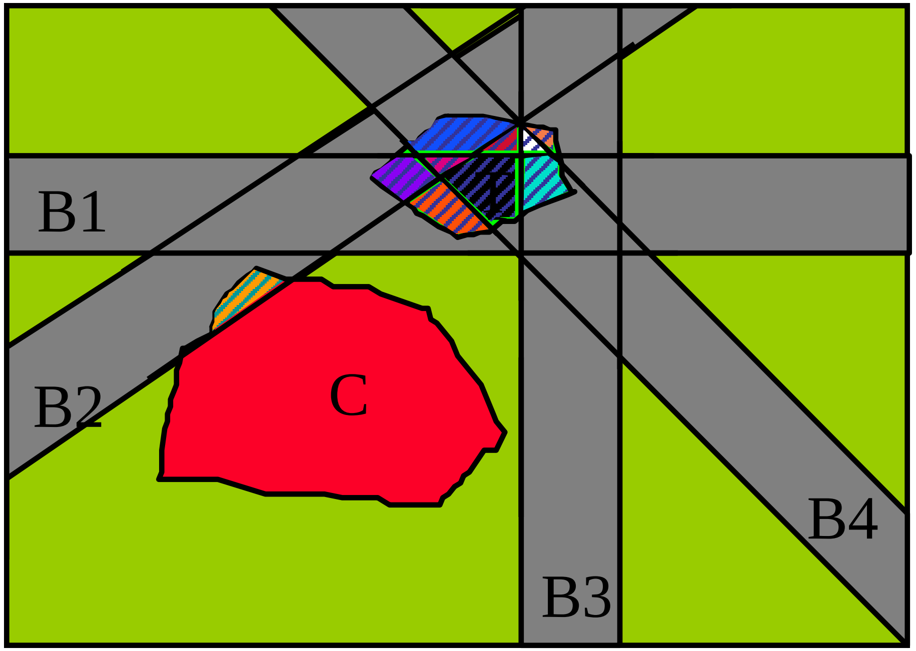
Falloff of dose in critical structure

Beam Sampling





Constraints



- $2000 \leq \text{Tumor} \leq 2200$
  - $2000 \leq B2 + B4 \leq 2200$
  - $2000 \leq B4 \leq 2200$
  - $2000 \leq B3 + B4 \leq 2200$
  - $2000 \leq B3 \leq 2200$
  - $2000 \leq B1 + B3 + B4 \leq 2200$
  - $2000 \leq B1 + B4 \leq 2200$
  - $2000 \leq B1 + B2 + B4 \leq 2200$
  - $2000 \leq B1 \leq 2200$
  - $2000 \leq B1 + B2 \leq 2200$
- $0 \leq \text{Critical} \leq 500$
  - $0 \leq B2 \leq 500$

$2000 < \text{Tumor} < 2200$

$2000 < B2 + B4 < 2200$
$2000 < B4 < 2200$
$2000 < B3 + B4 < 2200$
$2000 < B3 < 2200$
$2000 < B1 + B3 + B4 < 2200$
$2000 < B1 + B4 < 2200$
$2000 < B1 + B2 + B4 < 2200$
$2000 < B1 < 2200$
$2000 < B1 + B2 < 2200$

$\Rightarrow$

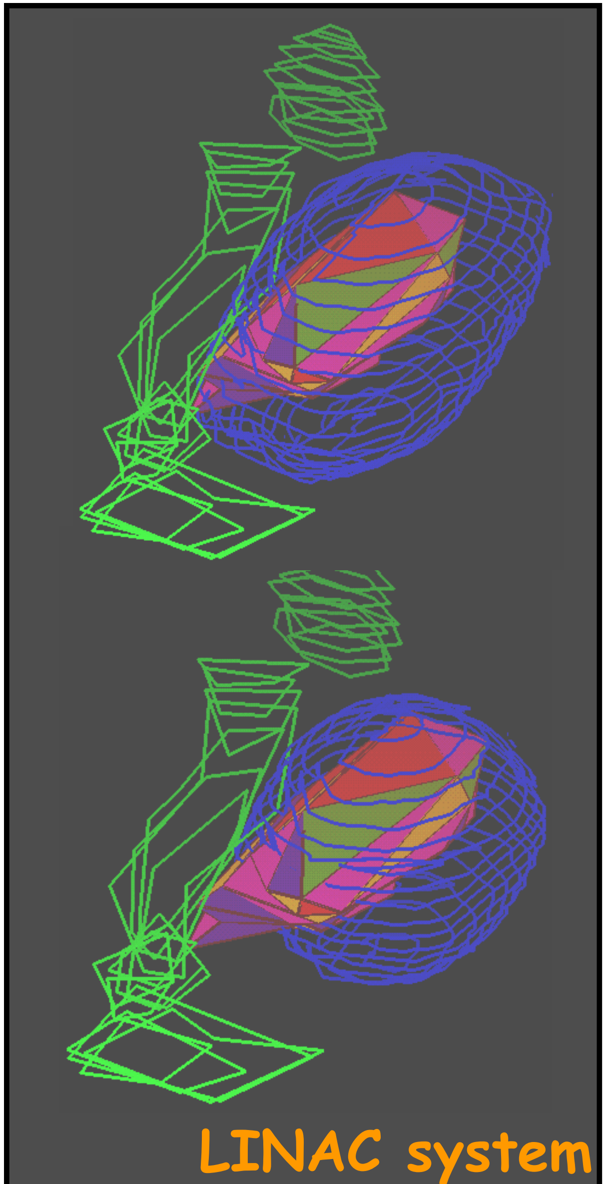
$2000 < \text{Tumor} < 2200$

$2000 < B4$

$2000 < B3$
$B1 + B3 + B4 < 2200$

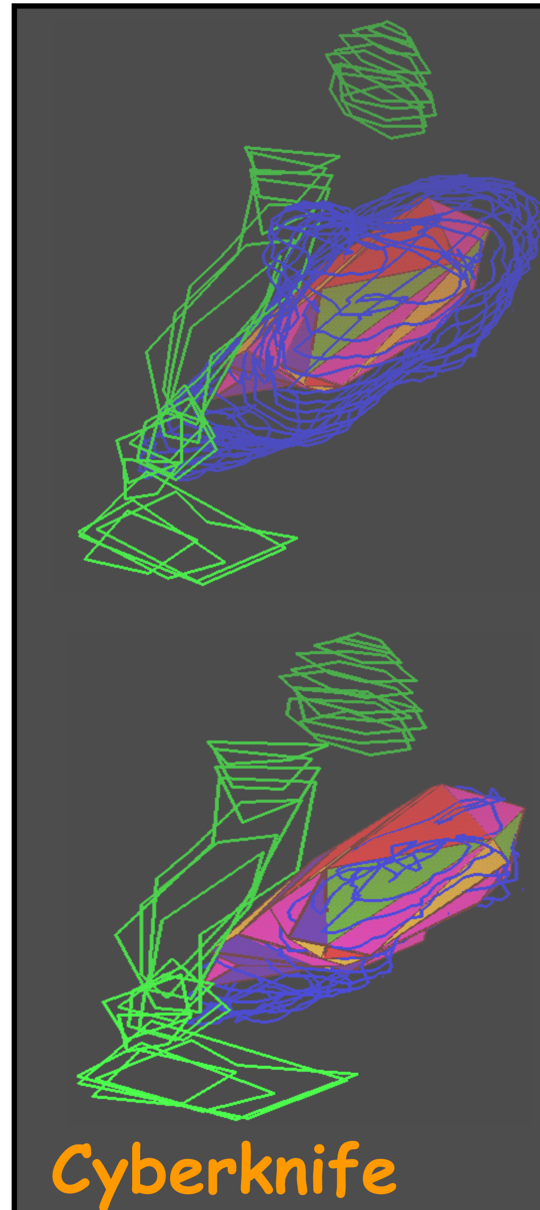
$B1 + B2 + B4 < 2200$
$2000 < B1$

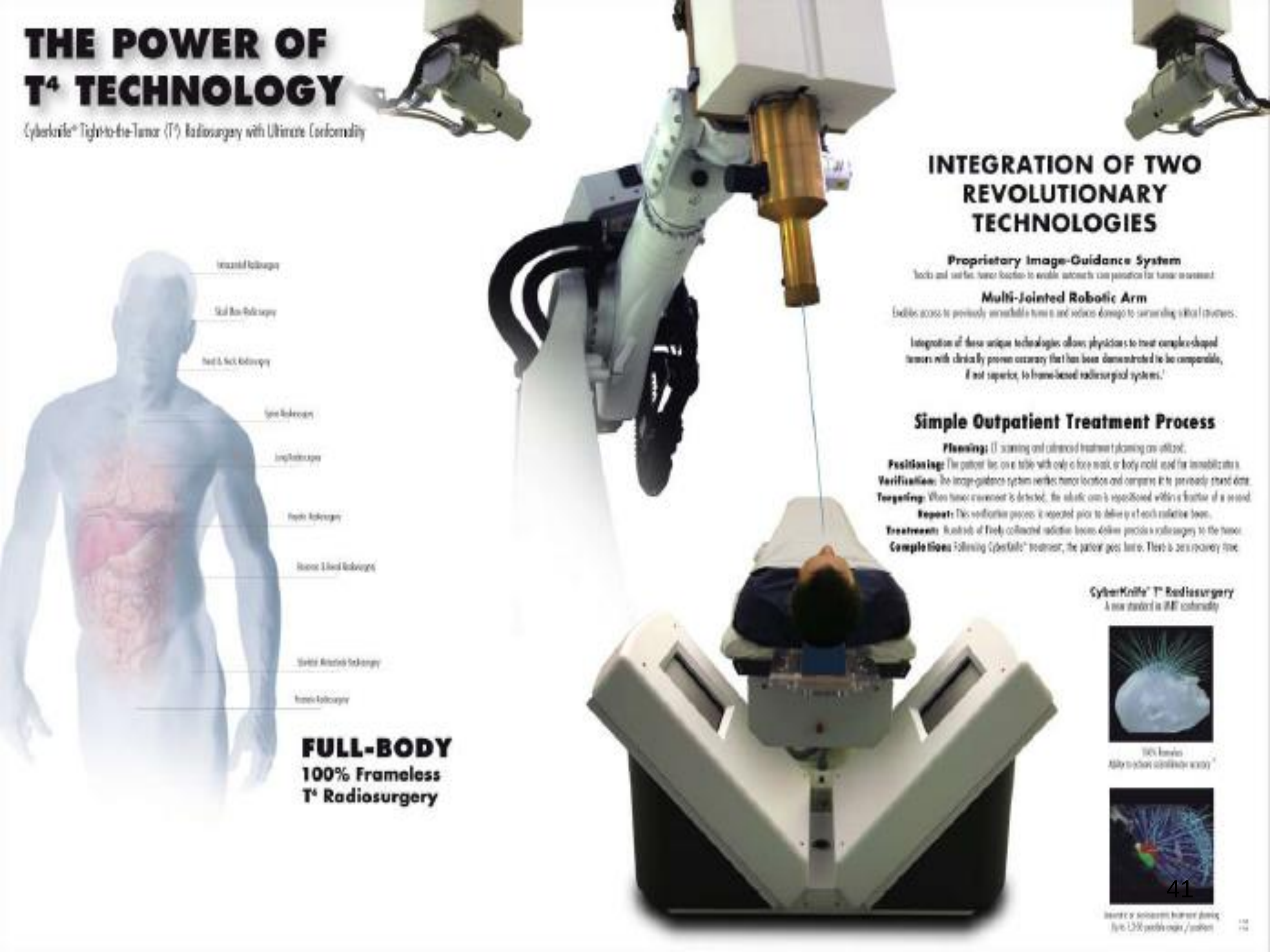
Case Results



50% Isodose Surface

80% Isodose Surface




THE POWER OF
T4 TECHNOLOGY
CyberKnife® Tight-to-the-Tumor (T4) Radiosurgery with Ultimate Conformality
FULL-BODY
100% Frameless
T4 Radiosurgery
INTEGRATION OF TWO REVOLUTIONARY TECHNOLOGIES
Proprietary Image-Guidance System
Tracks and verifies tumor location to enable automatic compensation for tumor movement.
Multi-Jointed Robotic Arm
Enables access to previously unreachable tumors and reduces damage to surrounding critical structures.
Integration of these unique technologies allows physicians to treat complex-shaped tumors with clinically proven accuracy that has been demonstrated to be comparable, if not superior, to frame-based radiosurgical systems.
Simple Outpatient Treatment Process
Planning: CT scanning and advanced treatment planning are utilized.
Positioning: The patient lies on a table with only a face mask or body mold used for immobilization.
Verification: The image-guidance system verifies tumor location and compares it to previously stored data.
Targeting: When tumor movement is detected, the robotic arm is repositioned within a fraction of a second.
Repeat: This verification process is repeated prior to delivery of each radiation beam.
Treatment: Hundreds of finely collimated radiation beams deliver precision radiosurgery to the tumor.
Completion: Following CyberKnife® treatment, the patient goes home. There is zero recovery time.
CyberKnife® T4 Radiosurgery