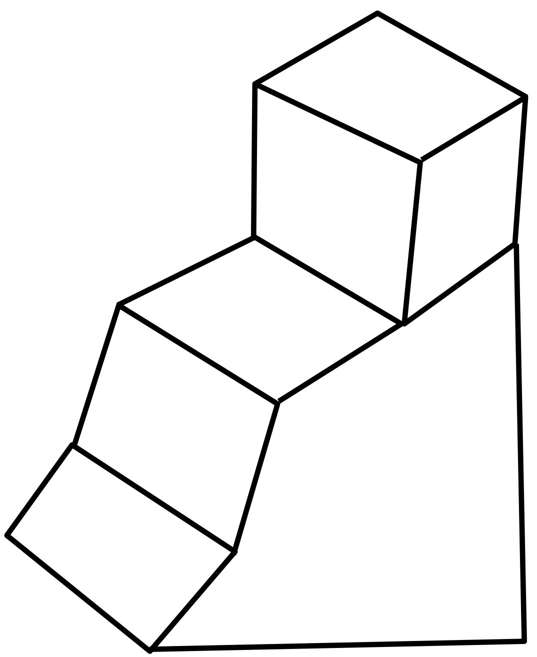
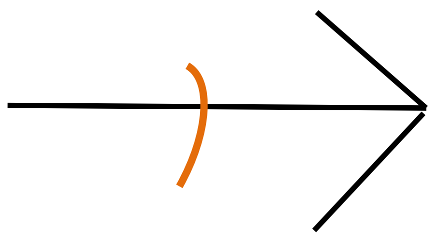
Artificial Intelligence Fundamentals

Constraints: Search, Domain Reduction

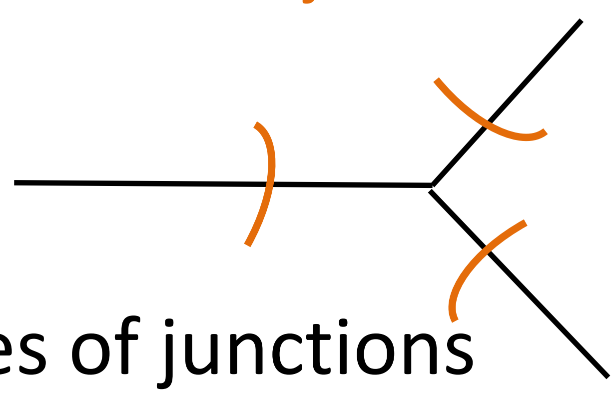
Constraints: Interpreting Line Drawings

- How many objects are in the drawing?
- Guzman ("Decomposition of a visual scene into three-dimensional bodies")- experimentalist



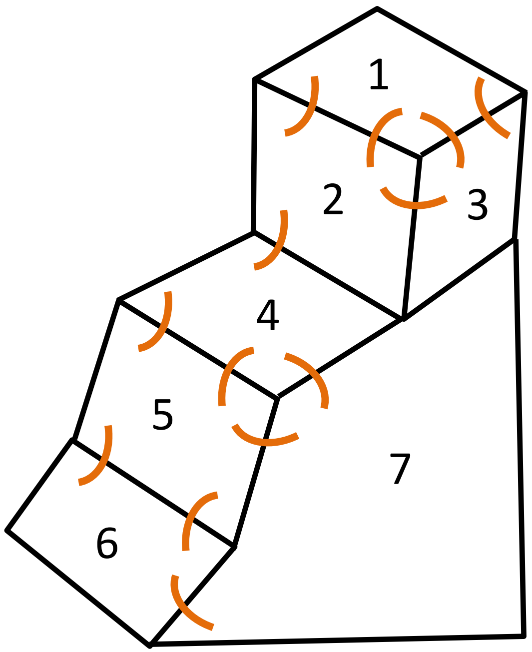


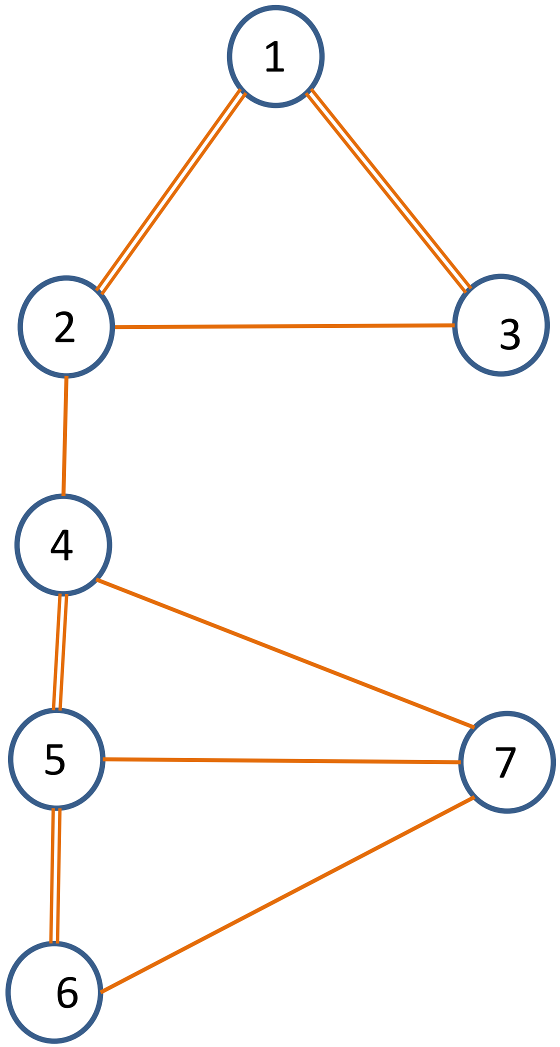
These 2 faces belongs to the same object



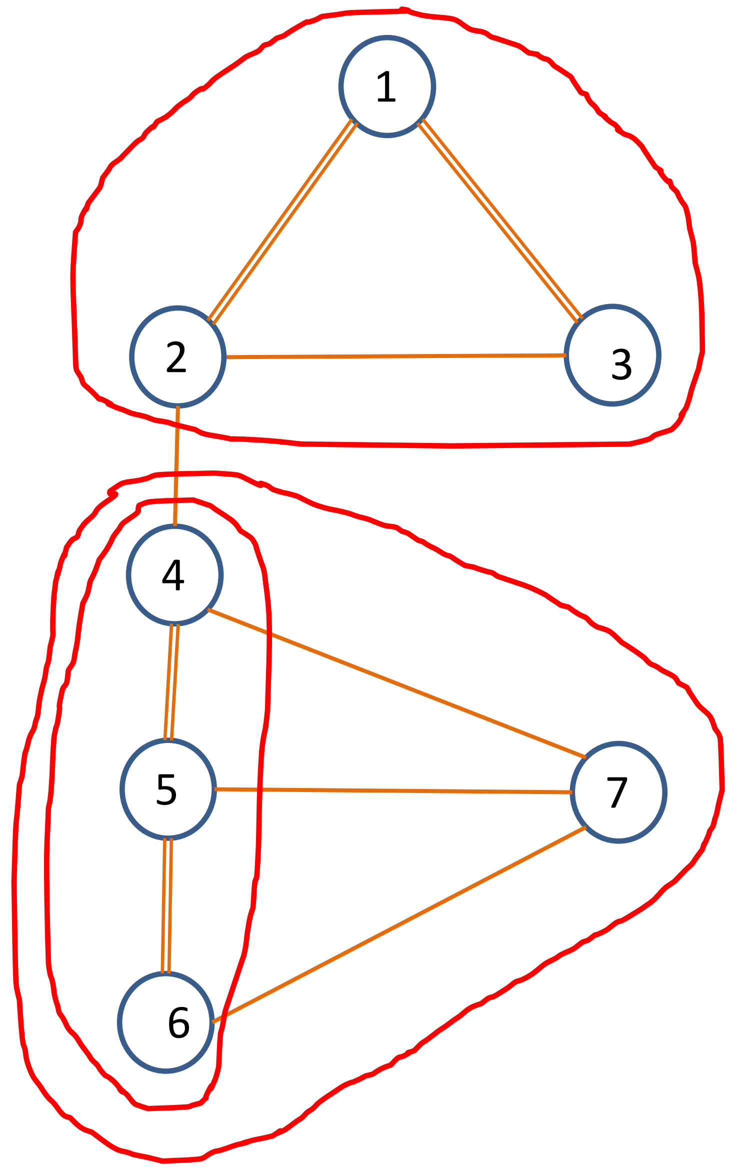
- Types of junctions

Constraints: Interpreting Line Drawings



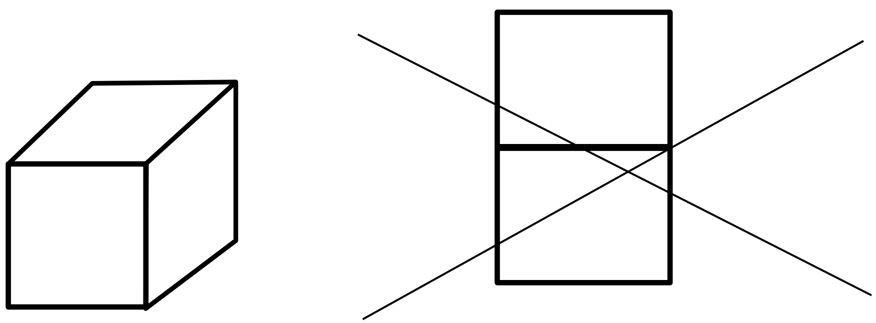


- 1 link theory -> too liberal
- 2 link theory –> too conservative
- 2 link * (repeated) – link super regions that are connected more than 2 links

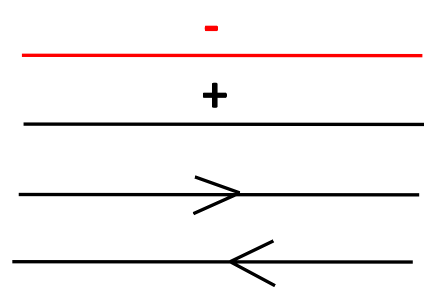


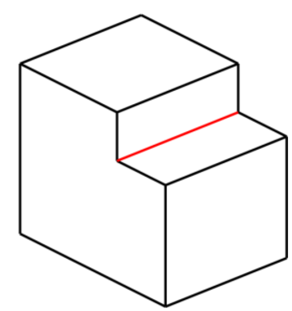
- In the final there are 2 objects
- It works because there are many 3 faced vertexes (junctions)

- Huffman – mathematician
    1. General position



    2. Trihedral – all vertexes are formed by 3 planes (3 faces)
    3. Four kinds of lines:
        1. Concave
        2. Convex
        3. Boundaries



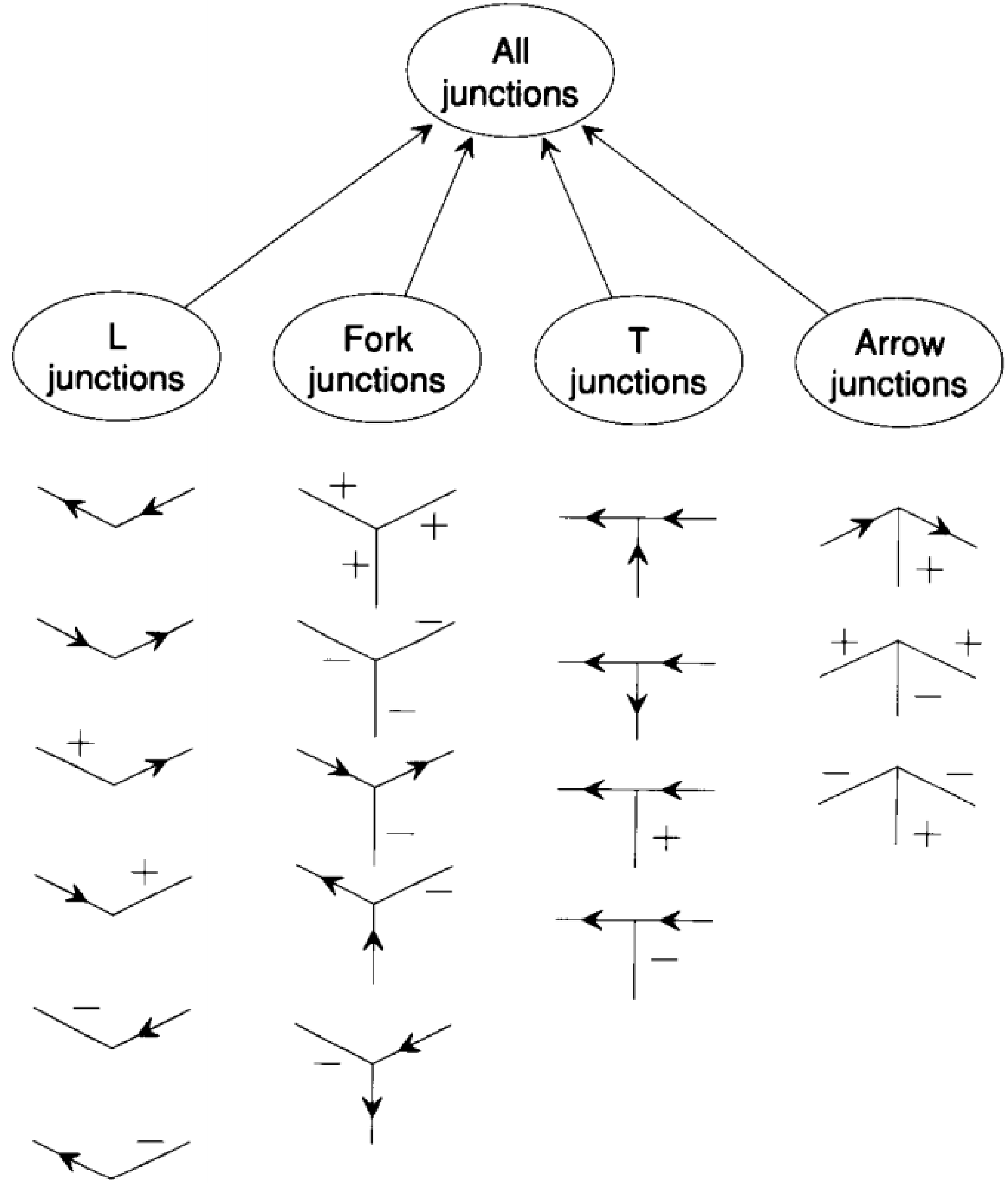


When you walk on the direction of the line you have the object in the right

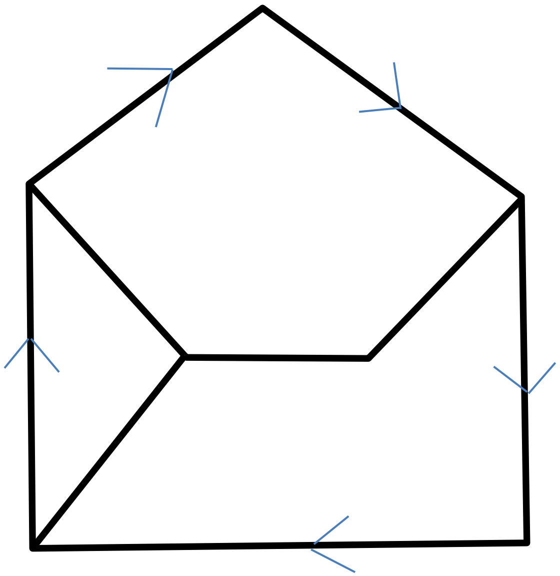
4. Without shadows and cracks
5. There are only 18 ways to arrange the labels around the junction

Vertexes Junctions

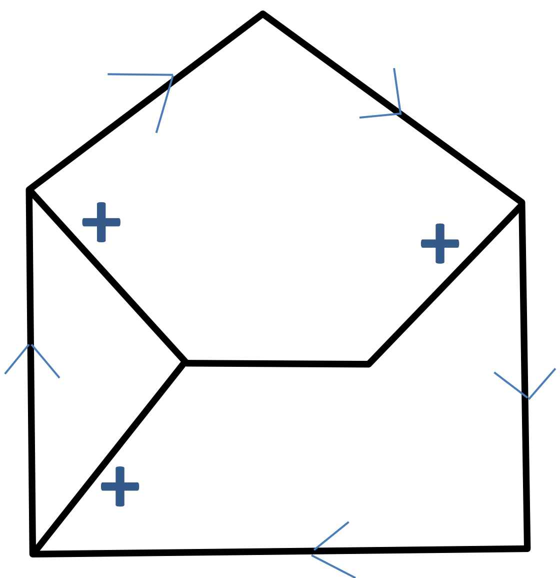
Edges Lines



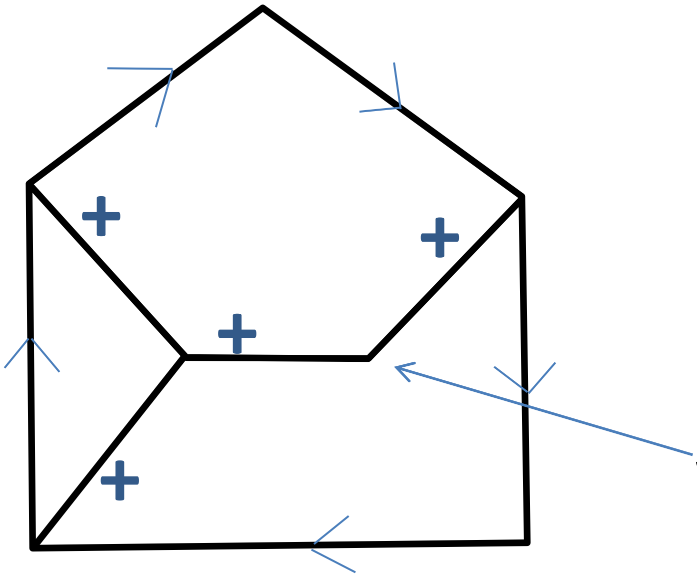
Can you build one of those object?



The object floating in space -> all boundaries are boundary lines

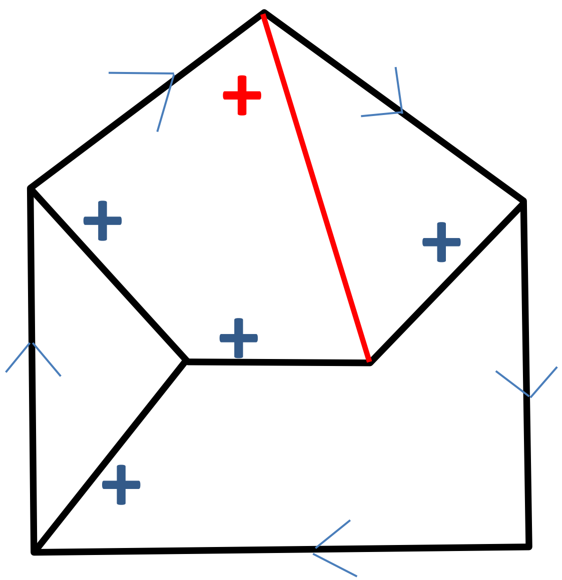


A line can't change his nature along his length -> so if it’s a + line at an end must be a + line at the other end



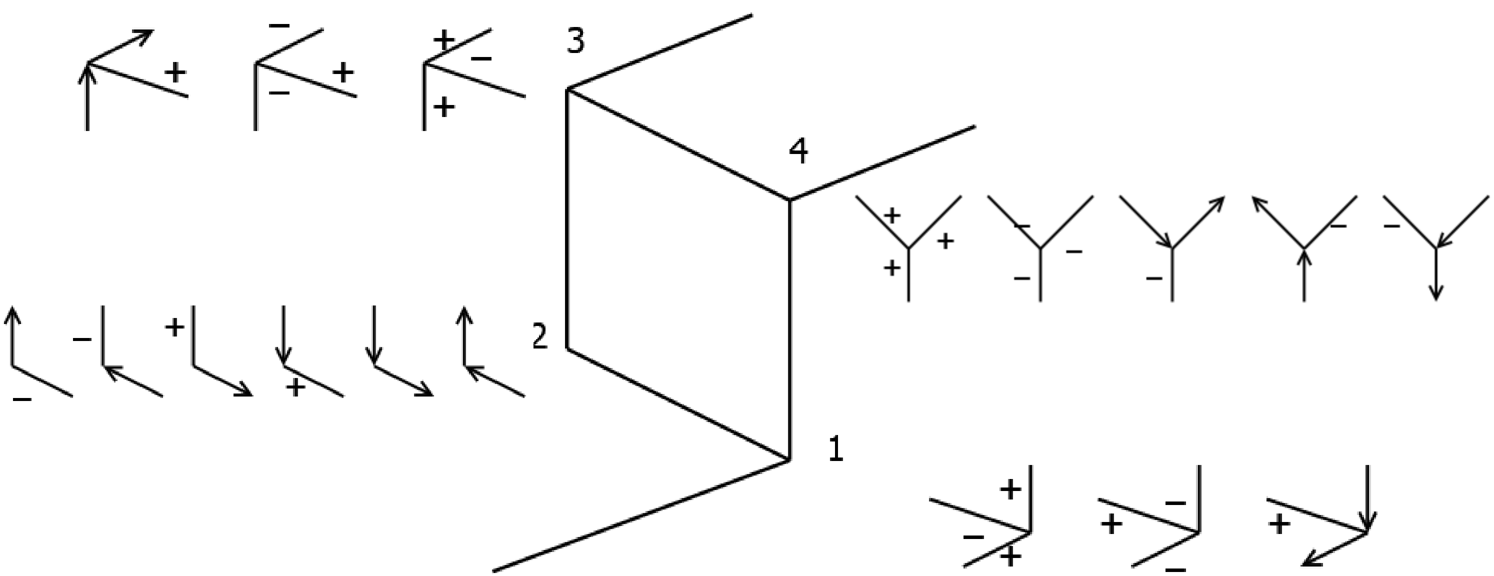
What about that junction?

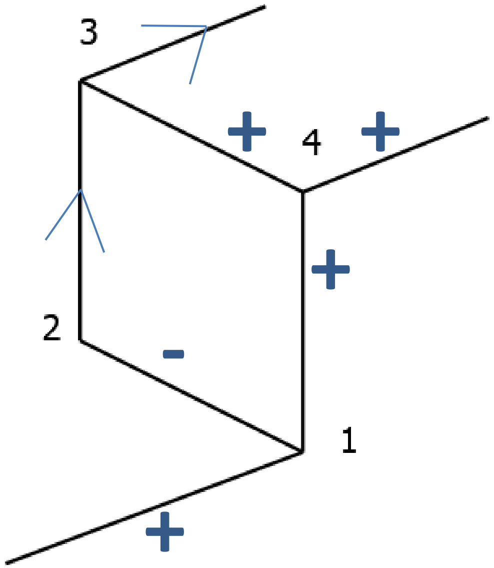
It’s an L junction with pluses at both of the lines, but there isn’t exist in the catalog -> this object can't exist in the real world



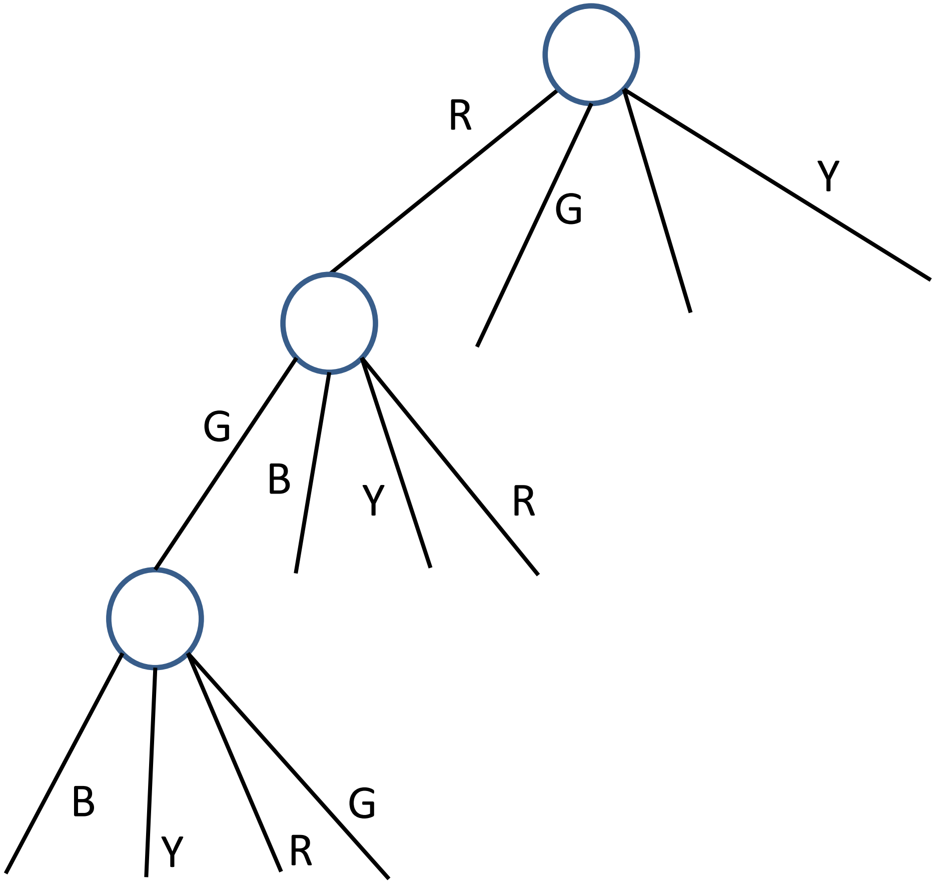
Helps to have a line like that?

Answer: No


3
4
2
1

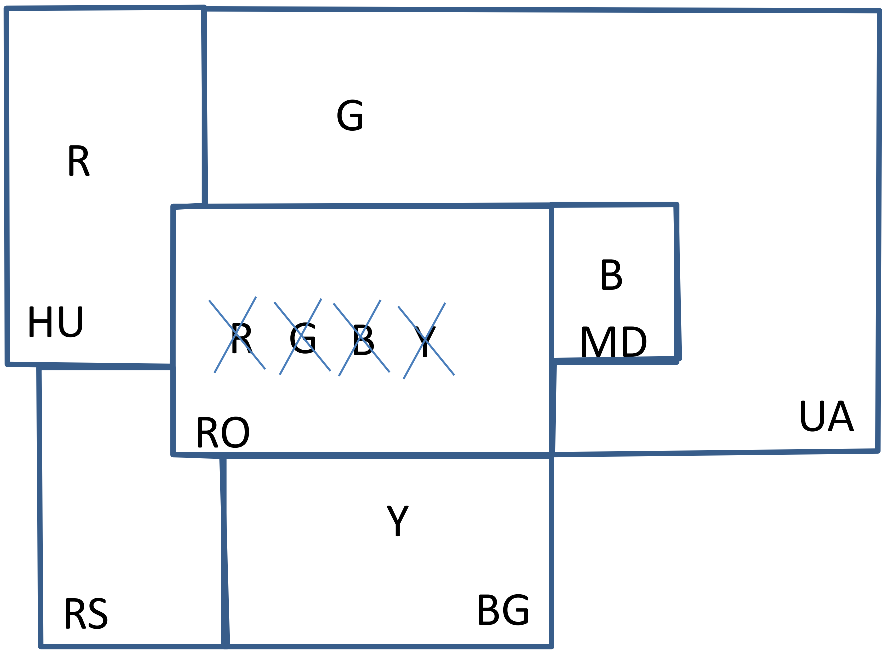

3
4
2
1
+
+
+
-
+

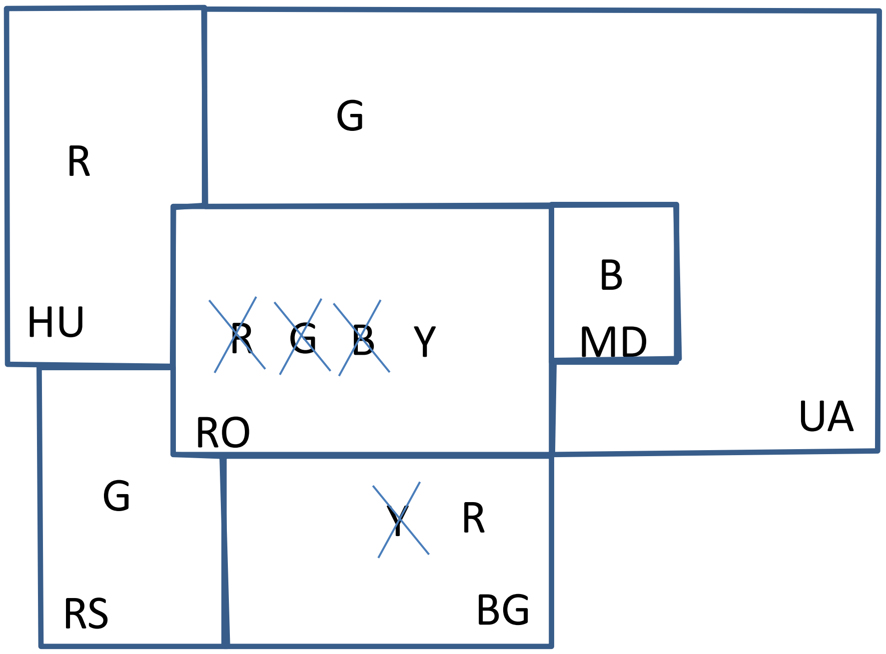
Map Coloring



- Depth first search
- 4 colors

Map coloring - Romania





R
HU
G
B
MD
~~R~~ ~~G~~ ~~B~~ Y
RO
UA
G
RS
~~Y~~ R
BG

Domain reduction vocabulary

- Variable *X* – something that can have an assignment
- Value *V* – something that can be assigned
- Domain *D* – bag of values
- Constraint *C* – a limit on variable values

  - Countries – variables
  - Colours – values
  - Domains – the remaining colours
  - Constraints – map constraints

Domain reduction agorithm

for each Depth First Search assignment

    for each variable X_i $\boldsymbol{considered}$

        for each v_i in D_i

            for each constraint $C\left(v_i, v_j\right)$ where $v_j \in D_j$

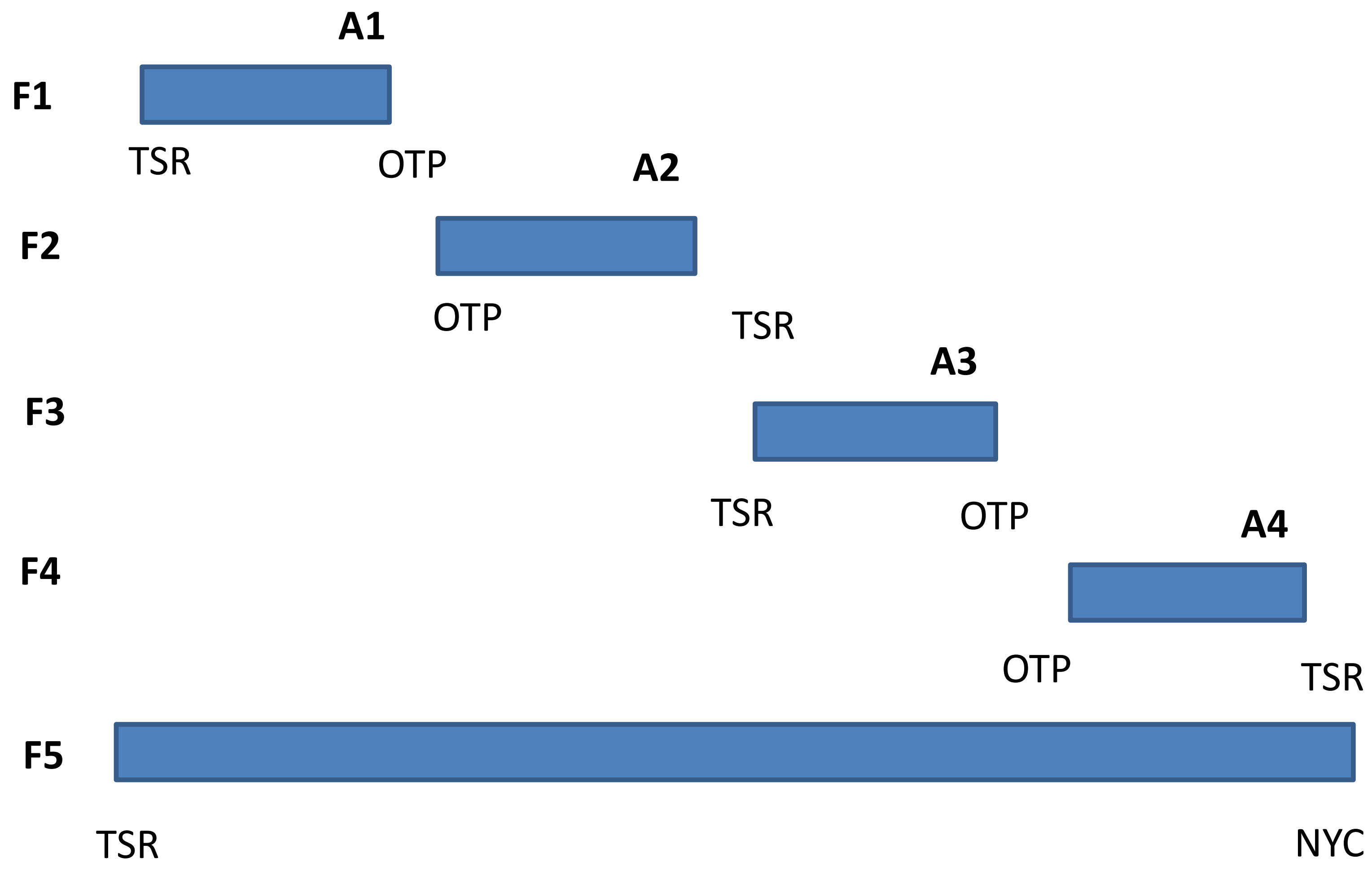
                if not $\exists v_j \ni \; C(v_i, v_j)$ satisfied remove v_i from D_i

                if D_i is empty then BACKTRACK

Domain reduction algorithm

- We can consider:
  1. Nothings
  2. Assignments only
  3. Check neighbors only – 406
  4. Propagate checking through *Variables* with *D* reduced to 1 value
  5. Propagate checking through *Variables* with reduced *Domains – 0*
  6. Everything

Example – airline scheduling



- Constraints
  - 2 planes cannot fly in the same moment in two different flights
- Minimum grount time constraint
  - Other types of constraints

- Question ?
  - How many planes are needed in order to satisfy a schedule?

Rules for good resource allocation

- Always use the most constraints first
- Propagate through domain reduce to a single value
- If you try to figure out what is the minimum number of resources needed, converge from overresource and from underresource and see what interval remains (squeeze to a small interval)

Related resources

- http://ocw.mit.edu/courses/electrical-engineering-and-computer-science/6-034-artificial-intelligence-fall-2010/exams/MIT6_034F10_quiz2_2007.pdf

Readings

- Artificial Intelligence (3rd Edition), Patrick Winston, Chapter 12