Artificial Intelligence Fundamentals

Course materials

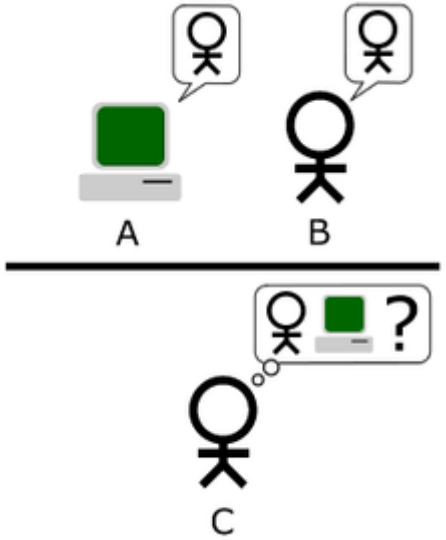
- Online course – Peter Norvig and Sebastian Thrun, “Intro to Artificial Intelligence”, https://www.udacity.com/course/cs271
- Stuart Russell, Peter Norvig, “Artificial Intelligence – A modern approach”,(3rd Edition)
- Online course – Patrick Winston, “Artificial Intelligence”, http://ocw.mit.edu/courses/electrical-engineering-and-computer-science/6-034-artificial-intelligence-fall-2010/
- Artificial Intelligence (3rd Edition) Paperback, Patrick Winston, 1992, http://courses.csail.mit.edu/6.034f/ai3/

What is AI ?

Systems that think like humans.	Systems that think rationally.
Systems that act like humans	Systems that act rationally

- a tension exists between approaches centered around humans and approaches centered around rationality
- we are not suggesting that humans are necessarily “irrational” in the sense of “emotionally unstable” or “insane”.
- we often make mistakes; we are not all chess grandmasters even though we may know all the rules of chess

The Turing Test – Can machines think?



- The "standard interpretation" of the Turing Test, in which player C, the interrogator, is tasked with trying to determine which player – A or B – is a computer and which is a human. The interrogator is limited to using the responses to written questions to make the determination

The Turing Test - computer capabilities

- The Turing test
  - Natural language processing
  - Knowledge representation
  - Automated reasoning
  - Machine learning
- The Total Turing Test
  - All of the above
  - Computer Vision
  - Robotics
- Captcha –Completely Automated Public Turing test to tell Computers and Humans Apart

Turing test success marks milestone in computing history

- Q. Hi Eugene. What is your favorite breed of dog?
- A. No, I hate dog's barking.

- Q. What color is your hair?
- A. Red, of course! Oh, what a fruitful conversation;-)

- Q. What is your favorite subject in school?
- A. My friends and teachers say I am totally stupid and untalented, so I only like music lessons because music teacher tells me only that I'm tone-deaf and at least doesn't call me stupid.

- Q: How many roads must a man walk down, before you can call him a man?
- A. No more than four. Some another topic?

- Q. Do you have a girlfriend?
- A. I don't have a girlfriend 'cause I still can't understand what a pleasure to spend my money on somebody else without and instant or delayed result.

http://www.reading.ac.uk/news-and-events/releases/PR583836.aspx

Introduction

- Artificial Intelligence is a branch of Science which deals with helping machines find solutions to complex problems in a more human-like fashion.
- AI is the technique of uncertainty management in computer software. AI is the discipline that you apply when you want to know what to do when you don't know what to do.
- AI has successfully been used in:
  - Medicine
  - Games
  - Robotics
  - Entertainment
  - Many more applications that makes your computer software smart

AI through history

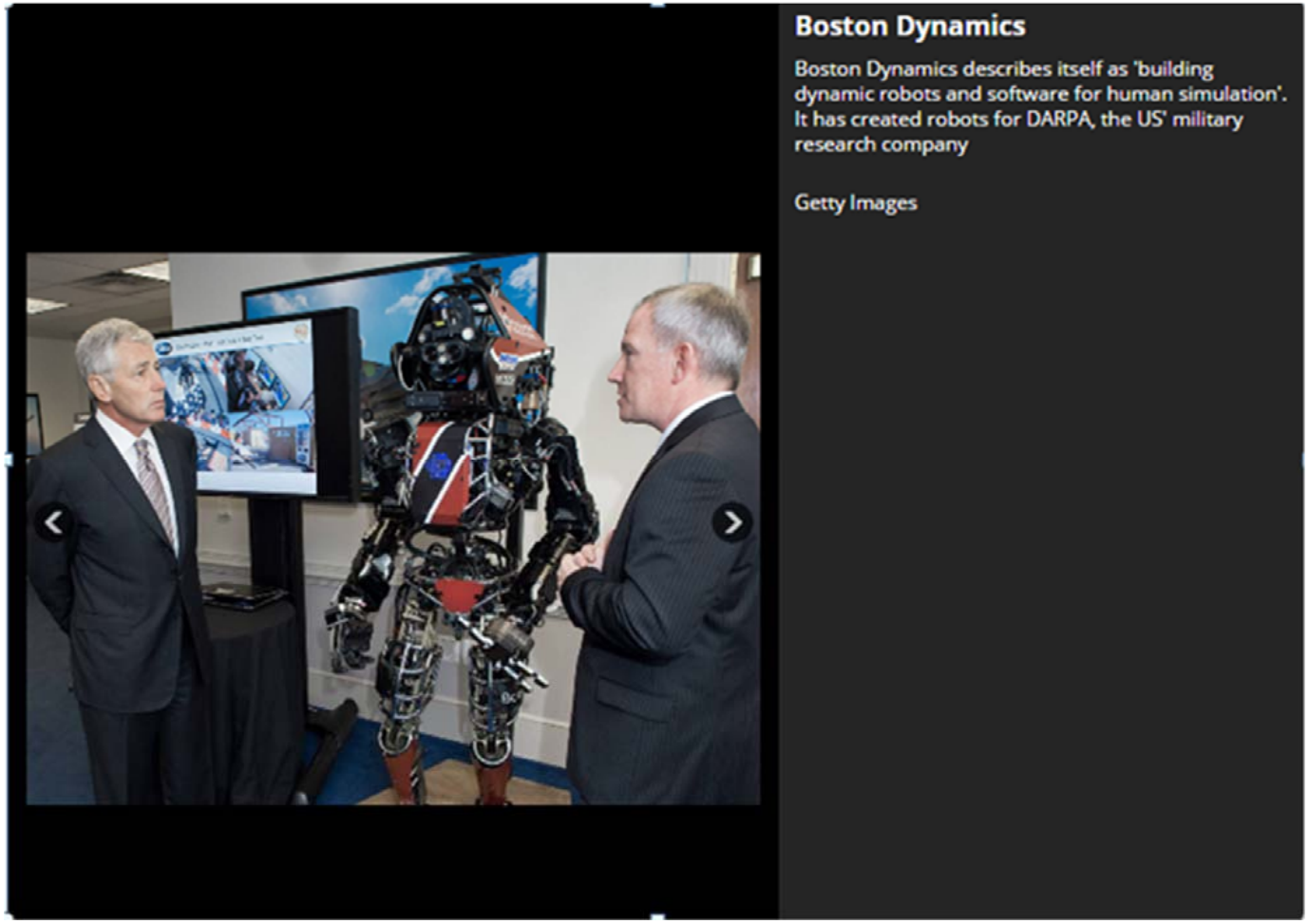
Deep Blue beats Kasparov

Deep Blue, a computer created by IBM, won a match against world champion Garry Kasparov in 1997. The computer could evaluate 200 million positions per second, and Kasparov accused it of cheating after the match was finished

Getty Images

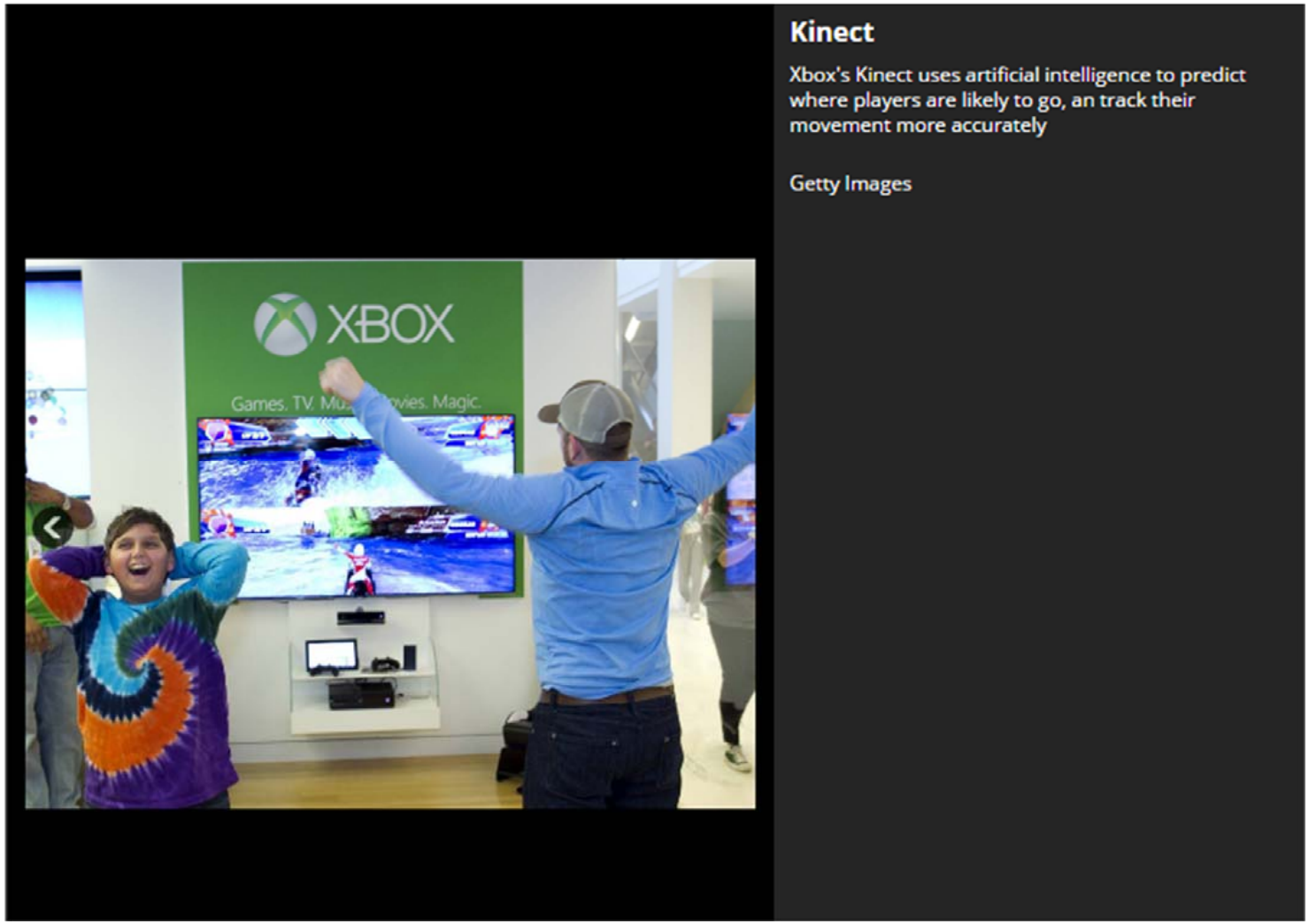
http://www.independent.co.uk/

AI through history



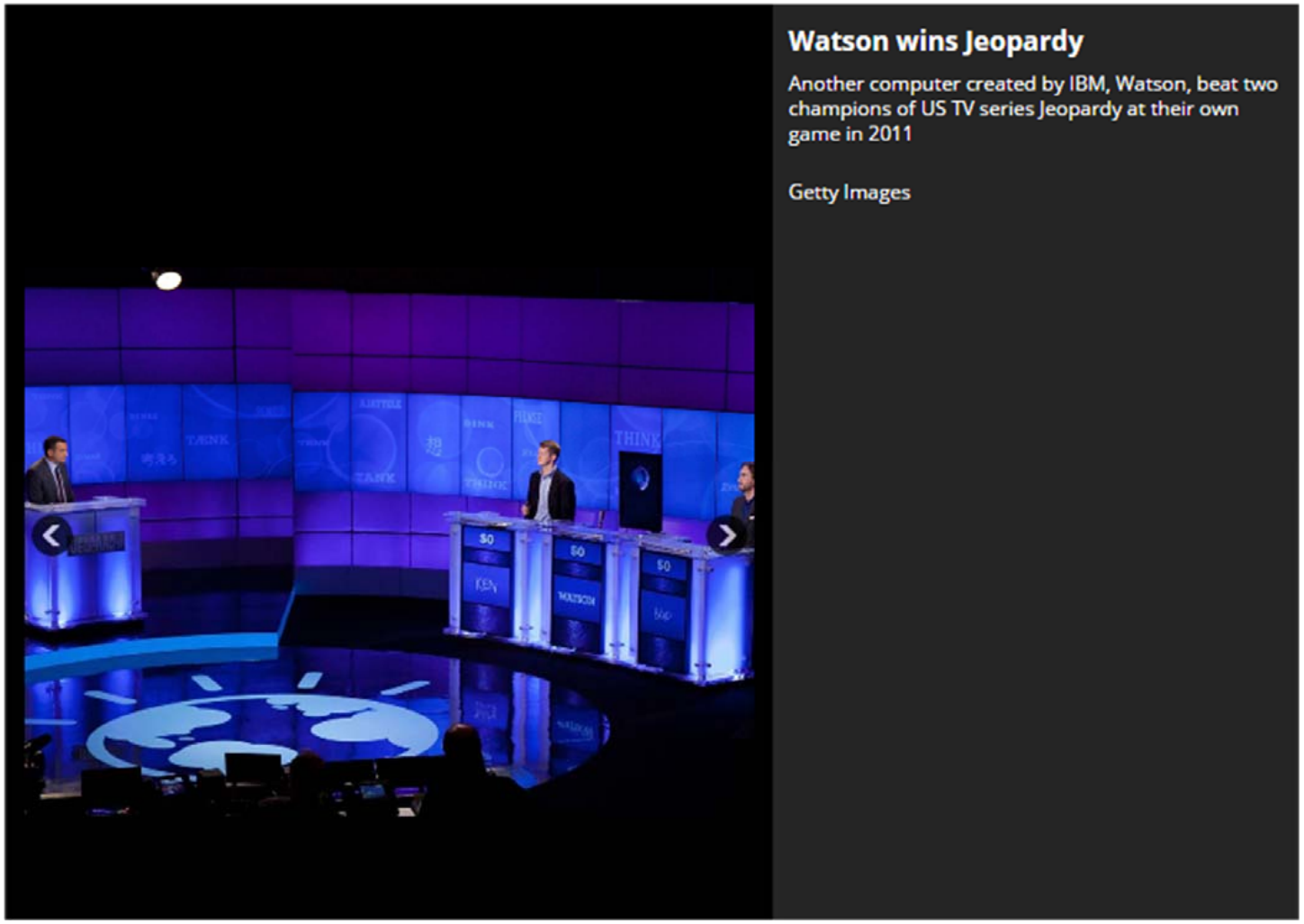
http://www.independent.co.uk/

AI through history



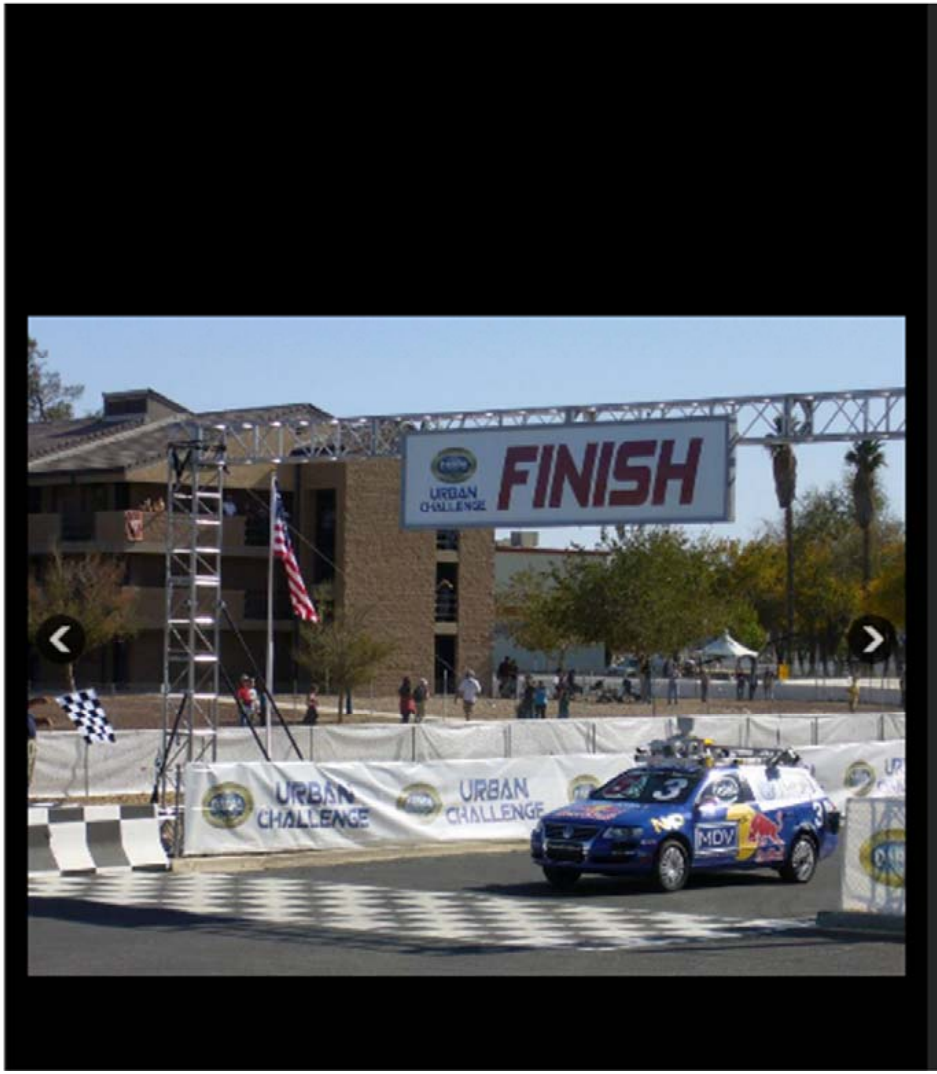
http://www.independent.co.uk/

AI through history



http://www.independent.co.uk/

AI through history



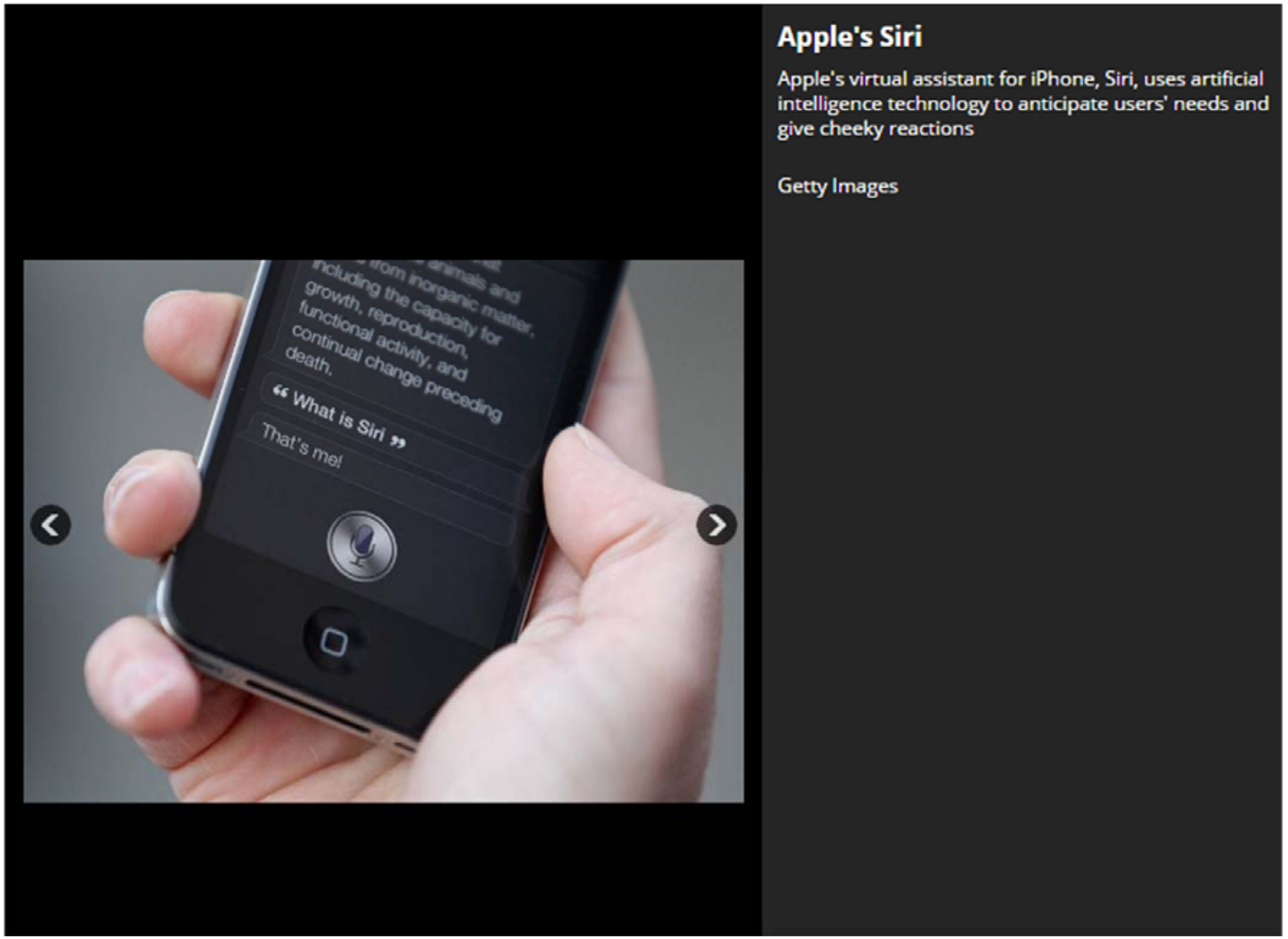
DARPA Urban Challenge

The DARPA Urban Challenge, set up by the US Department of Defense, challenges driverless cars to navigate a 60 mile course in an urban environment that simulates guerilla warfare

Getty Images

http://www.independent.co.uk/

AI through history



http://www.independent.co.uk/

AI now

AlphaGo: using machine learning to master the ancient game of Go

- There are 1,000,000,000,000,000,000,000,000,000,000,000,000,000,000,000,000,000,000,000,000,000,000,000,000,000,000,000,000,000,000,000,000,000,000,000,000,000,000,000,000,000,000,000,000,000,000,000,000,000,000,000,000,000,000,000,000,000 possible positions
- that’s more than the number of atoms in the universe
- more than a googol (10^{100}) times larger than chess.
- https://www.alphagomovie.com/

Versions	Hardware	Elo rating	Matches
AlphaGo Fan	176 GPUs, distributed	3144	5:0 against Fan Hui
AlphaGo Lee	48 TPUs,distributed	3739	4:1 against Lee Sedol
AlphaGo Master	4 TPUs, single machine	4858	60:0 against professional players; Future of Go Summit
AlphaGo Zero	4 TPUs, single machine	5185	100:0 against AlphaGo Lee 89:11 against AlphaGo Master
AlphaZero	4 TPUs, single machine	N/A	60:40 against AlphaGo Zero

AI now

- Adobe demos “photoshop for audio,” lets you edit speech as easily as text https://arstechnica.com/information-technology/2016/11/adobe-voco-photoshop-for-audio-speech-editing/
- Amazon Echo
- Amazon Go - http://www.vox.com/new-money/2016/12/5/13842712/amazon-go-no-checkout
- Google Brain Super-resolution - https://arstechnica.com/information-technology/2017/02/google-brain-super-resolution-zoom-enhance/

AI now

- Boston Dynamics in present
- https://www.cnbc.com/video/2018/05/10/boston-dynamics-new-video-atlas-and-spotmini-have-new-tricks.html
- Google AI creates its own 'child' AI that's more advanced than systems built by humans – AutoML project (http://www.independent.co.uk/life-style/gadgets-and-tech/news/google-child-ai-bot-nasnet-automl-machine-learning-artificial-intelligence-a8093201.html)

AI – friend or foe?

Bill Gates - “First the machines will do a lot of jobs for us and not be super intelligent. A few decades after that though the intelligence is strong enough to be a concern. I don’t understand why some peoples are not concerned.”

Elon Musk – donates $10M to keep AI from turning Evil

Stephen Hawking – “Once humans develop artificial intelligence, it would take off on its own and re-design itself at an ever increasing rate. Humans, who are limited by slow biological evolution, couldn't compete and would be superceded.”

Stuart Russell –“AI potential to benefit humanity is enormous, even in defense. But allowing machines to choose to kill humans will be devastating to our security and freedom.” https://www.youtube.com/watch?v=HipTO_7mUOw

AI – friend or foe?

Google – incorporates AI into the very core of its current and future technologies. AI is no longer a subject for academia; it's out there in the consumer market.

You could ask Facebook "Which of my friends had babies recently?" and get a reasonable answer, complete with photos, even if the new parents had never explicitly said that a baby was born. (It can already figure out, who you're dating, and when you might break up, even if the relationship isn't listed on Facebook.)

SwiftKey™

Ben Medlock – "We need to discuss the likely ethical implications, from data security when machines further analyze our lives, to whether self-driving cars should prioritize the safety of the driver . But are we currently at risk of being extinguished by our own creations? Not for a long time in my view."

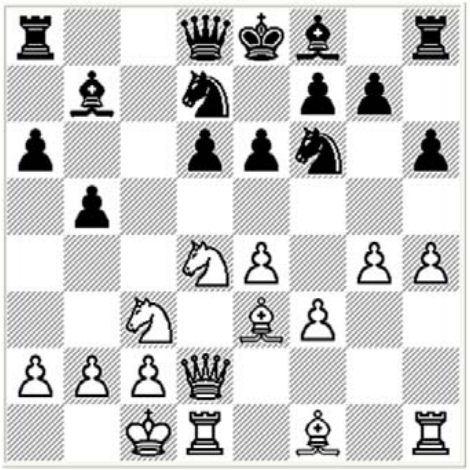
Building safe artificial intelligence

- https://medium.com/@deepmindsafetyresearch/building-safe-artificial-intelligence-52f5f75058f1
- https://www.theregister.co.uk/2017/02/16/deepmind_shows_ai_can_turn_aggressive/
- https://deepmind.com/blog/specifying-ai-safety-problems/

Terminology

- FULLY vs. PARTIALLY OBSERVABLE

Game of Chess



In any point of time, the information is completely sufficient to make the optimal decision.

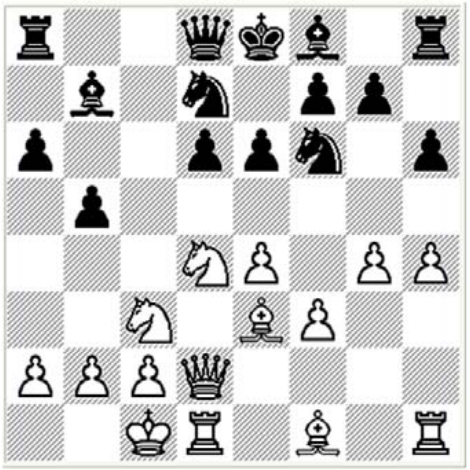
Game of Poker

You need memory to make the optimal decision

Terminology

- DETERMINISTIC vs. STOCHASTIC

Game of Chess



The taken actions determine the outcome in an unique way. (e.g. in chess no randomness is involved in the development of the future states)

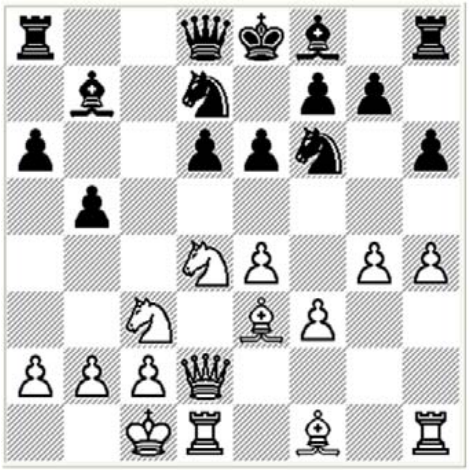
Games with dices

Even if movements are still deterministic, the outcome of an action involves the dices (a random variable), so the output is randomness.

Terminology

- DISCRETE vs. CONTINUOUS

Game of Chess



Involves a finite set of actions and outcomes.

Football

Involves an infinite actions and outcomes (depends of angle, acceleration, etc.).

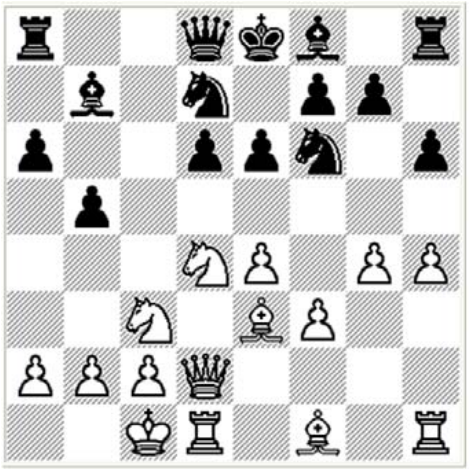
Terminology

- BENIGN vs. ADVERSARIAL environment

Whether prediction

Environment might be random (stochastic) but he has no objectives on its own.

Many games

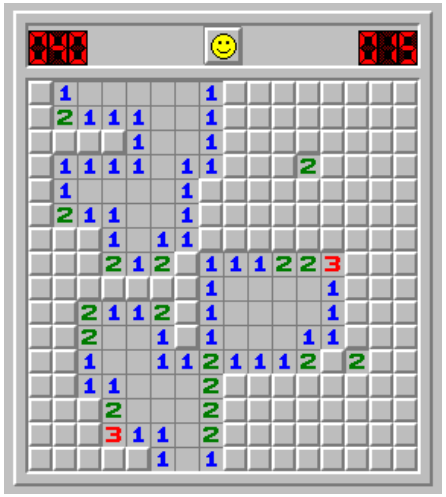


It's interested in "making your life worst".

QUIZ

	Partially observable	Stochastic	Continuous	Adversarial
Checkers				
Poker				
Autonomous robot car				
Minesweeper				

Checkers



Minesweeper

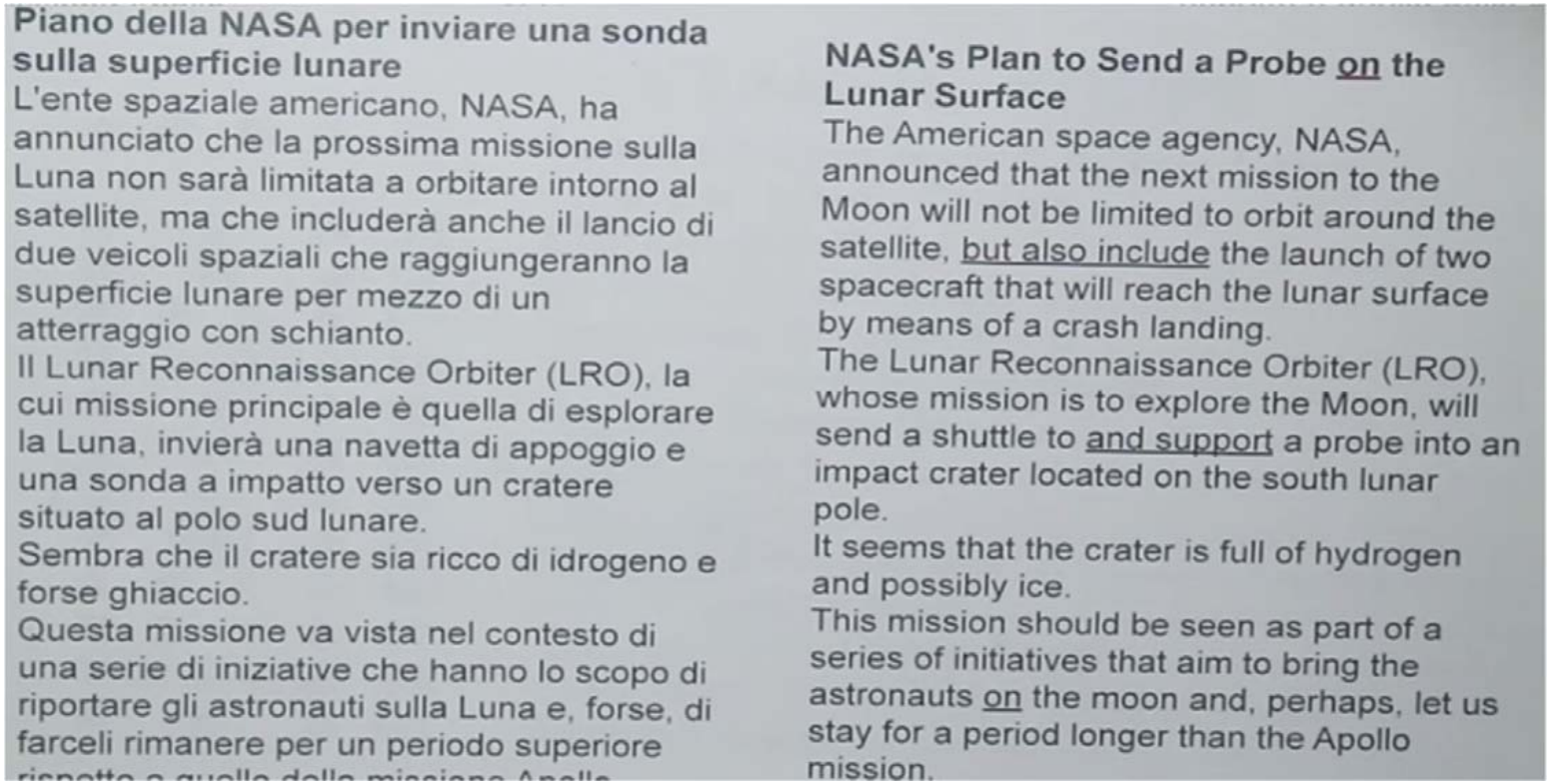
QUIZ ANSWERS

	Partially observable	Stochastic	Continuous	Adversarial
Checkers				●
Poker	●	●		●
Autonomous robot car	●	●	●	Depends where you live ☺
Minesweeper	●	Depends		

Example of AI – Machine Translation

- Google machine translation system
- Supports 90 languages
- Before Oct. 2007, Google Translate was based on SYSTRAN (uses Rule-based machine translation – RbMT)
- Since Oct. 2007 Google Translate uses statistical machine translation

How it works ?



Piano della NASA per inviare una sonda sulla superficie lunare

L'ente spaziale americano, NASA, ha annunciato che la prossima missione sulla Luna non sarà limitata a orbitare intorno al satellite, ma che includerà anche il lancio di due veicoli spaziali che raggiungeranno la superficie lunare per mezzo di un atterraggio con schianto.

Il Lunar Reconnaissance Orbiter (LRO), la cui missione principale è quella di esplorare la Luna, invierà una navetta di appoggio e una sonda a impatto verso un cratere situato al polo sud lunare.

Sembra che il cratere sia ricco di idrogeno e forse ghiaccio.

Questa missione va vista nel contesto di una serie di iniziative che hanno lo scopo di riportare gli astronauti sulla Luna e, forse, di farceli rimanere per un periodo superiore

NASA's Plan to Send a Probe on the Lunar Surface

The American space agency, NASA, announced that the next mission to the Moon will not be limited to orbit around the satellite, but also include the launch of two spacecraft that will reach the lunar surface by means of a crash landing.

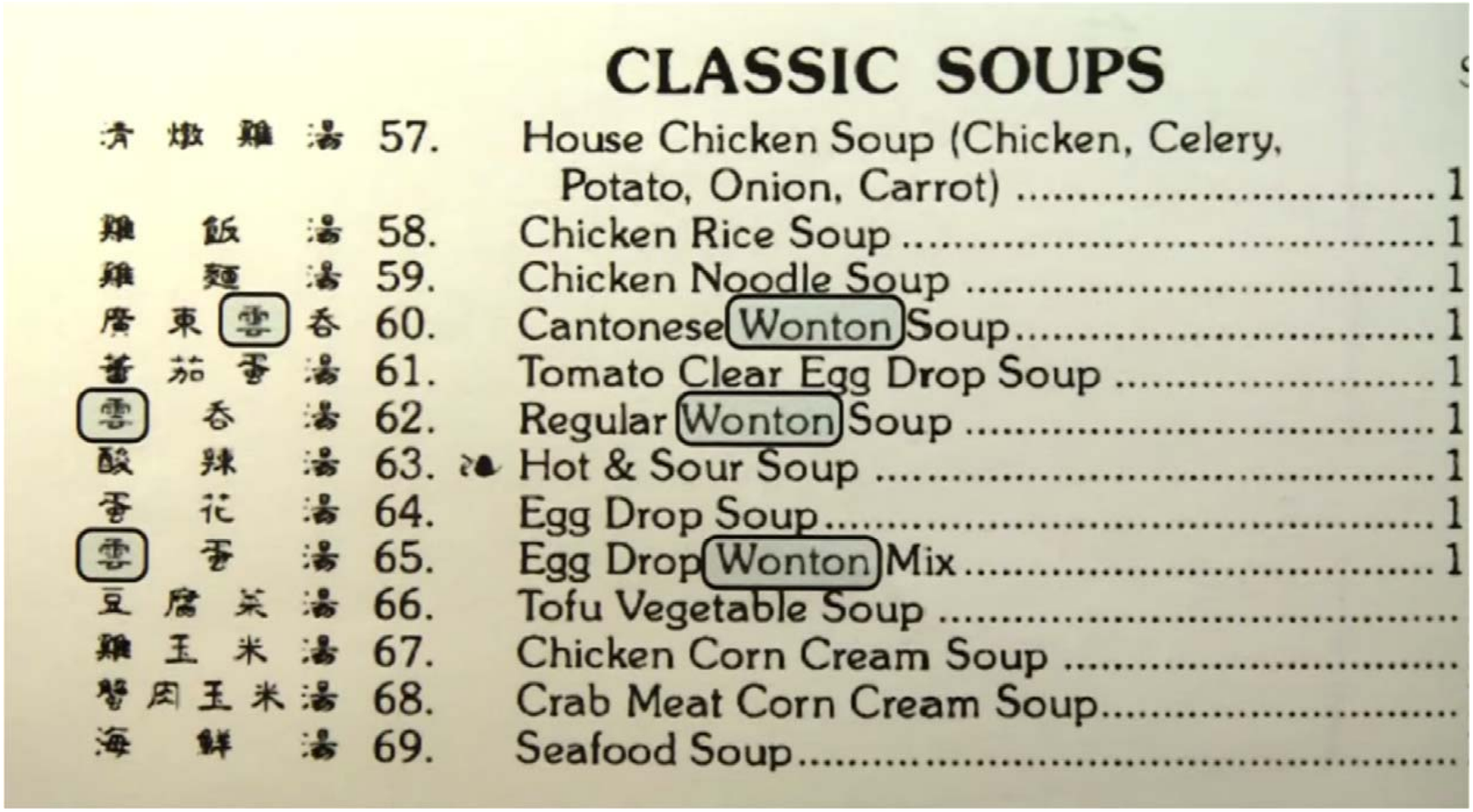
The Lunar Reconnaissance Orbiter (LRO), whose mission is to explore the Moon, will send a shuttle to and support a probe into an impact crater located on the south lunar pole.

It seems that the crater is full of hydrogen and possibly ice.

This mission should be seen as part of a series of initiatives that aim to bring the astronauts on the moon and, perhaps, let us stay for a period longer than the Apollo mission.

- Find newspaper that publishes 2 editions, and now we have examples of translations
- The process has 2 parts: offline (build the model based on examples) and online (given another text we use the model to translate it)

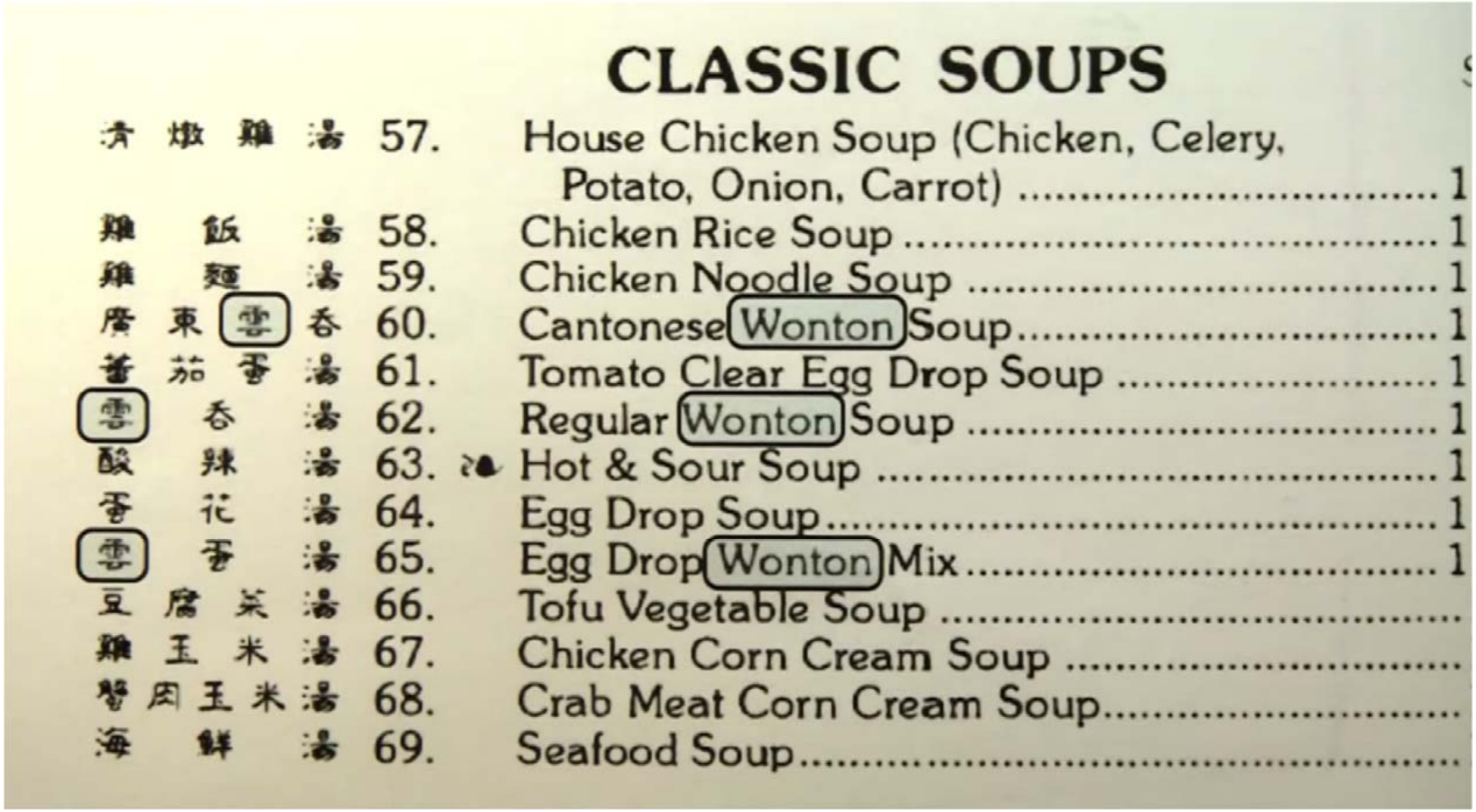
Chinese restaurant



CLASSIC SOUPS

清燉雞湯	57.	House Chicken Soup (Chicken, Celery, Potato, Onion, Carrot) 1
雞飯湯	58.	Chicken Rice Soup 1
雞麵湯	59.	Chicken Noodle Soup 1
廣東雲吞	60.	Cantonese Wonton Soup 1
蕃茄蛋湯	61.	Tomato Clear Egg Drop Soup 1
雲吞湯	62.	Regular Wonton Soup 1
酸辣湯	63.	Hot & Sour Soup 1
蛋花湯	64.	Egg Drop Soup 1
雲蛋湯	65.	Egg Drop Wonton Mix 1
豆腐菜湯	66.	Tofu Vegetable Soup .....
雞玉米湯	67.	Chicken Corn Cream Soup .....
蟹肉玉米湯	68.	Crab Meat Corn Cream Soup .....
海鮮湯	69.	Seafood Soup .....

QUIZ



CLASSIC SOUPS

清燉雞湯	57.	House Chicken Soup (Chicken, Celery, Potato, Onion, Carrot) 1
雞飯湯	58.	Chicken Rice Soup 1
雞麵湯	59.	Chicken Noodle Soup 1
廣東雲吞	60.	Cantonese Wonton Soup 1
番茄蛋湯	61.	Tomato Clear Egg Drop Soup 1
雲吞湯	62.	Regular Wonton Soup 1
酸辣湯	63.	Hot & Sour Soup 1
蛋花湯	64.	Egg Drop Soup 1
雲蛋湯	65.	Egg Drop Wonton Mix 1
豆腐菜湯	66.	Tofu Vegetable Soup .....
雞玉米湯	67.	Chicken Corn Cream Soup .....
蟹肉玉米湯	68.	Crab Meat Corn Cream Soup .....
海鮮湯	69.	Seafood Soup .....

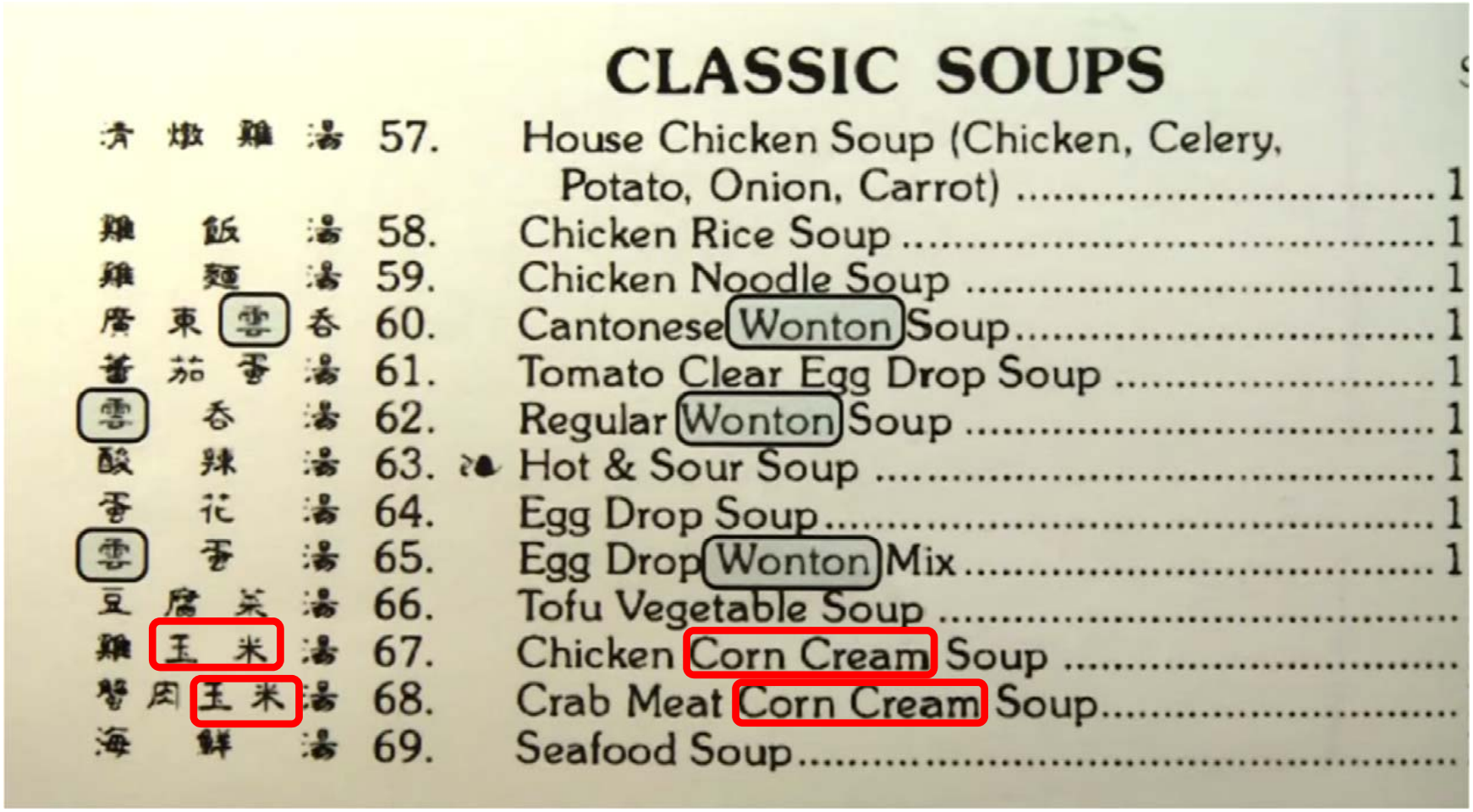
- Find the chinese character for **chicken, corn cream**
- How about the **soup ?**

Chinese restaurant – find chinese character for chicken

CLASSIC SOUPS

清燉雞湯	57.	House Chicken Soup (Chicken, Celery, Potato, Onion, Carrot) 1
雞飯湯	58.	Chicken Rice Soup 1
雞麵湯	59.	Chicken Noodle Soup 1
廣東雲吞	60.	Cantonese Wonton Soup 1
蕃茄蛋湯	61.	Tomato Clear Egg Drop Soup 1
雲吞湯	62.	Regular Wonton Soup 1
酸辣湯	63.	Hot & Sour Soup 1
蛋花湯	64.	Egg Drop Soup 1
雲蛋湯	65.	Egg Drop Wonton Mix 1
豆腐菜湯	66.	Tofu Vegetable Soup .....
雞玉米湯	67.	Chicken Corn Cream Soup .....
蟹肉玉米湯	68.	Crab Meat Corn Cream Soup .....
海鮮湯	69.	Seafood Soup .....

Chinese restaurant – find chinese character for corn cream



CLASSIC SOUPS

清燉雞湯	57.	House Chicken Soup (Chicken, Celery, Potato, Onion, Carrot) 1
雞飯湯	58.	Chicken Rice Soup 1
雞麵湯	59.	Chicken Noodle Soup 1
廣東雲吞	60.	Cantonese Wonton Soup 1
蕃茄雞湯	61.	Tomato Clear Egg Drop Soup 1
雲吞湯	62.	Regular Wonton Soup 1
酸辣湯	63.	Hot & Sour Soup 1
雞花湯	64.	Egg Drop Soup 1
雲雞湯	65.	Egg Drop Wonton Mix 1
豆腐菜湯	66.	Tofu Vegetable Soup .....
雞玉米湯	67.	Chicken Corn Cream Soup .....
蟹肉玉米湯	68.	Crab Meat Corn Cream Soup .....
海鮮湯	69.	Seafood Soup .....

Chinese restaurant – find chinese character for soup

CLASSIC SOUPS

清燉雞湯	57.	House Chicken Soup (Chicken, Celery, Potato, Onion, Carrot) 1
雞飯湯	58.	Chicken Rice Soup 1
雞麵湯	59.	Chicken Noodle Soup 1
廣東雲吞湯	60.	Cantonese Wonton Soup 1
蕃茄雞蛋湯	61.	Tomato Clear Egg Drop Soup 1
雲吞湯	62.	Regular Wonton Soup 1
酸辣湯	63.	Hot & Sour Soup 1
雞蛋花湯	64.	Egg Drop Soup 1
雲吞雞蛋湯	65.	Egg Drop Wonton Mix 1
豆腐菜湯	66.	Tofu Vegetable Soup .....
雞玉米湯	67.	Chicken Corn Cream Soup .....
蟹肉玉米湯	68.	Crab Meat Corn Cream Soup .....
海鮮湯	69.	Seafood Soup .....

We don’t have a 100% correlation

Artificial Intelligence Fundamentals

Problem Solving

Goal Trees

Rule-Based Expert Systems

Example 1

- Resolve the following indefinite integral

$$\int \frac{-5x^4}{(1-x^2)^{5/2}} dx$$

- Can you do it in your head?
- A program that can do that, it's an *intelligent* program?
- In 1963, Slagle James, "A Heuristic Program that Solves Symbolic Integration Problems in Freshman Calculus.", presents SAINT (Symbolic Automation INTegrator)

The Problem-Reduction method

- Convert difficult goals into one or more easier-to-achieve subgoals
- Each subgoal may be divided still more finely into one or more lower-level subgoals
- Recognize goals -> convert them into appropriate subgoals
- Problem reduction -> goal reduction

Standards forms

- If an integral is into a “standard form” the goal is immediately achieved by substitution
- SAINT uses 26 standard forms

$$(a)\ \int \frac{1}{x}dx = \ln(x)$$

$$(b)\ \int x^n dx = \frac{x^{n+1}}{n+1}$$

$$(c)\ \int \cos(x)\,dx = \sin(x)$$

$$\vdots$$

Simple (Safe) transformation

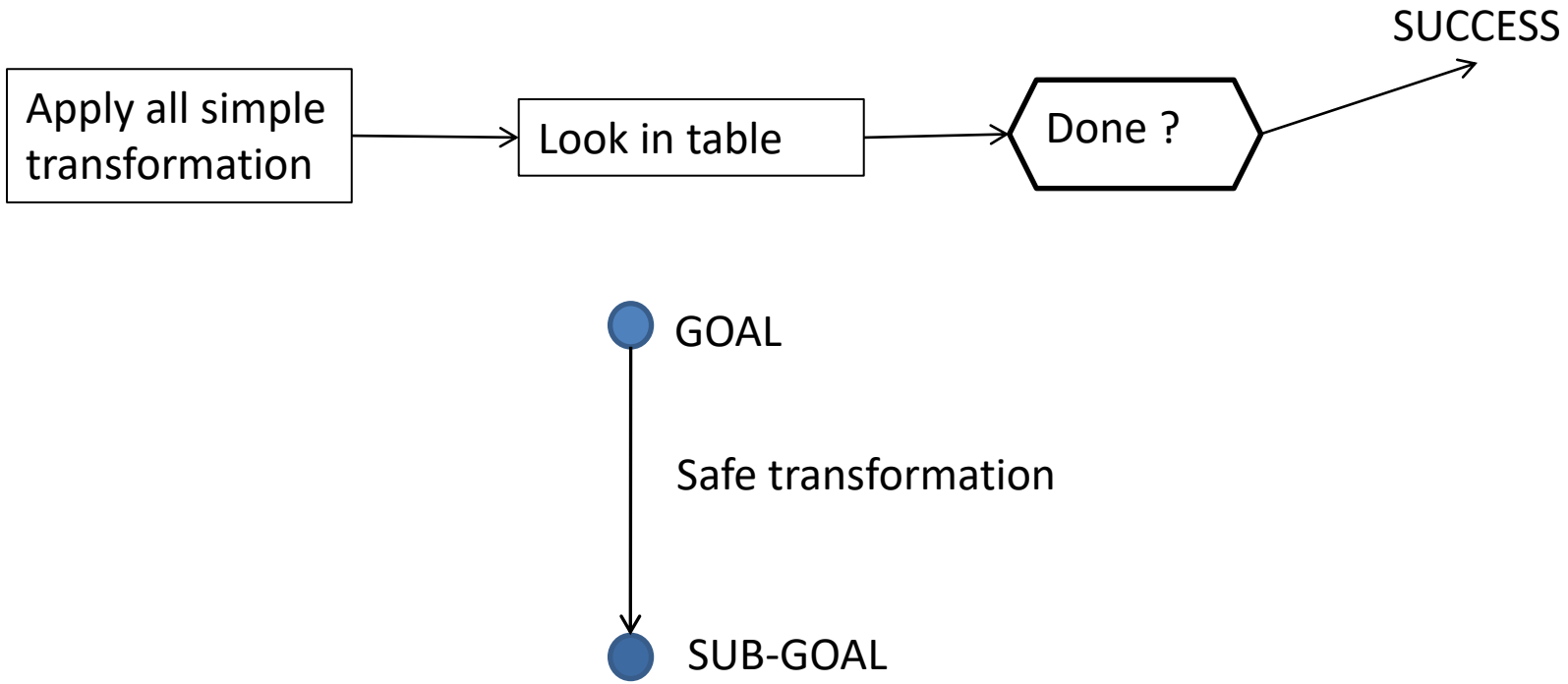
1. Deal with *minus* sign $\int -g(x)dx = -\int g(x)dx$
2. Take the constant out $\int cg(x)dx = c\int g(x)dx$
3. Decompose the sum $\int \sum g_i(x)dx = \sum \int g_i(x)dx$
4. Polynomial division $\int \frac{P(x)}{Q(x)}dx \rightarrow DIVIDE$

⋮

Simple transformation

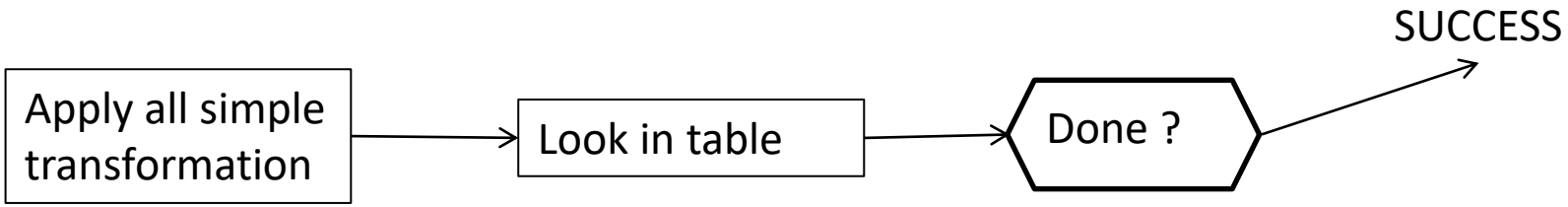
- a safe transformation – a transformation which, when applicable, is always or almost always appropriate
- for a goal, a transformation is appropriate if it's the correct next step to bring that goal nearer to achievement
- SAINT uses 8 simple transformations

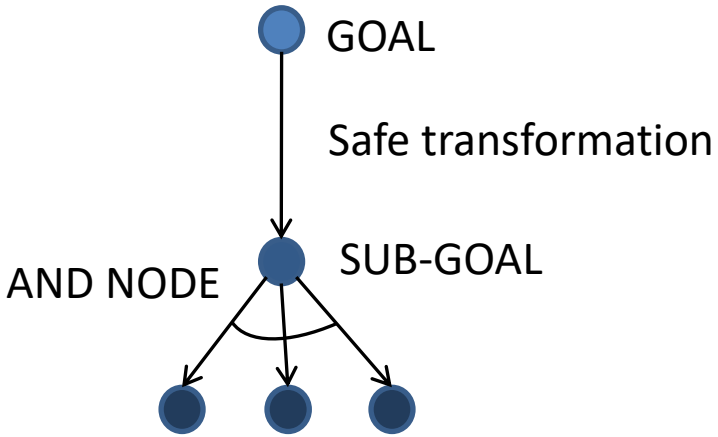
The framework



- This version of framework doesn't tell us anything about what happens for transformation no. 3

The framework - update





Problem reduction

- Apply all simple transformations

$$\int \frac{-5x^4}{(1-x^2)^{5/2}} \xrightarrow{1} \int \frac{5x^4}{(1-x^2)^{5/2}} \xrightarrow{2} \int \frac{x^4}{(1-x^2)^{5/2}}$$

Heuristic transformations

- Heuristic method - method that often work; isn’t guaranty that will always works
- Is not an algorithm – it’s an attempt

Heuristic transformation

A. Transformation from trigonometric form into other trigonometric form

$$f(sin(x), cos(x), tg(x), ctg(x), sec(x), cosec(x))$$
$$= g_1(sin(x), cos(x))$$
$$= g_2(tg(x), cosec(x))$$
$$= g_3(ctg(\mathrm{x}), sec(x))$$

B. Transformation from trigonometric form into polynomial form

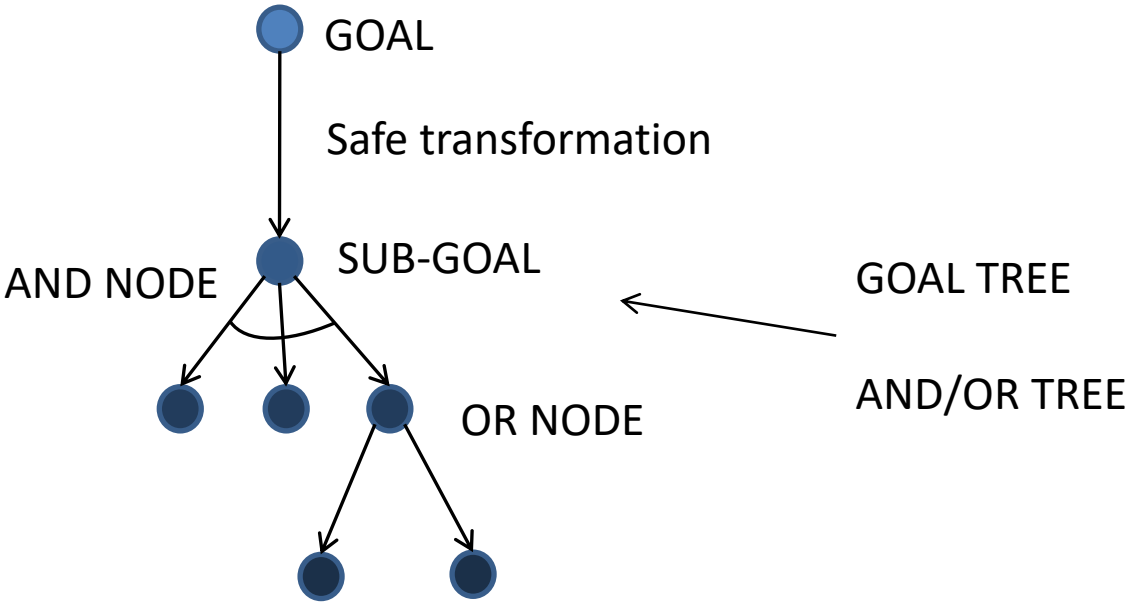
$$\int f(tg(x))dx = \int \frac{f(y)}{1+y^2}dy$$

C. Transformation from polynomial form into trigonometric form

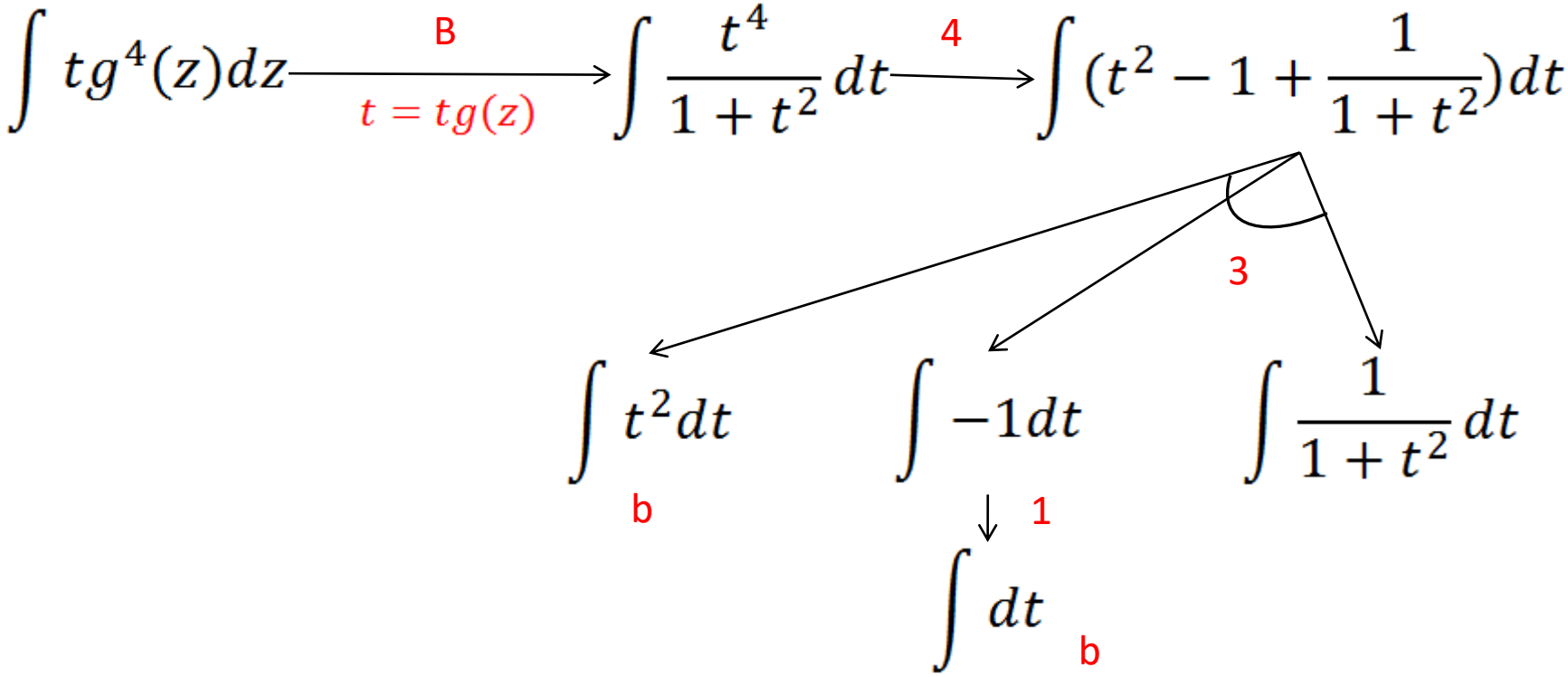
$$1 - x^2 \rightarrow x = sin(y) \qquad 1 + x^2 \rightarrow x = tg(y)$$

Problem reduction - continue

$$\int \frac{x^4}{(1-x^2)^{5/2}}\,dx \xrightarrow[x=\sin(y)\;\; dx=\cos(y)dy]{C} \int \frac{\sin^4(y)}{\cos^4(y)}\,dy \xrightarrow{A} \int \frac{1}{ctg^4(z)}\,dz \;\;\text{or}\;\; \int tg^4(z)\,dz$$



- Measure the depth of function composition
- The winner is *tg(x)*

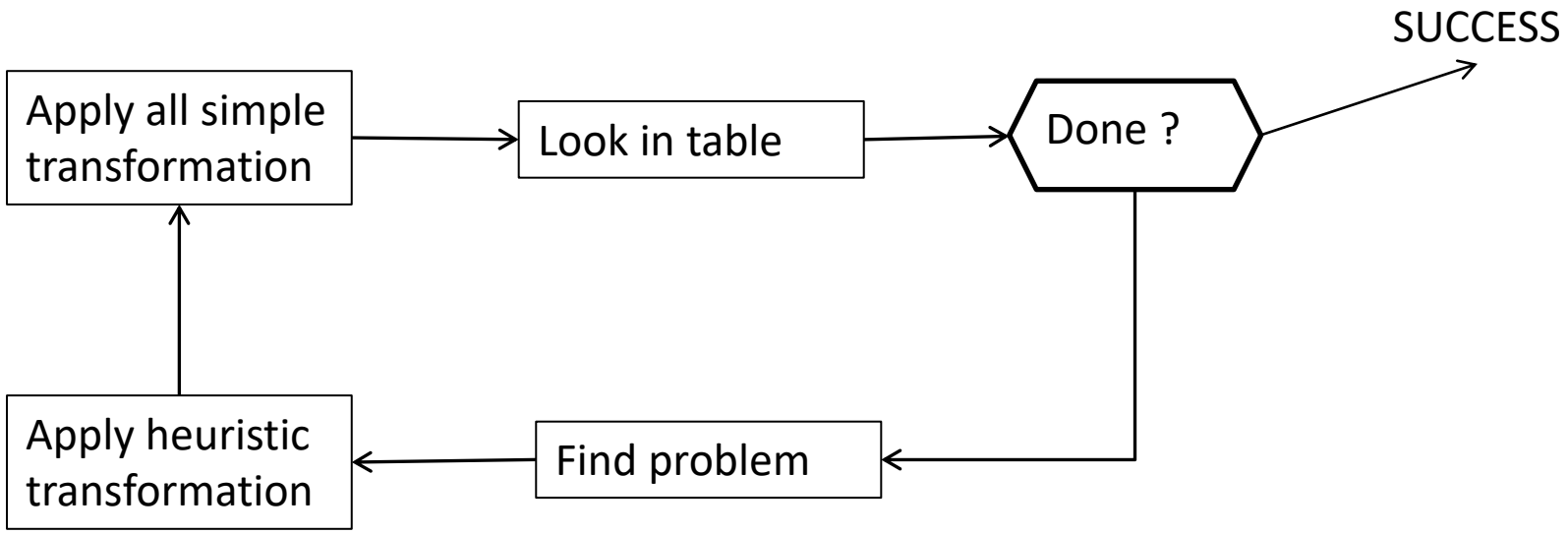


Problem reduction - continue

$$\int \frac{1}{1+t^2}\,dt \xrightarrow[w = arctg(t)]{C} \int dw \quad b$$

- Before doing this transformation, SAINT compute the function composition (which is 3) and return to a previous choice $\int \frac{1}{ctg^4(z)}\,dz$ which have 2
- After doing some transformation, the function composition raises to 4, and because of that, SAINT returns to this transformation

The framework - update



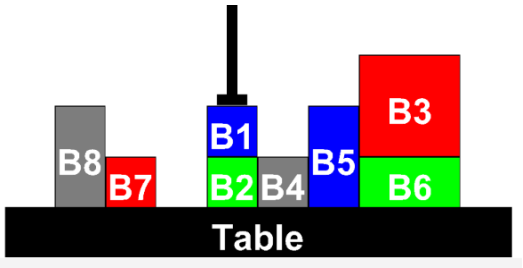
Questions

- How good the integration program is it?
  - 32K memory
  - Resolves 54 from 56 of the hardest problem (lacks 2 transformations)
  - Depth of the tree – 7 levels
  - Average depth – approx. 3
  - Branches unused – approx. 1
- What kind of knowledge is involved in this?
  - Knowledge about transformations
  - How goal tree works
  - Knowledge about standard forms
  - Knowledge about domain

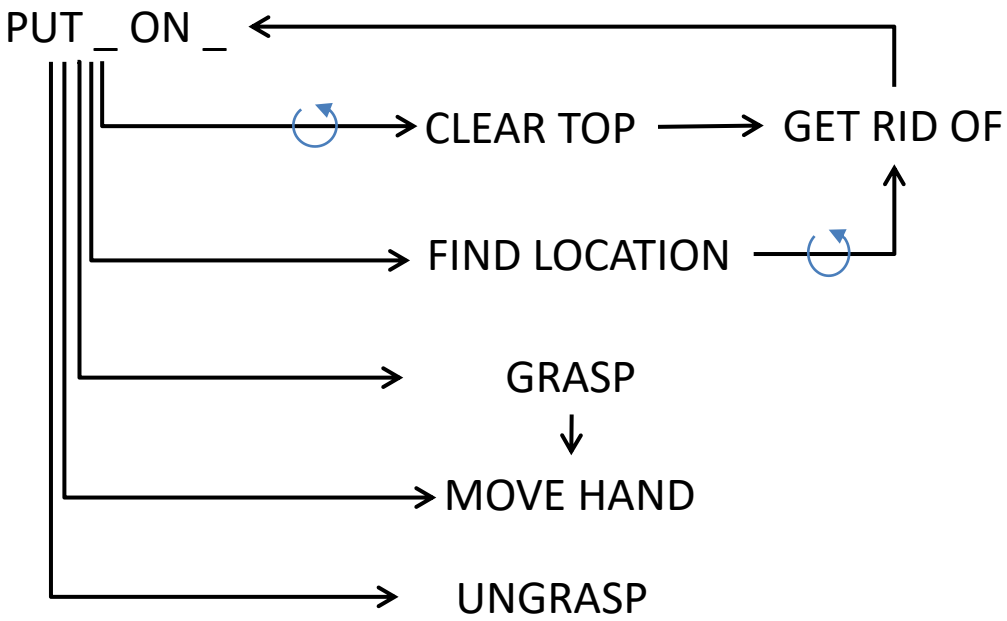
Questions

- How is the knowledge represented?
  - Table expressions
  - LISP relations
  - Goal tree information is embedded into procedures
- How much knowledge is required?
  - Table of integrals – standard form – 26
  - Simple transformations – 8
  - Heuristics transformations - 12

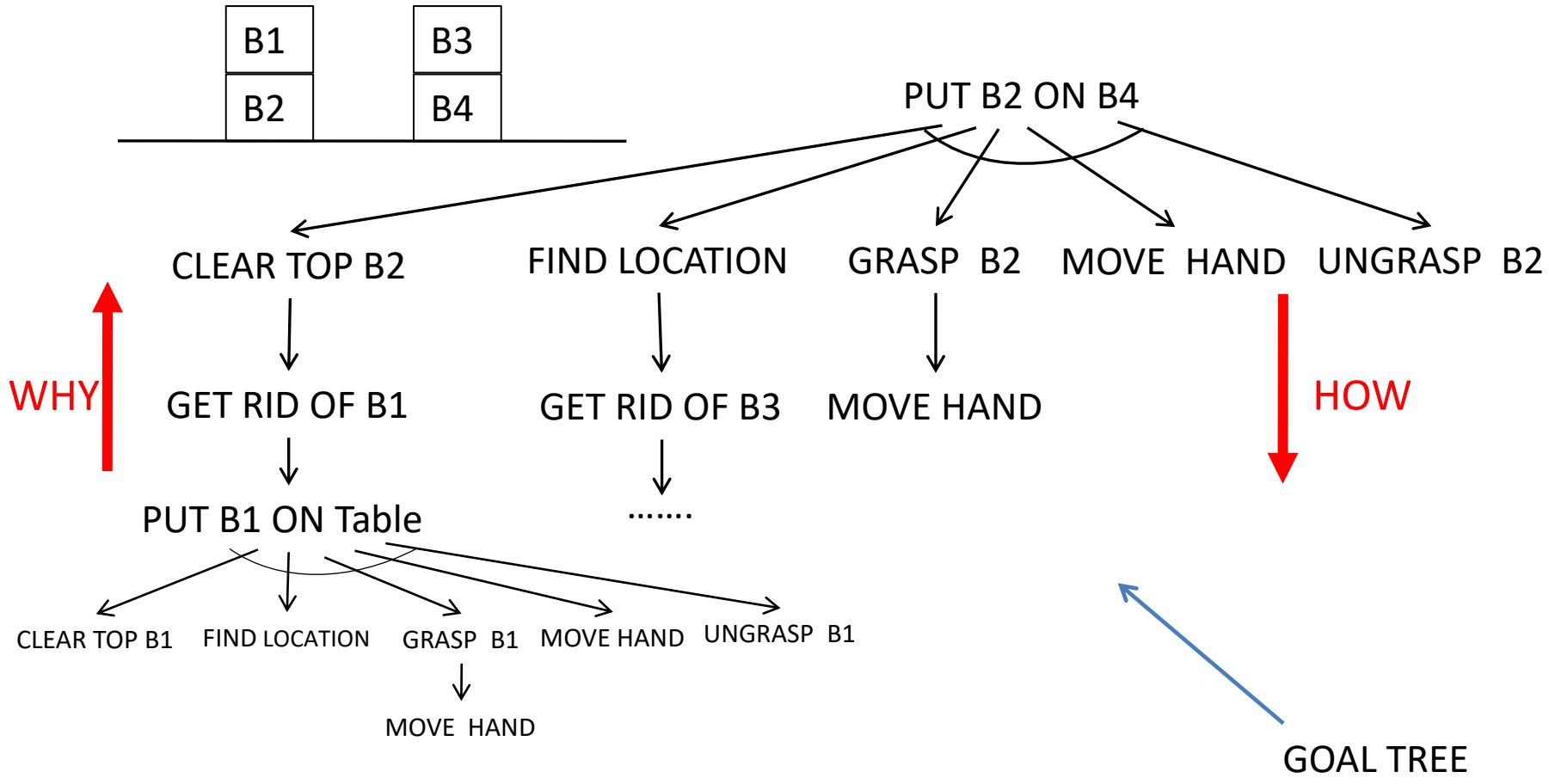
Blocks world



- Demo
- Written by Terry Winograd – professor at Stanford



Example

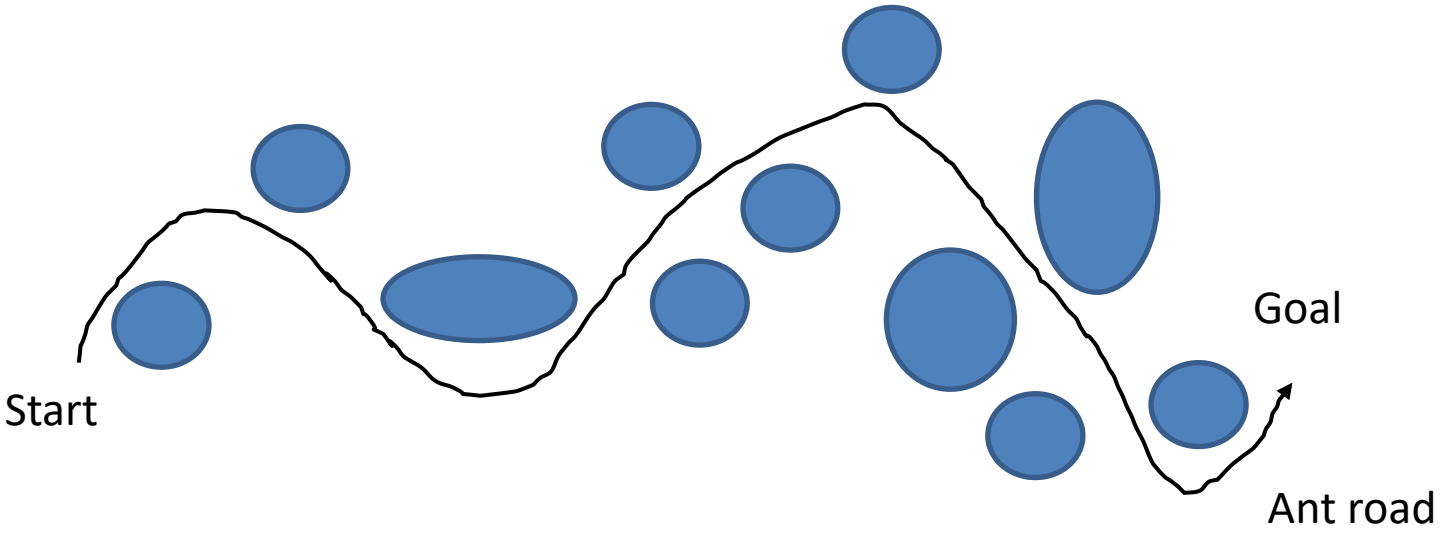

B1
B2
B3
B4
PUT B2 ON B4
CLEAR TOP B2
FIND LOCATION
GRASP B2
MOVE HAND
UNGRASP B2
WHY
HOW
GET RID OF B1
GET RID OF B3
MOVE HAND
PUT B1 ON Table
.....
CLEAR TOP B1
FIND LOCATION
GRASP B1
MOVE HAND
UNGRASP B1
MOVE HAND
GOAL TREE

How and Why questions

- Deal with "*how*" questions:
  - Identify the goal involved in the goal tree
  - If the goal is an AND node report all immediate subgoals
  - If the goal is an OR node report the immediate subgoal that was achieved
- Deal with "*why*" questions:
  - Identify the goal involved in the goal tree
  - Report the immediate supergoal

Complexity

- The complexity of the behavior is largely a consequence not of the complexity of the program, but the complexity of a problem (the environment)

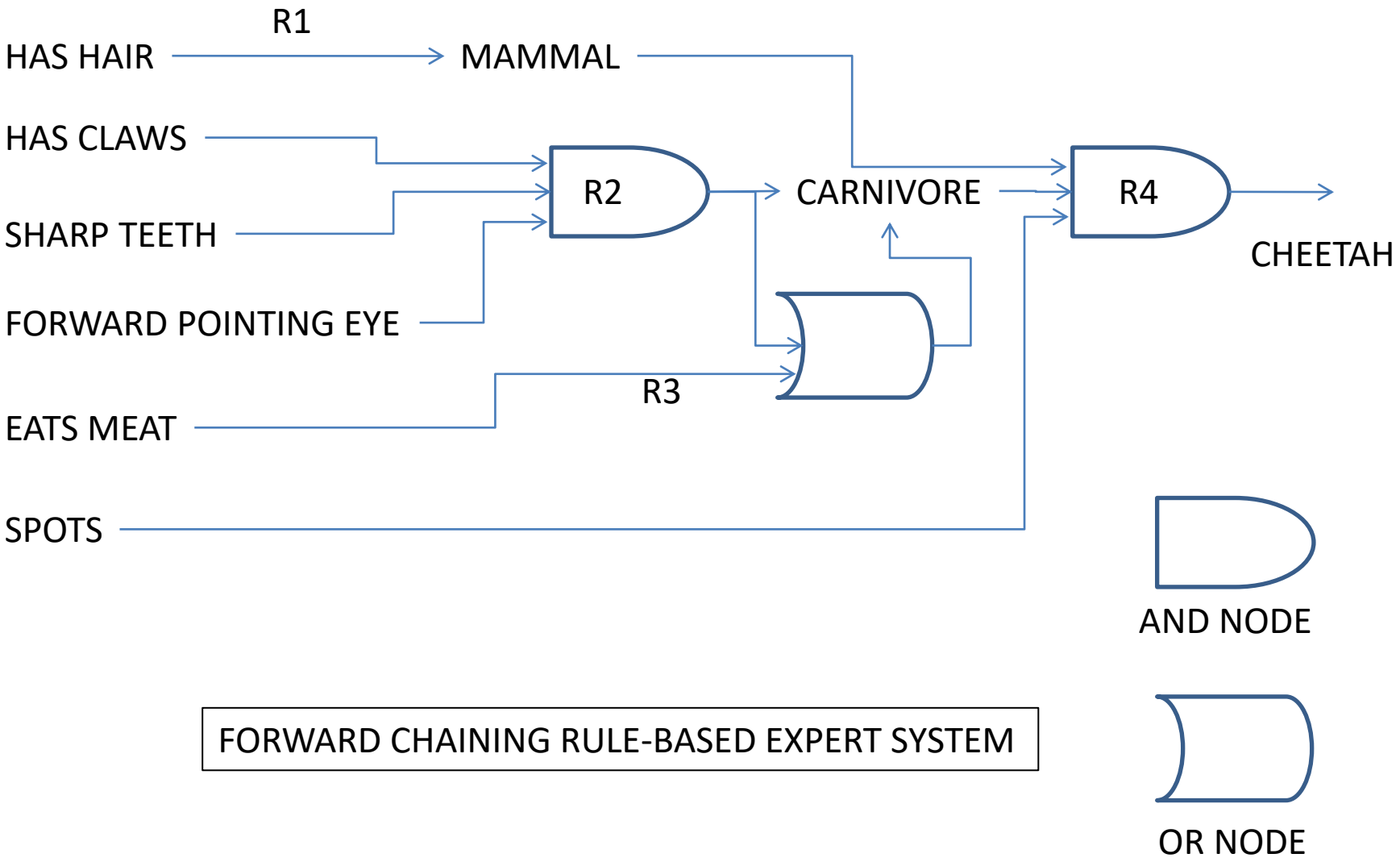


C(behavior) = max(C(program), C(environment))

Rule-based expert systems

- Emulate the decision-making ability of a human expert
- All the knowledge in a form of simple rules: "*If …..then…*"

Example – Animals from Zoo

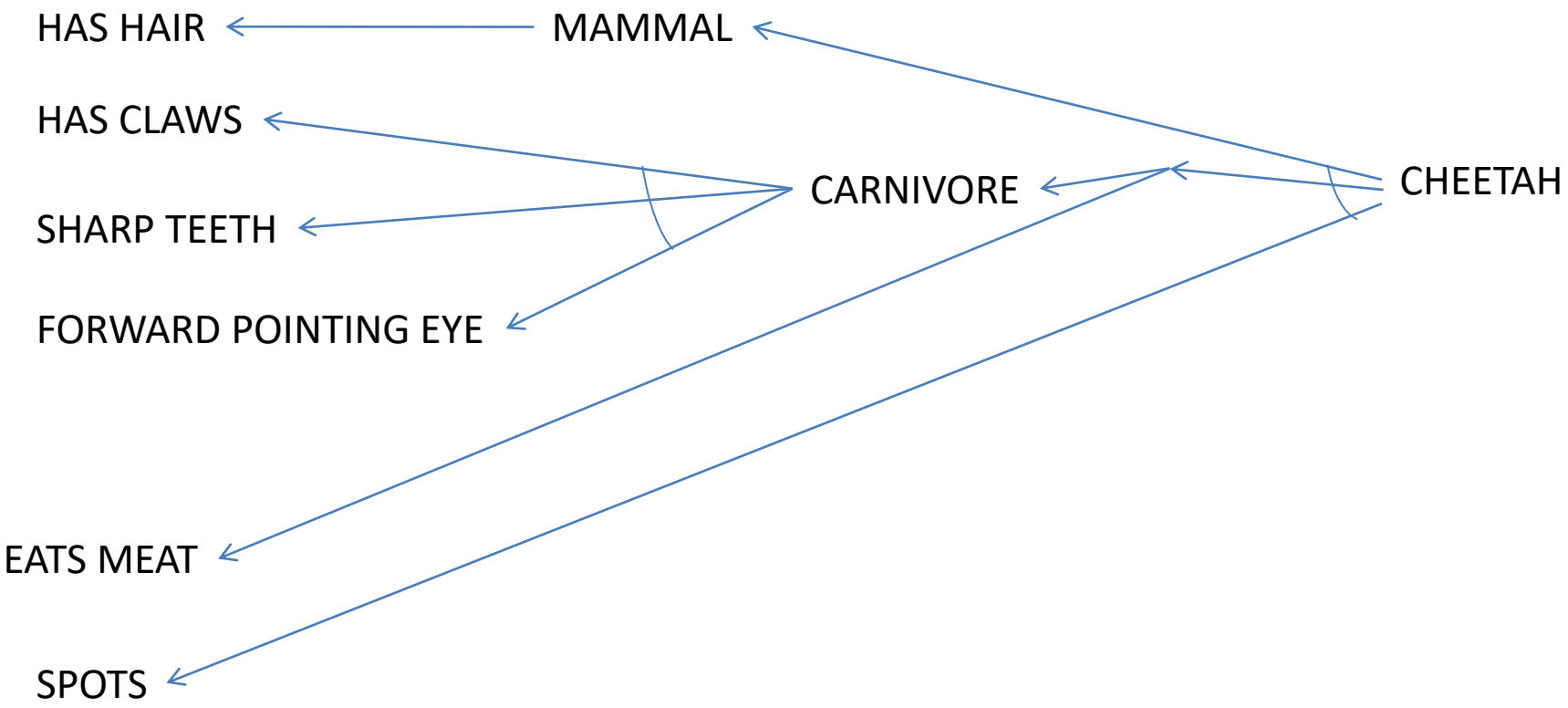


FORWARD CHAINING RULE-BASED EXPERT SYSTEM

Rule based expert systems

- Is not an expert in the real meaning of the word because it didn't have the experience of previous cases
- Only follows some already defined rules
- The systems can answers to the questions about the behavior
  - *Why are you interested in the animal claws?*
  - *Because I will try to see if it's a carnivore.*

Backward chaining



BACKWARD CHAINING RULE-BASED EXPERT SYSTEM

Expert systems

- Are deduction systems – working with facts to produce new facts
- One of the first expert system – MYCIN
  - Rule example: **If** x's type is primary bacteremia and x's suspected portal is gastrointestinal and the site of the culture of x is sterile **Then** there is evidence that x is bacteroides

Principles of knowledge engineering

- Deal with specific cases – learn knowledge they otherwise they missed
- Ask questions about the things that appear to be the same but actually they are handled differently -> new words in my domain
- Build a system and see when it crack -> missing rules

Questions

- Is an expert system real smart ?
- He doesn’t know about the knowledge involves into an expert system
- Rule based systems doesn’t have anything to do with common sense

Sample exam problem

- Due to constant pressure from the AIF staff, J. K. Rowling decides to write an 8th Harry Potter book. But, she's suffering from a bad case of writer's block and decides to use a rule-based system to help her with the plot for Harry Potter and the Deadhorse Principle. She's given you a set of rules and assertions and would like your help with developing key plot points.

Rules:

P0:	IF (AND('(?x) is ambitious', '(?x) is a squib') THEN '(?x) has a bad term')
P1:	IF ('(?x) lives in Gryffindor Tower') THEN ('(?x) is a protagonist')
P2:	IF (('(?x) lives in Slytherin dungeon') THEN ('(?x) is a villain'), '(?x) is ambitious))
P3:	IF (AND(OR('(?x) is a protagonist', '(?x) is a villain'), '(?x) is ambitious') THEN ('(?x) studies a lot'))
P4:	IF (AND('(?x) studies a lot', '(?x) is a protagonist') THEN ('(?x) becomes Hermione's friend'))
P5:	IF (AND('(?x) snogs (?y)', '(?x) lives in Gryffindor Tower', '(?y) lives in Slytherin dungeon') THEN ('(?x) has a bad term'))

- ?x, ?y – variables waiting to be bound
- after IF we have some *antecedents* that must be true in order to match the rule
- after THEN we have *consequences* that will be added into DB

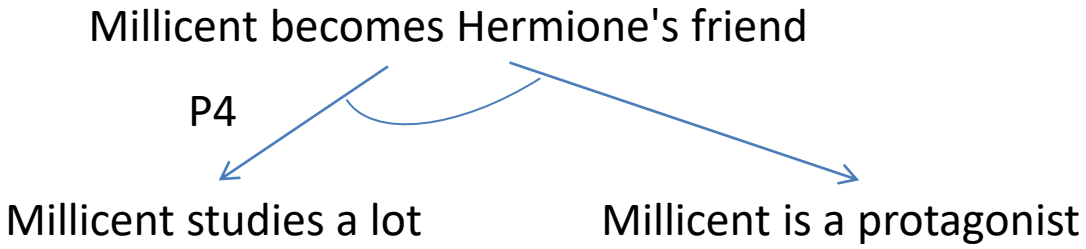
Assertions:

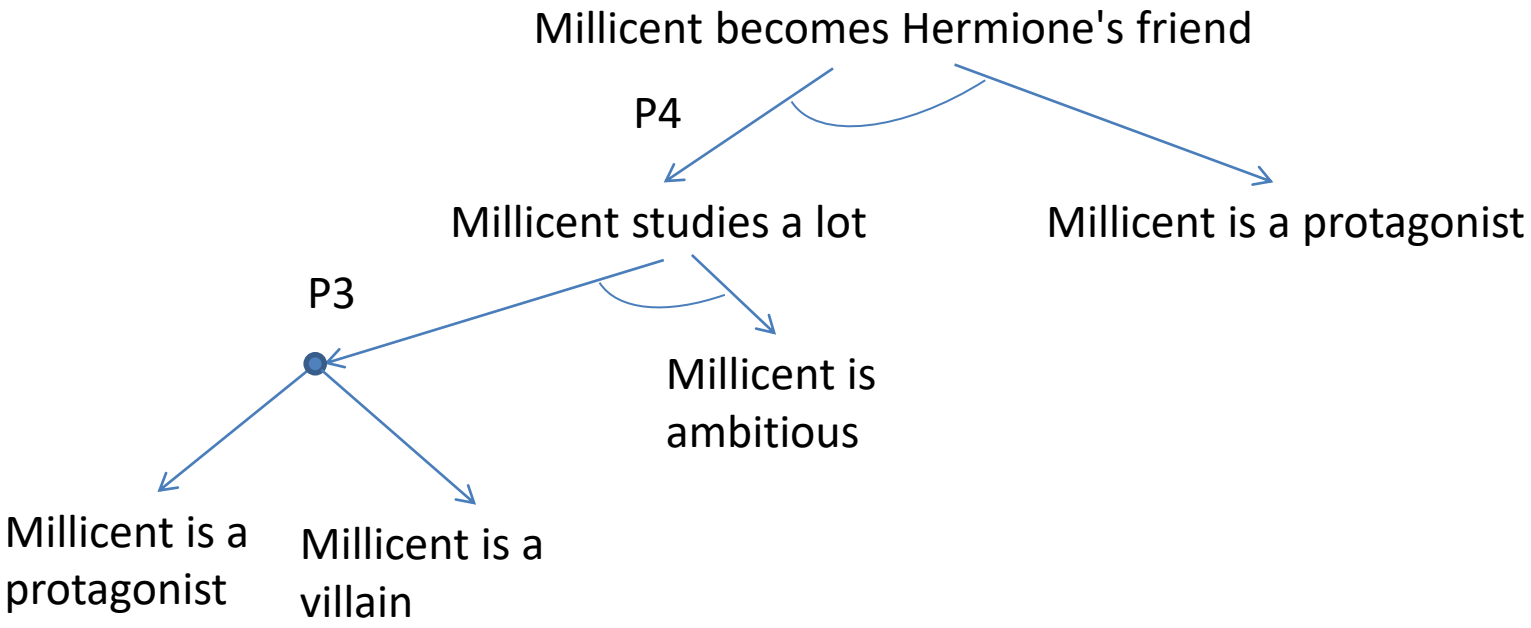
A0:	(Millicent lives in Slytherin dungeon)
A1:	(Millicent is ambitious))
A2:	(Seamus lives in Gryffindor Tower)
A3:	(Seamus snogs Millicent)

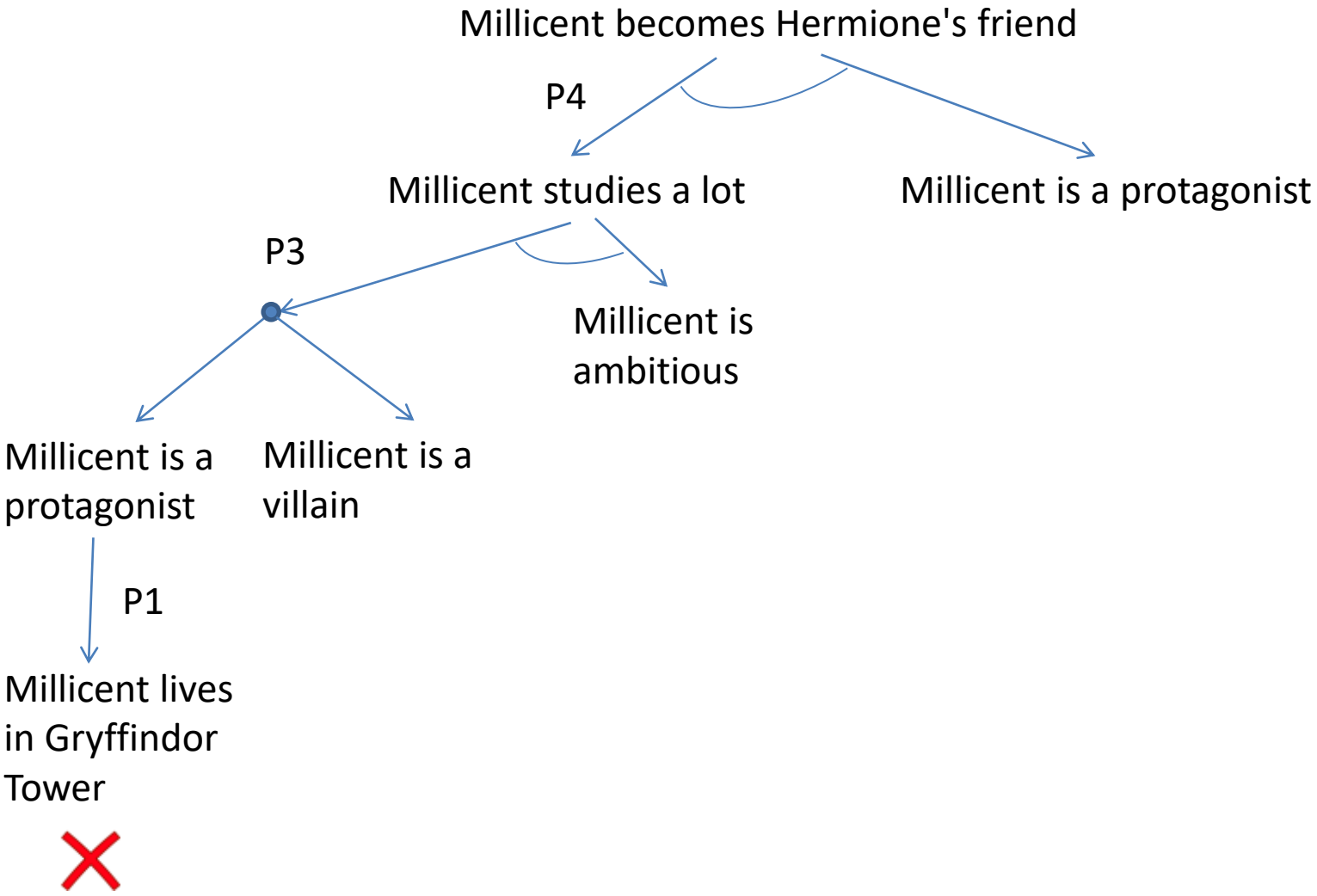
- Check assertions before using a rule

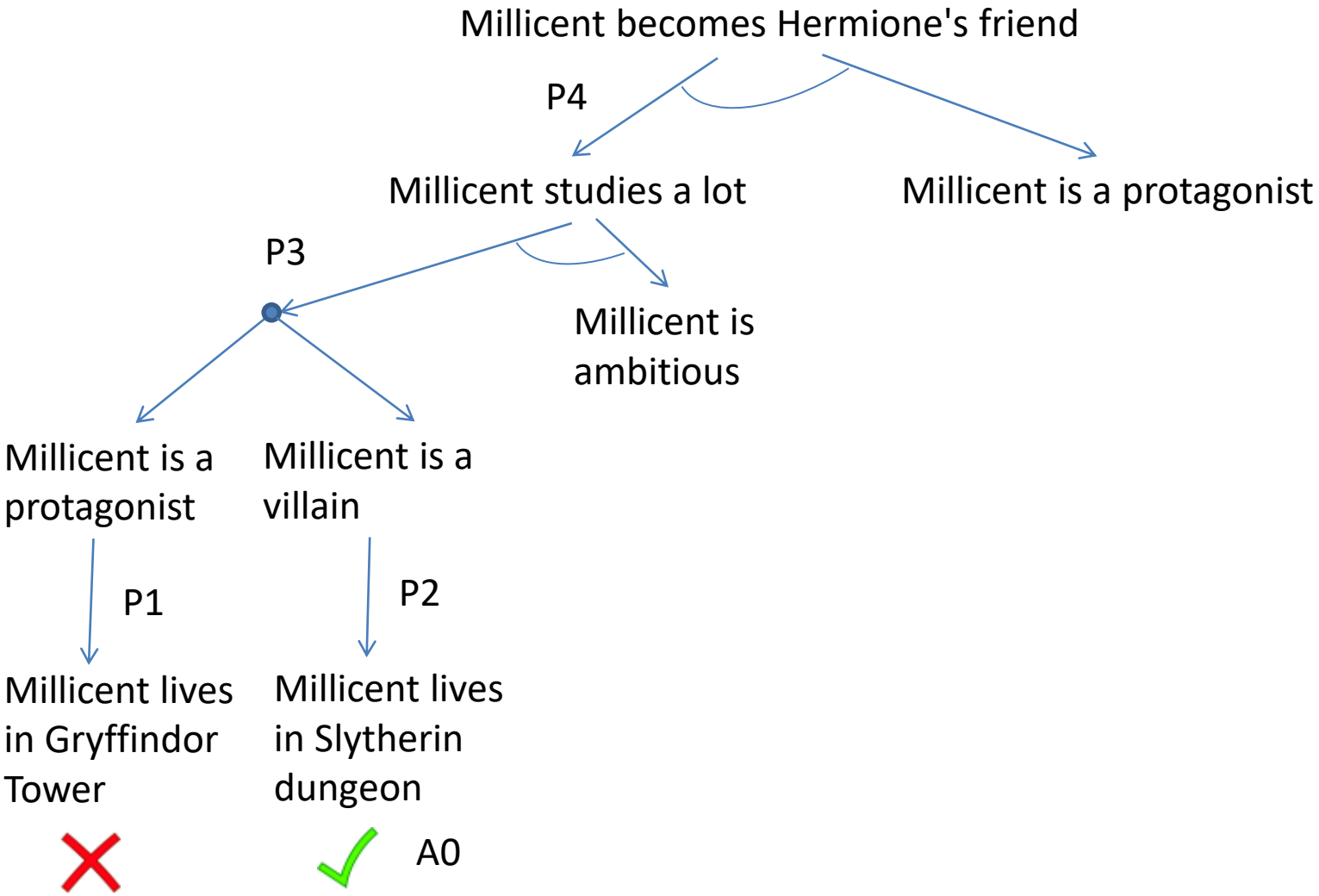
Backward chaining

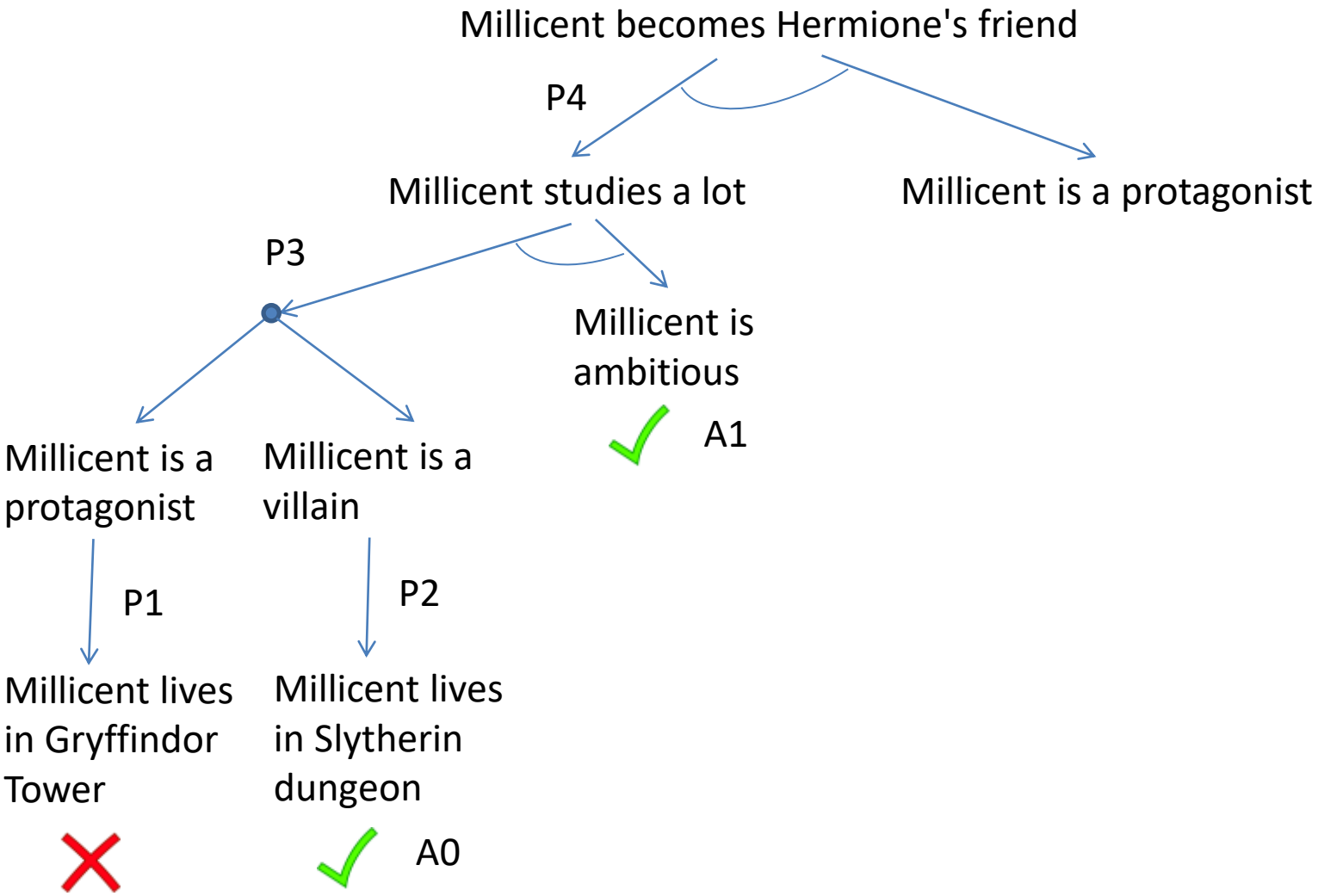
- When working on a hypothesis, the backward chainer tries to find a matching assertion in the list of assertions.
  - If we must demonstrate that “Seamus snogs Millicent” then we done. Because it’s in assertion list then we prove it.
- If no matching assertion is found, the backward chainer tries to find a rule with a matching consequent.
  - If we must demonstrate that “Seamus is a protagonist” then based on P1 we can demonstrate that someone is a protagonist if is living in GT, and we have that Seamus lives in GT, so again we done.
- In case none are found, then the backward chainer assumes the hypothesis is false.
- The backward chainer never alters the list of assertions, so it can derive the same result multiple times.
- Rules are tried in the order they appear.
- Antecedents are tried in the order they appear
- The goal tree is traverse in depth-first order

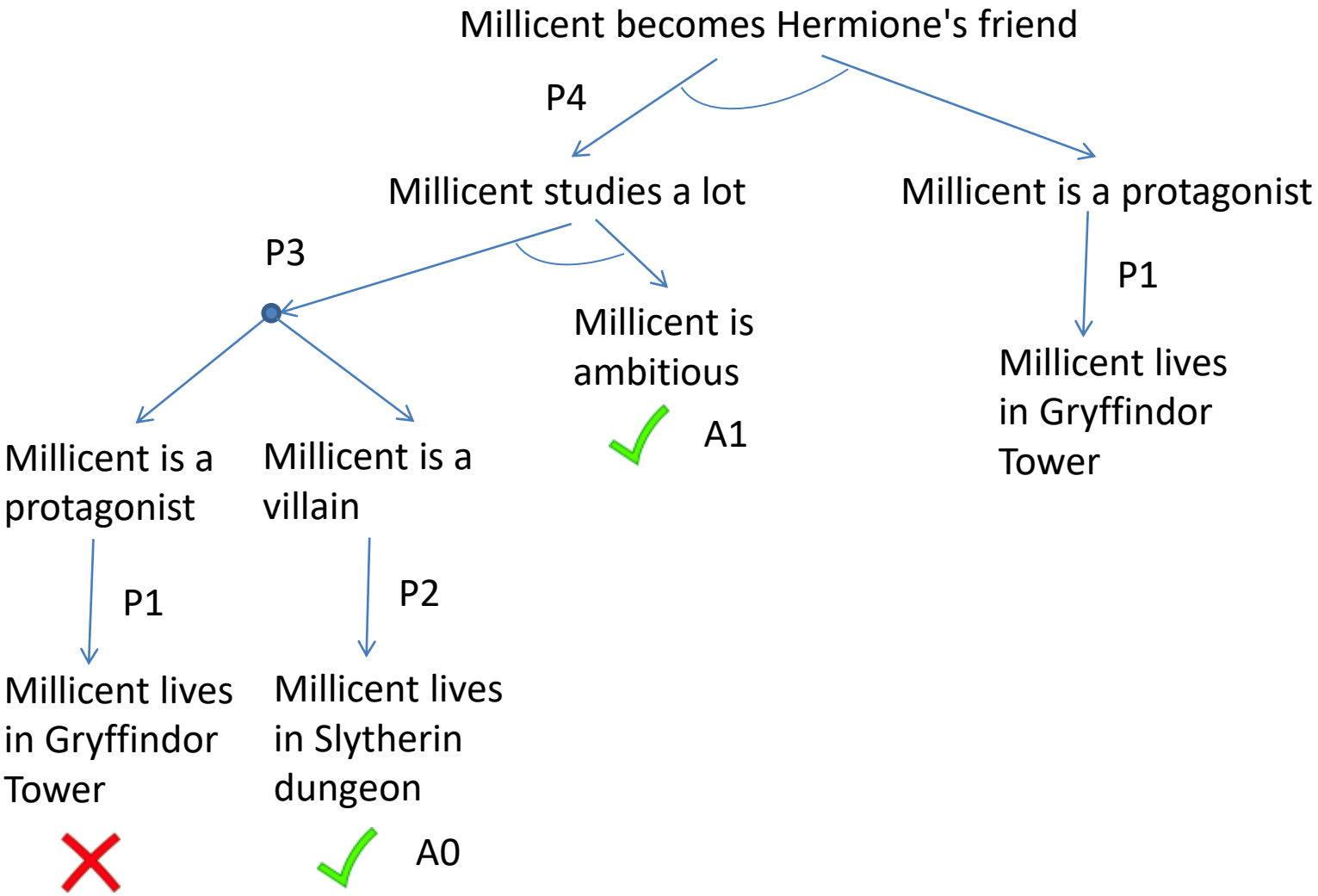

Millicent becomes Hermione's friend
P4
Millicent studies a lot
Millicent is a protagonist

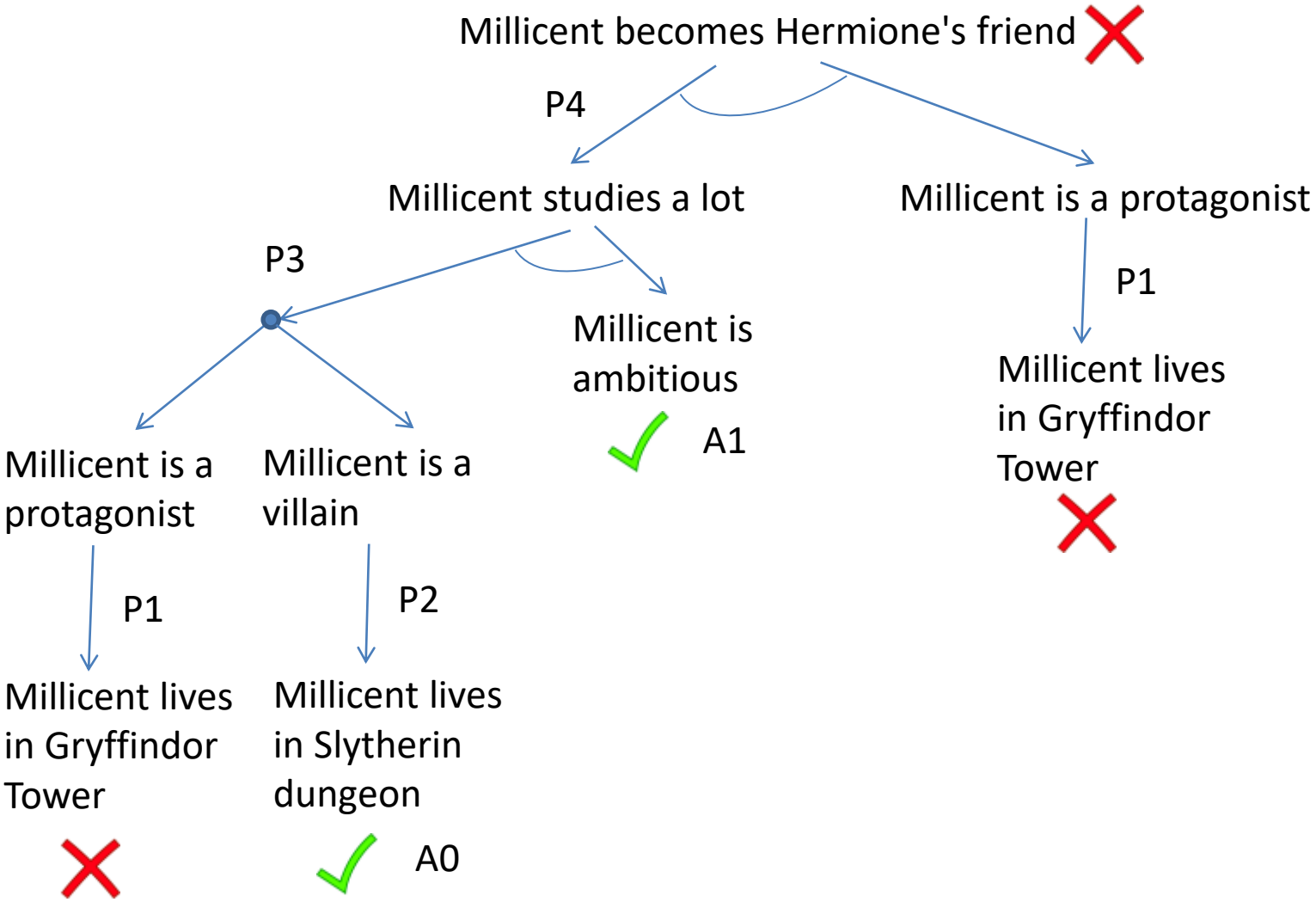

Millicent becomes Hermione's friend
P4
Millicent studies a lot
Millicent is a protagonist
P3
Millicent is ambitious
Millicent is a protagonist
Millicent is a villain


Millicent becomes Hermione's friend
P4
Millicent studies a lot
Millicent is a protagonist
P3
Millicent is ambitious
Millicent is a protagonist
Millicent is a villain
P1
Millicent lives in Gryffindor Tower


Millicent becomes Hermione's friend
P4
Millicent studies a lot
Millicent is a protagonist
P3
Millicent is ambitious
Millicent is a protagonist
Millicent is a villain
P1
P2
Millicent lives in Gryffindor Tower
Millicent lives in Slytherin dungeon
A0


Millicent becomes Hermione's friend
P4
Millicent studies a lot
Millicent is a protagonist
P3
Millicent is ambitious
A1
Millicent is a protagonist
Millicent is a villain
P1
P2
Millicent lives in Gryffindor Tower
Millicent lives in Slytherin dungeon
A0


Millicent becomes Hermione's friend
P4
Millicent studies a lot
Millicent is a protagonist
P3
P1
Millicent is ambitious
A1
Millicent lives in Gryffindor Tower
Millicent is a protagonist
Millicent is a villain
P1
P2
Millicent lives in Gryffindor Tower
Millicent lives in Slytherin dungeon
A0


Millicent becomes Hermione's friend
P4
Millicent studies a lot
Millicent is a protagonist
P3
P1
Millicent is ambitious
A1
Millicent lives in Gryffindor Tower
Millicent is a protagonist
Millicent is a villain
P1
P2
Millicent lives in Gryffindor Tower
Millicent lives in Slytherin dungeon
A0

A2 part

- Now, determine the minimum number of additional assertions required for Millicent to become Hermione's friend and list those assertions. Include no assertion that matches the consequent of a rule.

Millicent lives in Gryffindor Tower

- Your solution to Part A2 creates an uncommon situation. What is that uncommon situation and if J. K. considers the situation to be a problem, what should she do to the list of assertions to solve the problem?

remove the A0

Forward chaining

- Assume rule-ordering conflict resolution
- New assertions are added to the bottom of the list of assertions.
- If a particular rule matches assertions in the list of assertions in more than one way, the matches are considered in the order corresponding to the top-to-bottom order of the matched assertions. Thus, if a particular rule has an antecedent that matches both A1 and A2, the match with A1 is considered first.

- Run forward chaining on the rules and assertions. For the first two iterations, fill out the first two rows in the table below, noting the rules whose antecedents match the data, the rule that fires, and the new assertions that are added by the rule. For the remainder, supply only the fired rules and new assertions.

Matched	Fired	New Assertions Added to List of Assertions
P1,P2,P5	P1	Seamus is a protagonist
P1,P2,P5	P2	Millicent is a villain
P1,P2,P3,P5	P3	Millicent studies a lot
P1,P2,P3,P5	P5	Seamus has a bad term

Related resources

- http://aitopics.org/sites/default/files/classic/Feigenbaum_Feldman/Computers_And_Thought-Part_1_SAINT.pdf
- http://ocw.mit.edu/courses/electrical-engineering-and-computer-science/6-034-artificial-intelligence-fall-2010/exams/MIT6_034F10_quiz1_2009.pdf

Readings

- Artificial Intelligence (3rd Edition), Patrick Winston, pp. 53-60

Problem: Rule Systems

(http://ocw.mit.edu/courses/electrical-engineering-and-computer-science/6-034-artificial-intelligence-fall-2010/exams/MIT6_034F10_quiz1_2009.pdf)

Due to constant pressure from the AIF staff, J. K. Rowling decides to write an 8th Harry Potter book. But, she's suffering from a bad case of writer's block and decides to use a rule-based system to help her with the plot for Harry Potter and the Deadhorse Principle. She's given you a set of rules and assertions and would like your help with developing key plot points.

Rules:

P0:	IF (AND('(?x) is ambitious', '(?x) is a squib') THEN '(?x) has a bad term')
P1:	IF ('(?x) lives in Gryffindor Tower') THEN ('(?x) is a protagonist')
P2:	IF (('(?x) lives in Slytherin dungeon') THEN ('(?x) is a villain'), '(?x) is ambitious))
P3:	IF (AND(OR('(?x) is a protagonist', '(?x) is a villain'), '(?x) is ambitious') THEN ('(?x) studies a lot'))
P4:	IF (AND('(?x) studies a lot', '(?x) is a protagonist') THEN ('(?x) becomes Hermione's friend'))
P5:	IF (AND('(?x) snogs (?y)', '(?x) lives in Gryffindor Tower', '(?y) lives in Slytherin dungeon') THEN ('(?x) has a bad term'))

Assertions:

A0:	(Millicent lives in Slytherin dungeon)
A1:	(Millicent is ambitious))
A2:	(Seamus lives in Gryffindor Tower)
A3:	(Seamus snogs Millicent)

Part A: Backward Chaining

Make the following assumptions about backwards chaining:

- When working on a hypothesis, the backward chainer tries to find a matching assertion in the list of assertions. If no matching assertion is found, the backward chainer tries to find a rule with a matching consequent. In case none are found, then the backward chainer assumes the hypothesis is false.
- The backward chainer never alters the list of assertions, so it can derive the same result multiple times.
- Rules are tried in the order they appear.
- Antecedents are tried in the order they appear.

Part A1

JK knows she would like Millicent to become friends with Hermione. To help her figure out what other assertions must be satisfied, draw the goal tree for the hypothesis:

(Millicent becomes Hermione's friend)

Part A2

Now, determine the minimum number of additional assertions required for Millicent to become Hermione's friend and list those assertions. **Include no assertion that matches the consequent of a rule.**

Part A3

Your solution to Part A2 creates an uncommon situation. What is that uncommon situation and if J. K. considers the situation to be a problem, what should she do to the list of assertions to solve the problem?

Part B: Forward Chaining

You may make the following assumptions about forward chaining:

- Assume rule-ordering conflict resolution
- New assertions are added to the bottom of the list of assertions.
- If a particular rule matches assertions in the list of assertions in more than one way, the matches are considered in the order corresponding to the top-to-bottom order of the matched assertions. Thus, if a particular rule has an antecedent that matches both A1 and A2, the match with A1 is considered first.

Run forward chaining on the rules and assertions provided on page 1. For the first two iterations, fill out the first two rows in the table below, noting the rules whose antecedents match the data, the rule that fires, and the new assertions that are added by the rule. For the remainder, supply only the fired rules and new assertions.

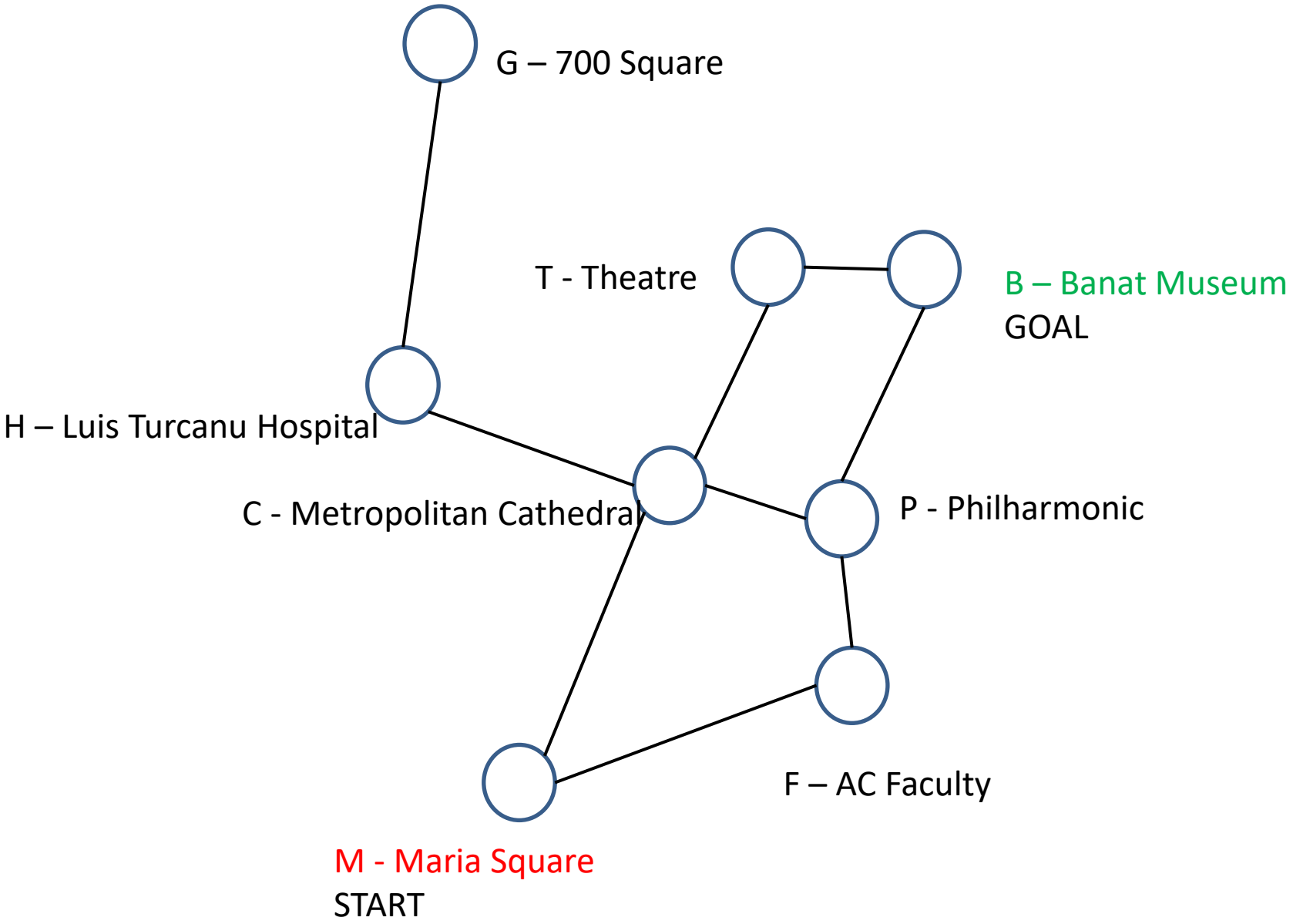
Artificial Intelligence Fundamentals

Search: Depth-First, Hill Climbing, Beam, Optimal, Branch and Bound, A*

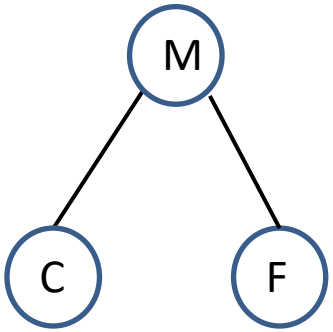
Search - Terminology

- Search algorithm – an algorithms that takes a problem as input and returns a **solution** in the form of an **action sequence**.
- A problem can be defined formally by four components:
  - **Initial state**
  - Possible actions available defined by a **successor function**
  - The **goal test** – determines whether a given state is a goal state
  - A **path cost** – function that assigns a numeric cost to each path.
- Informed vs. Uninformed search – there is some evaluation function that guide your search
- Complete vs. Incomplete search – if there is a solution the algorithm will find it
- Optimal vs. Non-optimal – the solution found is also the best one

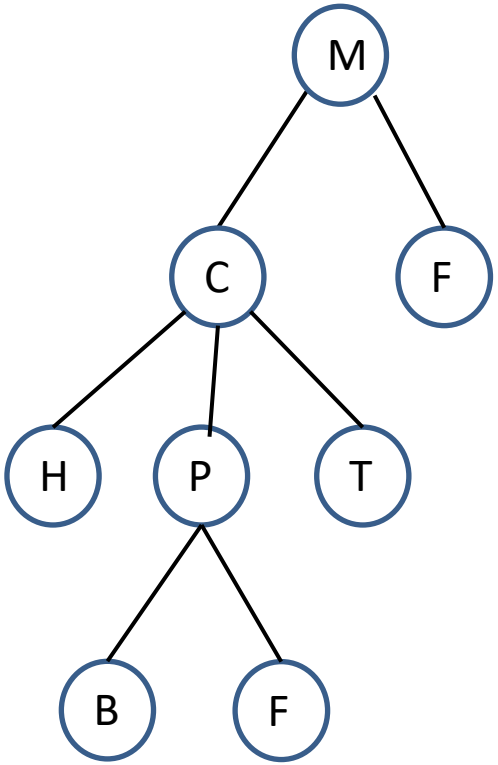
Problem – Navigation in Timisoara



Problem – Navigation in Timisoara



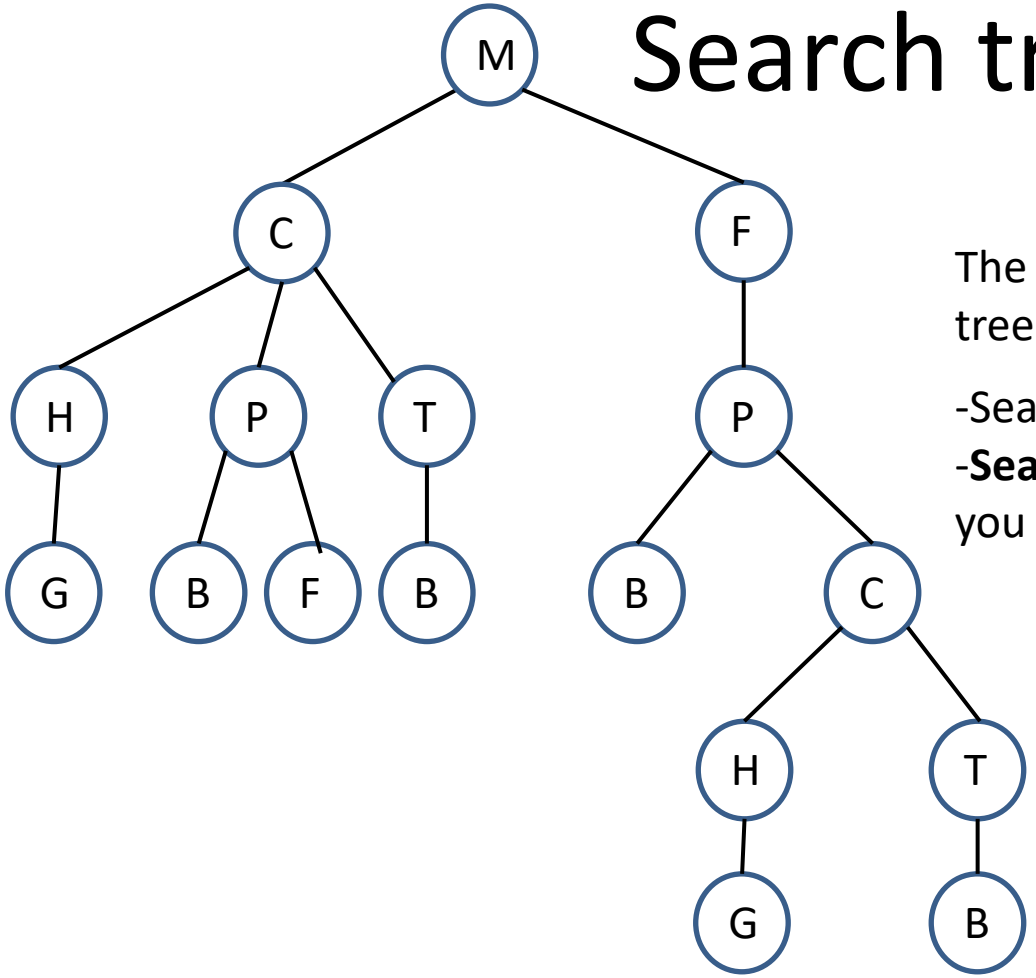
Convention 1 : The nodes appear in lexical order



Convention 2 : When expanding a node don't put an already visited node as a valid successor. (E.g. M is not a valid successor for C and M is not a valid successor for F)

Problem – Navigation in Timisoara

Search tree



The all possible paths – expansion search tree

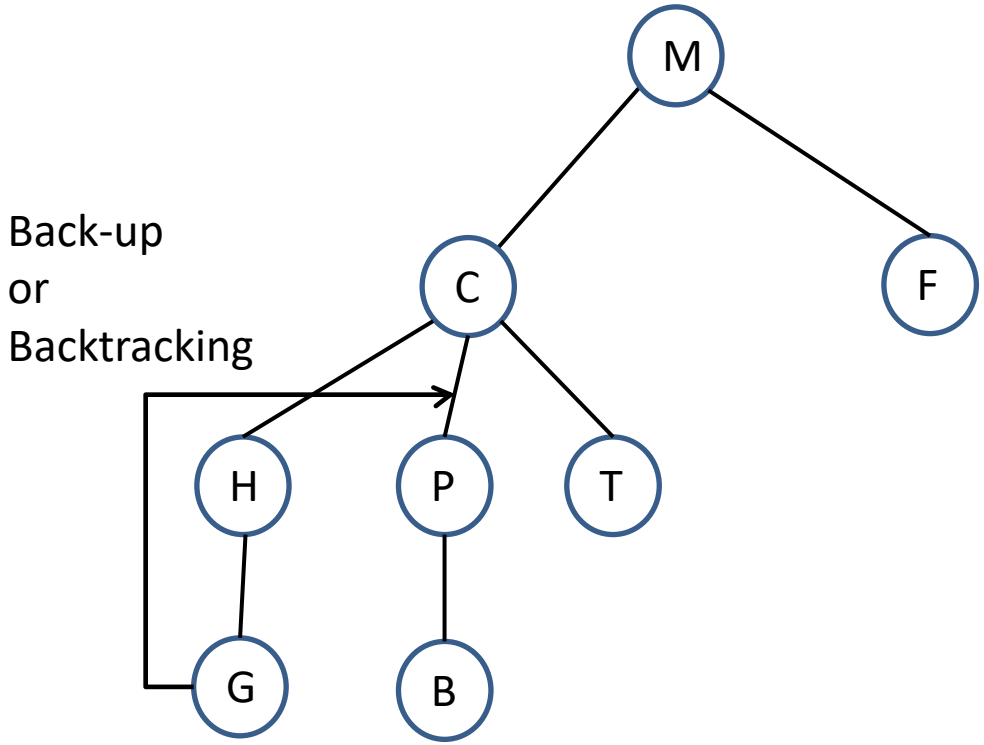
-Search is not equal with maps

-**Search is about choices** you make when you exploit a map

Depth First Search

- Always expands the **deepest** node
- When all nodes have the same deep, expand the **node from the left** (by convention)
- If a node have no successors, the search **backs-up** to the place we made the last decision and choose another branch
- The process is knows as **back-up** or **backtracking**

Depth First Search



Depth First Search – The algorithm

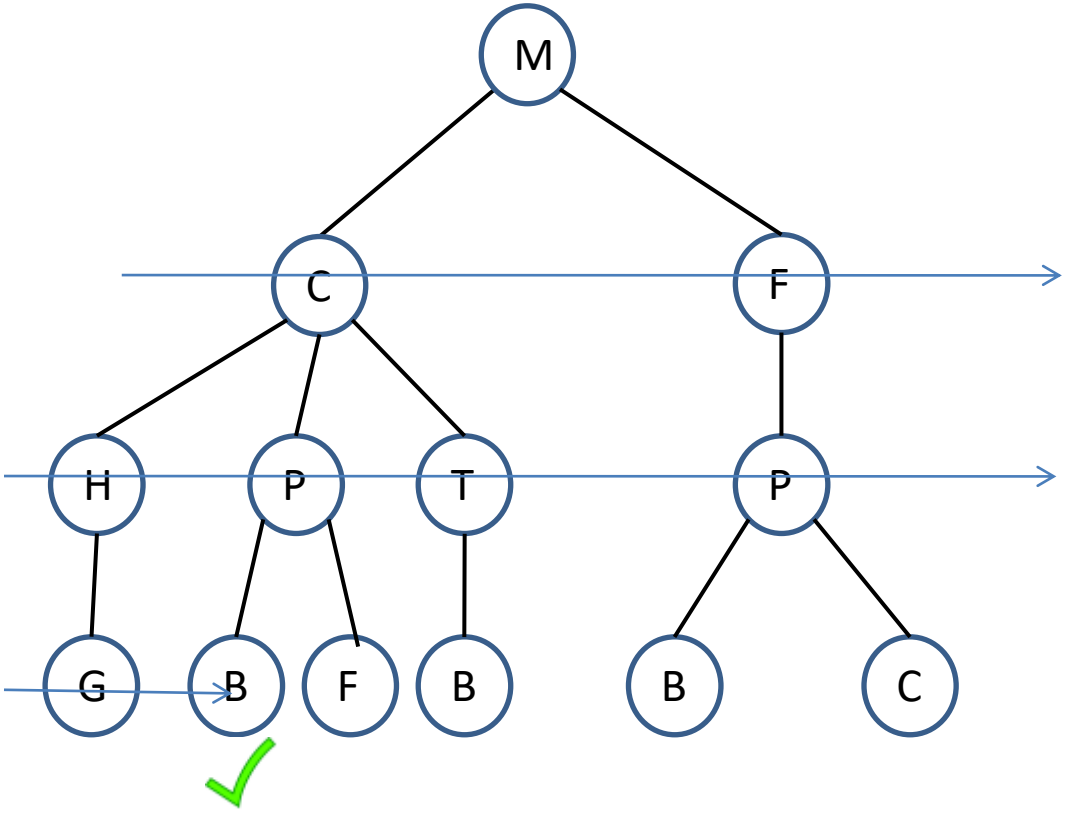
- Initialize the queue with the *Initial state*, having a zero-length path
- **REPEAT** until at least one path in the queue terminates at the goal node OR queue is empty
    - Remove the first path from queue -> *firstPath*
    - Extends the *firstPath* to all the neighbours (except the neighbors that are already in the *firstPath*) of the terminal node -> *newPaths*
    - Add the *newPaths* to the front of the queue
- If the goal is found return SUCCESS

Search characteristics

	Backtracking
Extended search	
Depth First	

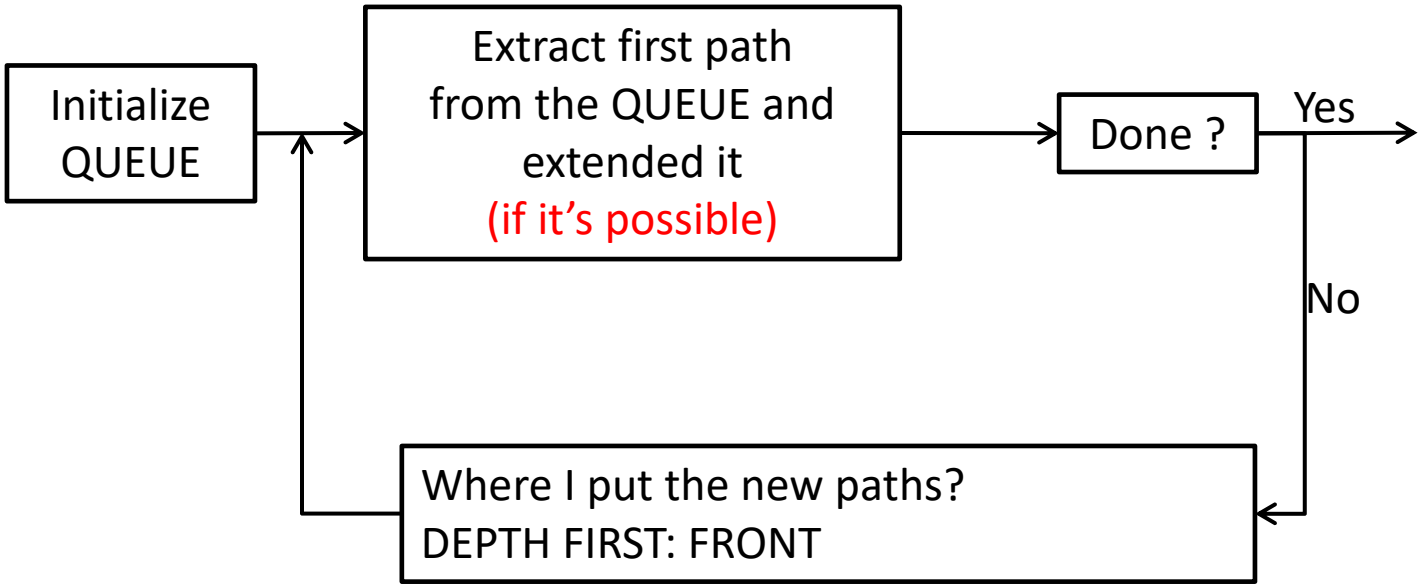
Breadth-first Search

- The **root** node is **expanded** first
- All the node at a **given level are expanded**
- Build up the tree **level by level**

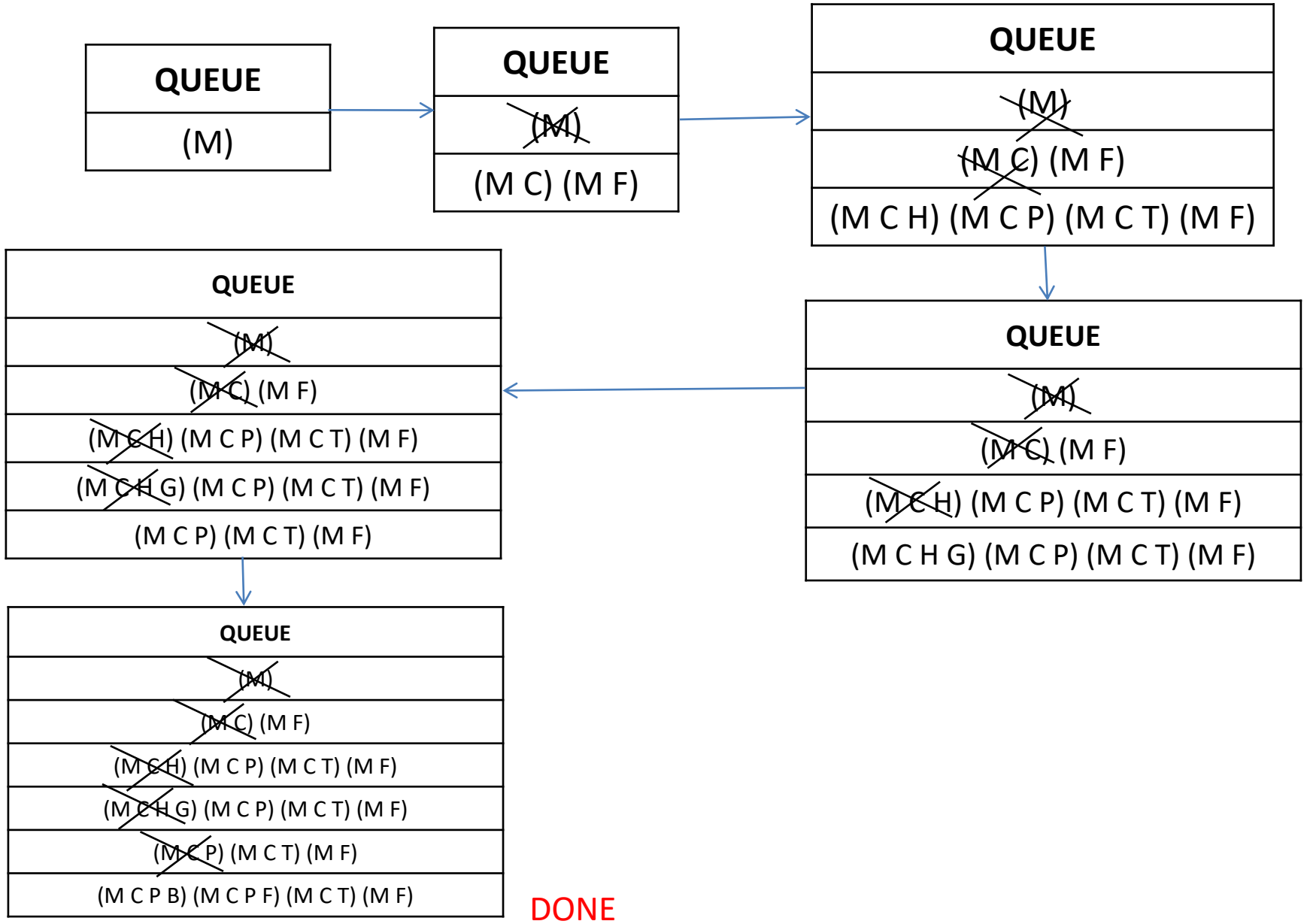


How the search works?

- Helps us to understand the differences between the search algorithms
- Develop a waiting list (**QUEUE**) of paths that are under consideration



Depth First Search

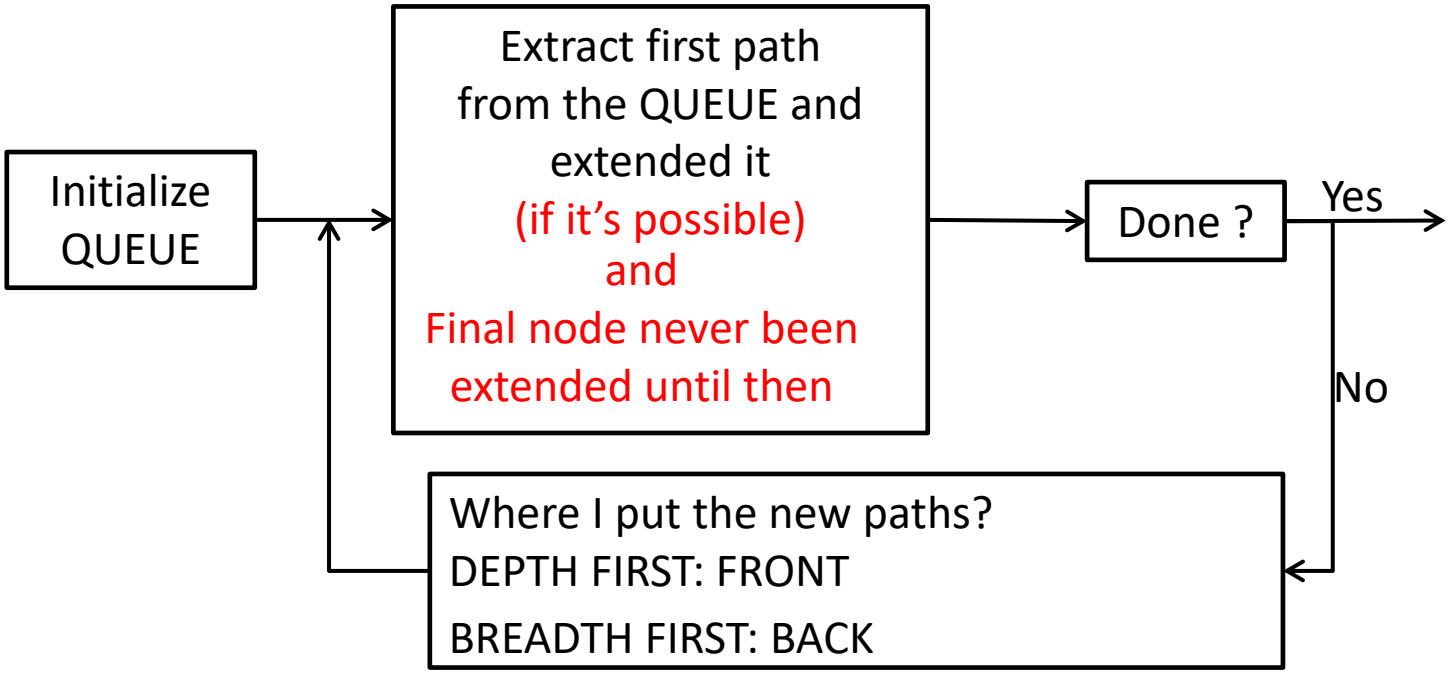


Breadth First Search – The algorithm

- Initialize the queue with the *Initial state*, having a zero-length path
- **REPEAT** until at least one path in the queue terminates at the goal node OR queue is empty
    - Remove the first path from queue -> *firstPath*
    - Extends the *firstPath* to all the neighbours (except the neighbors that are already in the *firstPath*) of the terminal node -> *newPaths*
    - Add the *newPaths* to the back of the queue
- If the goal is found return SUCCESS

How the search works?

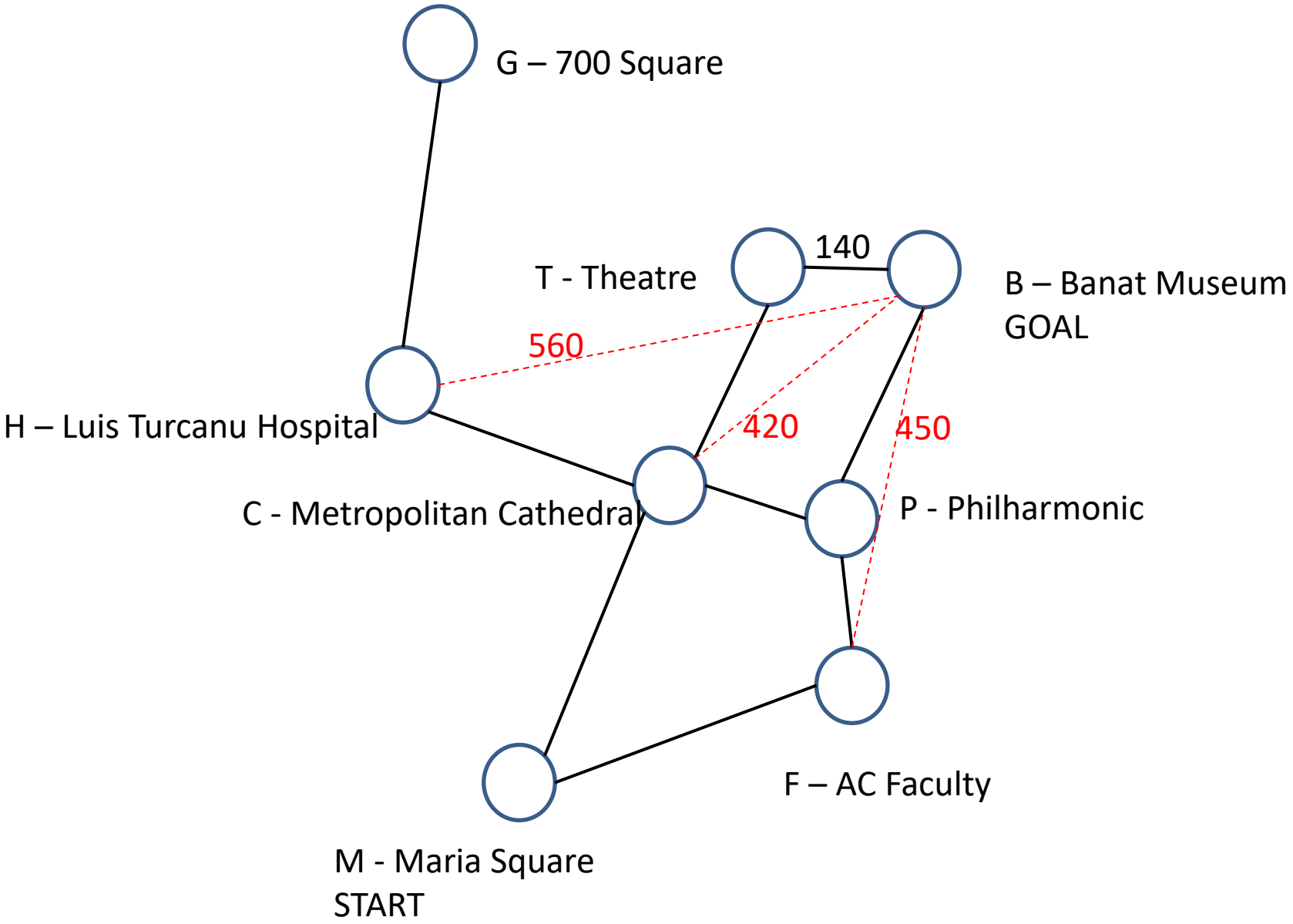
- Helps us to understand the differences between the search algorithms
- Develop a waiting list (**QUEUE**) of paths that are under consideration



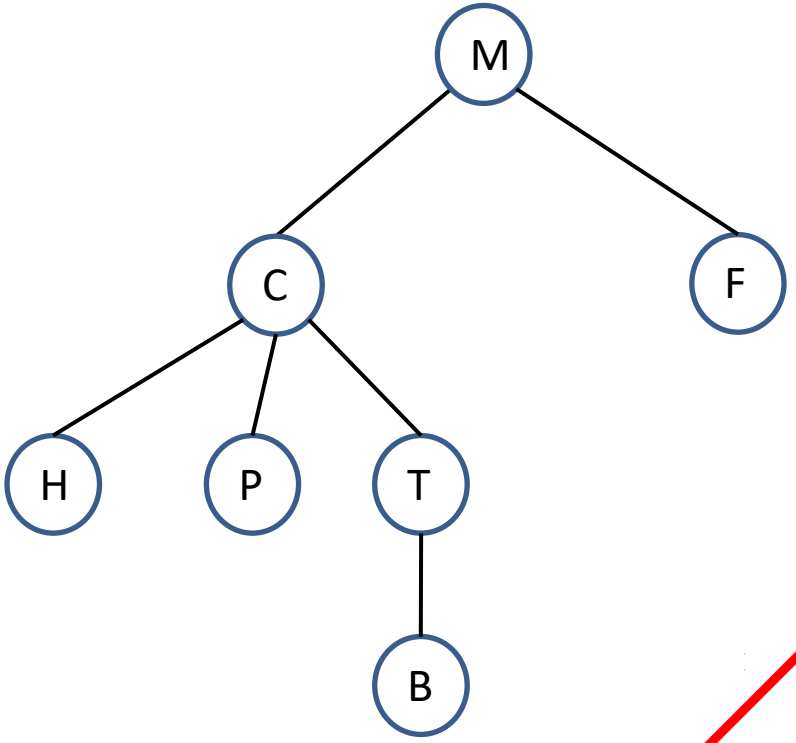
Hill Climbing

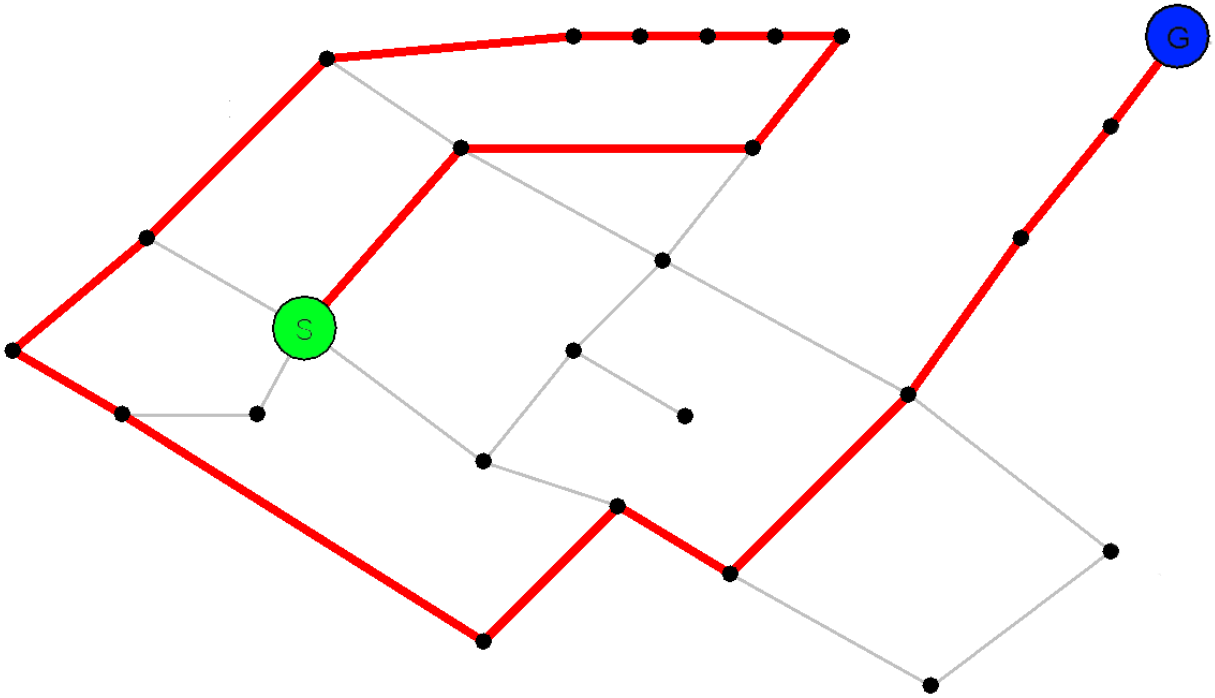
- Depth first search and breadth first search – **uninformed** search strategies
- Incredible **inefficient** in most of the cases
- **Informed(Heuristic)** search
  - uses problem-specific **knowledge ->** create a way to **order the choices**
  - can find solutions more **efficiently**
- Local search algorithms: **best**, **hill-climbing, beam**
- **Hill climbing** (greedy local search)
  - like depth search, but the successors are not listed lexically, they are ordered according to an **objective function**
  - It moves in the direction of **increasing value** (**uphill**)
  - It terminates when it reaches a *peak*
  - Does **not maintain** a search tree

Problem – Navigation in Timisoara



Hill climbing

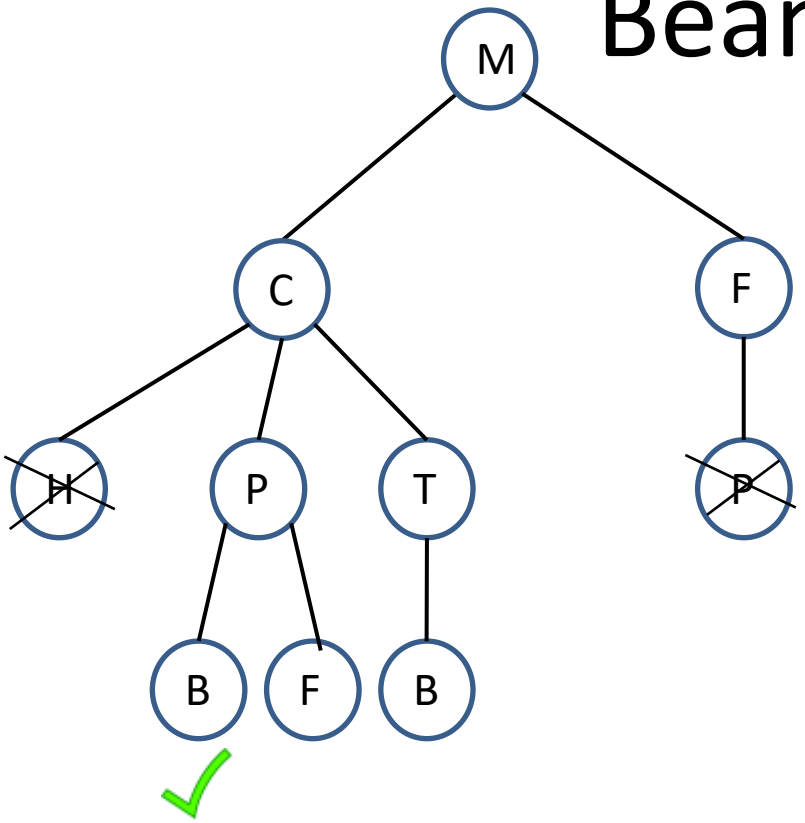




Hill Climbing Search – The algorithm

- Initialize the queue with the *Initial state*, having a zero-length path
- **REPEAT** until at least one path in the queue terminates at the goal node OR queue is empty
    - Remove the first path from queue -> *firstPath*
    - Extends the *firstPath* to all the neighbours (except the neighbors that are already in the *firstPath*) of the terminal node -> *newPaths*
    - Sort the *newPaths* by the estimated distance between the terminates node and the goal
    - Add the new path (if exists) to the front of the queue
- If the goal is found return SUCCESS

Beam search



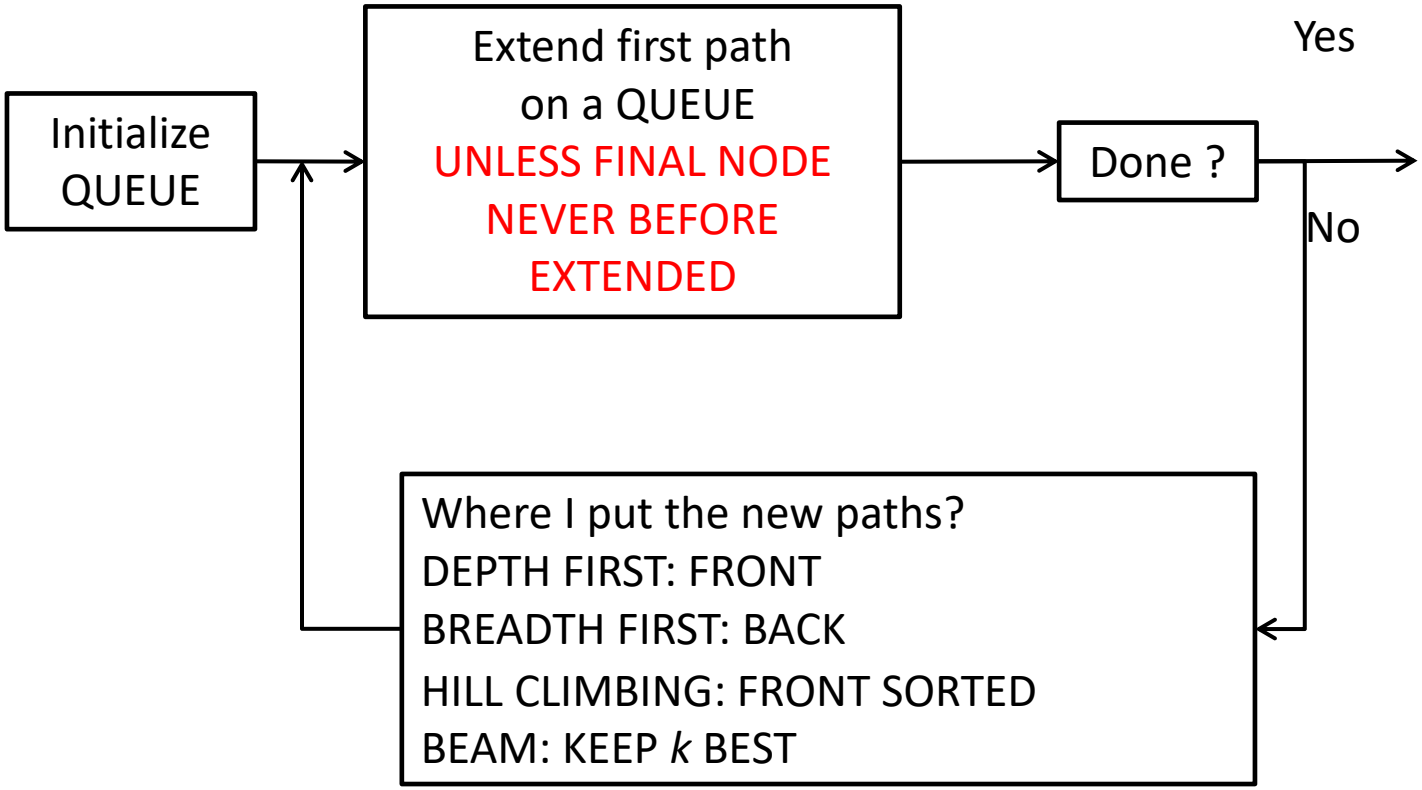
- Like breadth search but keep only a fixed number of possibilities – *k* (usualy small)
- Order the possibilities and take only *k* into consideration
- Example for *k* = 2

Beam Search – The algorithm

- Initialize the queue with the *Initial state*, having a zero-length path
- **REPEAT** until at least one path in the queue terminates at the goal node OR queue is empty
  - Remove the first path from queue -> *firstPath*
  - Extends the *firstPath* to all the neighbours (except the neighbors that are already in the *firstPath*) of the terminal node -> *newPaths*
  - Add the new path (if exists) to the end of the queue
  - If we must go down one level in the search tree, then sort the entire *queue* by the estimated distance between the terminates node and the goal, and keep only first *k* paths
- If the goal is found return SUCCESS

How the search works?

- Helps us to understand the differences between the search algorithms
- Develop a waiting list (**QUEUE**) of paths that are under consideration



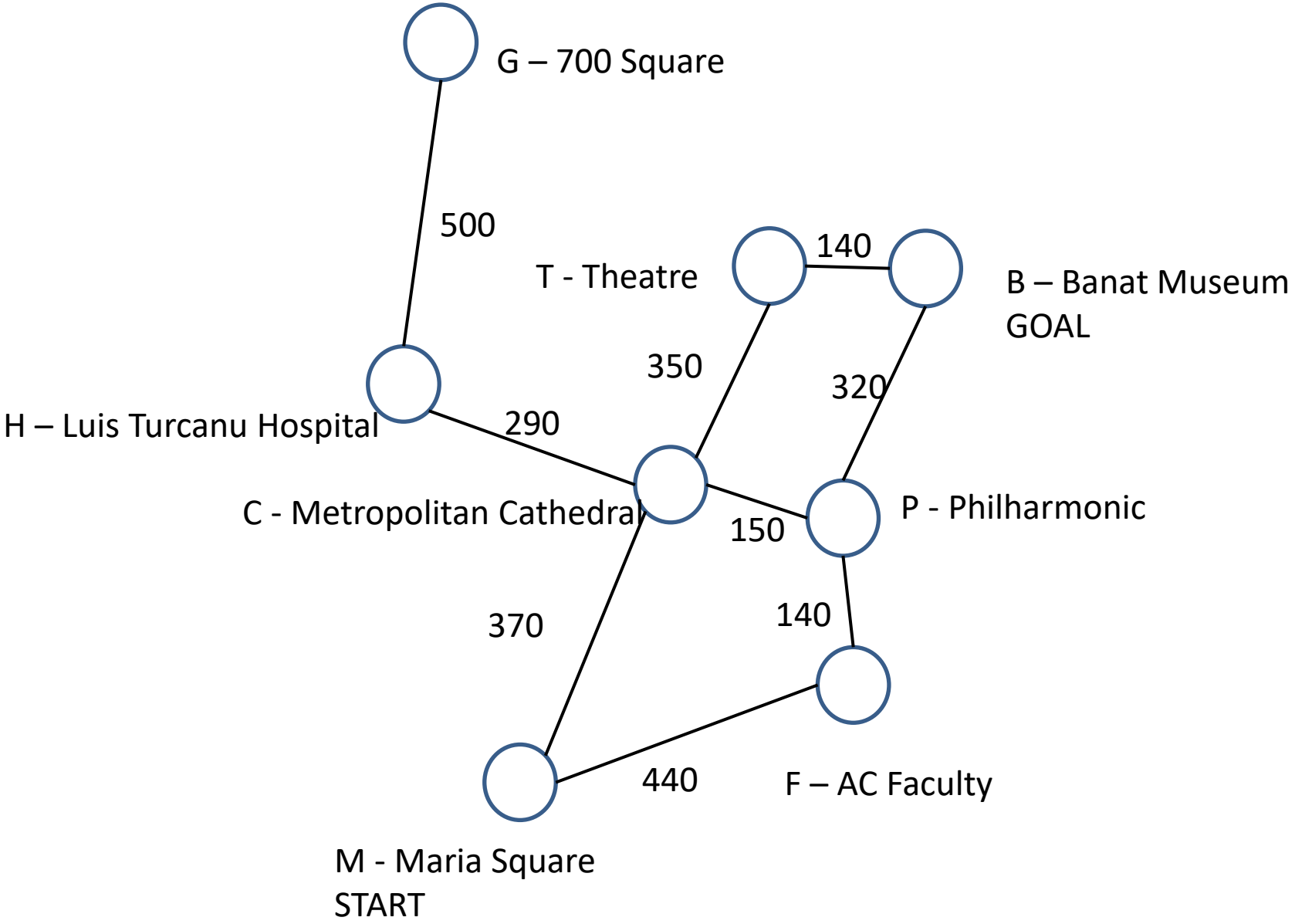
Hill climbing problems

- Problems:
  - Stuck into a local maximum

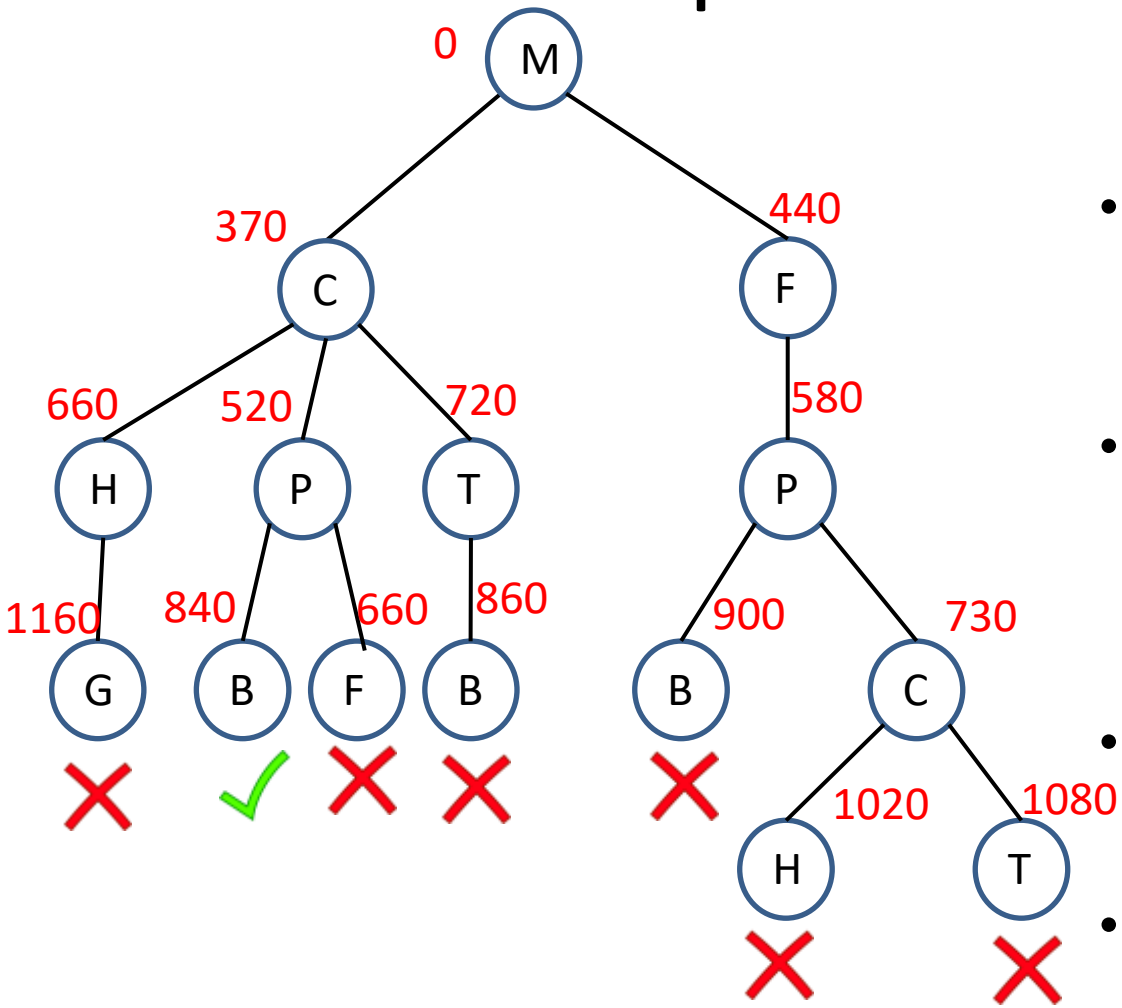
  - Hard to find the *peak*
    (plateau problem)

  - Miss the right direction
    (ridge problem)

Problem – Navigation in Timisoara



Optimal search

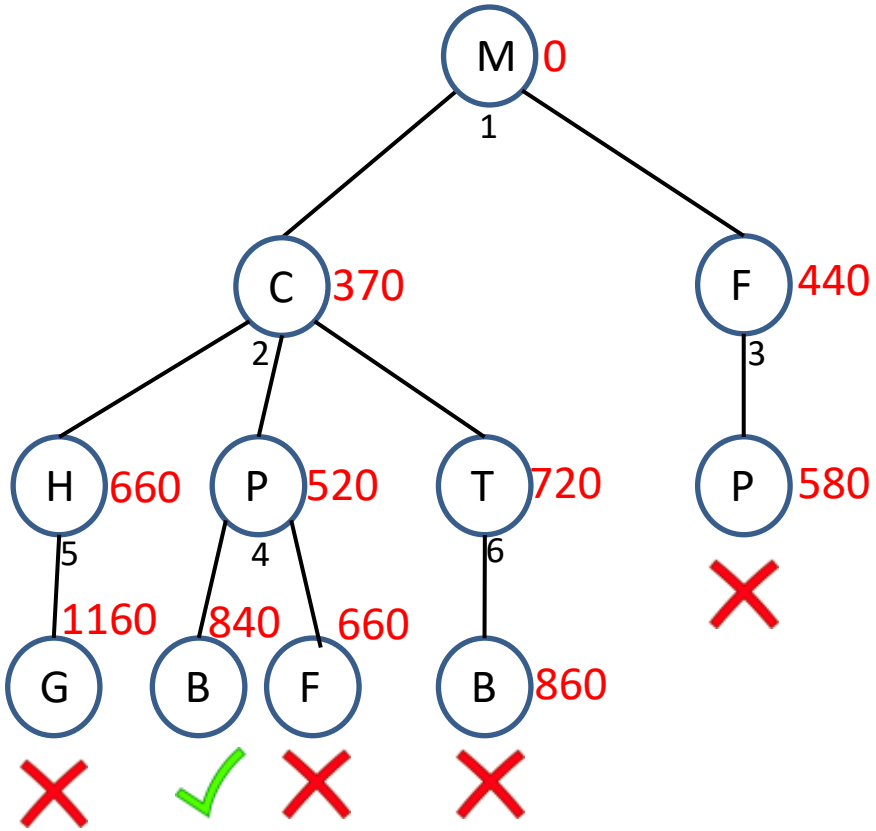


- Deals with search situations in which the cost of traversing a path is of a primary importance
- One procedure -> find all possible paths (depth first or breadth first) and select the best one -> works good for small search trees
- If the search tree is large we must apply an heuristic search
- One solution – branch and bound

Branch and bound search – The algorithm

- Initialize the queue with the *Initial state*, having a zero-length path
- **REPEAT** until at least one path in the queue terminates at the goal node OR queue is empty
  - Remove the first path from queue -> *firstPath*
  - Extends the *firstPath* to all the neighbours (except the neighbors that are already in the *firstPath*) of the terminal node -> *newPaths*
  - Add the *newPath* (if exists) to the queue
  - Sort the entire queue by path length with least-cost paths in front
- If the goal is found return SUCCESS

Branch and bound + Extended List

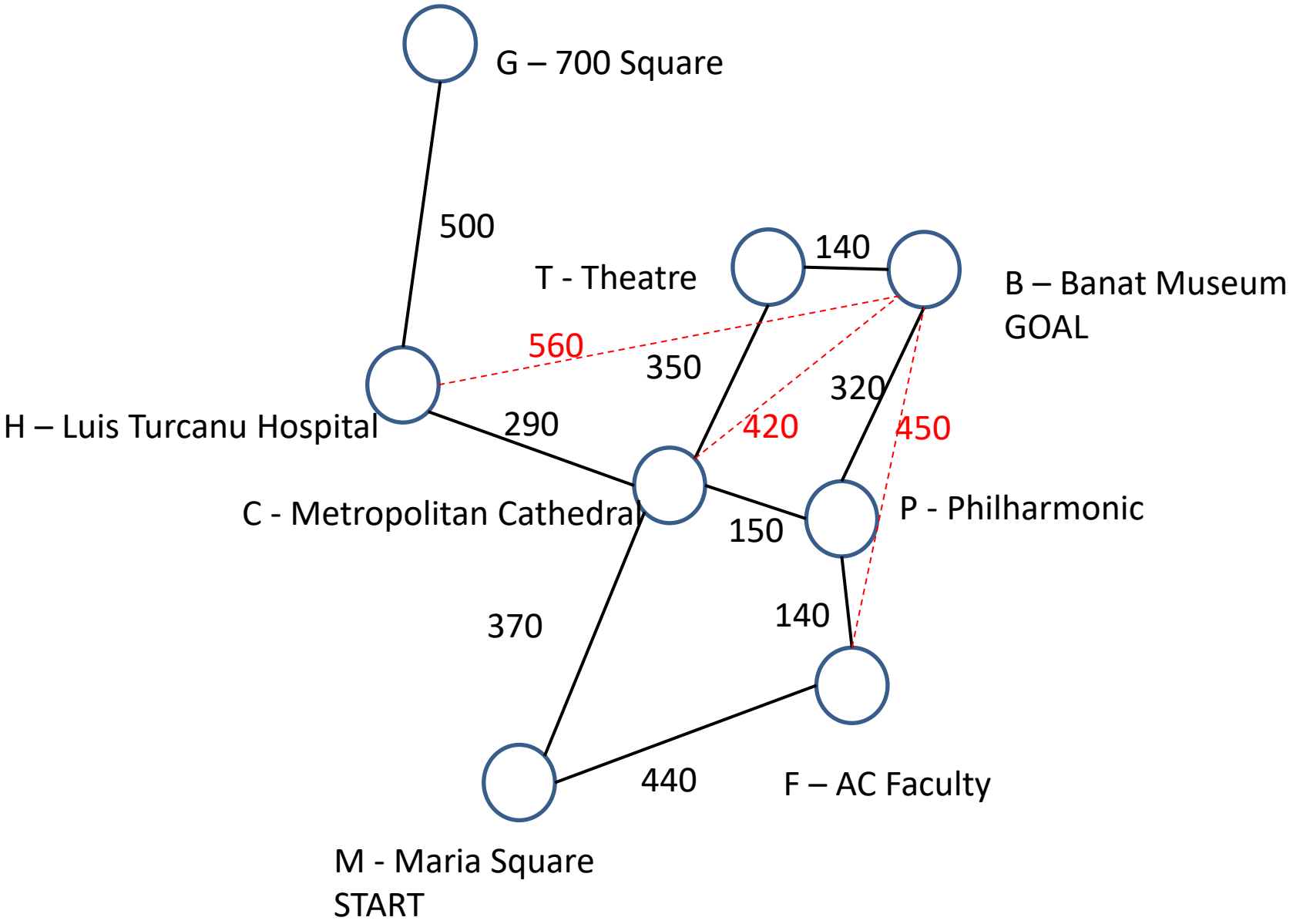


M C F P H T

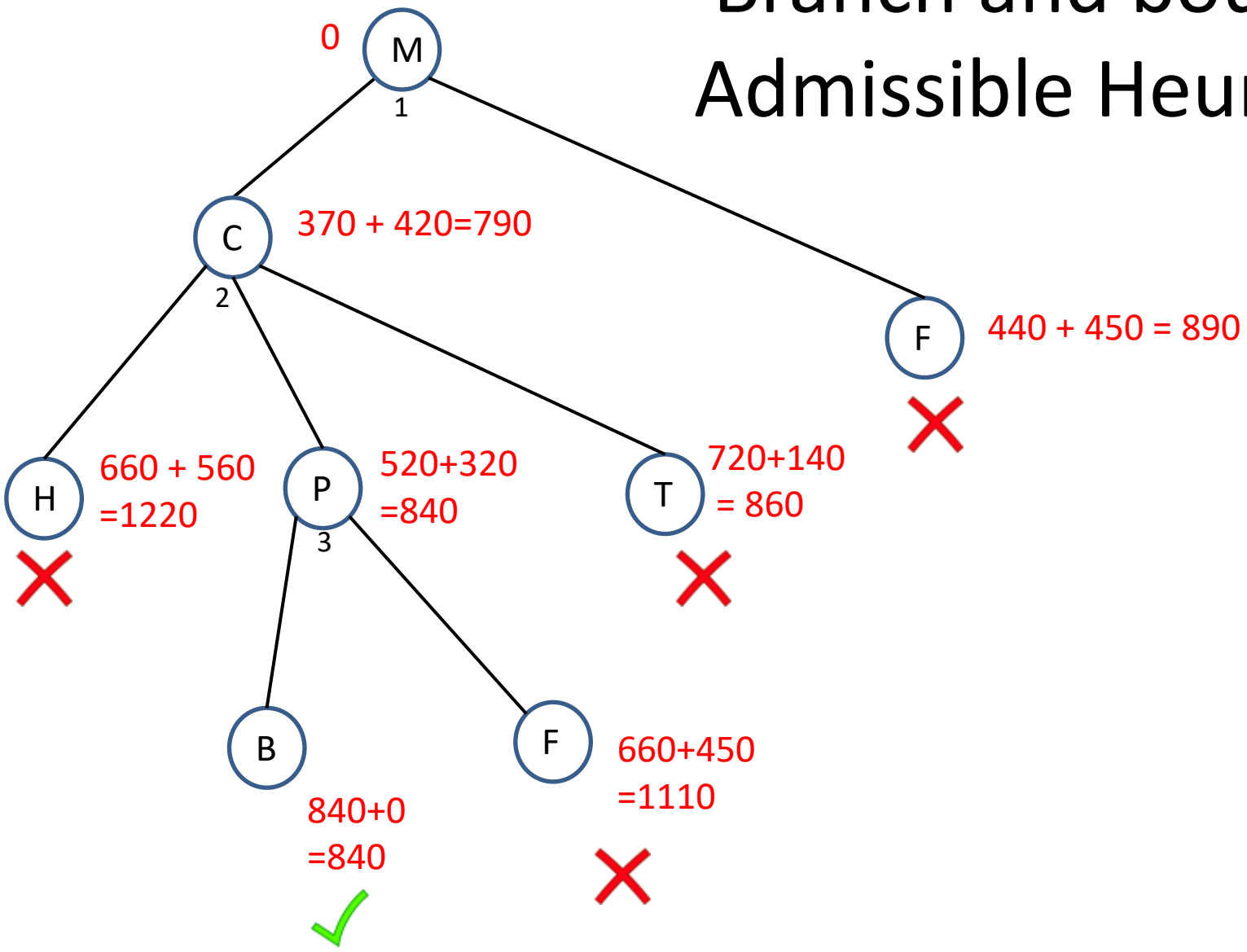
Branch and bound + Admissible heuristics

- We can improve the bound and branch search using guesses about the remaining distance from a node to the goal
- If our guess is good we have an estimation (*e*) of the total path length:
    - *e(total path length) = d(already traversed) + e(distance remaining)*
- In most of the cases guesses are not perfect, and a bad overestimate can cause errors (the right path can be overlooked)
- Admissible heuristics – heuristics that underestimate(*u*) the path; it's always smaller or equal than the real distance
    - *u(total path length) = d(already traversed) + u(distance remaining)*
- When the algorithm found a total path, you don't need to go further because all partial-path distance estimates are longer that the total path you already found (we *underestimate* the distances)

Problem – Navigation in Timisoara



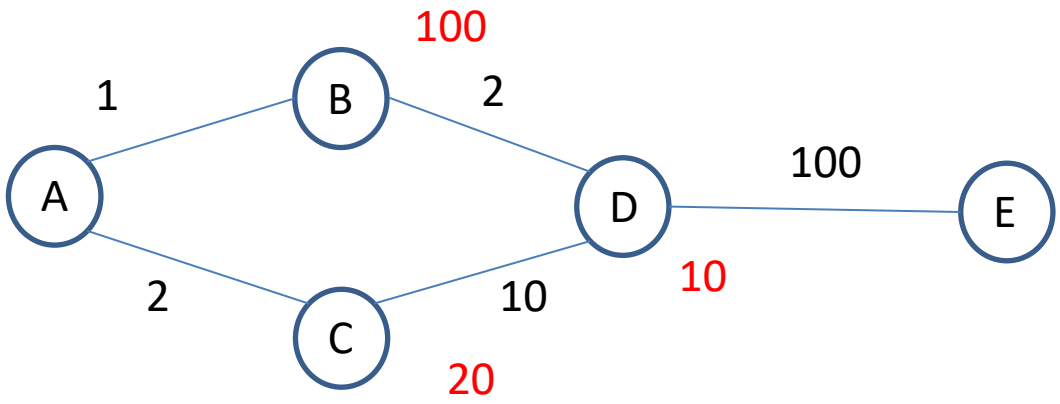
Branch and bound + Admissible Heuristics

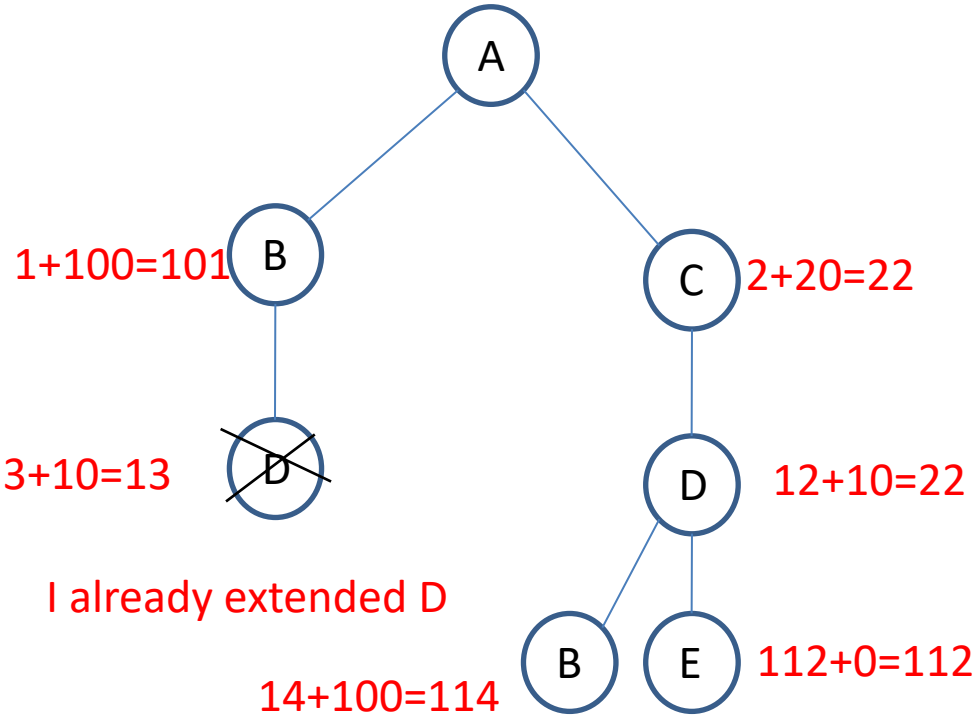


A*

- A* = Branch and bound + Extended list + Admissible heuristics
- Initialize the queue with the *Initial state*, having a zero-length path
- **REPEAT** until at least one path in the queue terminates at the goal node OR queue is empty
    - Remove the first path from queue -> *firstPath*
    - Extends the *firstPath* to all the neighbours (except the neighbors that are already in the *firstPath*) of the terminal node -> *newPaths*
    - Add the *newPath* (if exists) to the queue
    - Sort the entire queue by the sum of the path length and a lower-bound estimate of the cost remaining, with least-cost in front
- If the goal is found return SUCCESS

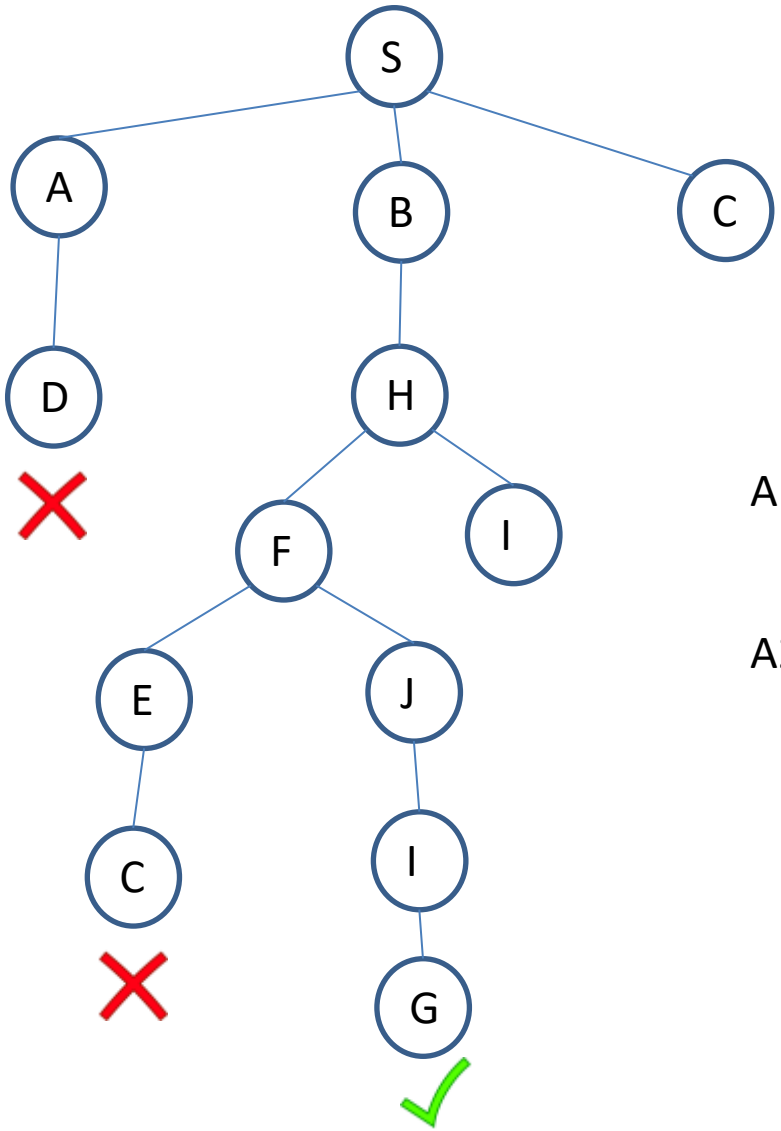
A* - admissibility problem





- ADMISSIBLE heuristic if the estimated distance (H) between any node X and the goal G is less or equal to the actual distance between the X and the goal (works if it's a map)
    - $H(X,G) \leq D(X,G)$
- CONSISTENCY for every 2 nodes X and Y if
    - $| H(X,G) - H(Y,G) | \leq D(X,Y)$

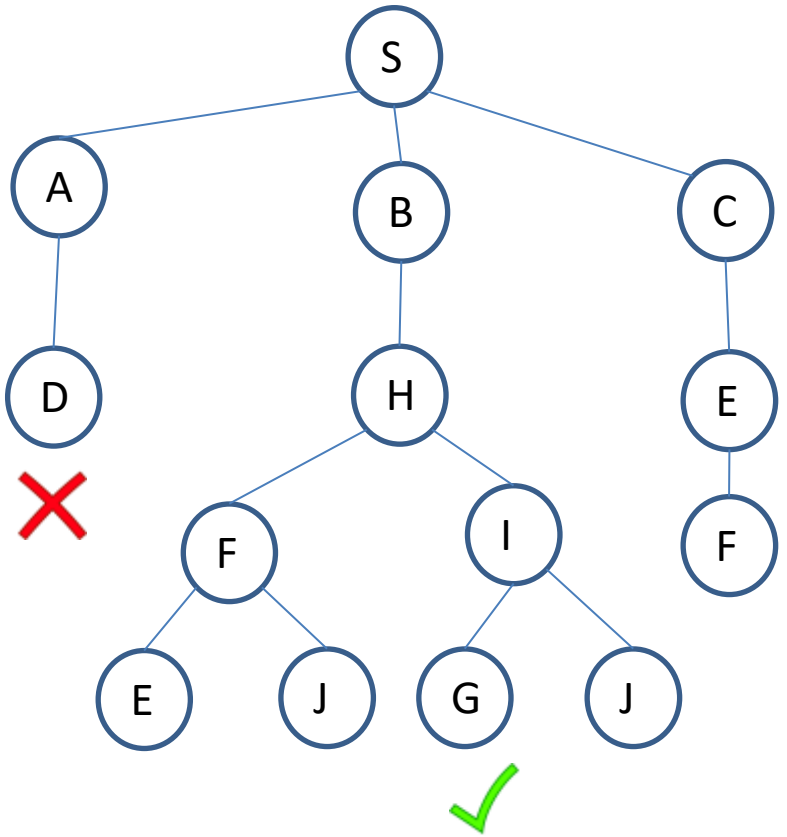
Sample search exam problem



A1 – S , A , D , B , H , F , E , C , J , I

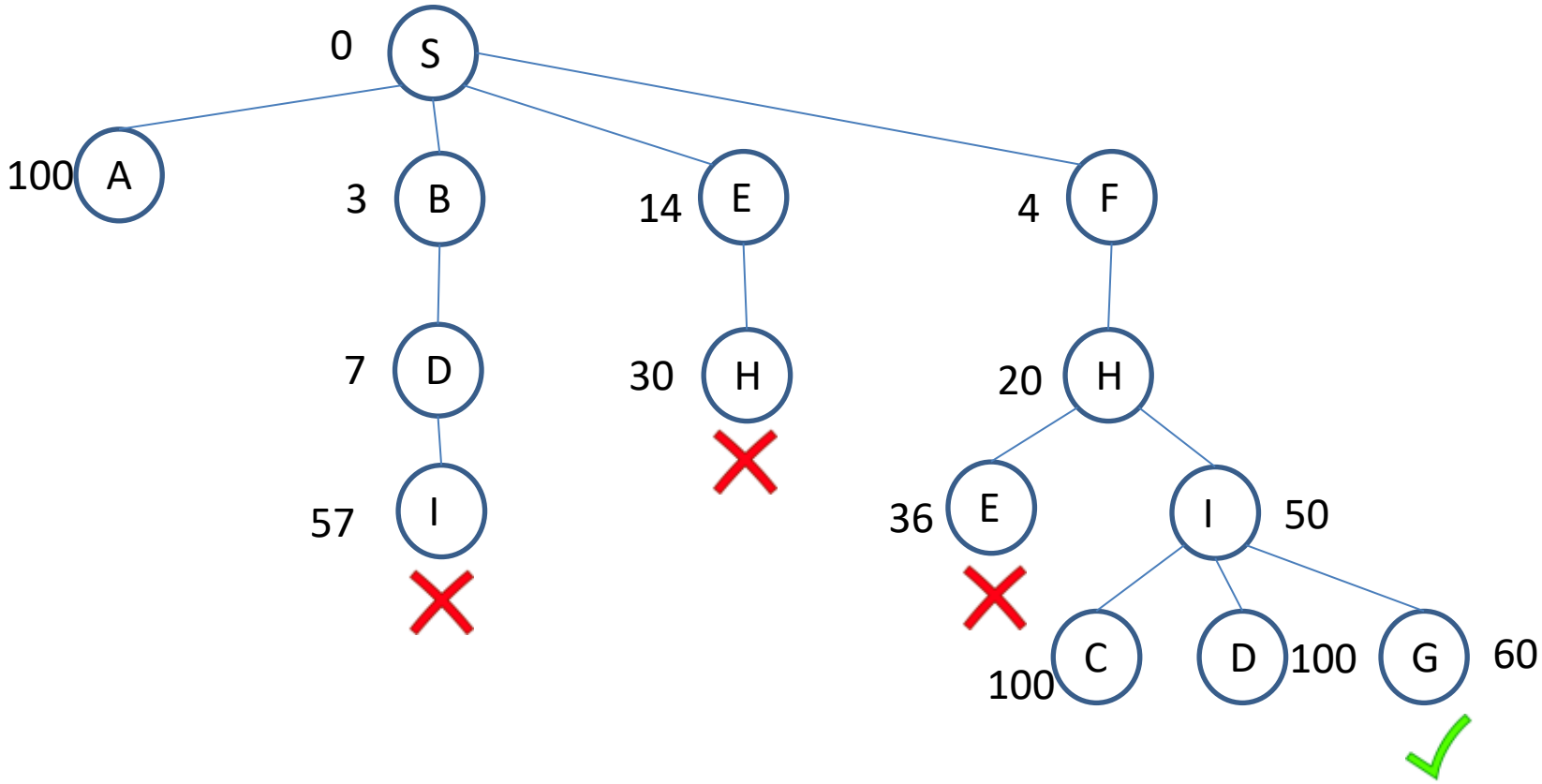
A2 – 2 times

Sample search exam problem



A3 – S , A , B , C , D , H , E , F , I

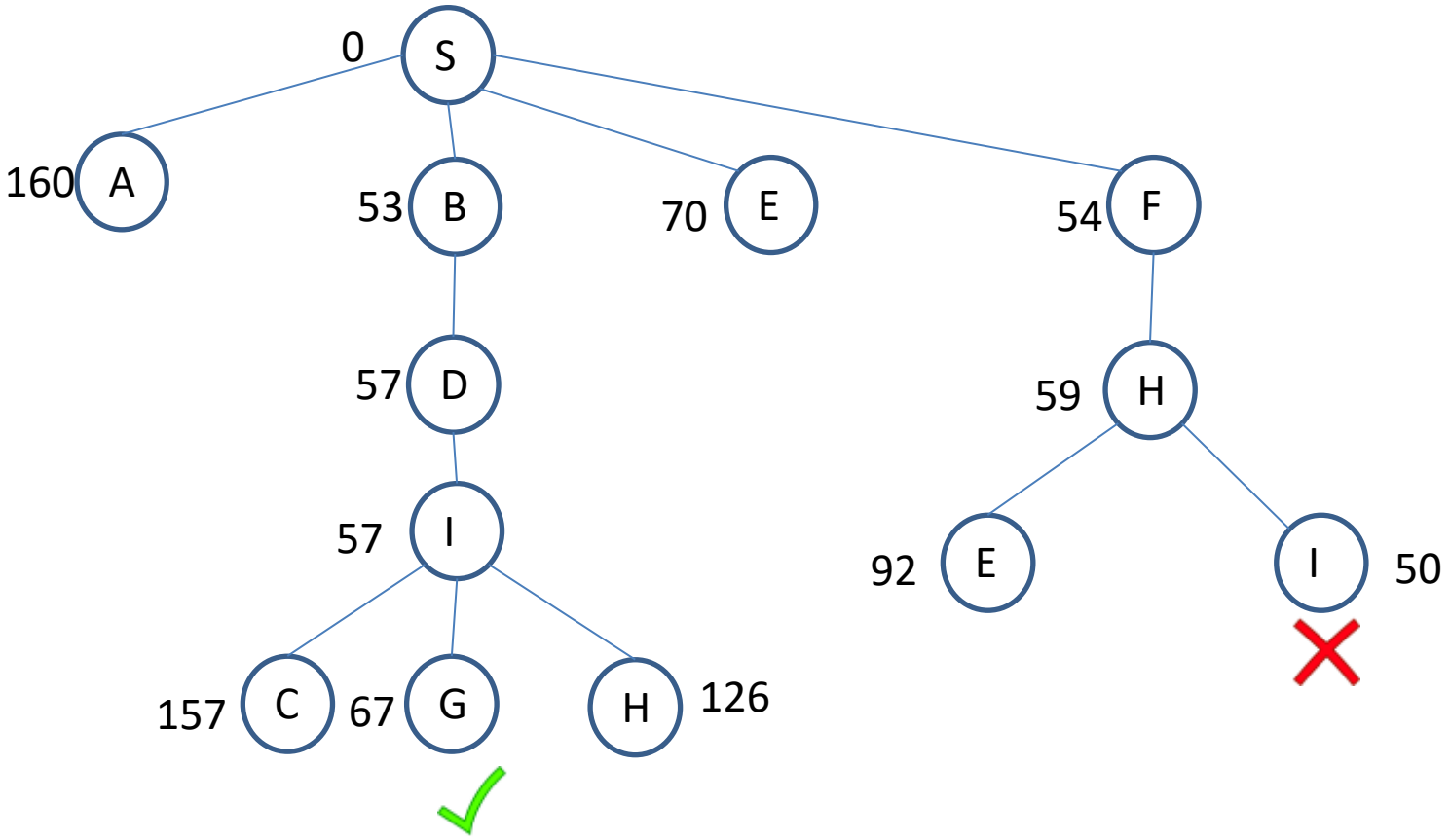
Sample search exam problem



B1 – S , B , F , D , E , H , I

S – F – H – I - G

Sample search exam problem



B2 – S , B , F , D , I , H
S – B – D – I - G

B3 - Heuristics is not consistent.

Related resources

- http://ocw.mit.edu/courses/electrical-engineering-and-computer-science/6-034-artificial-intelligence-fall-2010/exams/MIT6_034F10_quiz1_2008.pdf

Readings

- Artificial Intelligence (3rd Edition), Patrick Winston, Chapter 4 and Chapter 5

Problem : Search

Black Helmet Vader is shopping for a new evil stronghold. Starting from his current stronghold, the Depth-First-Search Star, he can explore the available models by either subtracting or adding a single feature. Fortunately, B.H. remembers how to perform the search techniques he learned in AIF from his mentor Emperor Cosmintine.

Part A: Basic Search

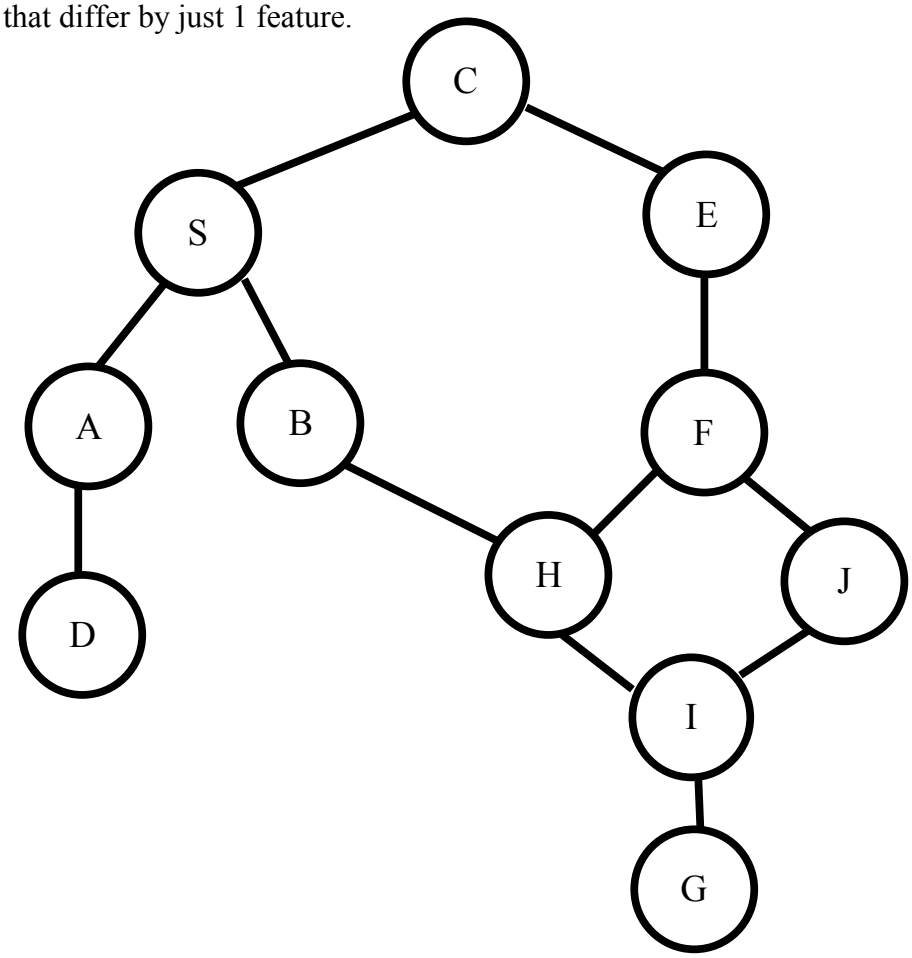
B.H. is looking for a stronghold that has the following qualities:

		Exhaust Pipe Weakness	“That's no Moon”	Race of Enslaved Minions	Secret Escape Route	Sharks with Laser Beams
G	AIF Fortress	-	+	+	+	+

Here is a table of 11 possible strongholds, in tie-breaking order.

		Exhaust Pipe Weakness	“That's no Moon”	Race of Enslaved Minions	Secret Escape Route	Sharks with Laser Beams
S	DFS Star	+	+	-	-	-
A	Shayol Ghul	-	+	-	-	-
B	Dol Guldor	+	+	+	-	-
C	Moonraker	+	-	-	-	-
D	Zeal Underwater Palace	-	+	-	+	-
E	Core of Zeromus	+	-	+	-	-
F	Whalers of the Moon Ride	+	-	+	-	+
G	AIF Fortress	-	+	+	+	+
H	Atlantis	+	+	+	-	+
I	Willy Wonka's Factory	+	+	+	+	+
J	Dr. Evil Moon Base	+	-	+	+	+

Being a clever Overlord, B.H. also produces this graph of exploration **choices** with edges joining the strongholds that differ by just 1 feature.



Part A1

B.H. uses **Depth-First Search with backtracking but no extended list.** He breaks ties according to the order in the table. List the strongholds that B.H. **extends**, in order, starting with the Depth-First-Search Star. Use the letters provided. If he extends a single stronghold more than once, list it multiple times.

Part A2

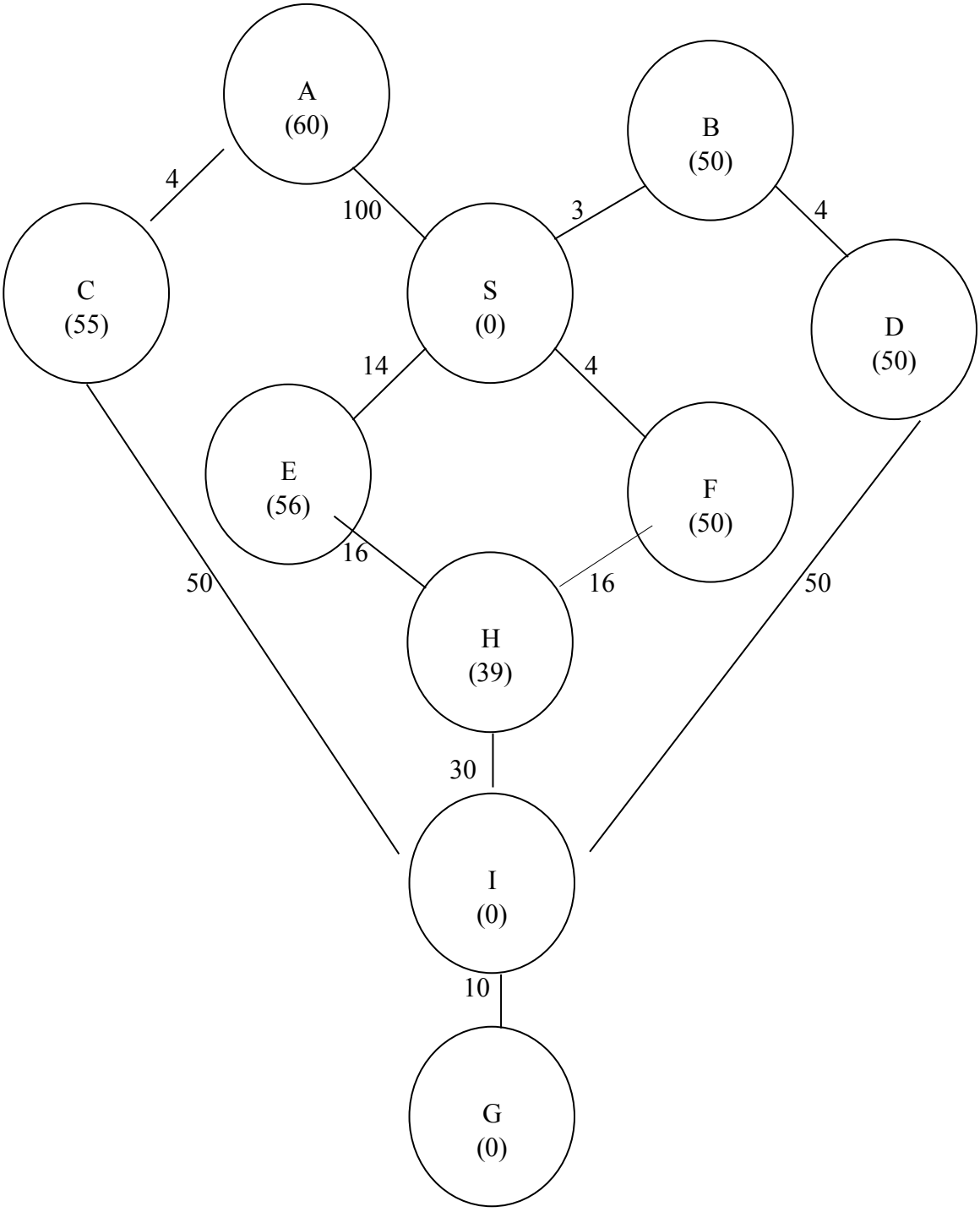
How many times did B.H. hit a dead end?

Part A3

B.H. repeats the process with a **Breadth-First Search with an extended list.** What path does he find?

Part B: Optimal Search

Now that B.H. has his new stronghold, he wants to invade parallel universes. So B.H. programs his evil supercomputer to find the shortest path of jumps from his starting universe S to his goal universe G.



Part B1

First, B.H. programs a simple **branch-and-bound search with an extended list**. As usual, he breaks ties of equal length in lexicographic order. List the nodes B.H.'s computer adds to the extended list, in order. Distances are shown next to edges. Ignore the numbers in parentheses for this part of the problem.

What path does B.H.'s computer find?

Part B2

Frustrated by branch-and-bound's speed, B.H. reprograms his computer to use **A***. B.H. counts the number of subspace anomalies between each universe and the goal and uses this count as the **heuristic** for A* (**these are the numbers in parentheses**). List the nodes B.H.'s computer adds to the extended list, in order.

What path does B.H.'s computer find now?

Part B3

B.H. is confused. Give a **brief** but **specific** explanation of what happened and why.

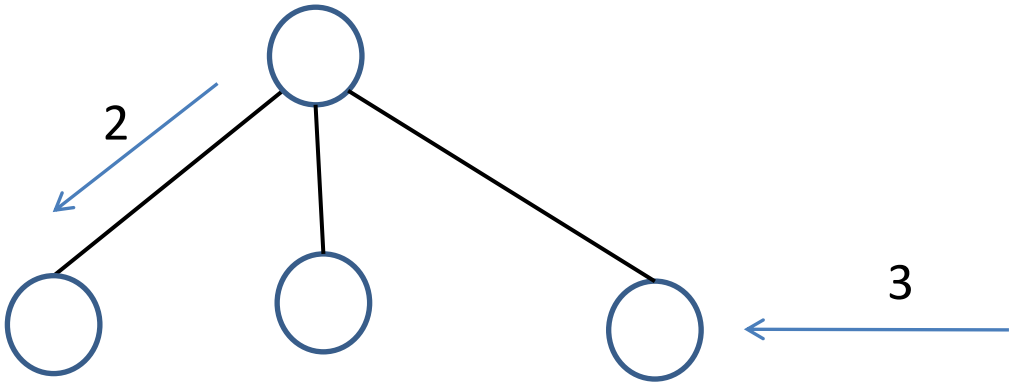
Artificial Intelligence Fundamentals

Games, Minimax and Alpha-Beta

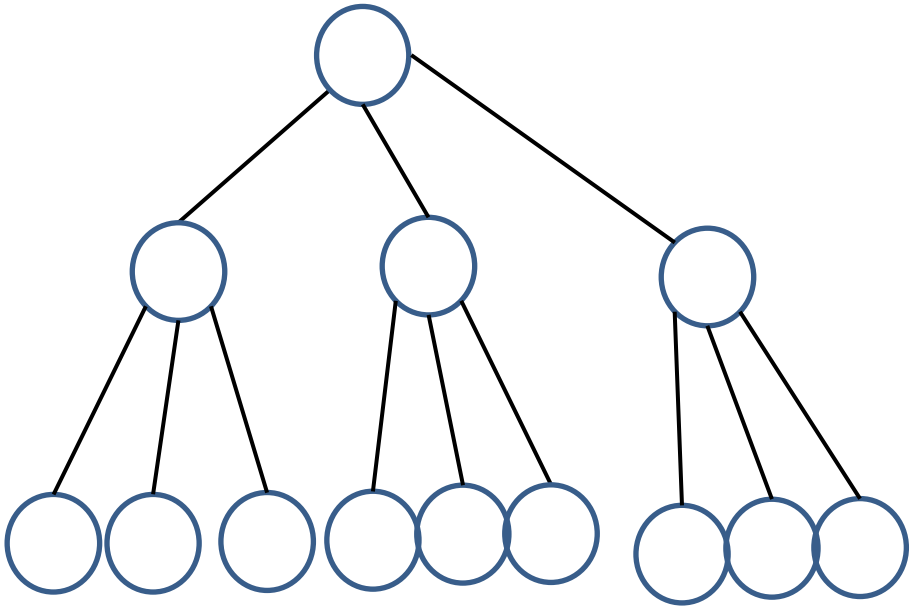
How we design a computer program to play a game ?

1. Analysis of the board + Strategy + Tactics -> Mixed up and somehow results a move
2. IF THEN Rules – If you can make a move then do it
3. Look ahead and evaluate – static function – linear scoring polynomial

$$S = g(f_1, f_2, \ldots, f_n), \text{ where } f_i \text{ are features}$$
$$= c_1 f_1 + c_2 f_2 + \ldots + c_n f_n$$


2
3

Game tree



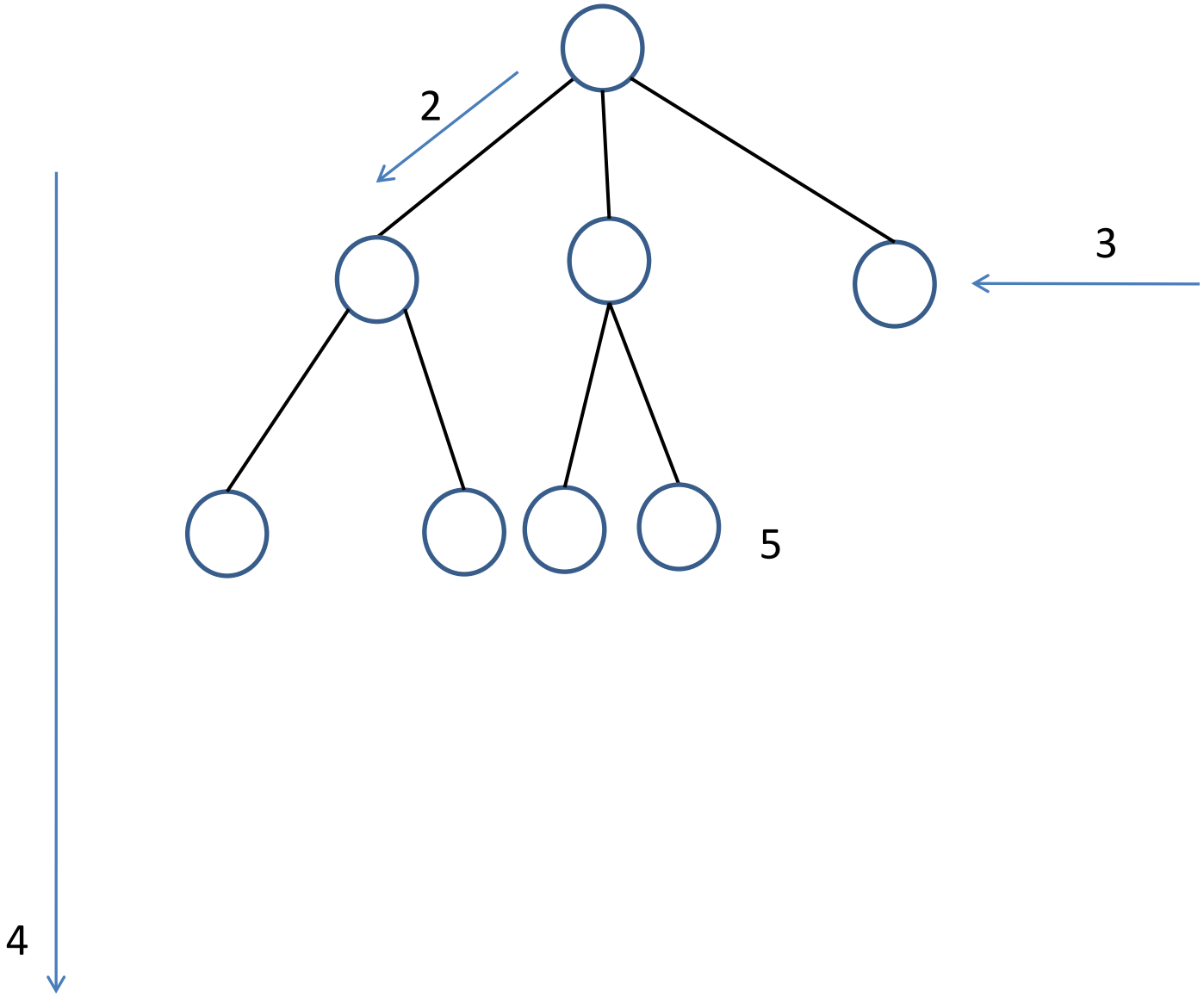
- Nodes – board configurations
- Branches – moves , transform one position into another
- d – depth of the tree (2)
- b – branching factor (3)
- b^d – terminal leaves (9)

4. Exhaustive search – Chess example

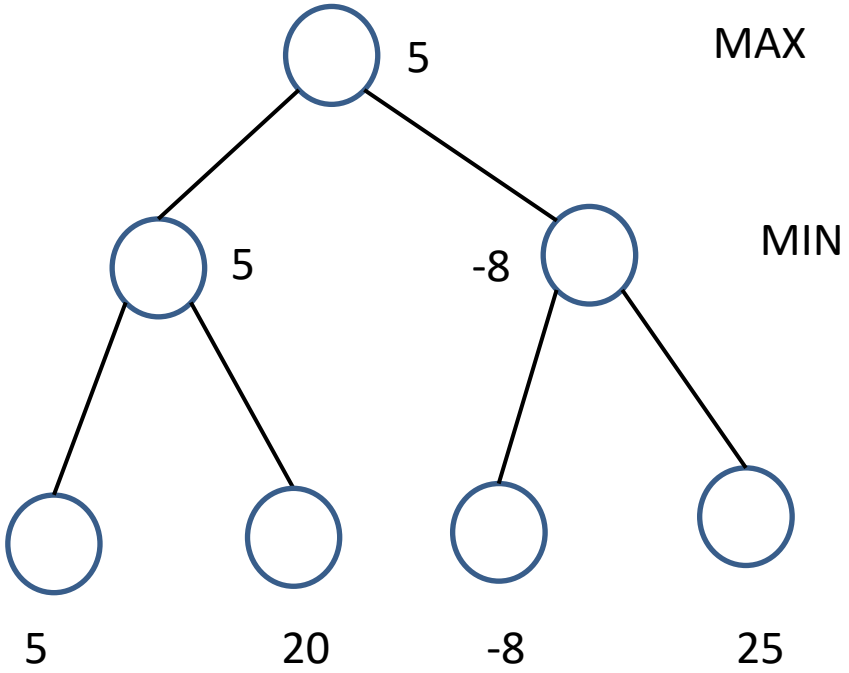
- Branching factor – 16
- Depth – 100
- Chess possibilities - 10^{120}
  - Universe contains - 10^{80} atoms
  - 1 year – 3* 10^{7} seconds
  - 1 sec - 10^{9} nano sec
  - Time from the beginning of the universe - 10^{10} years
  - Total – 10^{106}
- Analysis would be just getting started

5. Look ahead as far as possible

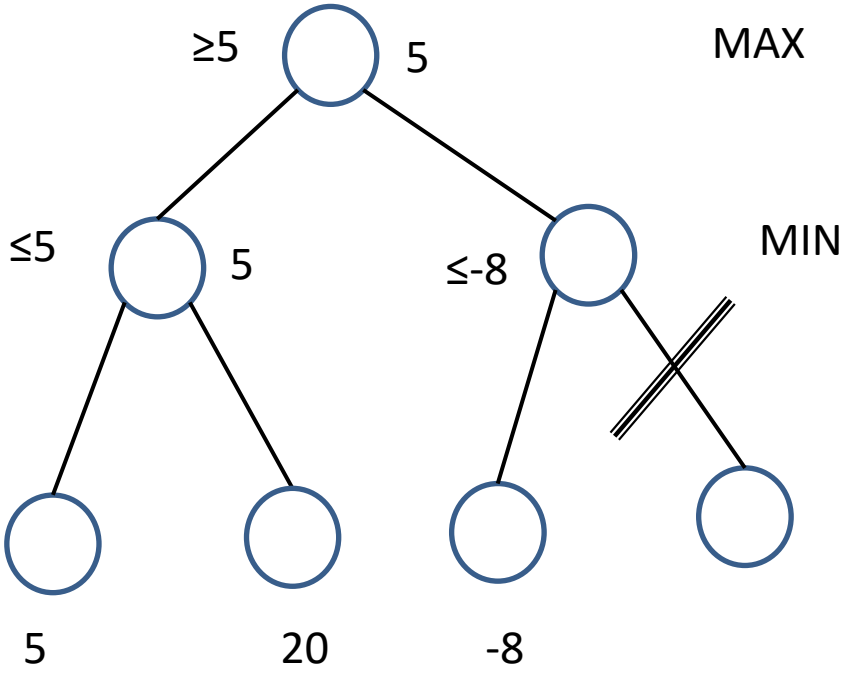
- Shannon & Turing
- Static evaluation – compute a number that reflects the board quality
  - Positive values favors one player (MAX player)
  - Negative values favors the other (MIN player)

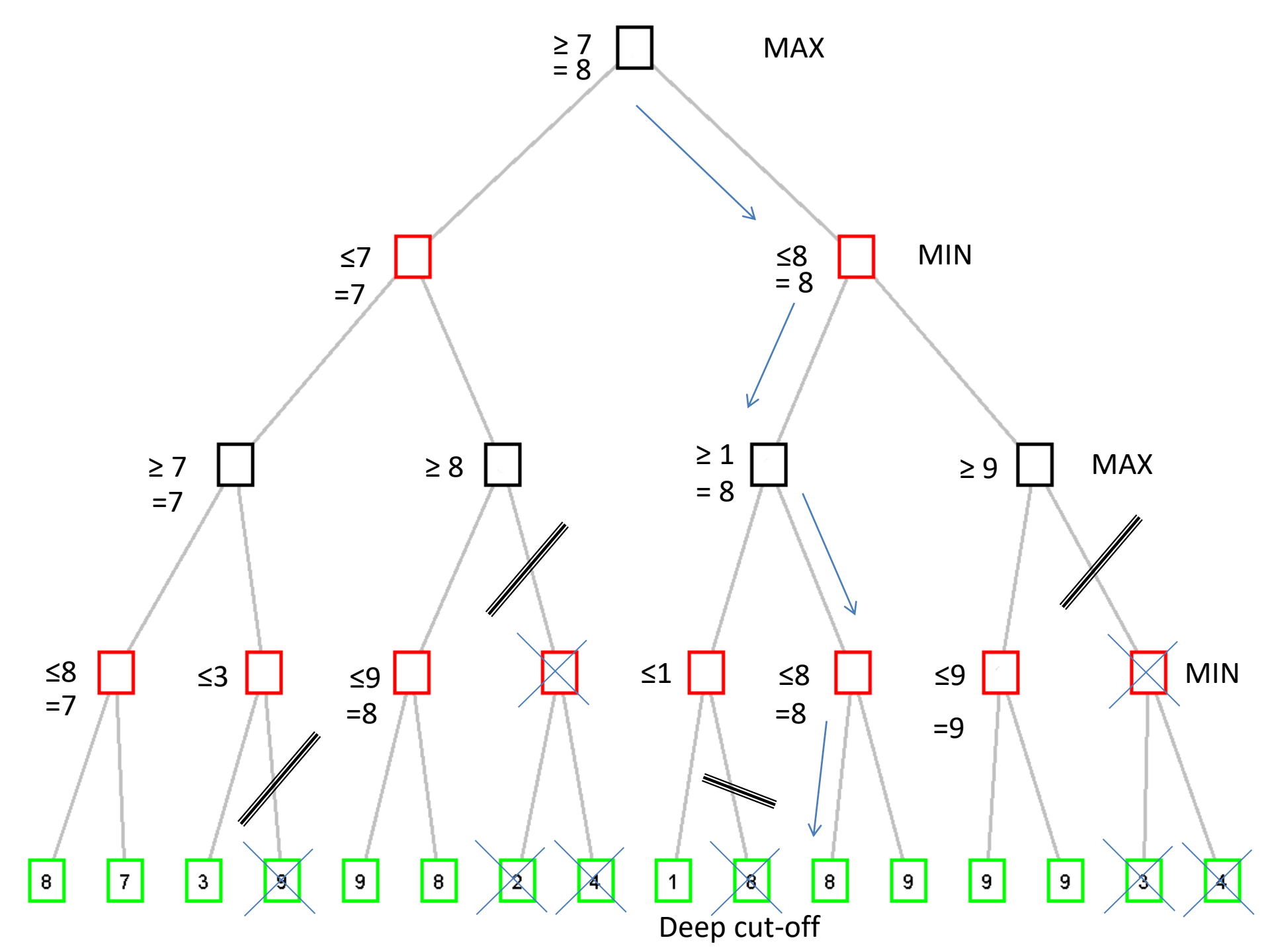

2
3
4
5

Minimax



Alpha Beta




≥ 7
= 8
MAX
≤7
=7
≤8
= 8
MIN
≥ 7
=7
≥ 8
≥ 1
= 8
≥ 9
MAX
≤8
=7
≤3
≤9
=8
≤1
≤8
=8
≤9
=9
MIN
8
7
3
9
9
8
2
4
1
8
8
9
9
9
3
4
Deep cut-off

ALPHA-BETA algorithm

- If the level is the top, let alpha be -∞ and let beta be ∞
- If the limit of search has been reached, compute the static value of the current position relative to the appropriate player. Report the result.
- If the level is a minimizing level
  - Until all children are examined with ALPHA-BETA or until alpha is equal to or grater than beta
    - Use ALPHA-BETA procedure, with the current alpha and beta values, on a child; note the value reported
    - Compare the value reported with the beta value; if the reported value is smaller, reset beta to the new value
  - Report beta
- Otherwise, the level is a maximizing level:
  - Until all children are examined with ALPHA-BETA or until alpha is equal to or grater than beta
    - Use ALPHA-BETA procedure, with the current alpha and beta values, on a child; note the value reported
    - Compare the value reported with the alpha value; if the reported value is larger, reset alpha to the new value
  - Report alpha

Best-case and worst-case

- With ALPHA-BETA and for optimal arrangement, the number of static evaluations has the following formulae:

$$s = 2 * b^{\frac{d}{2}} - 1$$

- Worst –case -> no cuttings

$$s = b^{d}$$

Progressive Deepening

- The branching factor is not always the same -> How deep can I go giving a certain amount of time?
- The number of nodes requiring static evaluation at the bottom of the tree is:

$$s = b^d$$

- The number of nodes in the rest of the tree is:

$$1 + b + b^2 + b^3 + \cdots + b^{d-1} = \frac{b^d - 1}{b - 1}$$

- The ratio of the number of nodes in the bottom level to the number of nodes up to the bottom level is:

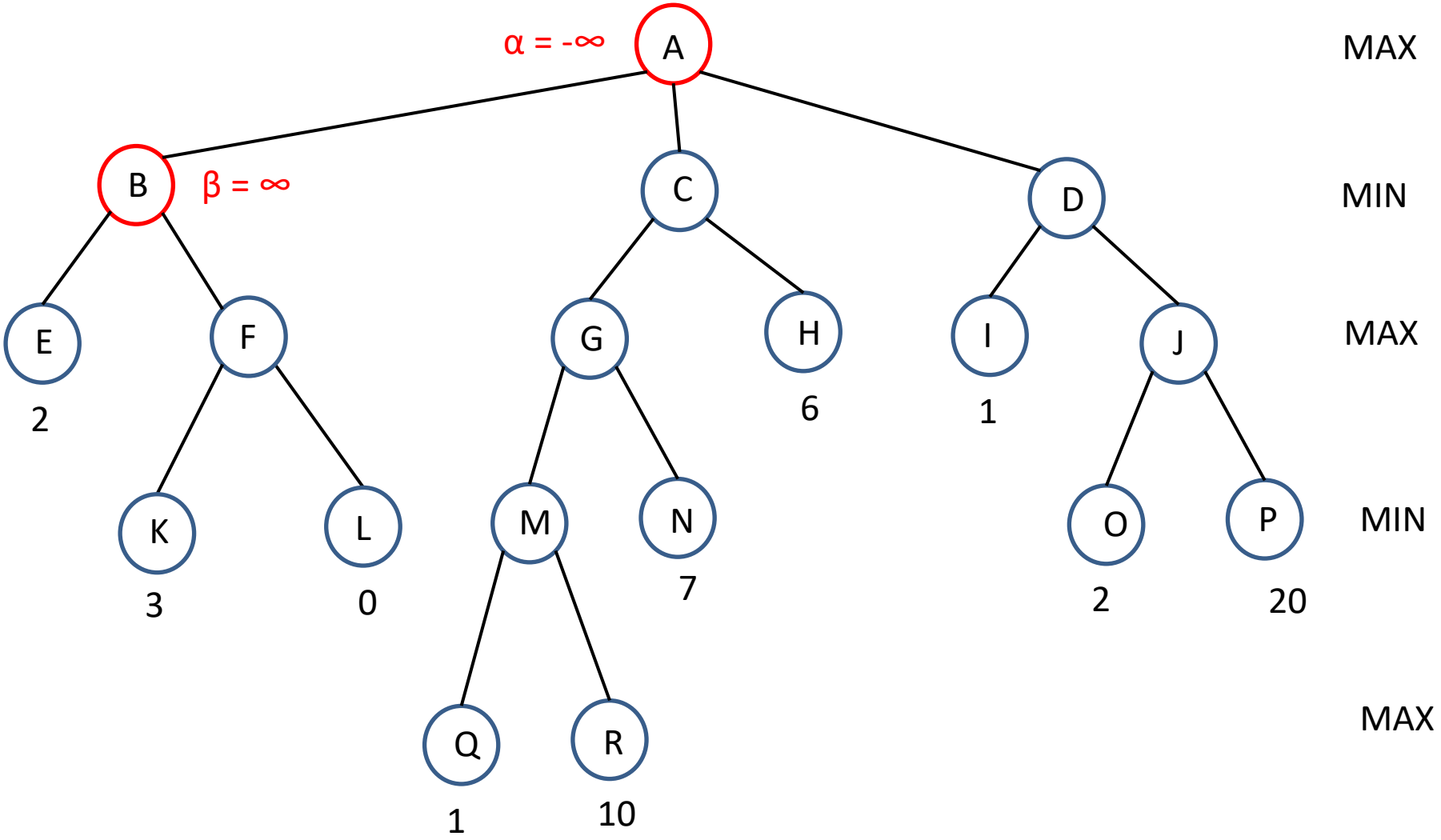
$$s = 1 + b + b^2 + b^3 + \cdots + b^{d-1} = \frac{b^d - 1}{b - 1} \qquad \text{ratio} = \frac{b^d}{\frac{b^d - 1}{b - 1}} \approx b - 1$$

- Improve the performance of ALPHA-BETA – reorder the nodes

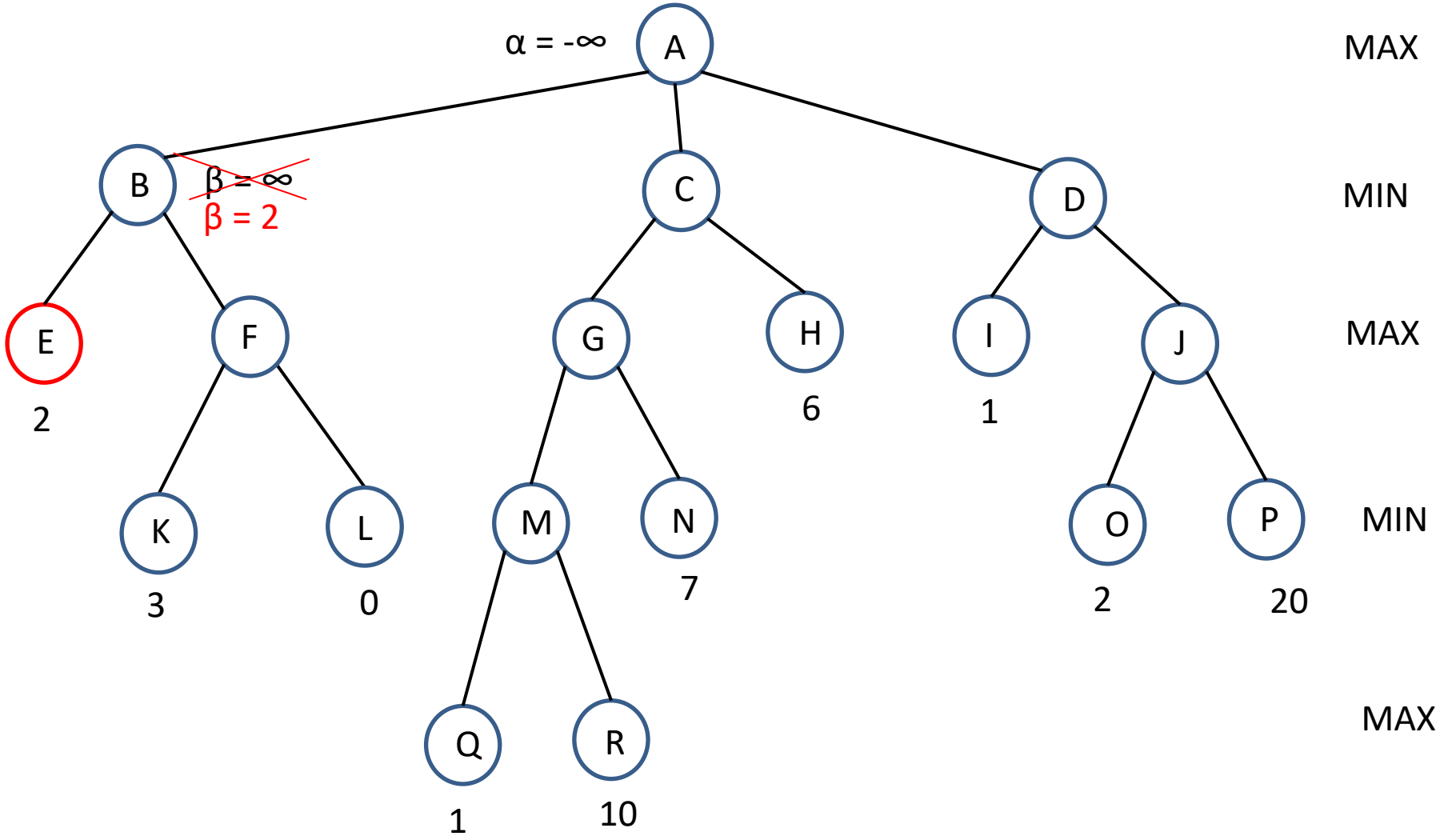
Deep-Blue

- Minimax – 14-15 levels
- Alpha Beta
- Progressive Deepening
- Parallel computing
- Opening book
- End game special situations
- Uneven tree development

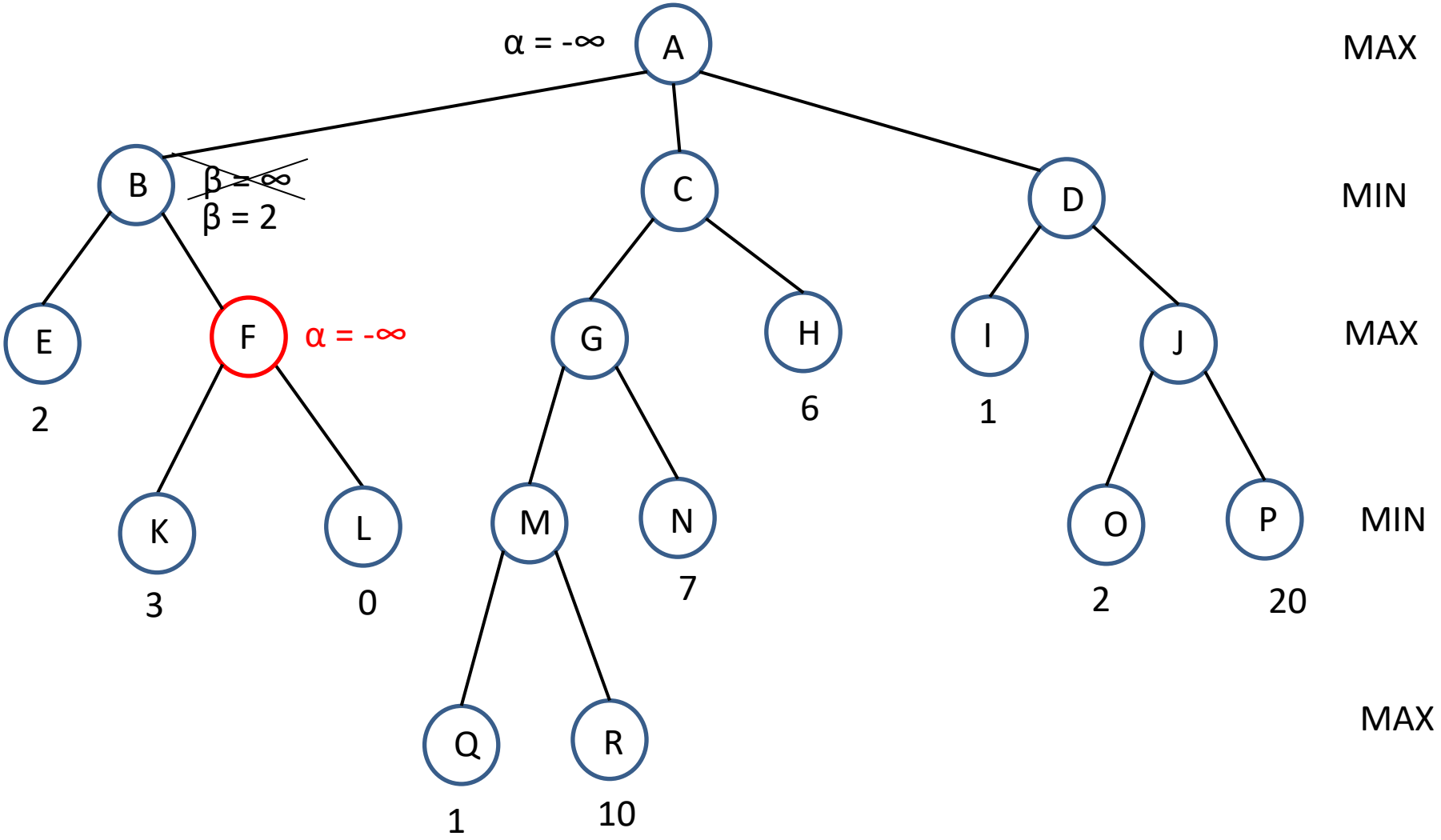
Alpha Beta example



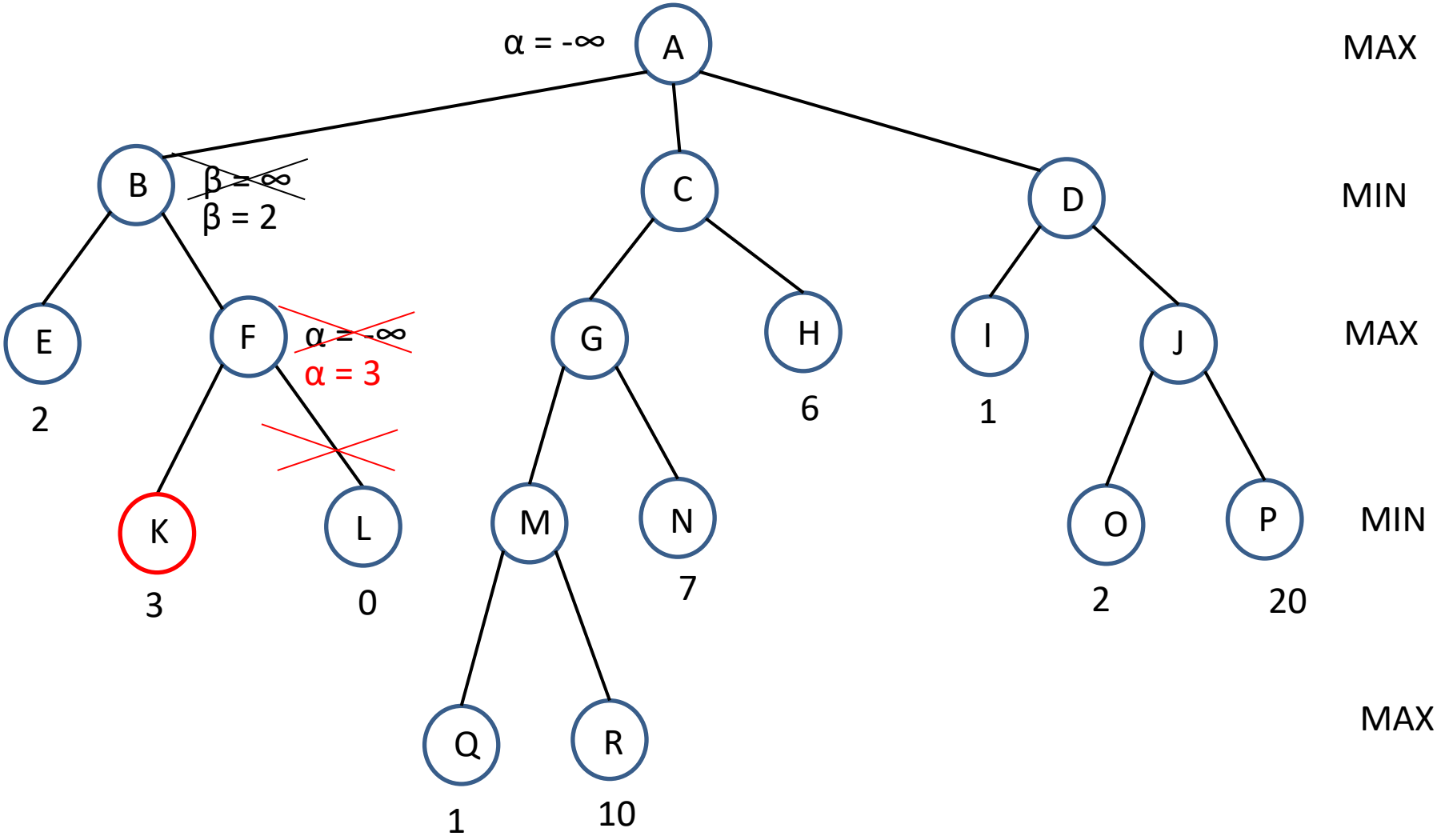
Alpha Beta example



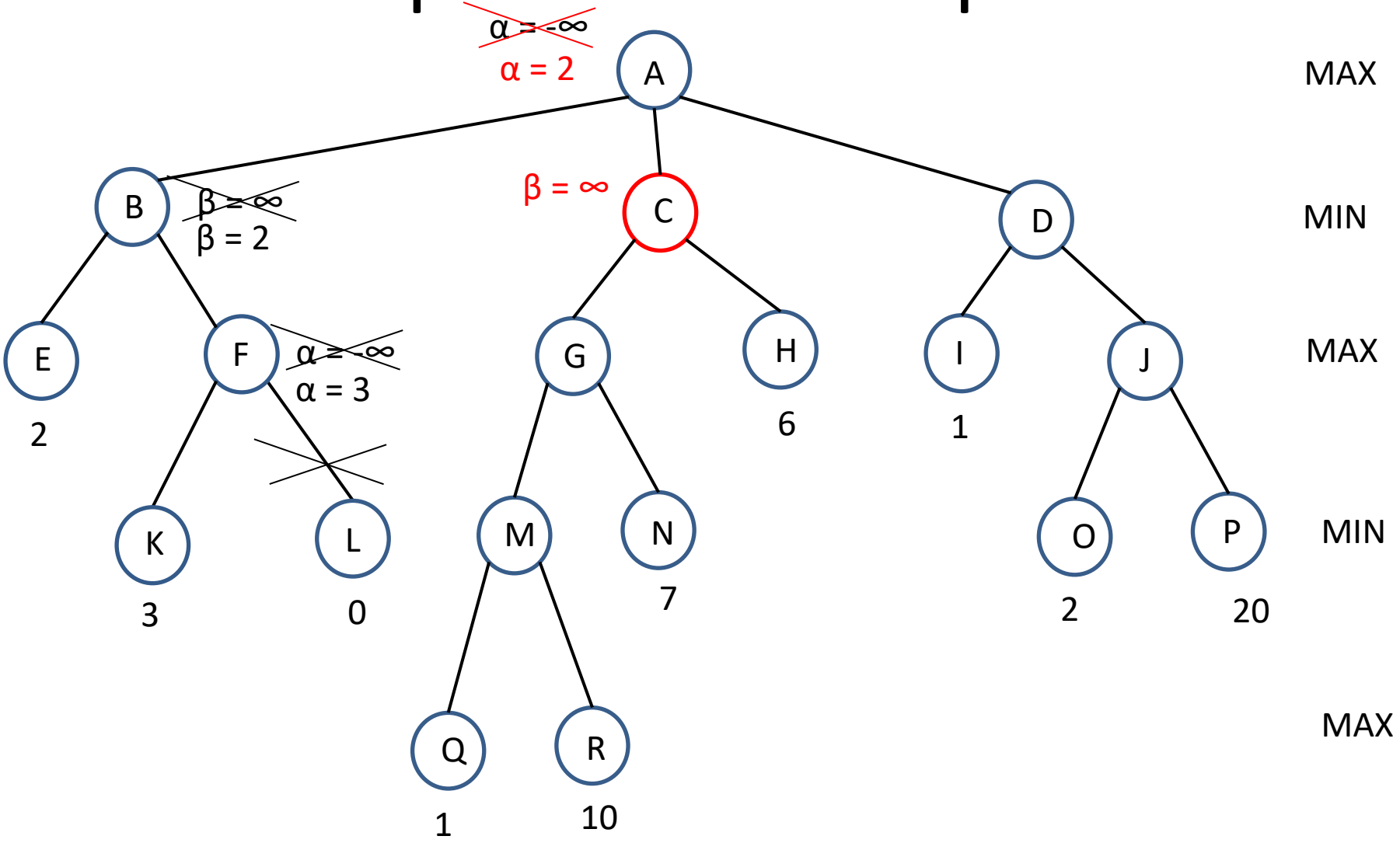
Alpha Beta example



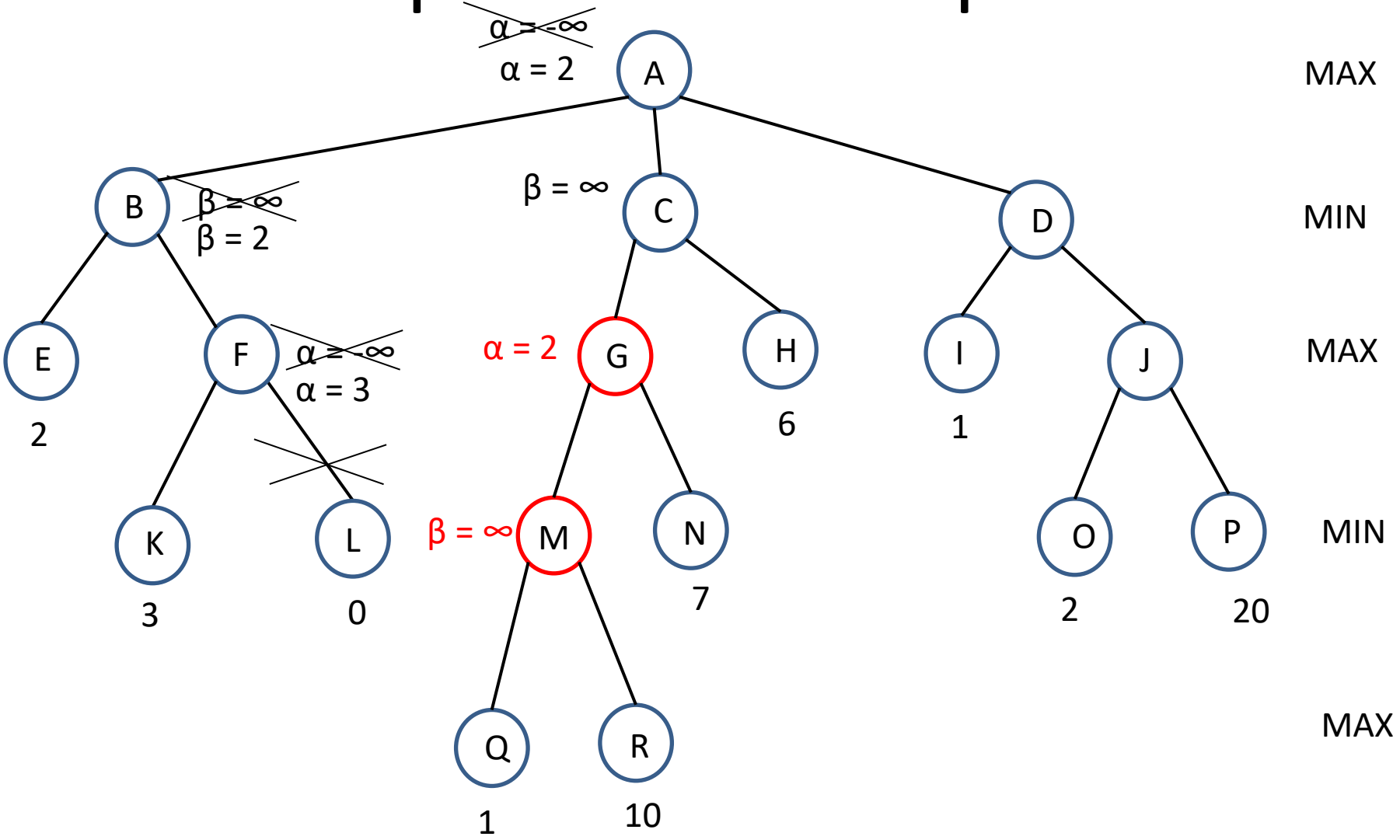
Alpha Beta example



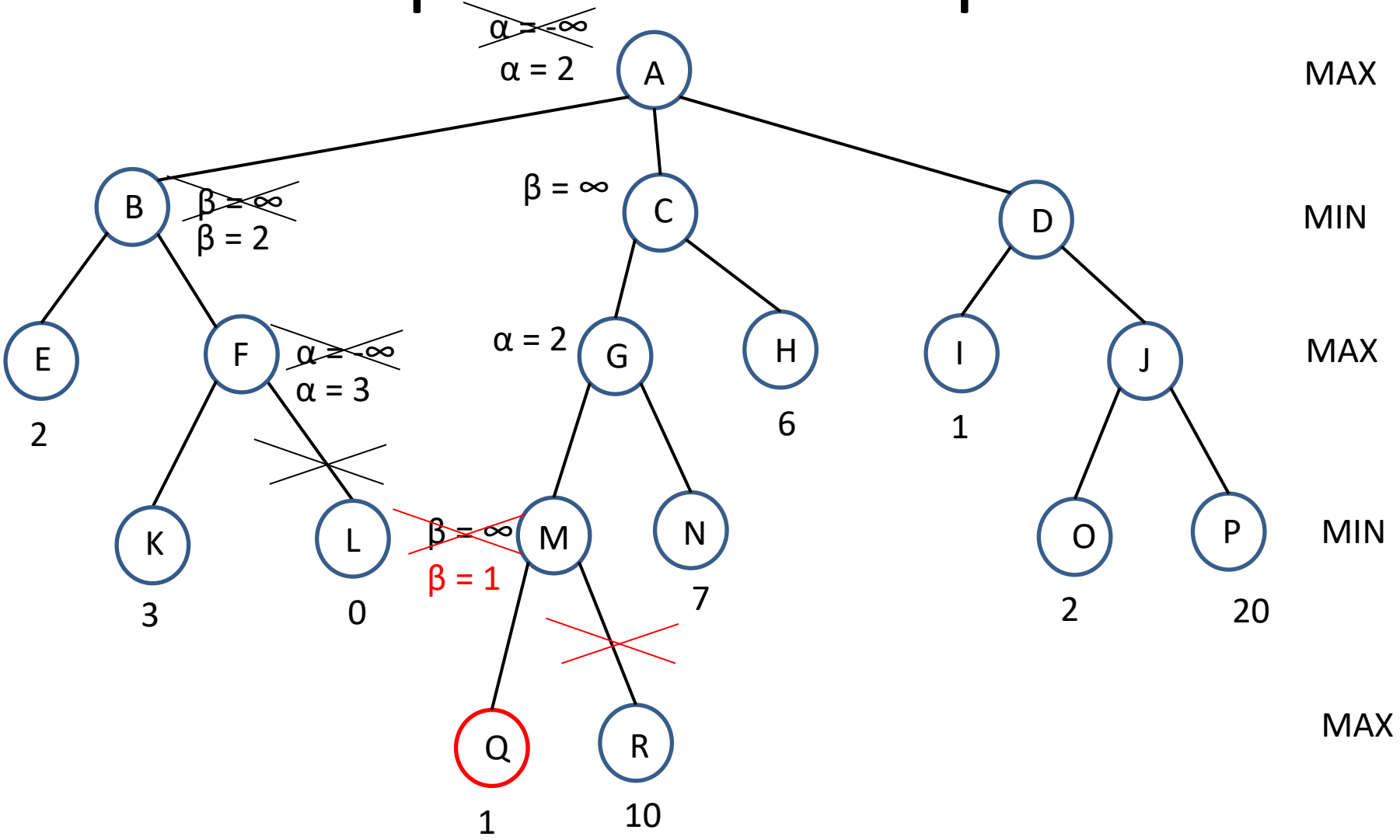
Alpha Beta example



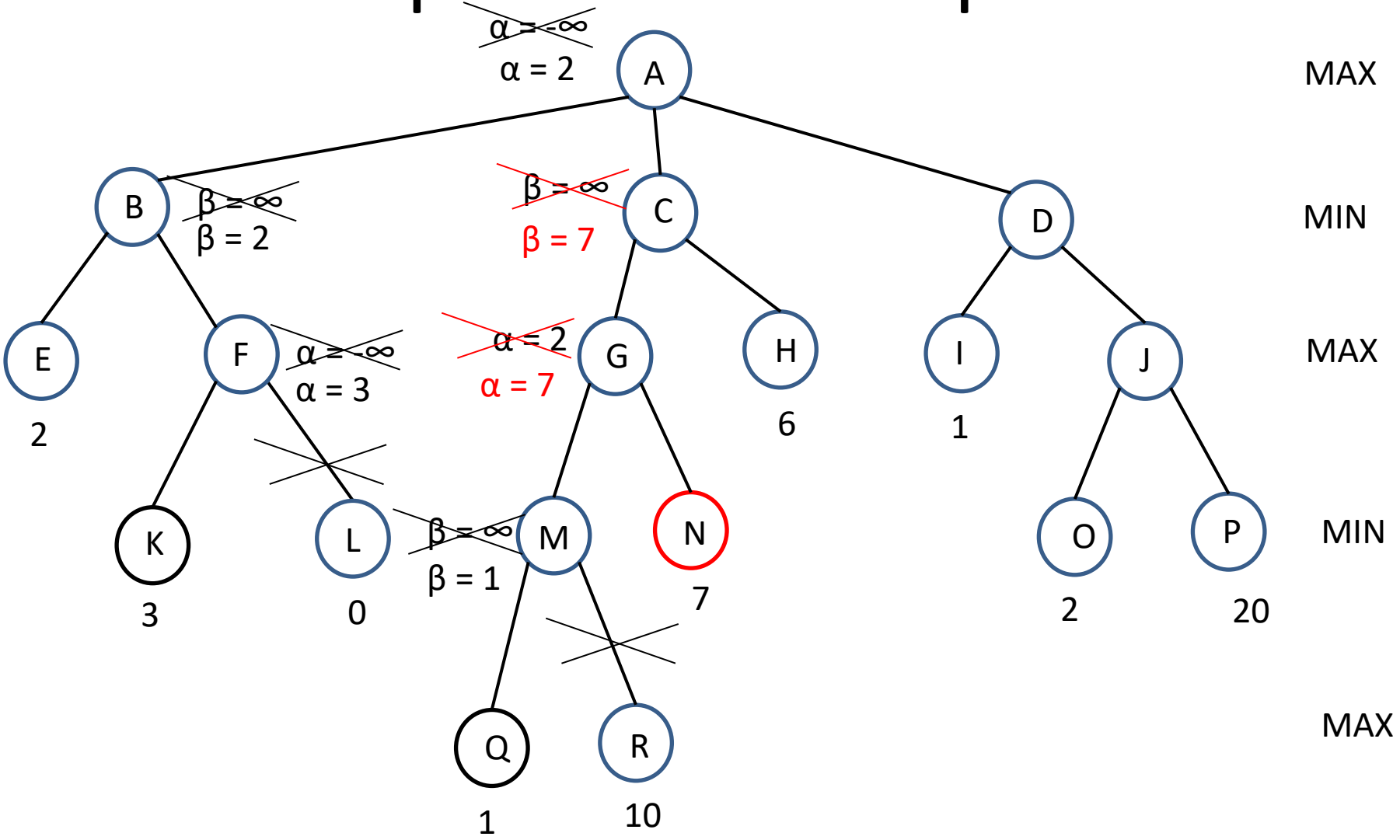
Alpha Beta example



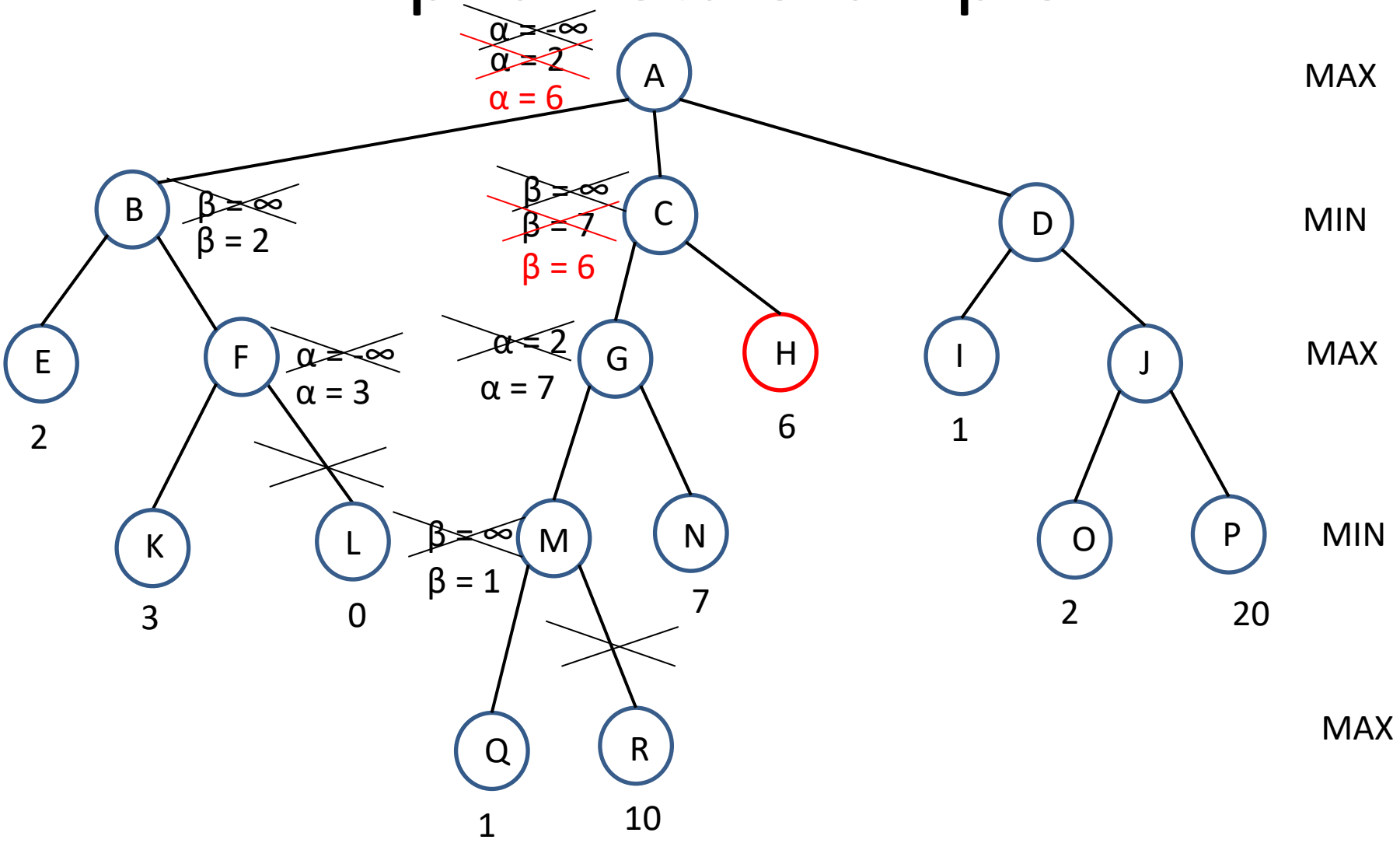
Alpha Beta example



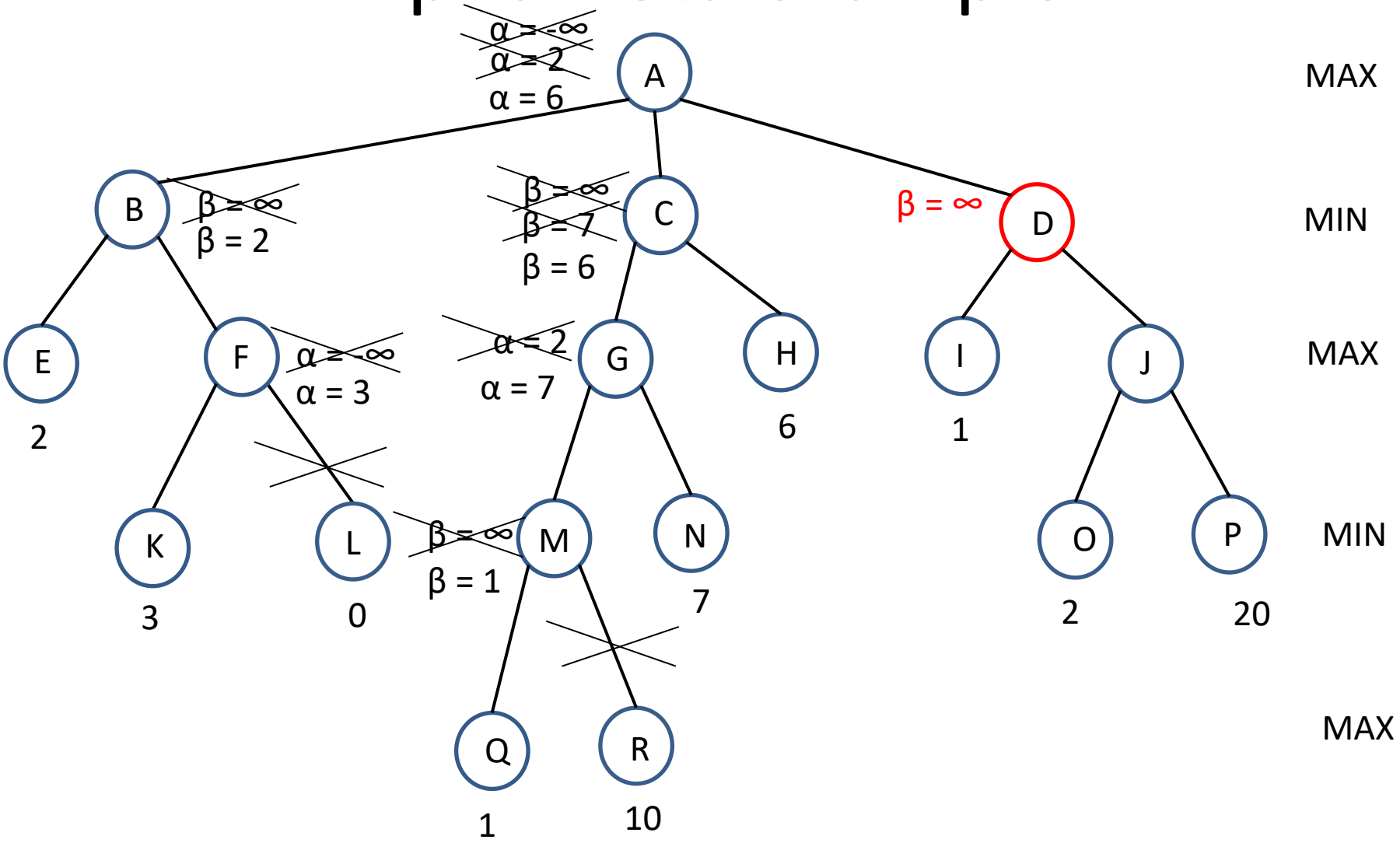
Alpha Beta example



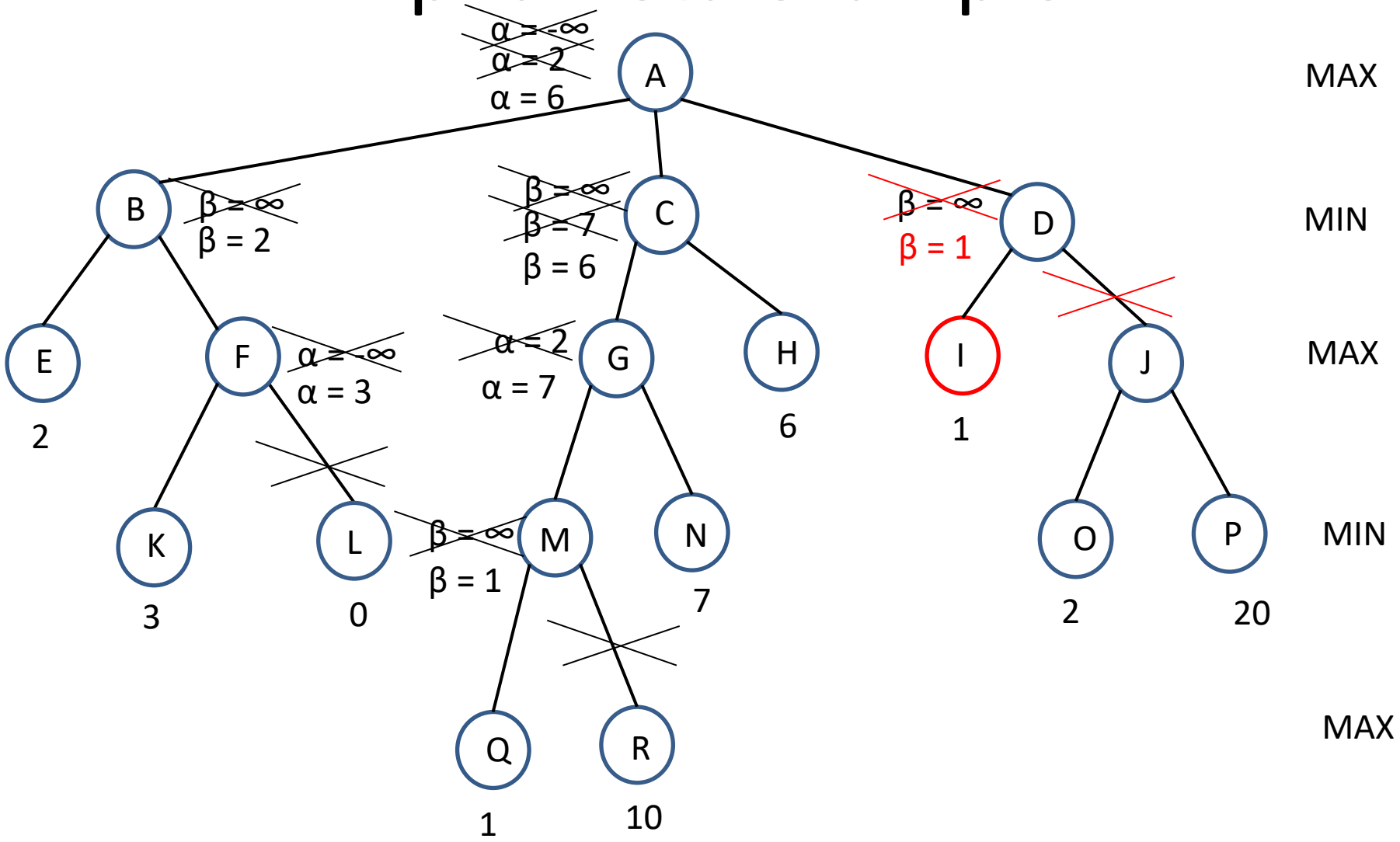
Alpha Beta example



Alpha Beta example



Alpha Beta example



List the leaf nodes in the order that they are statically evaluated: E, K, Q, N, H and I

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

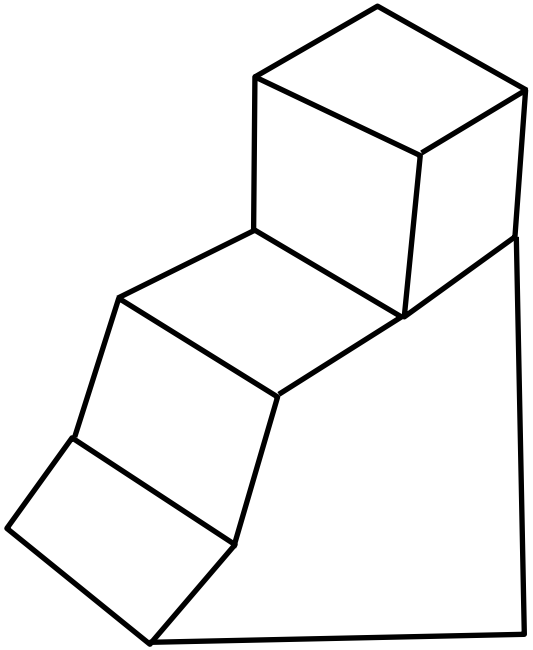
- Artificial Intelligence (3rd Edition), Patrick Winston, Chapter 6

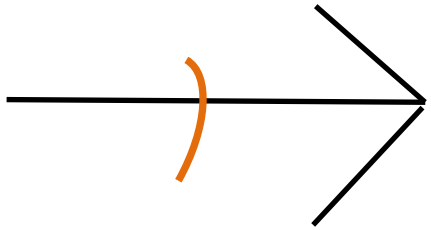
Artificial Intelligence Fundamentals

Constraints: Search, Domain Reduction

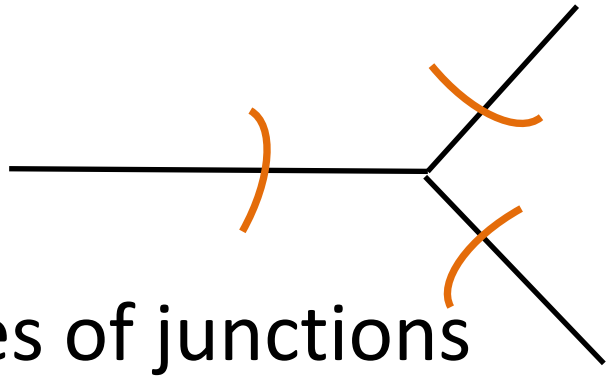
Constraints: Interpreting Line Drawings

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



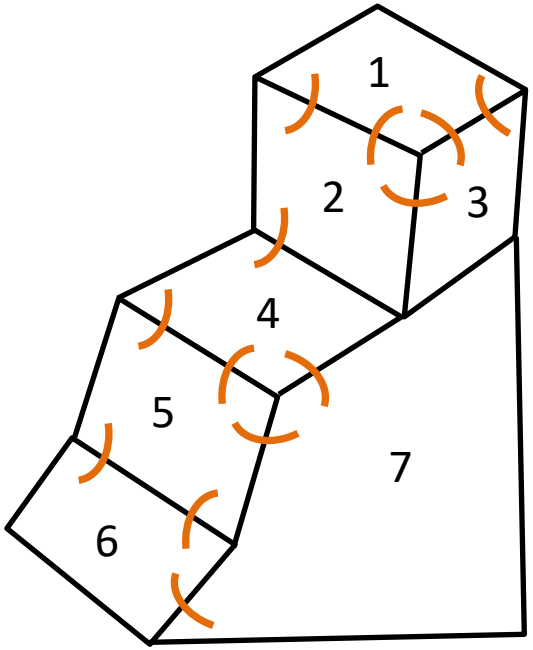


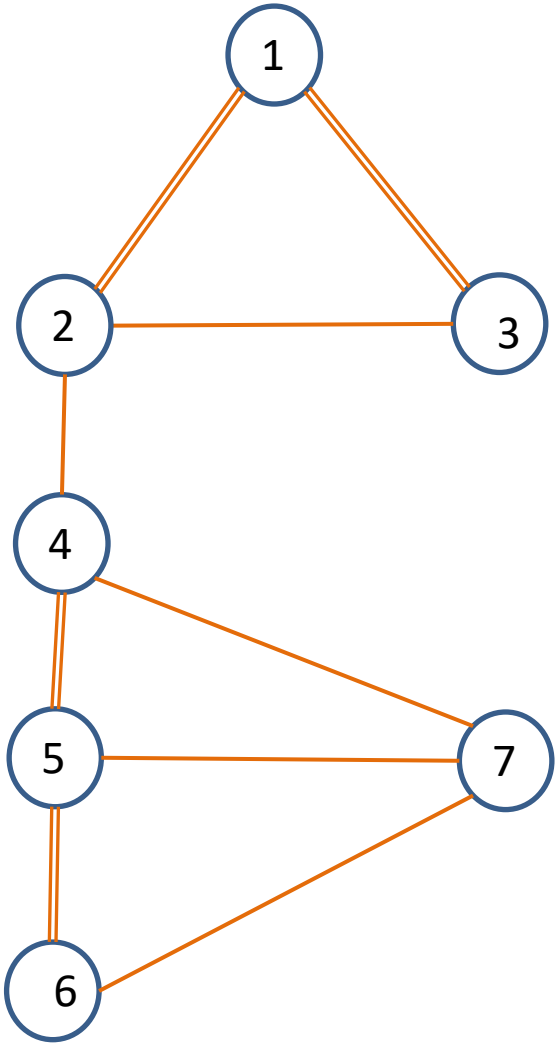
These 2 faces belongs to the same object



- Types of junctions

Constraints: Interpreting Line Drawings

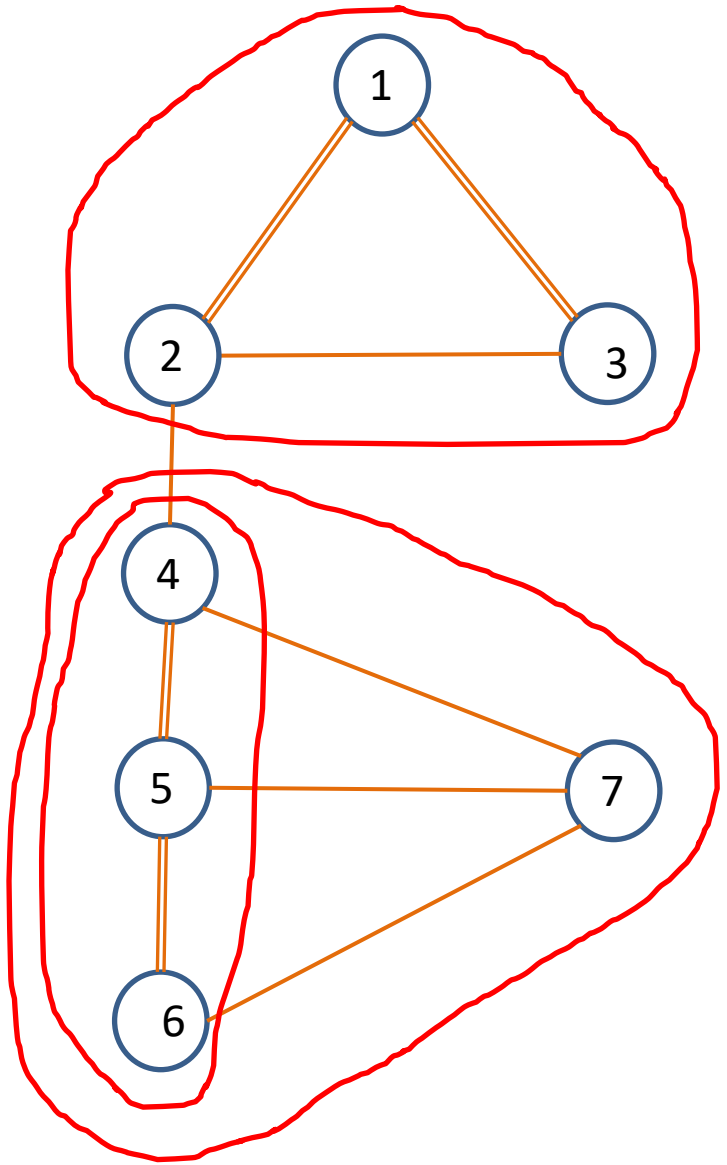




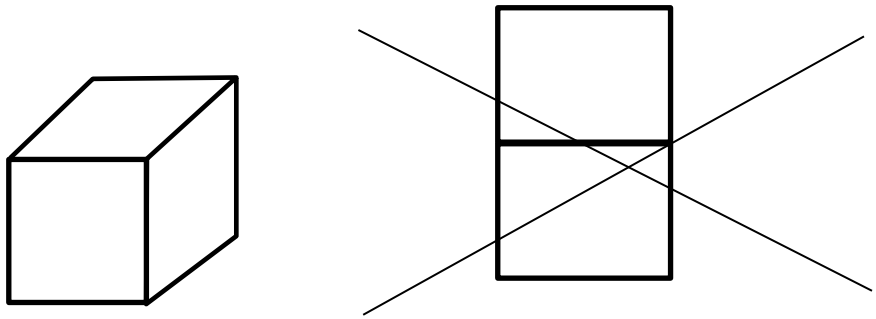
- 1 link theory -> too liberal
- 2 link theory –> too conservative
- 2 link * (repeated) – link super regions that are connected more than 2 links

- In the final there are 2 objects

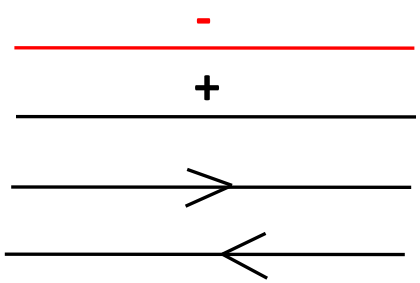
- It works because there are many 3 faced vertexes (junctions)

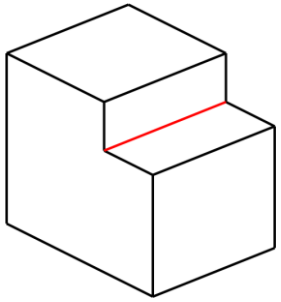


- Huffman – mathematician
  1. General position



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



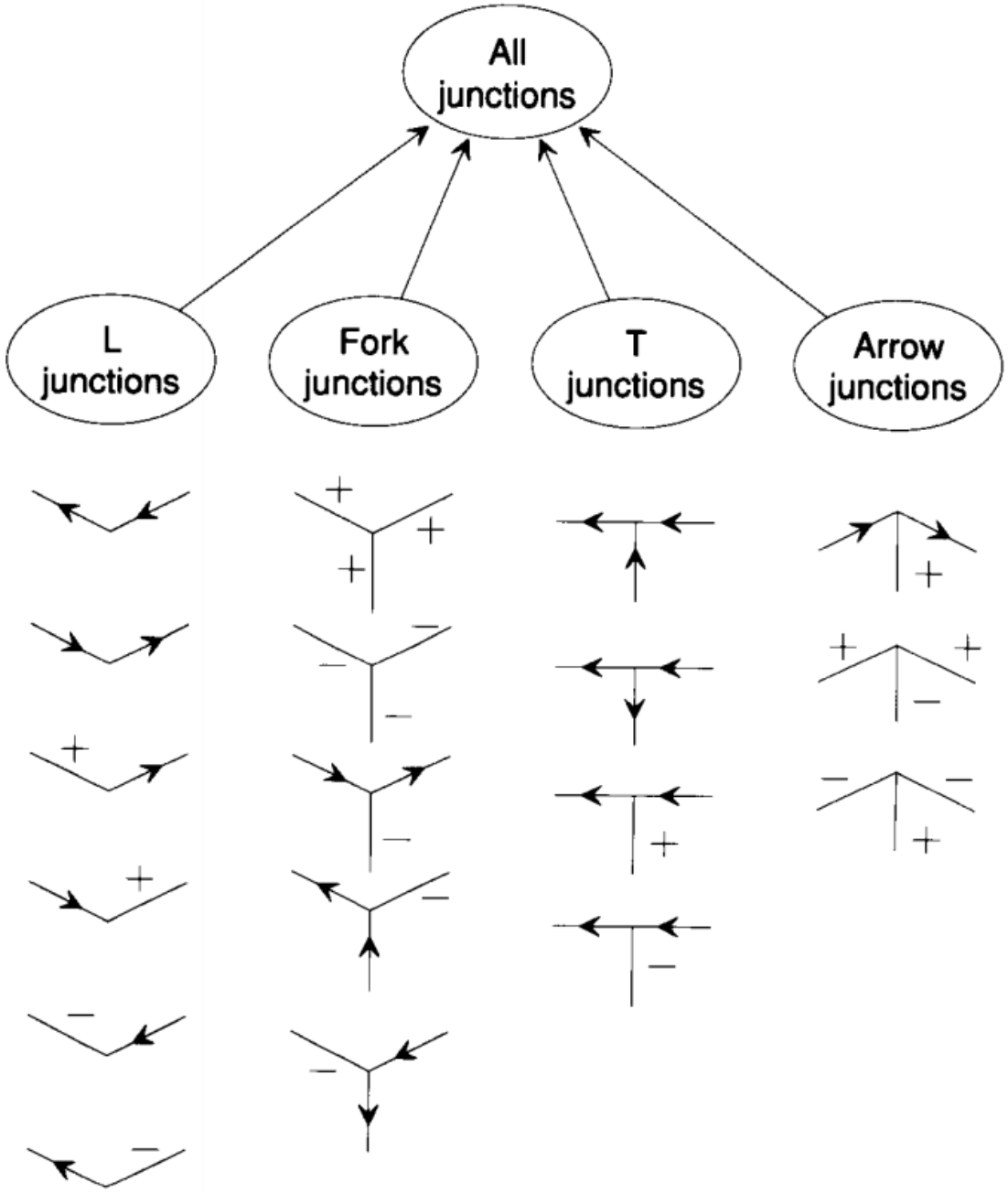


When you walk on the direction of the line you have the object in the right

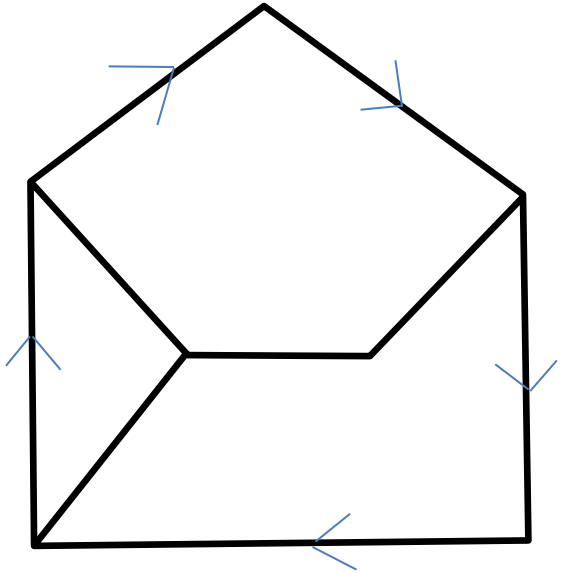
4. Without shadows and cracks
5. There are only 18 ways to arrange the labels around the junction

Vertexes Junctions

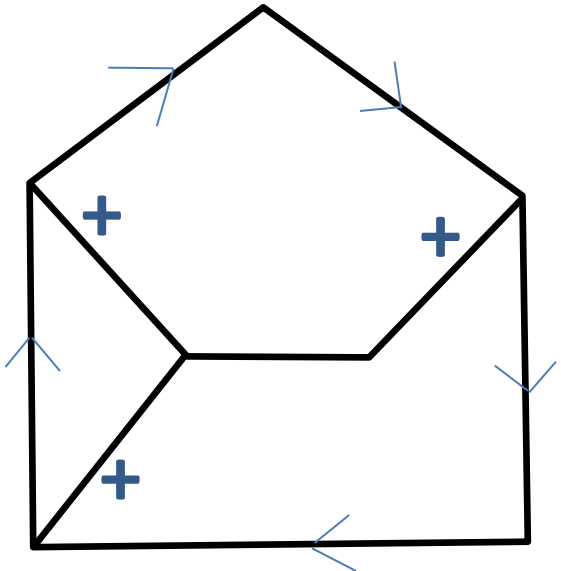
Edges Lines



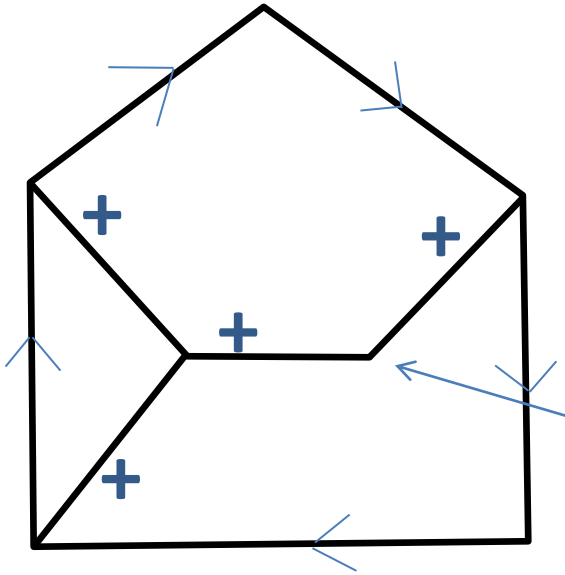
Can you build one of those object?



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

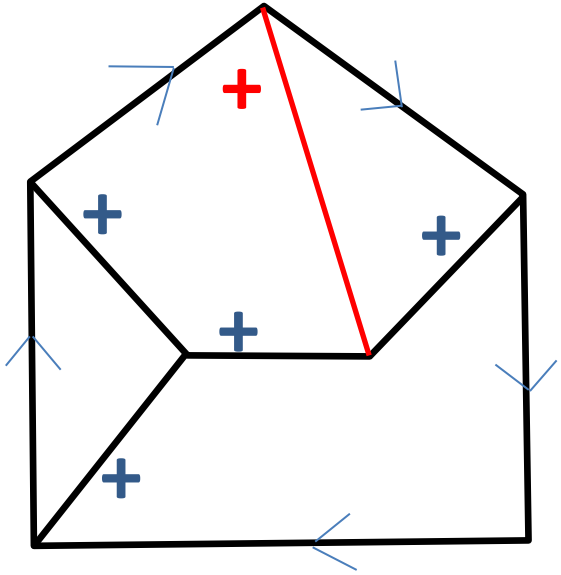


A line can't change his nature along his length -> so if it’s a + line at an end must be a + line at the other end



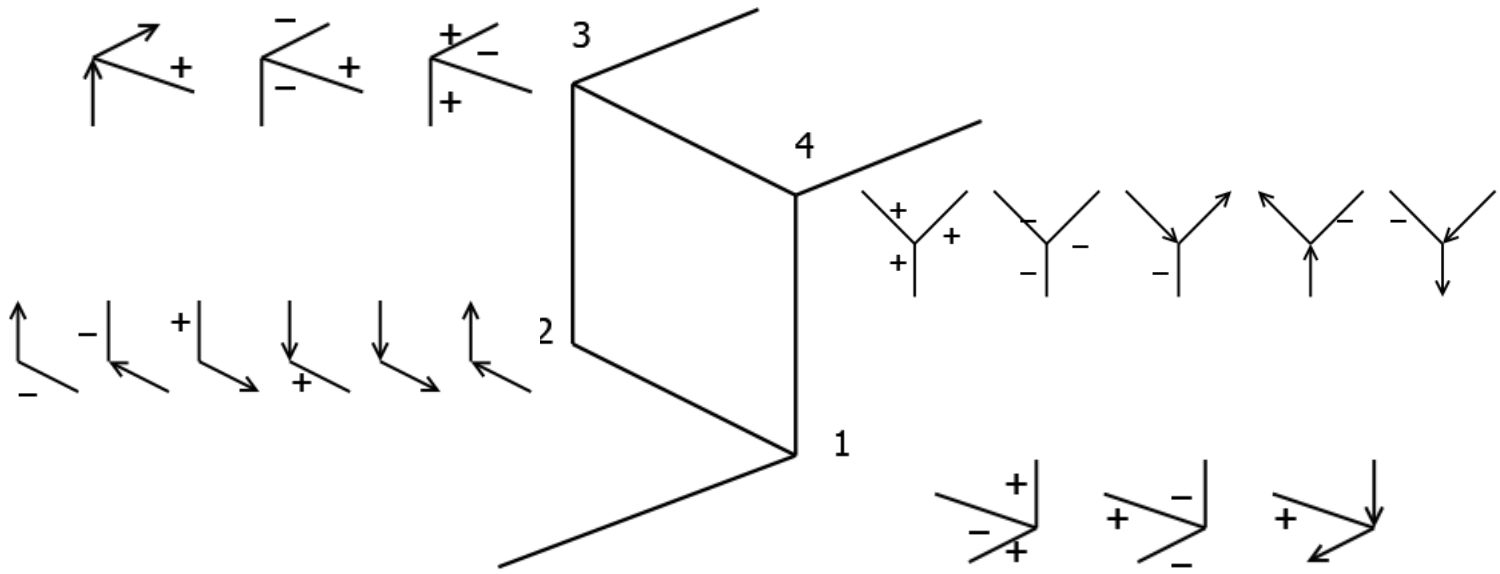
What about that junction?

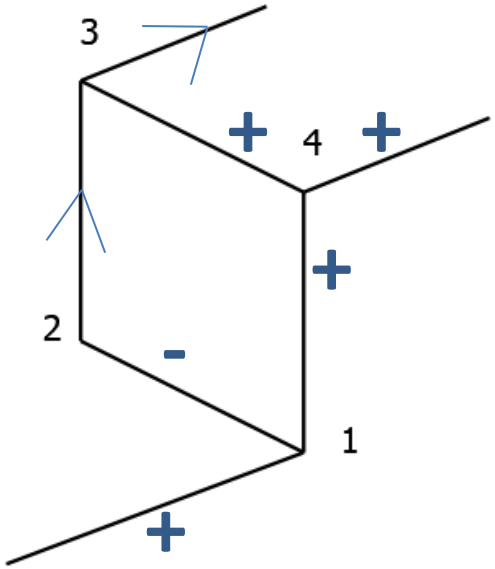
It’s an L junction with pluses at both of the lines, but there isn’t exist in the catalog -> this object can't exist in the real world



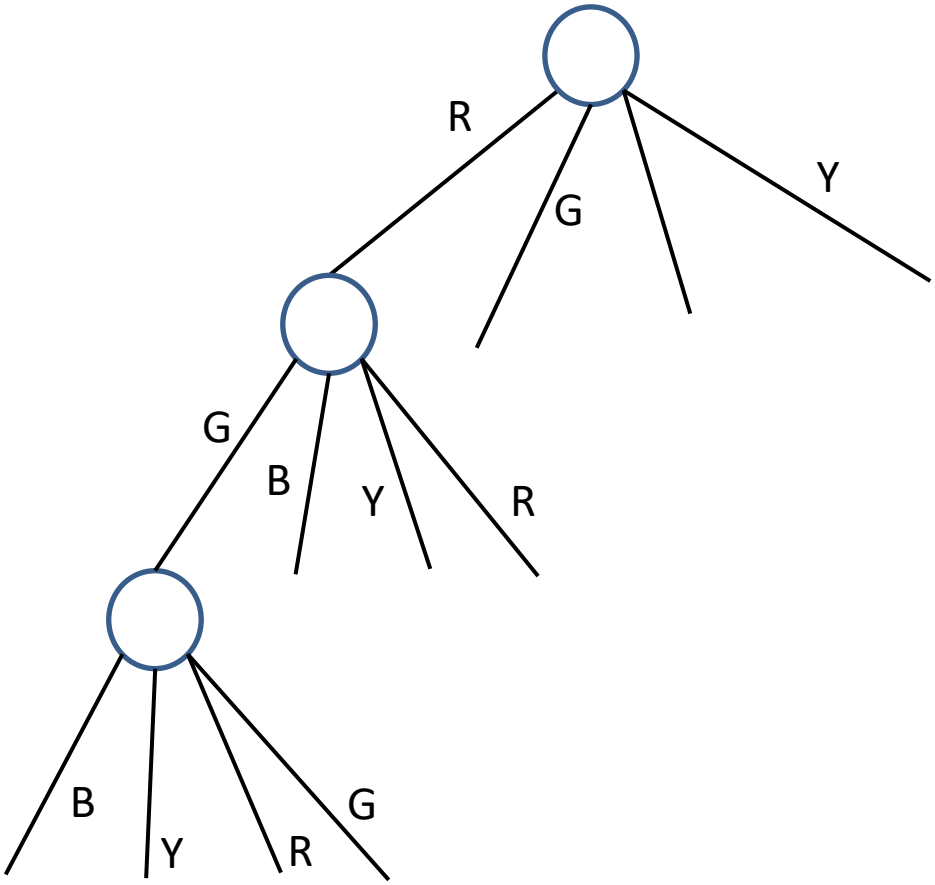
Helps to have a line like that?

Answer: No


3
4
2
1

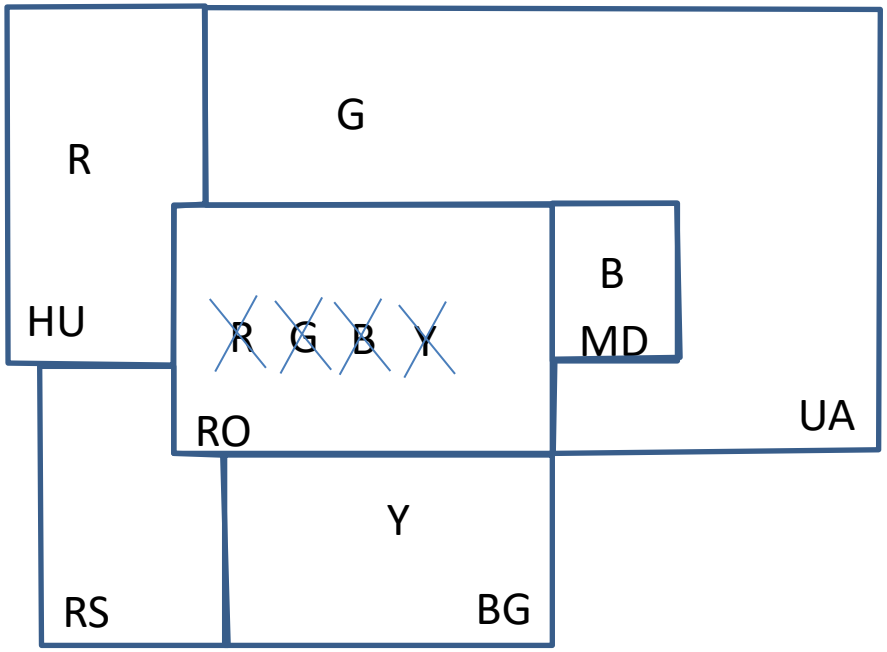

3
4
2
1

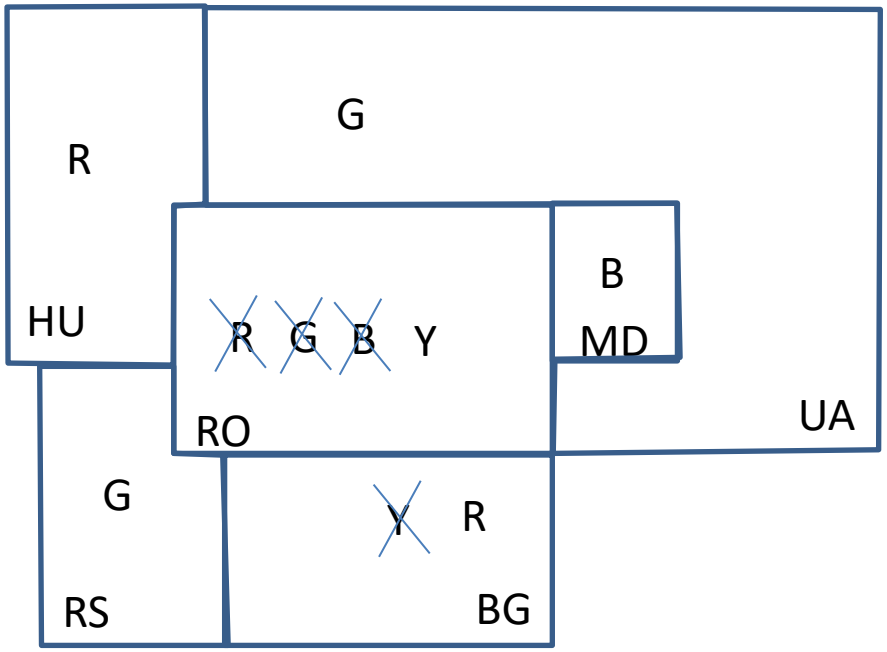
Map Coloring



- Depth first search
- 4 colors

Map coloring - Romania




R
G
HU
R G B Y
B
MD
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 ***considered***

for each v_i in D_i

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

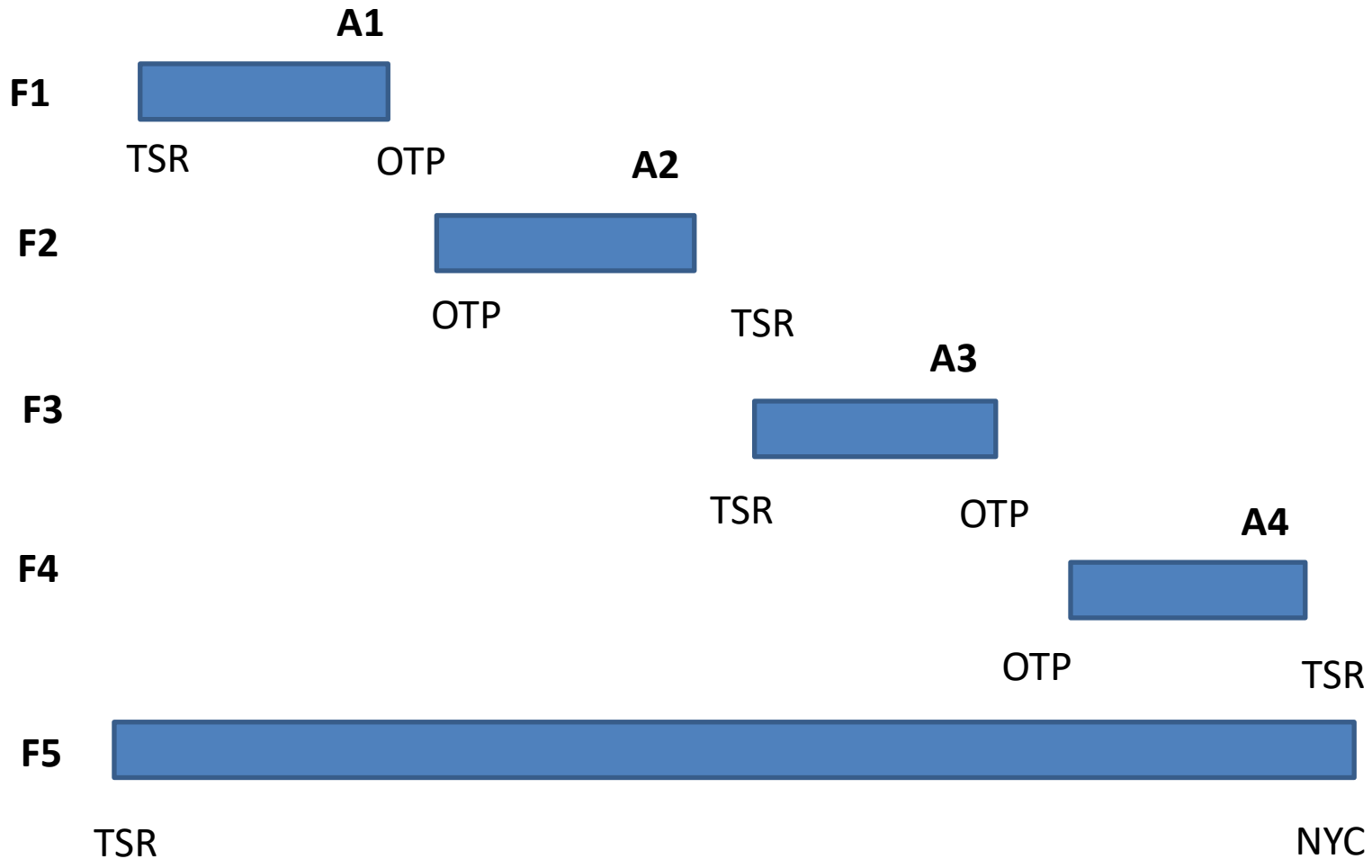
if not $\exists v_j \ni \ C(v_i, v_j)$ satisfied remove v_i from D_i

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

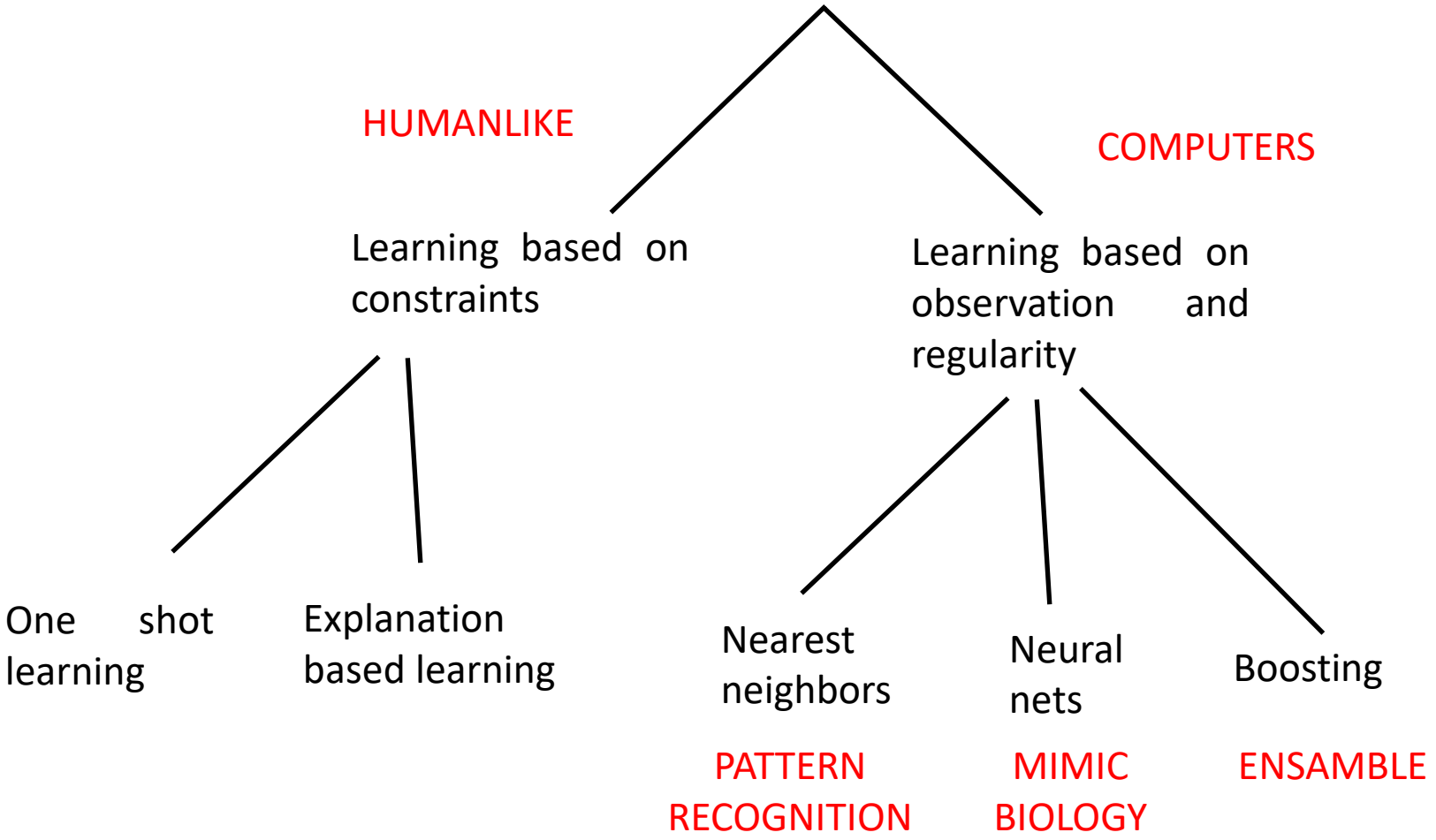
Readings

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

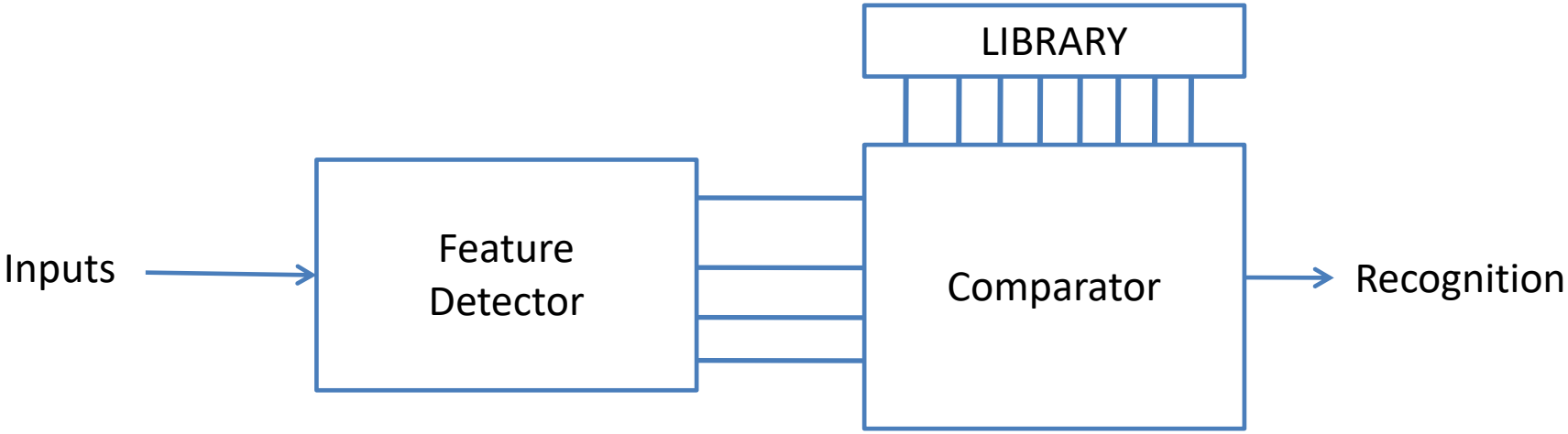
Artificial Intelligence Fundamentals

Learning: Nearest Neighbors

Learning

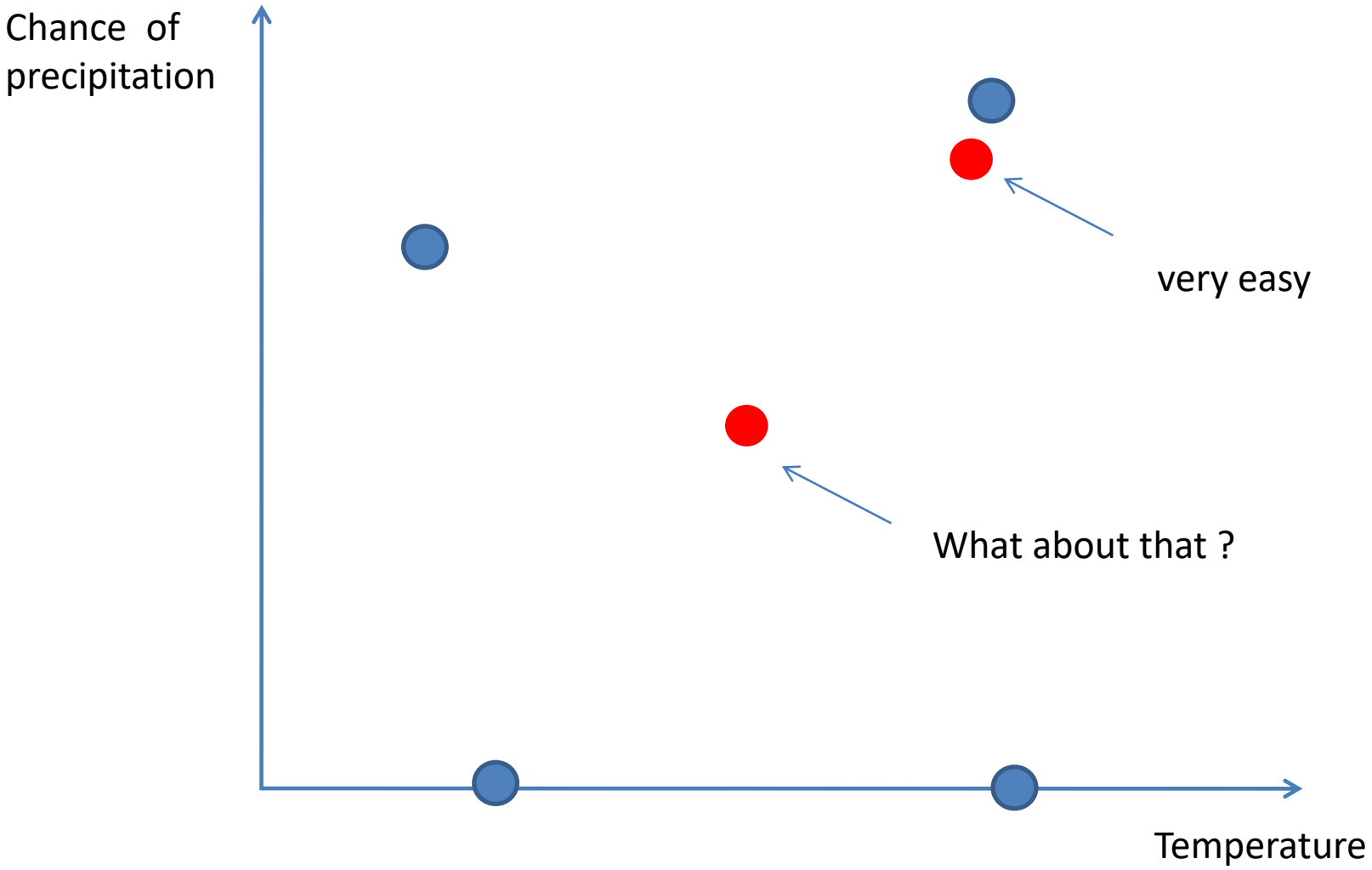

HUMANLIKE
COMPUTERS
Learning based on constraints
Learning based on observation and regularity
One shot learning
Explanation based learning
Nearest neighbors
Neural nets
Boosting
PATTERN RECOGNITION
MIMIC BIOLOGY
ENSAMBLE

Nearest neighbors

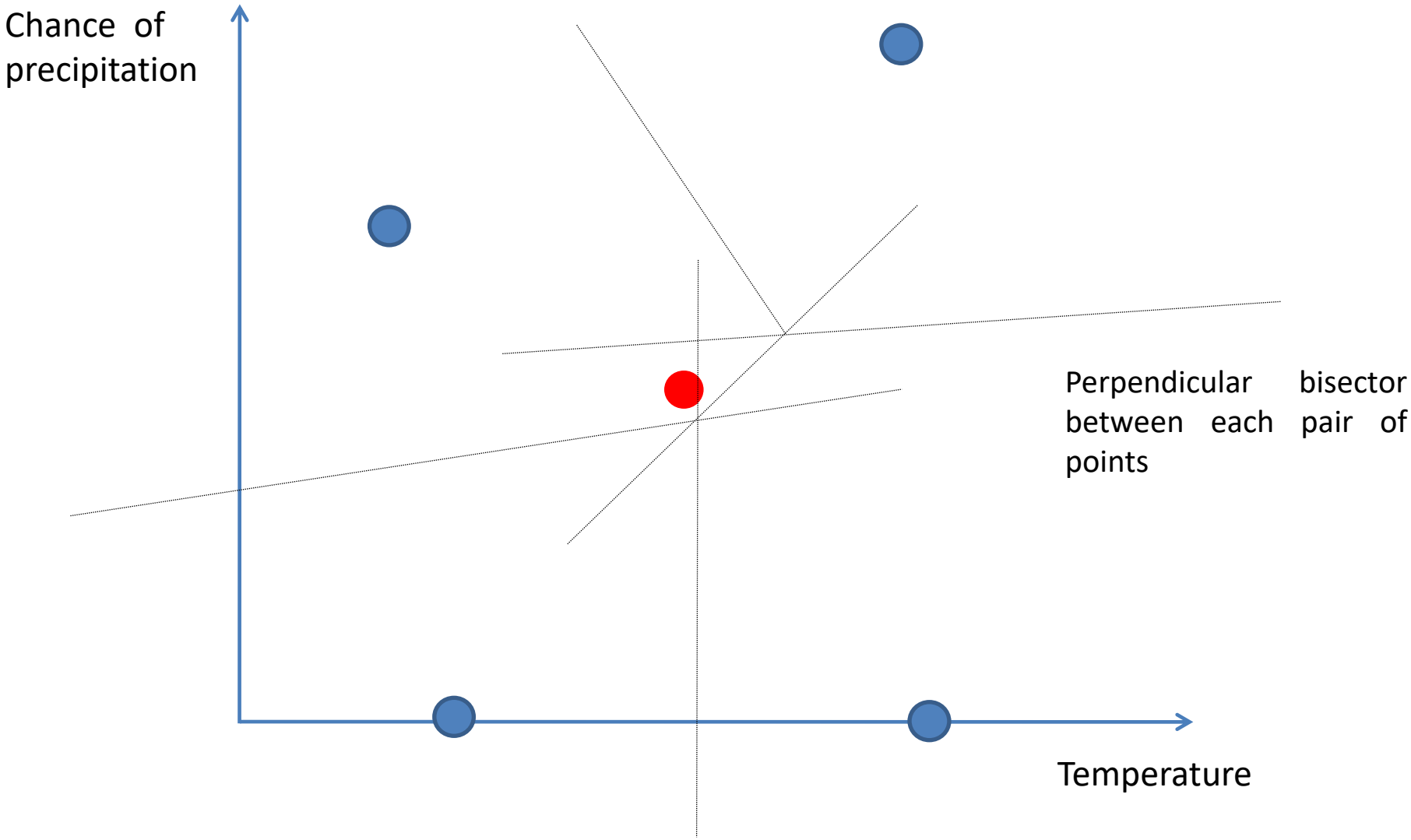


- Feature detector – extracts a vector of features from the inputs and for each feature compute a value
- LIBRARY – stores multiple feature vectors, one feature vector for one possibility
- Comparator – compare the feature vector with all the possibilities and return the nearest possibility -> recognition

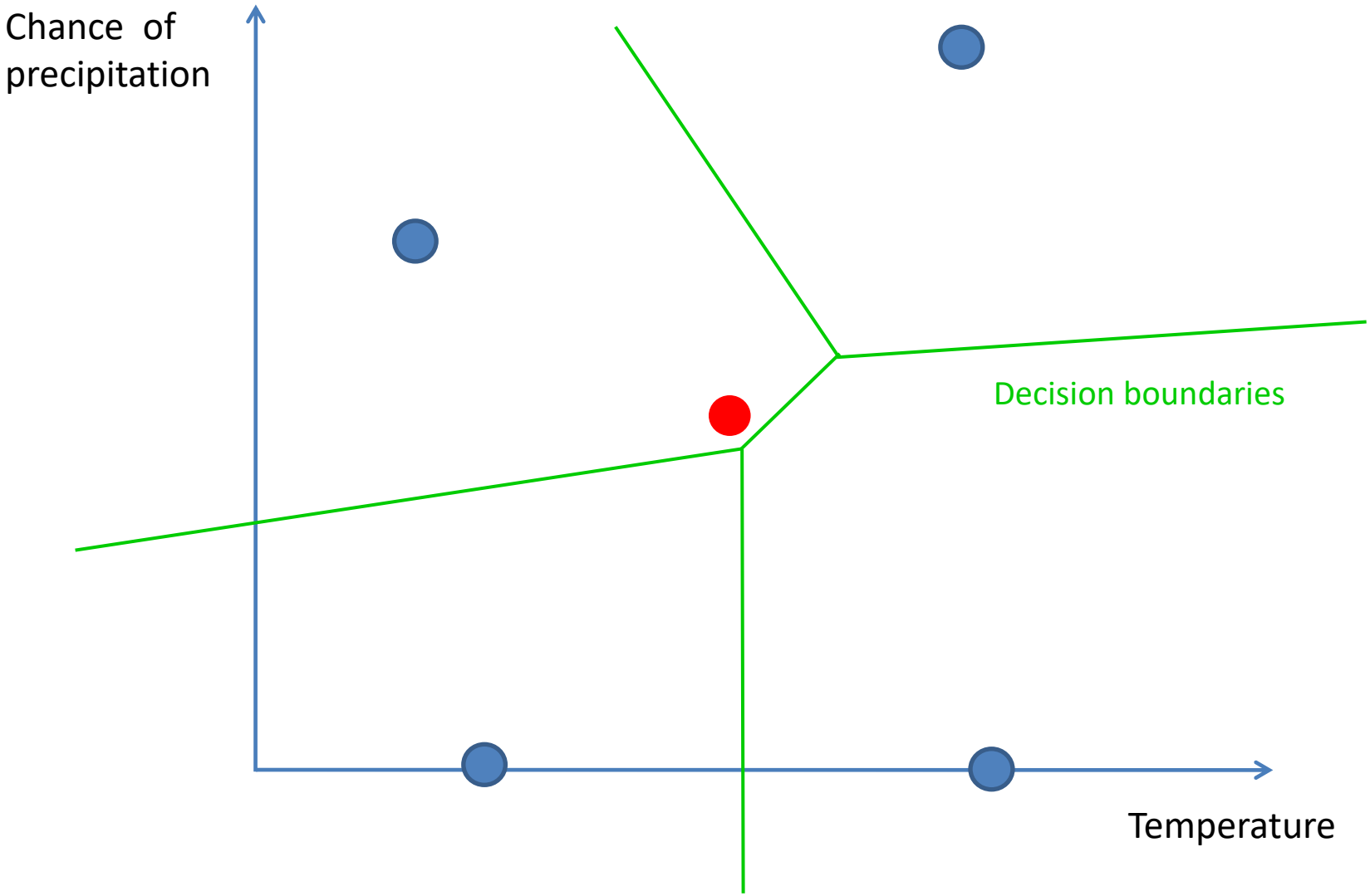
How we dress today?



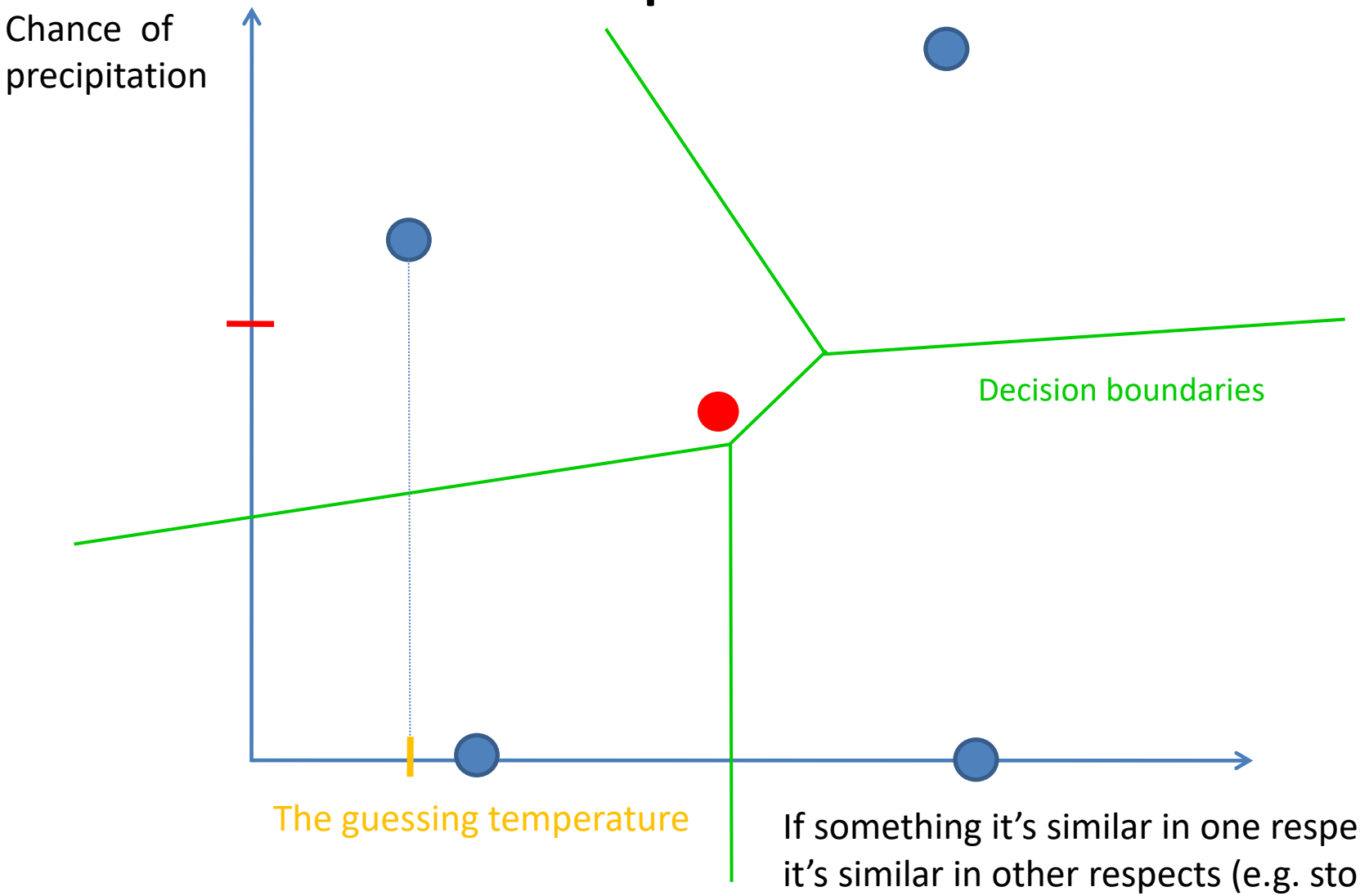
How we dress today?



How we dress today?

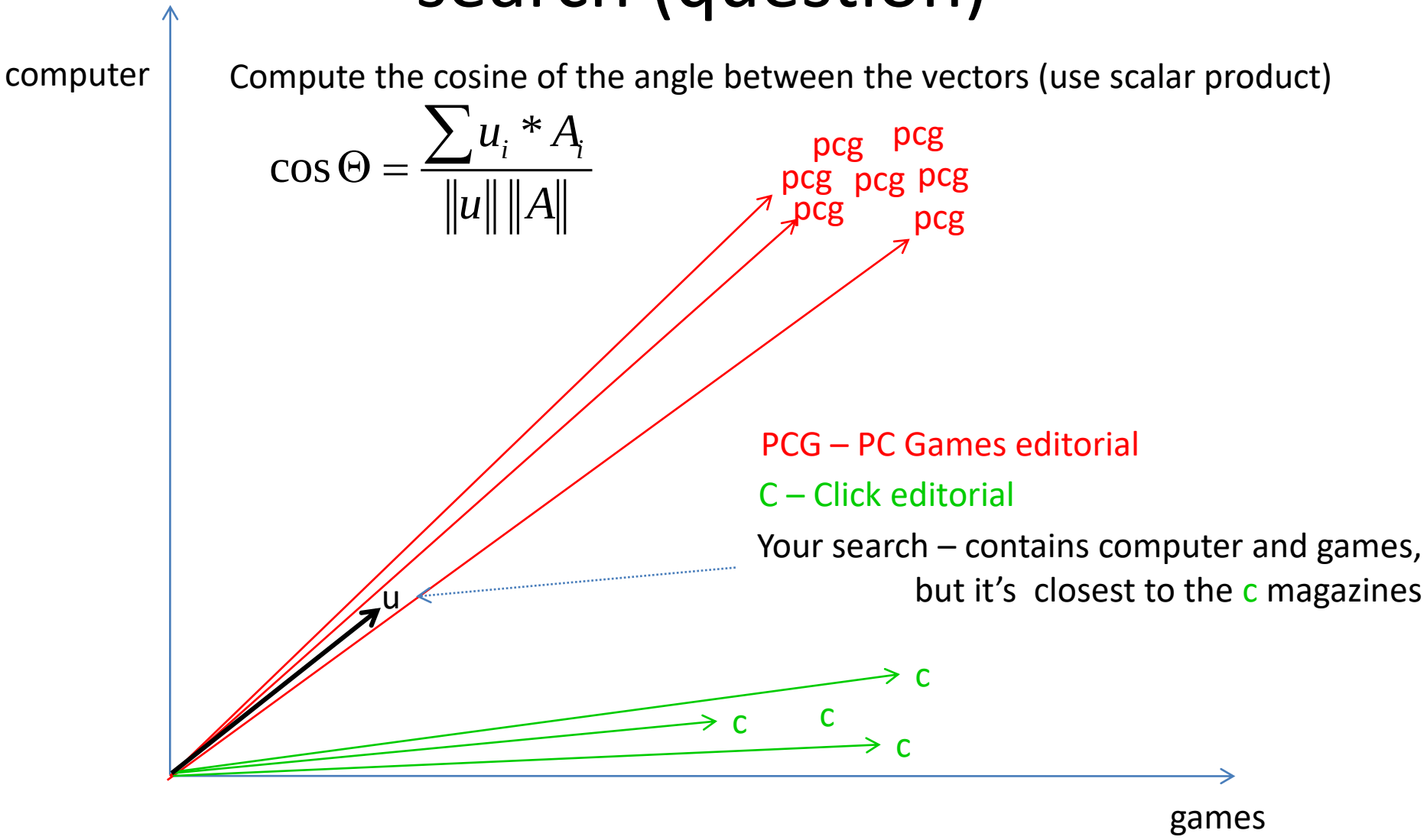


What if I measure only the temperature?

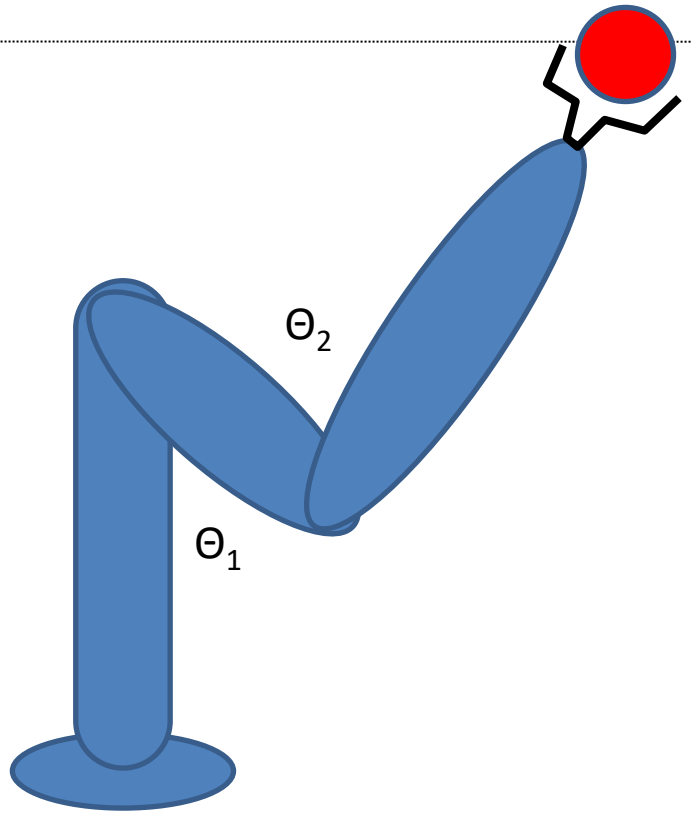


If something it's similar in one respect, it's similar in other respects (e.g. stories, legal cases, medical cases, etc.)

Finding magazines relevant to your search (question)



Robotic arm control-move the ball

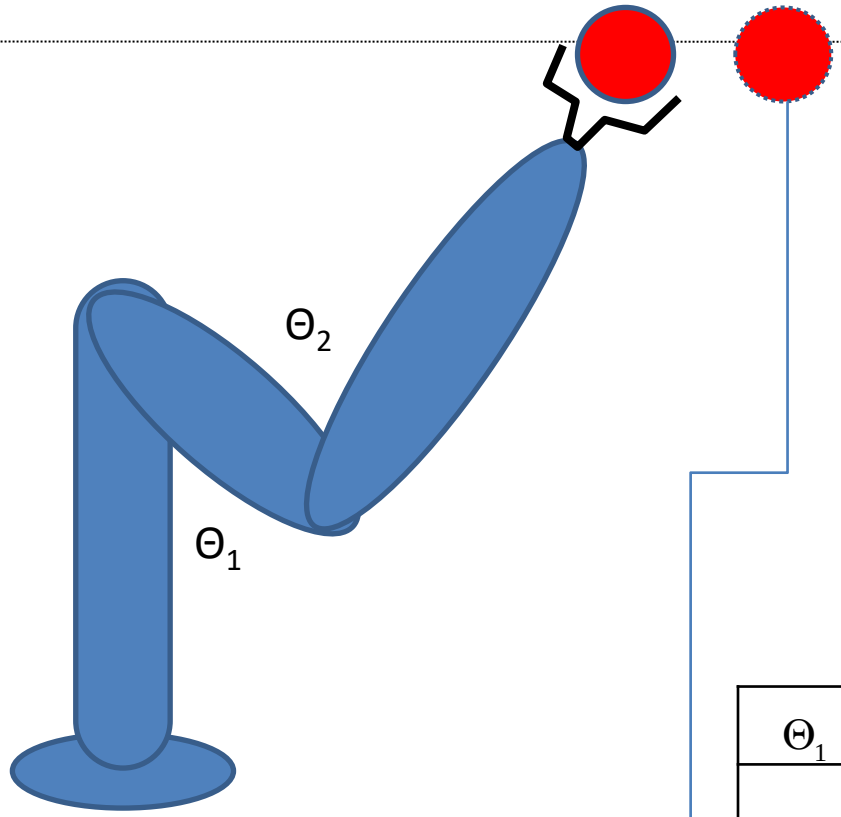


- Cinematic problem – translating the x coordinate of the ball into the Θ_1 and Θ_2 space
- Mechanical problem - Solve the equations – but there are very complicated

---

- Even so it doesn't work – there are frictions, precise of the measurements, etc.

We must fill the table



- For each position we have a corresponding row (the most similar one) in the table

torque

Θ_1	Θ_2	$\dot{\Theta}_1$	$\dot{\Theta}_2$	$\ddot{\Theta}_1$	$\ddot{\Theta}_2$	T_1	T_2

K Nearest Neighbors

- 1-NN
  - Given an unknown point, pick the closest 1 neighbor by some distance measure.
  - Class of unknown is the 1-nearest neighbor's label.
- k-NN
  - Given an unknown, pick the k closest neighbors by some distance function.
  - Class of unknown is the **mode** of the k-nearest neighbor's labels.
  - k is usually an odd number to facilitate tie breaking.

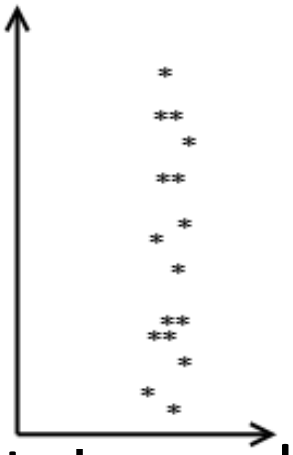
How to draw 1-NN decision boundaries

- Decision boundaries, lines on which it is equally likely to be in any of the classes.
  1. Examine the region where you think decision boundaries should occur.
  2. Find oppositely labeled points (+/-)
  3. Draw bisectors. (use pencil)
  4. Extend and join all bisectors. Erase extraneously extended lines.
  5. Remember to draw boundaries to the edge of the graph and indicate it with arrows! (a very common mistake)
  6. Your 1-NN boundaries generally should have sharp edges and corners (otherwise, you are doing something wrong or drawing boundaries for a higher k-nn.)

Problems

- SPREAD problem
  - Normalize the data

$$\sigma_x^2 = \frac{1}{N}\sum(x - \bar{x})^2 \quad , x' = \frac{x}{\sigma_x}$$



- WHAT MATTERS problem, the result depends only on *x* (not *y*)
  - The answer will be wrong

- NO PROBLEM problem, the answer doesn't depend the data at all

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

Artificial Intelligence Fundamentals

Learning: Identification Trees, Disorder

Egg fight problem

Champion ?	Multiple layers of painting ?	Big ?	Origin ?	Top?
No	?	No	Chicken	Round
No	Yes	Yes	Duck	Pointed
Yes	?	No	Guinea hen	Round
Yes	Yes	No	Guinea hen	Pointed
Yes	No	No	Chicken	Pointed
No	No	Yes	Duck	Flat
No	?	No	Chicken	Flat
No	?	Yes	Chicken	Flat

Why we cannot use a nearest neighbor algorithm?

Data set

- Non numeric
- Some characteristics don't matter
- Some characteristics do matter, but only part of the time
- Cost – some of the tests can be more expensive that others

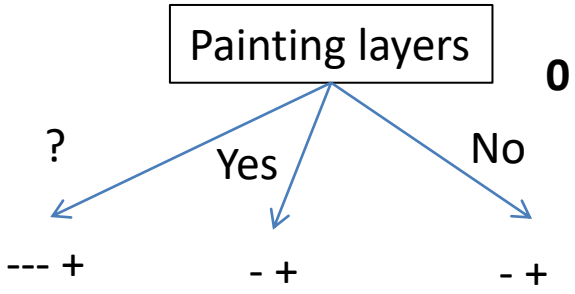
We need a method that enables computers to learn by assembling tests into an *identification tree*.

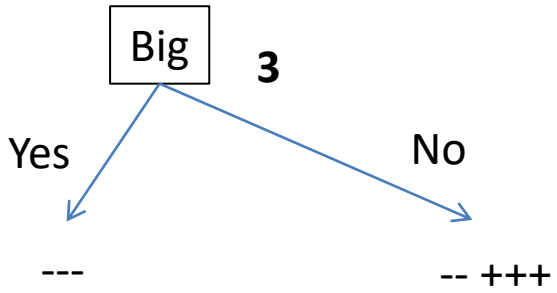
How we build that tree ? What will be a good characteristic of that tree?

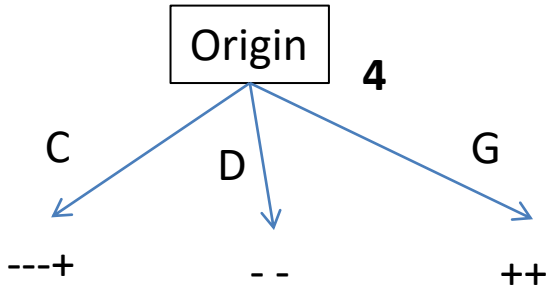
Occam's razor

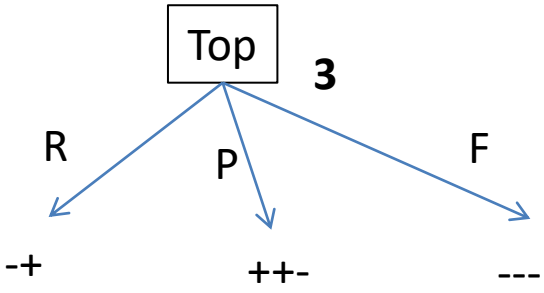
- The word is inherently simple. Therefore the smallest identification tree that is consistent with the samples is the one that is most likely to identify unknown objects correctly.
- How we can construct the smallest identification tree?

Tests

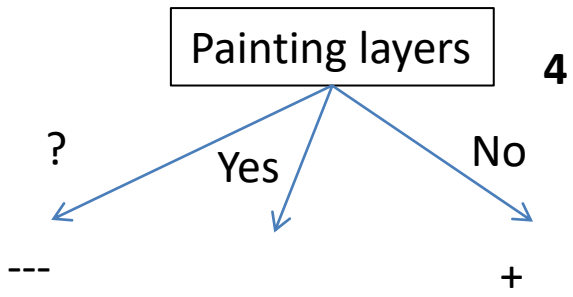




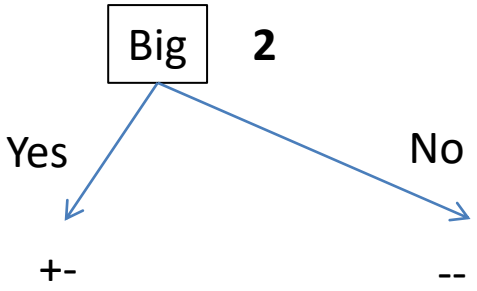


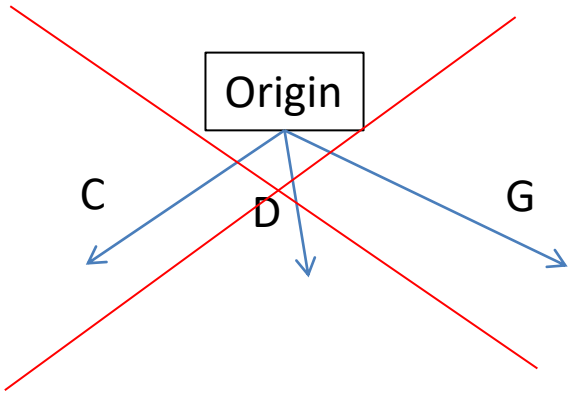


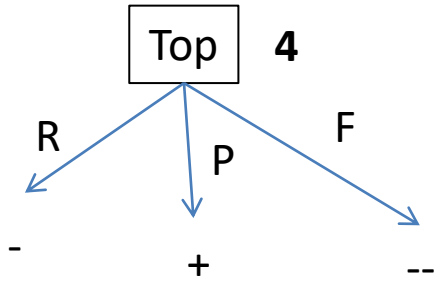
Champion ?	Multiple layers of painting ?	Big ?	Top?
No	?	No	Round
Yes	No	No	Pointed
No	?	No	Flat
No	?	Yes	Flat


Painting layers
4
?
Yes
No

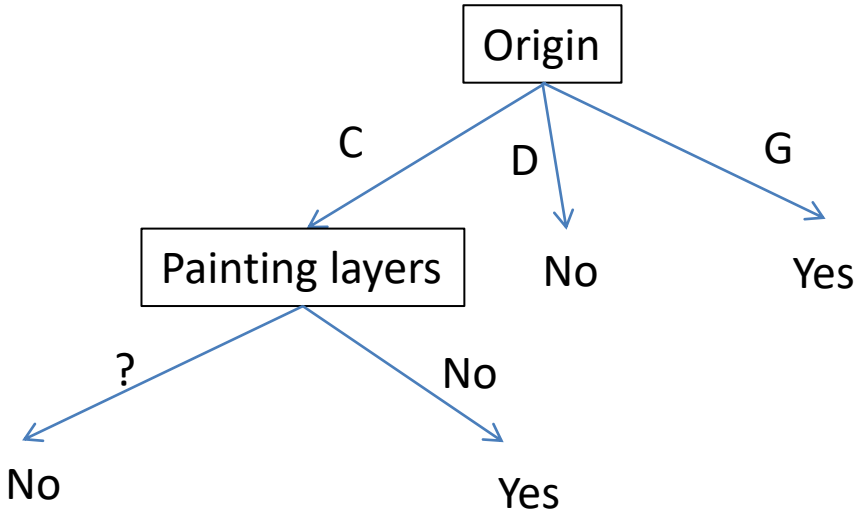
+


Big
2
Yes
No
+-
--


Origin
C
D
G

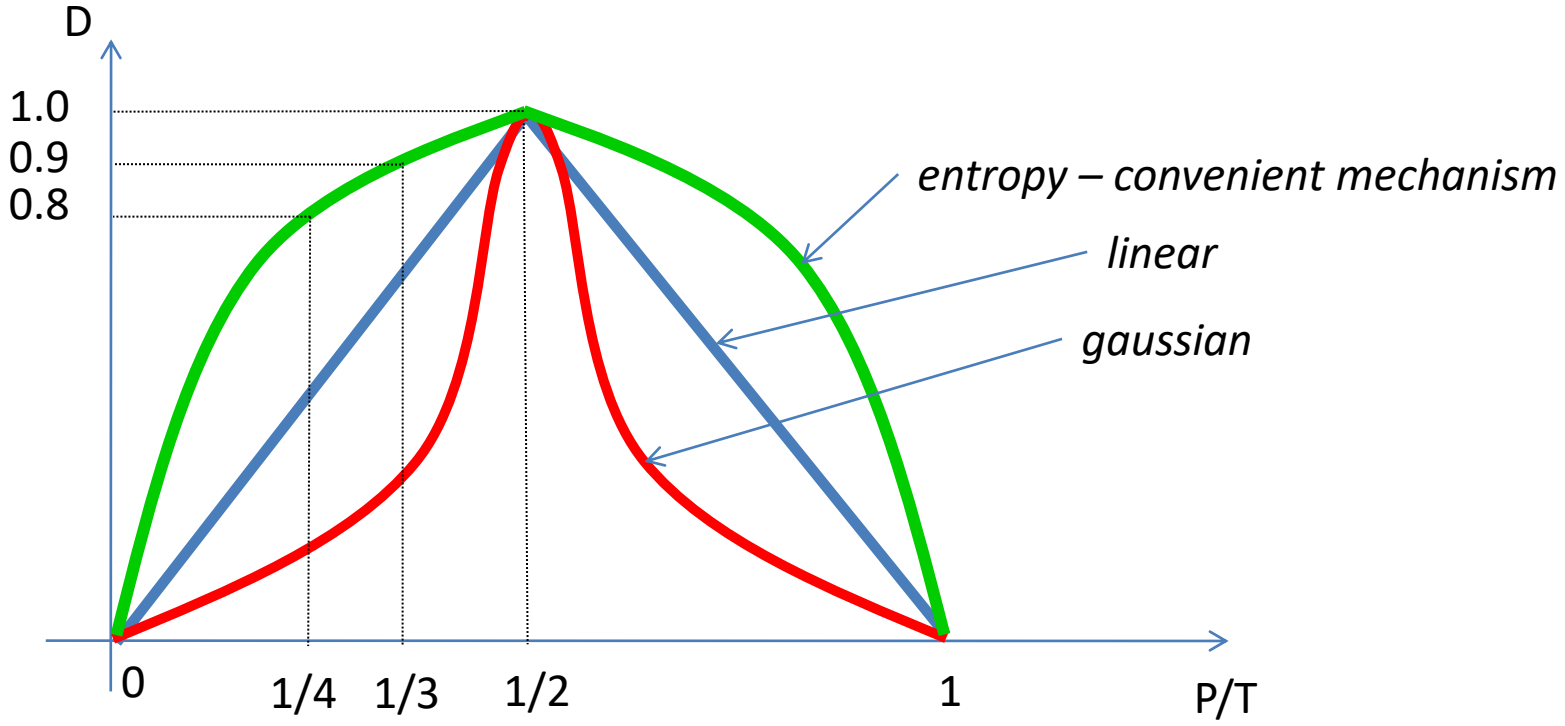

Top
4
R
P
F
-
+
--

Final identification tree



Measuring the disorder - entropy

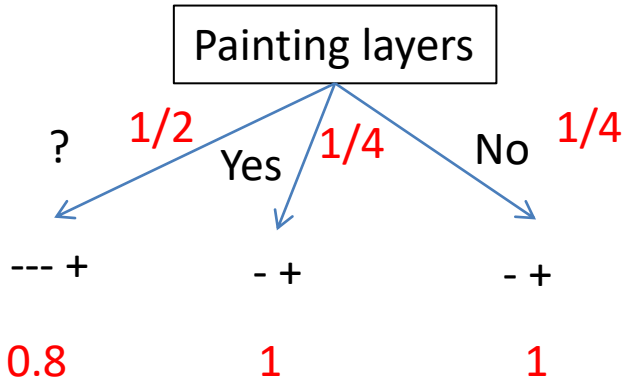
$$D(set) = -\frac{P}{T}\log_2\frac{P}{T} - \frac{N}{T}\log_2\frac{N}{T}$$



P/T = 1/2 , D = 1 P/T = 1 , D = 0 P/T = 0 , D = 0

Measuring the error of the test

$$E(Test) = \sum_{\substack{SETS \\ PRODUCED}} D(set) * \frac{\#\ of\ samples\ in\ set}{\#\ of\ samples\ handled\ by\ test}$$



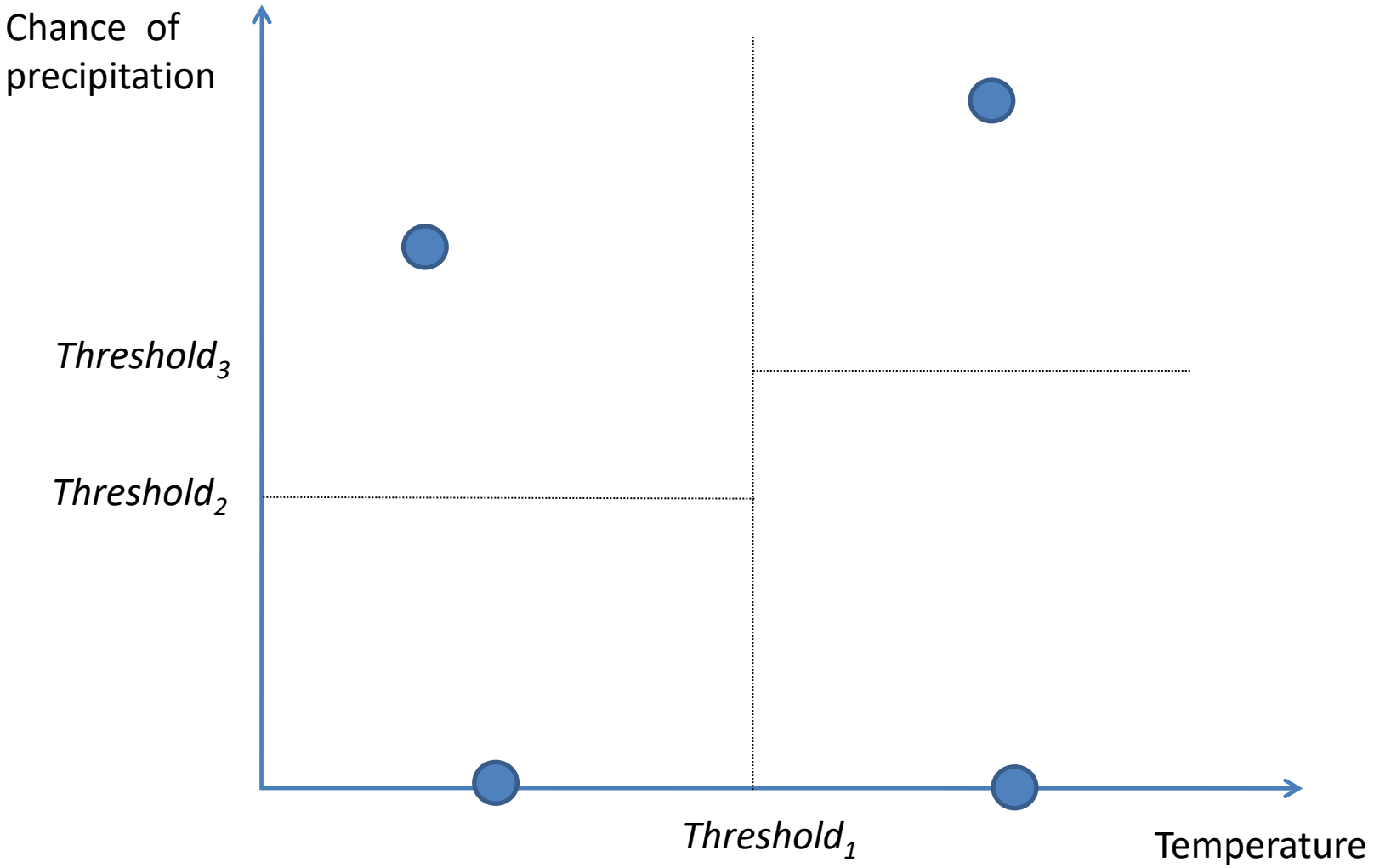
$$E(Test\ \text{Painting}) = 0.8 * \frac{1}{2} + 1 * \frac{1}{4} + 1 * \frac{1}{4} = 0.9$$

$$E(Test\ \text{Big}) = 0 * \frac{3}{8} + 0.9 * \frac{5}{8} \approx 0.56$$

$$E(Test\ \text{Origin}) = 0.8 * \frac{1}{2} + 0 * \frac{1}{4} + 0 * \frac{1}{4} = 0.4$$

$$E(Test\ \text{Top}) = 1 * \frac{1}{4} + 0.9 * \frac{3}{8} + 0 * \frac{3}{8} = 0.5875$$

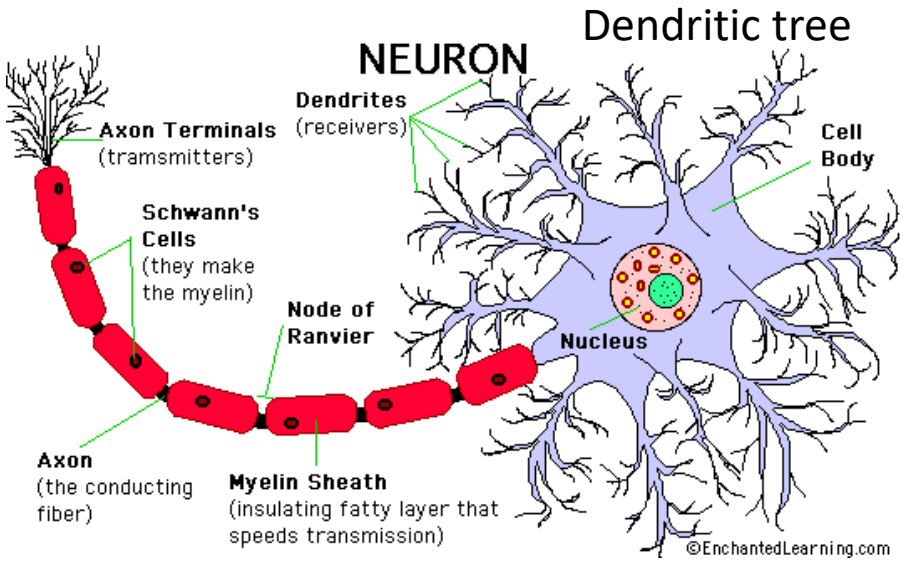
Numeric data



Artificial Intelligence Fundamentals

Learning: Neural Networks

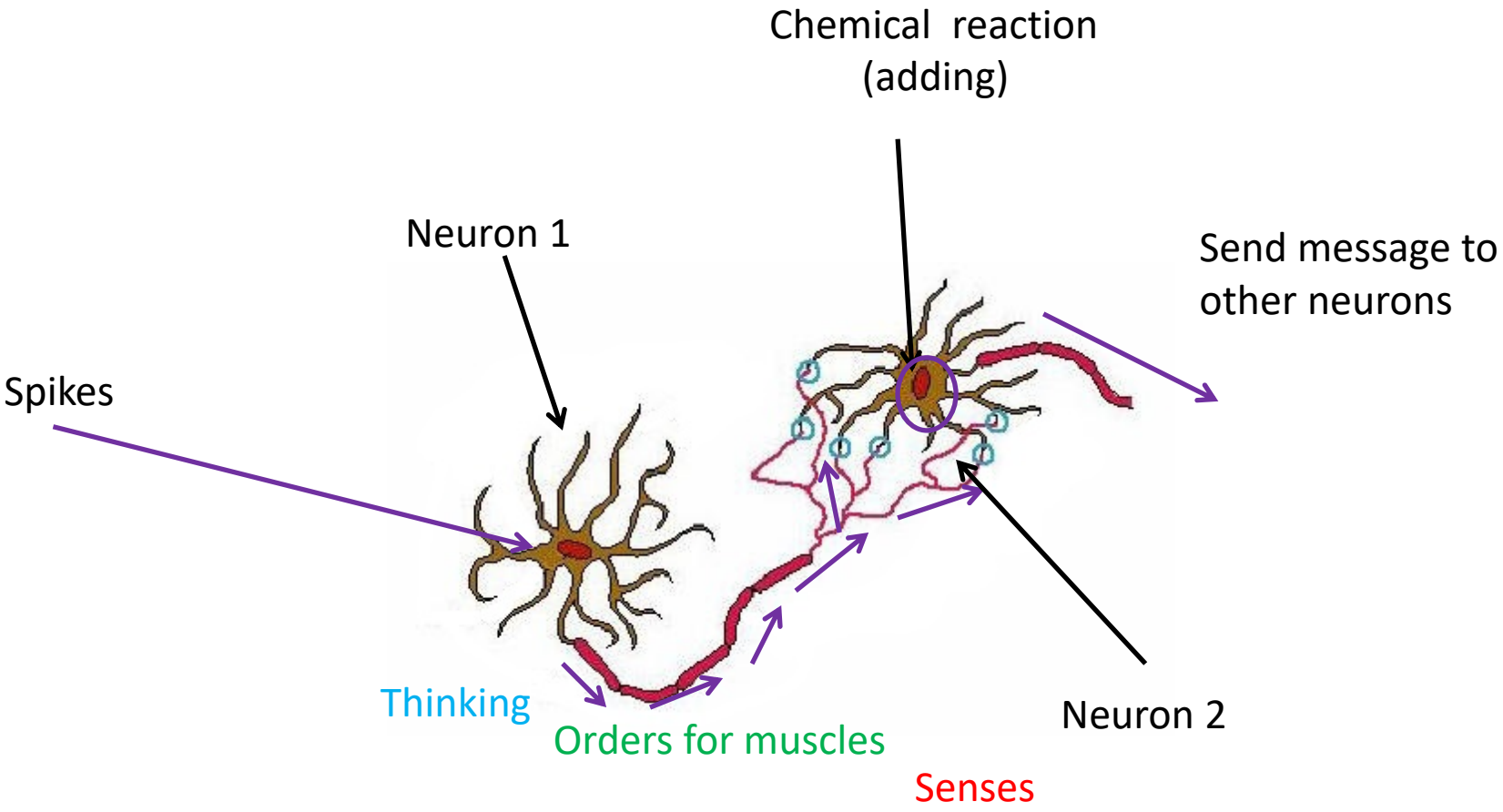
Biology



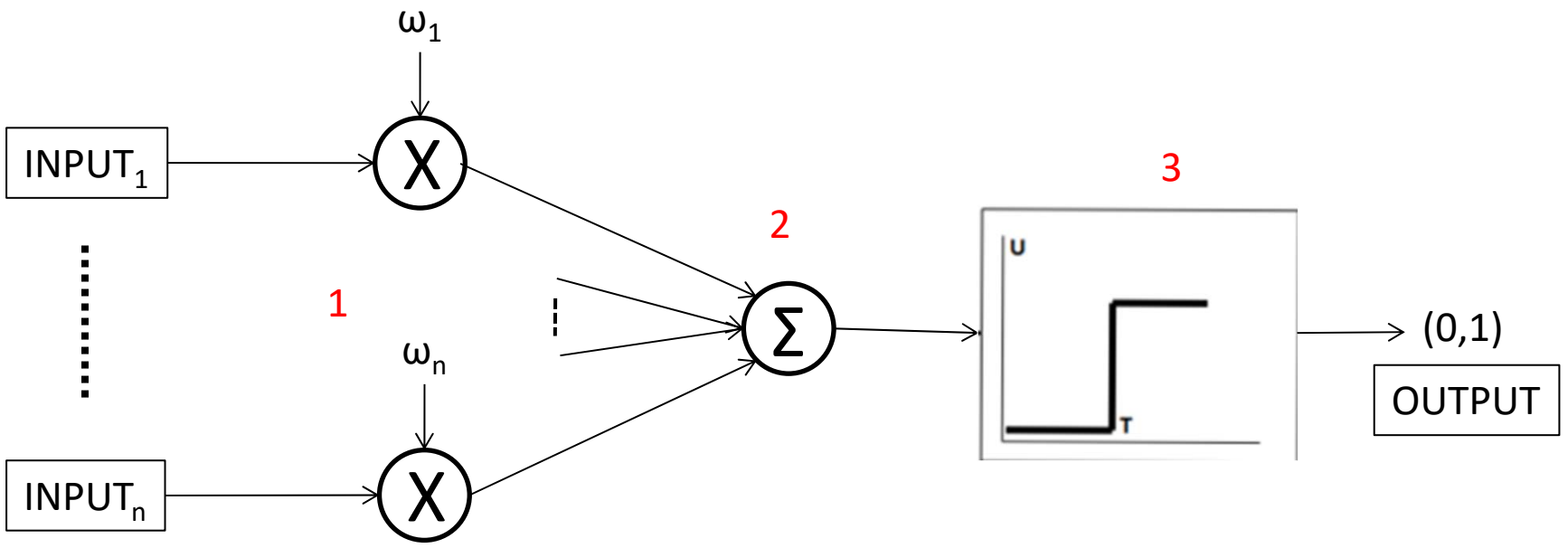
Major observations:

1. Synaptic weights
2. Cumulative effect
3. All or none

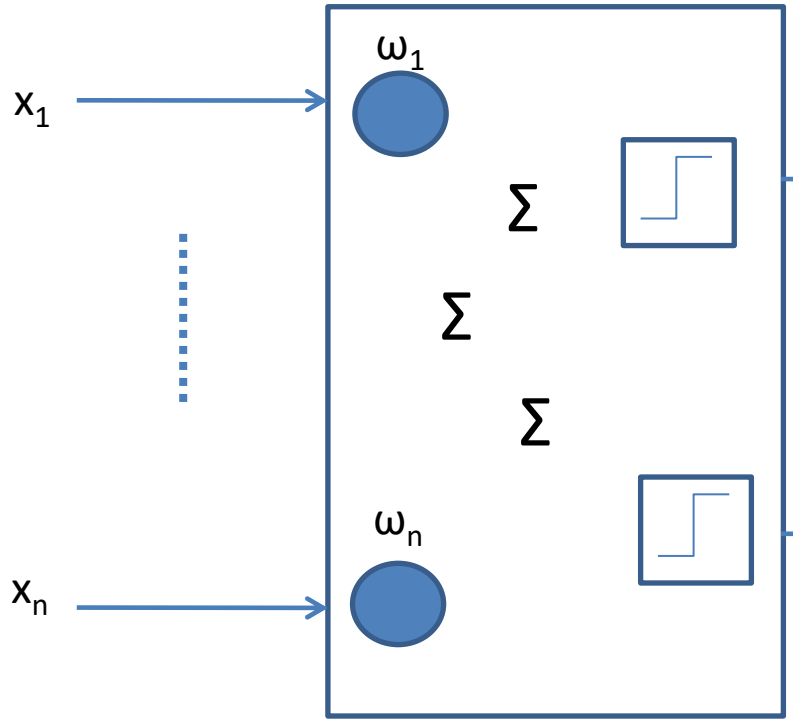
Neurons connectivity



Artificial neuron model



Neural Network



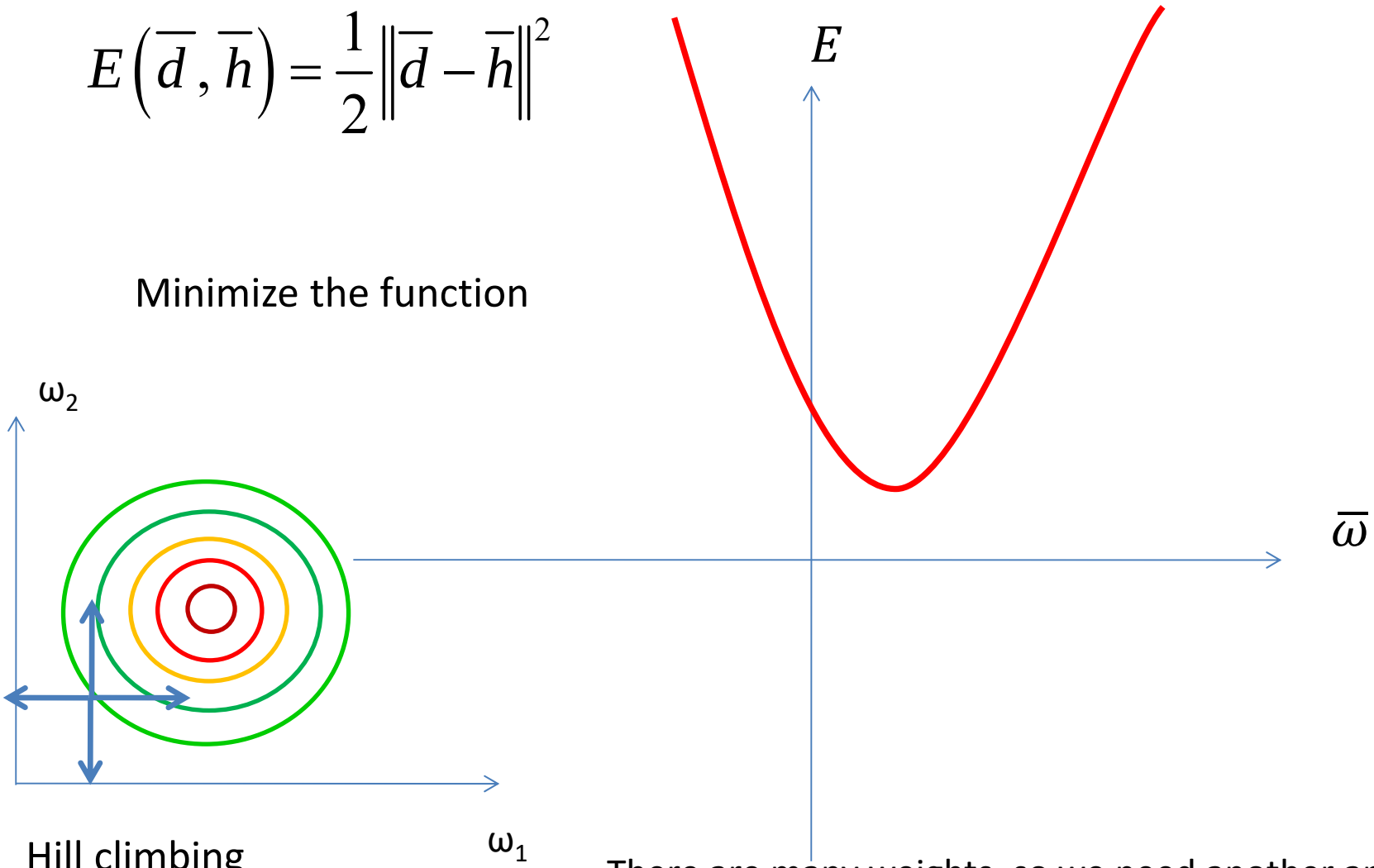
$$\overline{h} = f\left(\overline{x}, \overline{\omega}\right)$$

Activation function $\overline{d} = g\left(\overline{x}\right)$

Performance – error function

$$E(\overline{d},\overline{h}) = \frac{1}{2}\left\|\overline{d} - \overline{h}\right\|^2$$

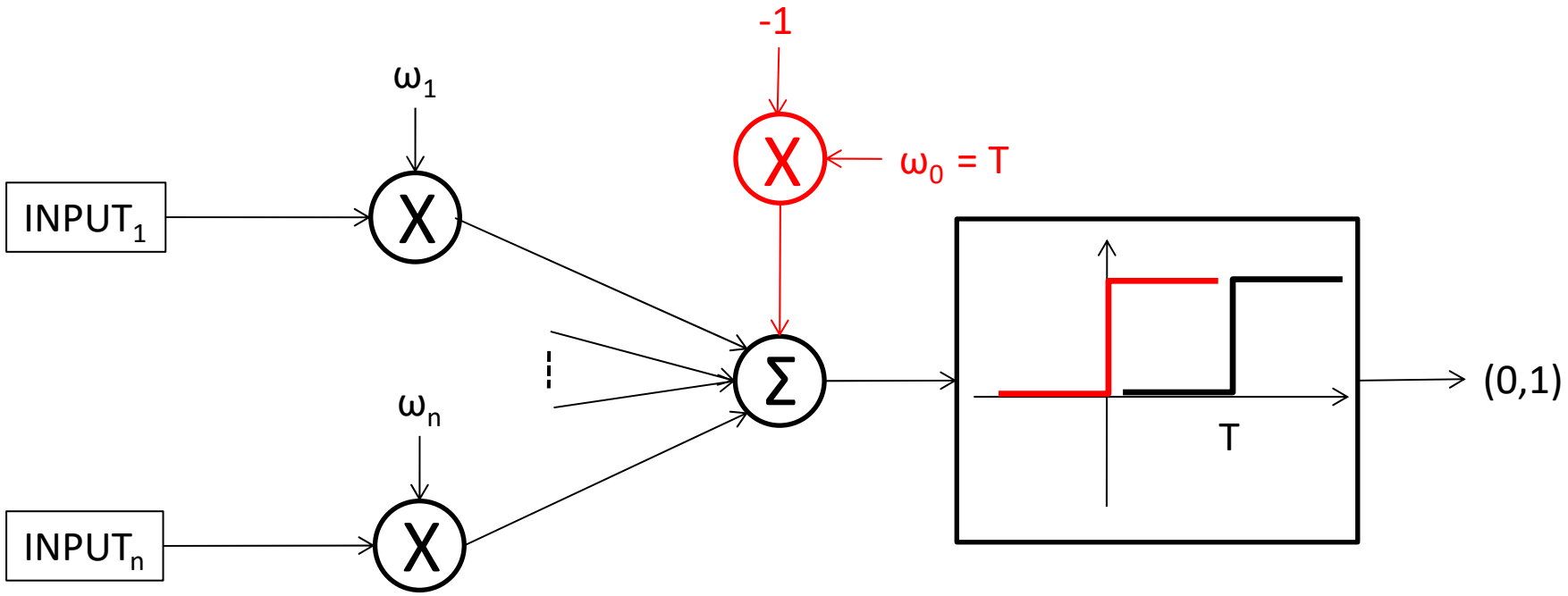
Minimize the function



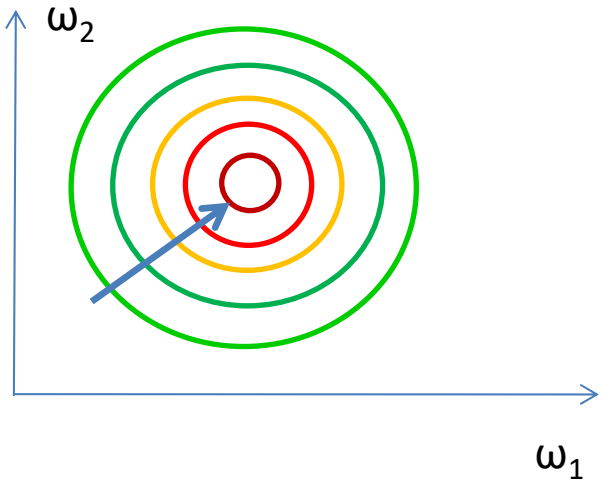
Hill climbing

There are many weights, so we need another approach.

The threshold



Gradient descent

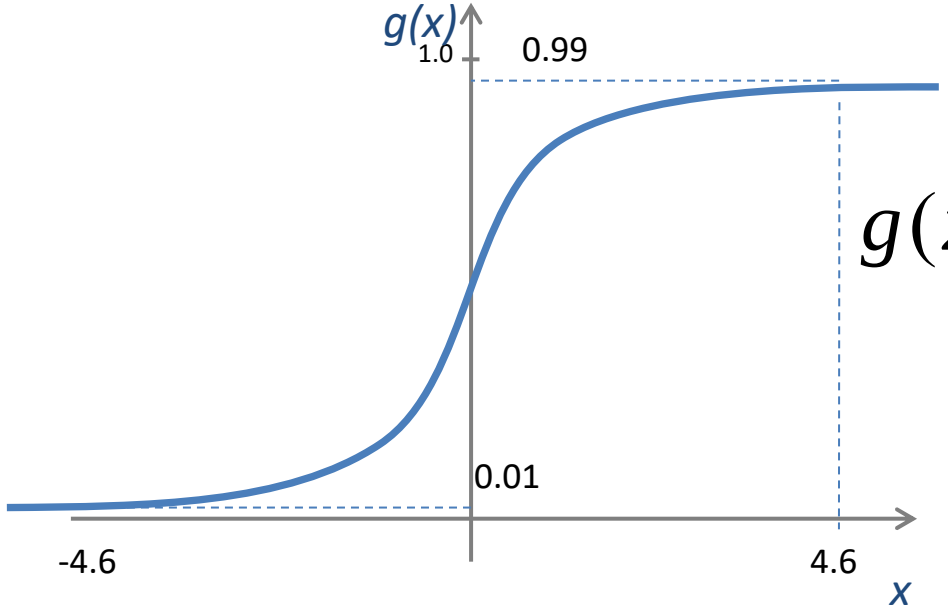


Gradient descent – moving toward the minimum.

learning rate

$$\Delta\omega = \sigma\left(\frac{\partial E}{\partial \omega_1} i + \frac{\partial E}{\partial \omega_2} j\right)$$

We need a continuous function – sigmoid function.

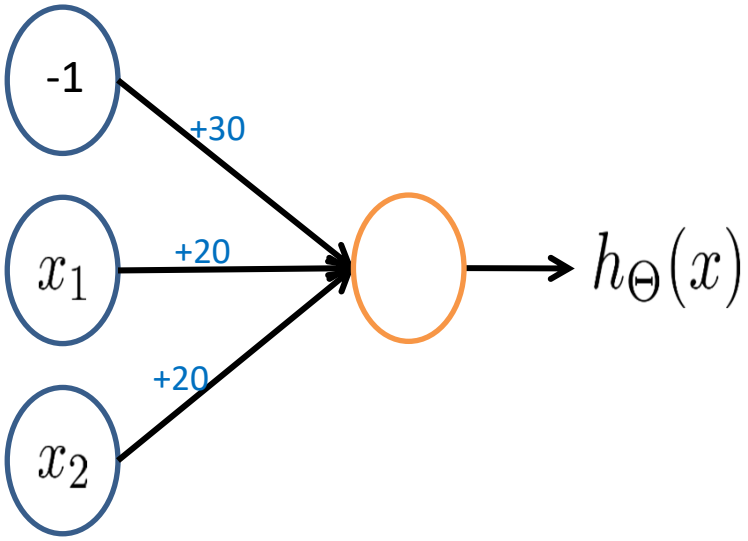


$$g(x) = \frac{1}{1+e^{-x}}$$

Example: logical AND

$x_1, x_2 \in \{0, 1\}$

$y = x_1 \text{ AND } x_2$

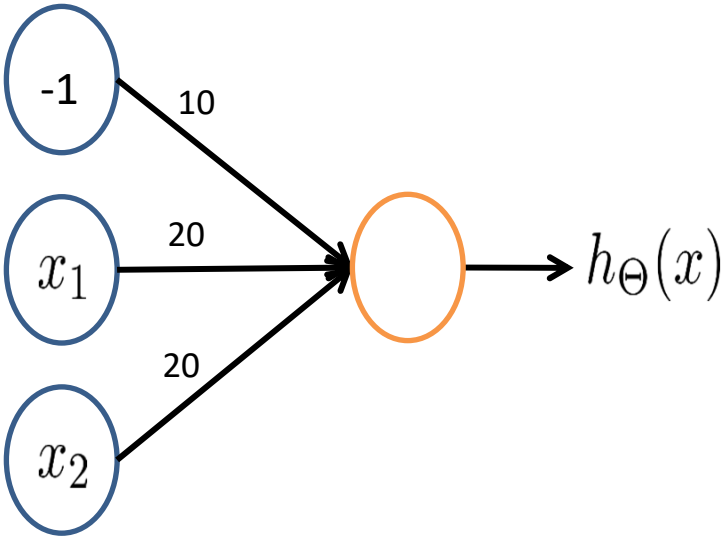


x_1	x_2	$h_\Theta(x)$
0	0	g(-30) ≈ 0
0	1	g(-10) ≈ 0
1	0	g(-10) ≈ 0
1	1	g(10) ≈ 1

$h(x) = x_1 \text{ AND } x_2$

$h(x) = g(-30 + 20\, x_1 + 20\, x_2)$

Example: logical OR

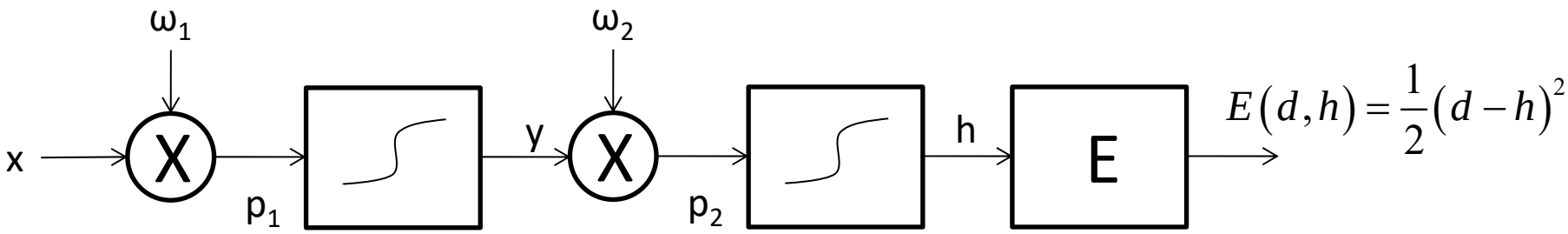


x_1	x_2	$h_\Theta(x)$
0	0	g(-10) ≈ 0
0	1	g(10) ≈ 1
1	0	g(10) ≈ 1
1	1	g(30) ≈ 1

$h(x) = g(-10 + 20\,x_1 + 20\,x_2)$

$h(x) = x_1$ OR x_2

A simple neural network



$$\frac{\partial E}{\partial \omega_2} = \frac{\partial E}{\partial h}\frac{\partial h}{\partial \omega_2} = -(d-h)\frac{\partial h}{\partial \omega_2} = -(d-h)\frac{\partial h}{\partial p_2}\frac{\partial p_2}{\partial \omega_2} = -(d-h)\frac{\partial h}{\partial p_2}y$$

$$\frac{\partial E}{\partial \omega_1} = \frac{\partial E}{\partial h}\frac{\partial h}{\partial \omega_1} = -(d-h)\frac{\partial h}{\partial \omega_1} = -(d-h)\frac{\partial h}{\partial p_2}\frac{\partial p_2}{\partial \omega_1} = -(d-h)\frac{\partial h}{\partial p_2}\frac{\partial p_2}{\partial y}\frac{\partial y}{\partial \omega_1}$$

$$= -(d-h)\frac{\partial h}{\partial p_2}\omega_2\frac{\partial y}{\partial p_1}\frac{\partial p_1}{\partial \omega_1} = -(d-h)\frac{\partial h}{\partial p_2}\omega_2\frac{\partial y}{\partial p_1}x$$

Derivative of the sigmoid

$$g = \frac{1}{1+e^{-x}}$$

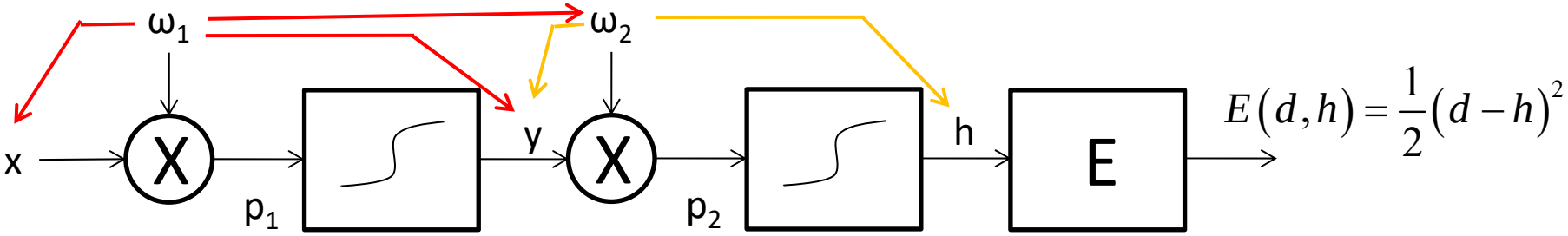
$$\frac{dg}{dx} = \frac{d}{dx}(1+e^{-x})^{-1} = -(1+e^{-x})^{-2}(-e^{-x}) = \frac{e^{-x}}{(1+e^{-x})^{-2}}$$

$$= \frac{1+e^{-x}-1}{(1+e^{-x})^{-2}} = \frac{1}{(1+e^{-x})}\left[\frac{1+e^{-x}}{(1+e^{-x})} - \frac{1}{(1+e^{-x})}\right] = g(1-g)$$

$$\frac{\partial E}{\partial \omega_2} = -(d-h)\frac{\partial h}{\partial p_2} y \quad = -(d-h)h\,(1-h)\,y$$

$$\frac{\partial E}{\partial \omega_1} = -(d-h)\frac{\partial h}{\partial p_2}\omega_2 \frac{\partial y}{\partial p_1} x = -(d-h)h(1-h)\,\omega_2 y(1-y)x$$

Backpropagation algorithm



$$\frac{\partial E}{\partial \omega_2} = (h-d)h(1-h)y$$

$$\frac{\partial E}{\partial \omega_1} = (h-d)h(1-h)\omega_2 y(1-y)x$$

It's only a local computation - depends by the staff on the vicinity

Changing the weights

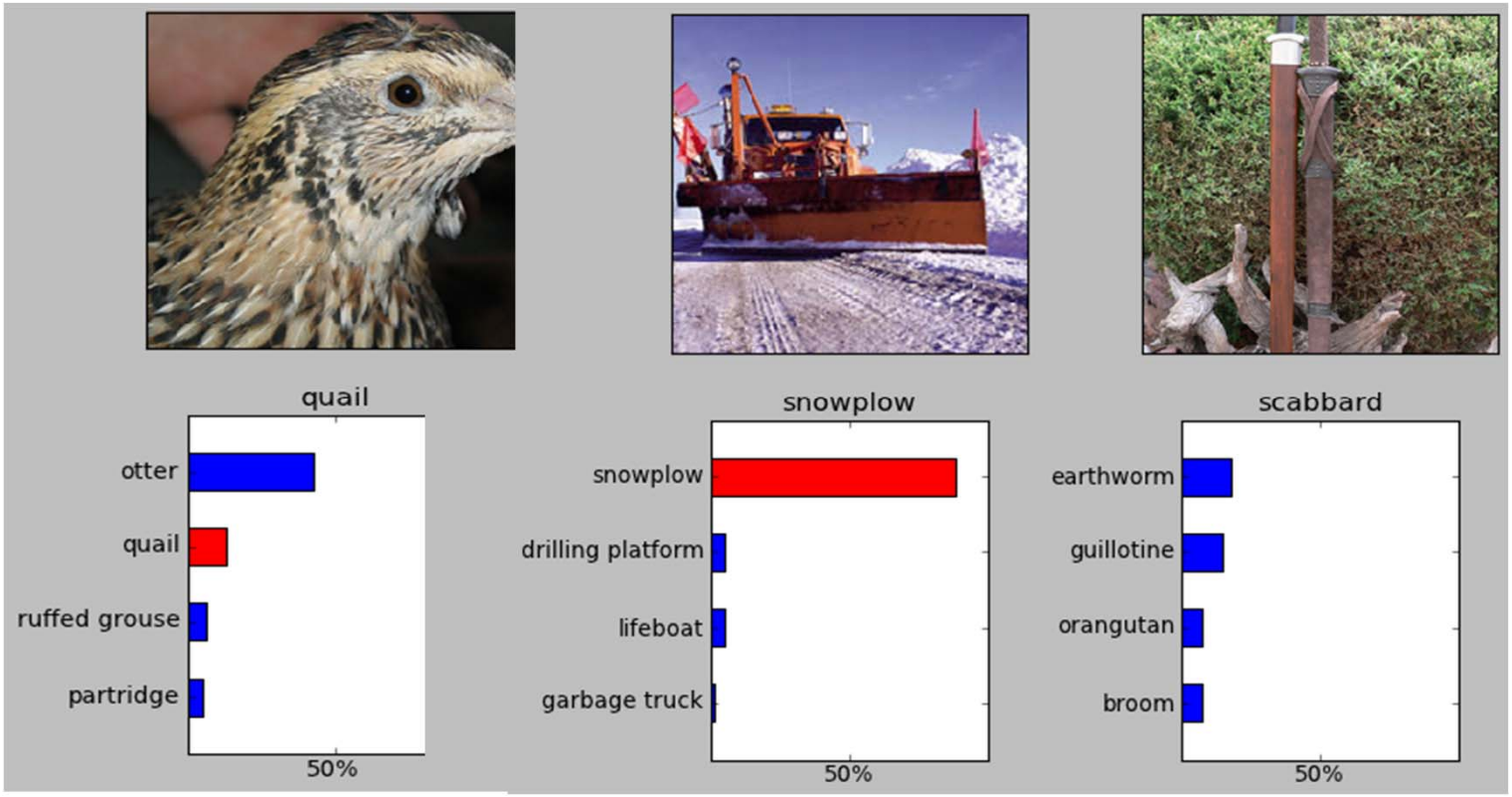
$$\omega_1 = \omega_{1old} + \sigma \frac{\partial E}{\partial \omega_1}$$

$$\omega_2 = \omega_{2old} + \sigma \frac{\partial E}{\partial \omega_2}$$

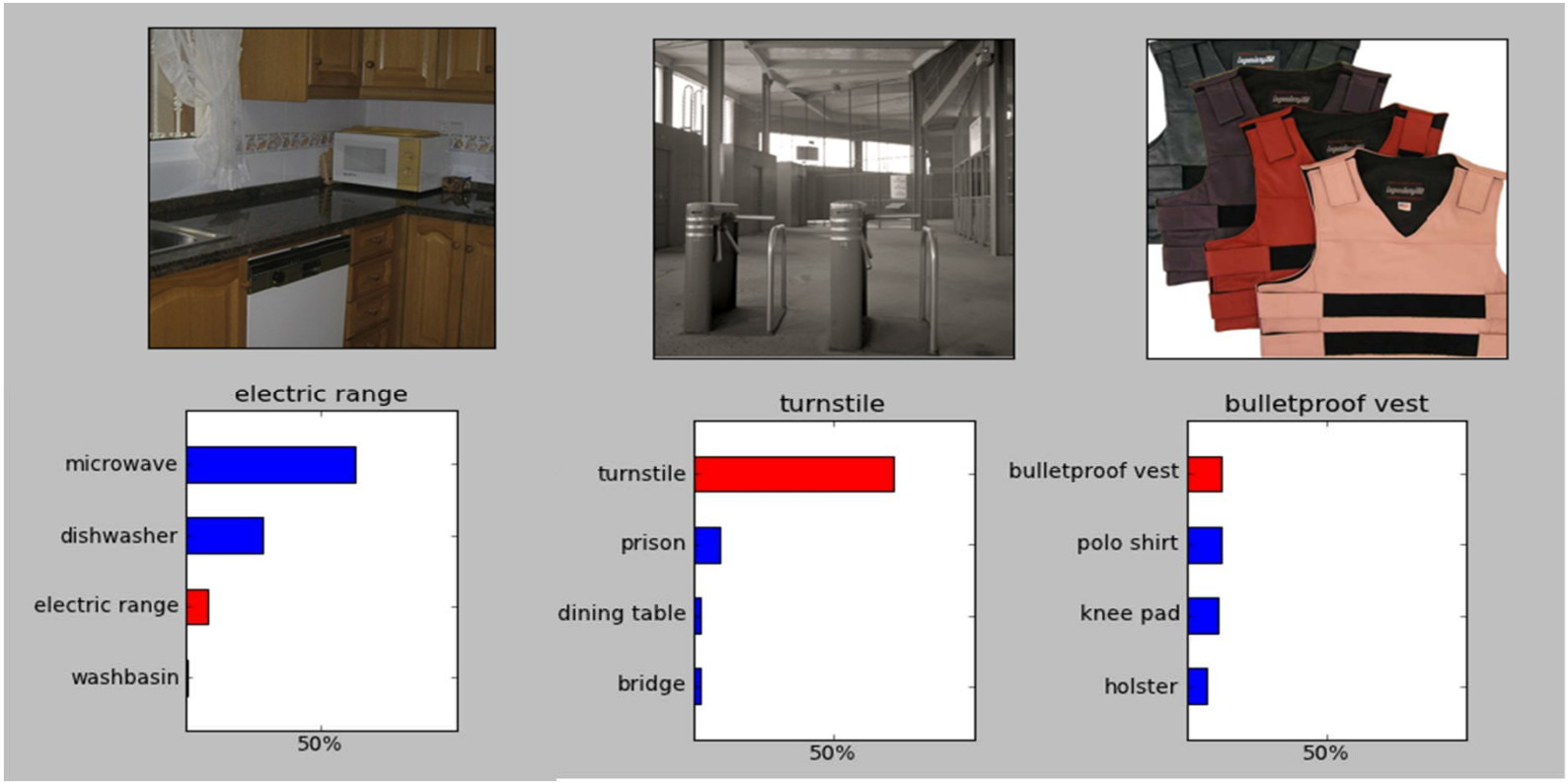
- It’s very important to choose a right learning rate σ (e.g. usually 0.01 or 0.001)

AlexNet

Some examples from an earlier version of the net

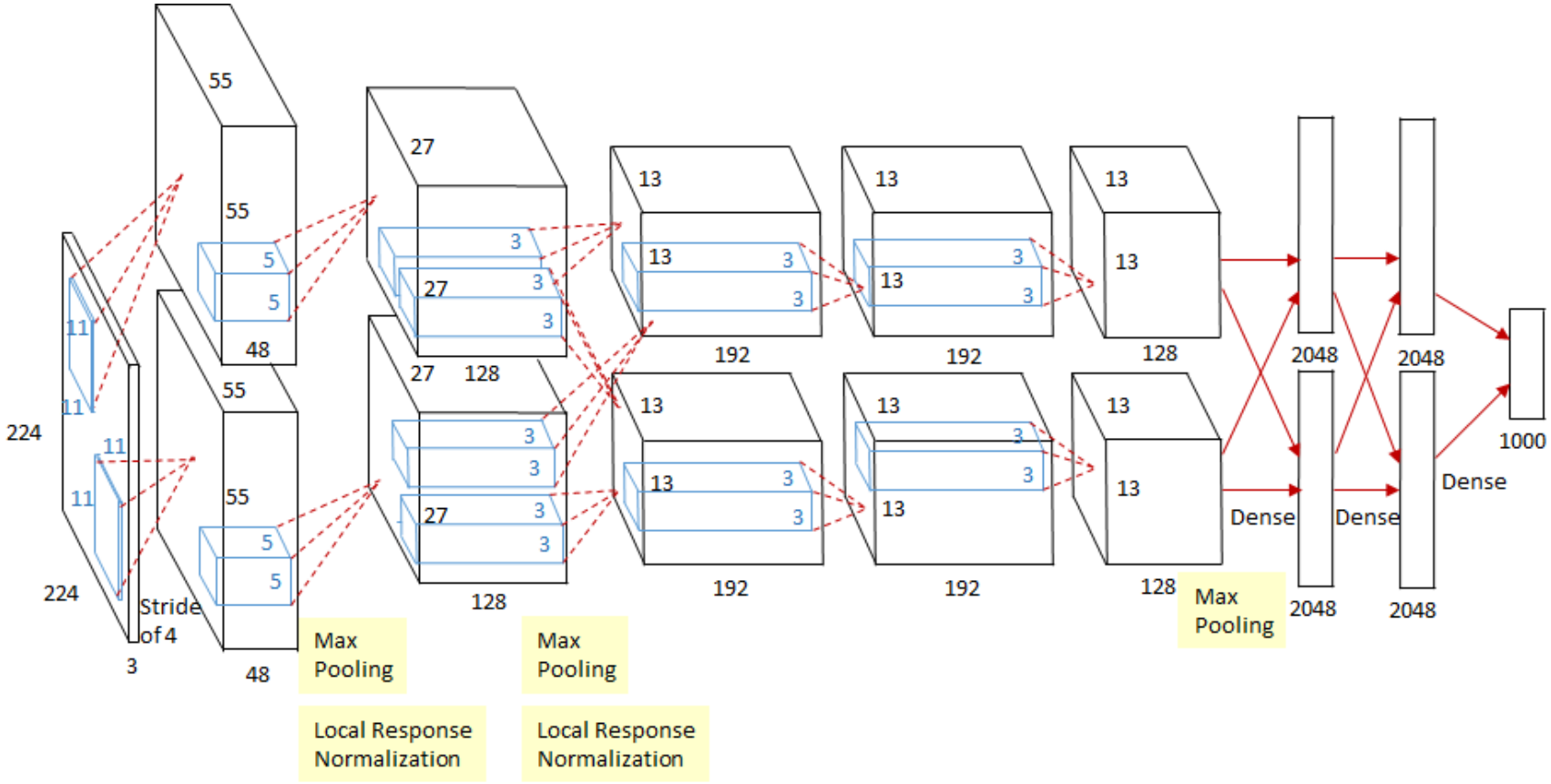


It can deal with a wide range of objects



Geoffrey Hinton

AlexNet

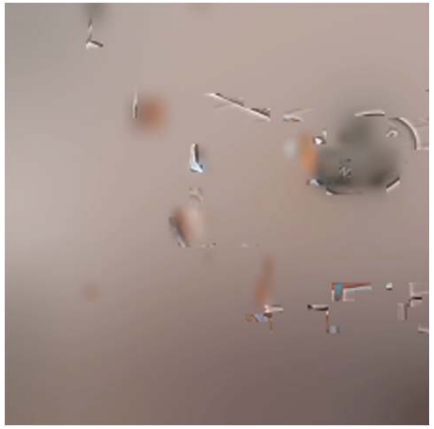


- 60 millions parameters
- 224 x 224 RGB input
- 1000 classes output

Problems

- Fitting a curve. Why it's better than other methods?
- How we encode the problems parameters ? (e.g. weather prediction – the hard problem)
- Over fitting
- Oscillations


DW C



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

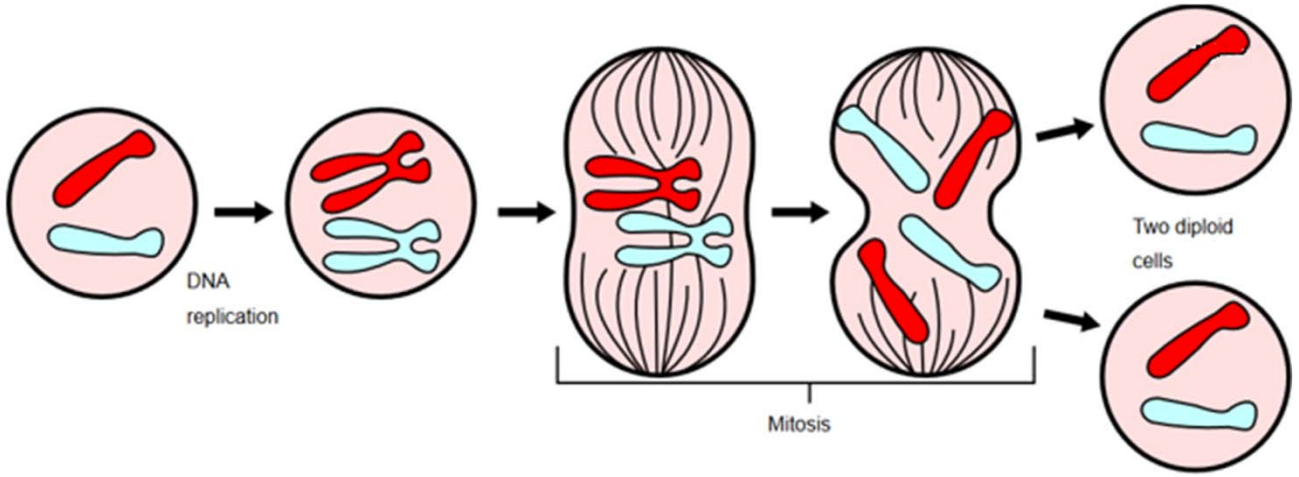
- Artificial Intelligence (3rd Edition), Patrick Winston, Chapter 12

Artificial Intelligence Fundamentals

Learning: Genetic Algorithms

Mitosis

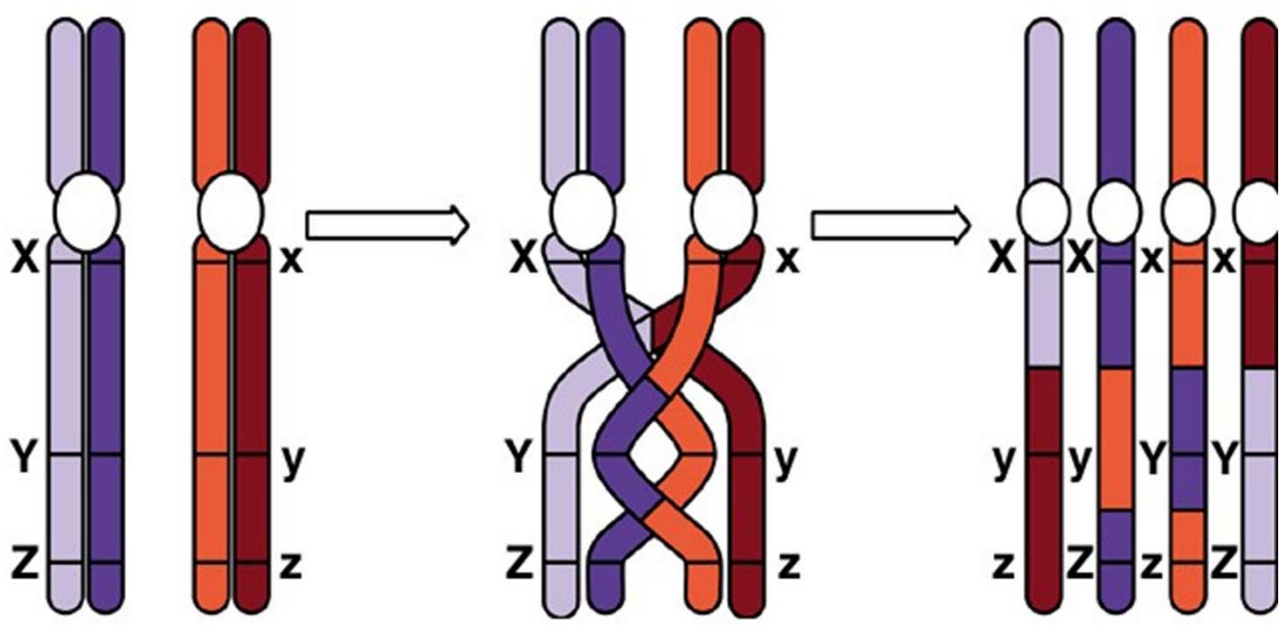
- Process when new cells are generated



Chromosome number	46	46	46	46
Chromatid number	46	92	92	46

Reproduction

- Meiosis - Special type of cell division that occurs in sexually reproducing organisms



Chromosome - mutation

…..0110001000100….. A string of things

Population	**Mutation**
…..0110001000100…..	…..0110101000100…..
…..0111010101100…..	…..0111011101100…..
…..1110110111100…..	…..1110110111100…..
…..0001001101100…..	…..0101001101100…..

Parameters – how many mutations per chromosome ; how many chromosome we allow to be mutated?

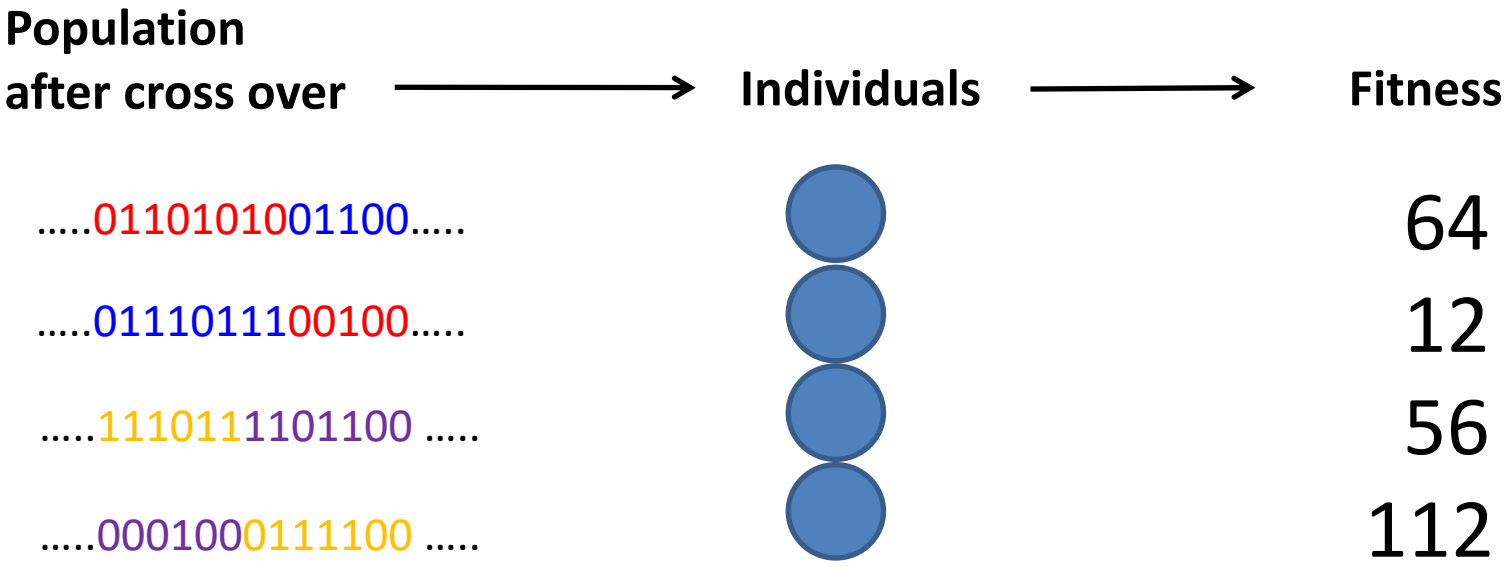
Chromosome – cross over

Population after mutation	**Population after cross over**
…..01101010000100…..	…..01101010001100…..
…..01110111011100…..	…..01110111000100…..
…..11101101111100…..	…..11101111011100 …..
…..00010011011100…..	…..00010000111100 …..

Parameters – how many cross over operations per chromosome ;?

Chromosome - genotype to phenotype transition

- The chromosome must be interpreted in order to be something
- Each chromosome create an individual



Fitness – Probability - Selection

Fitness	→	Probability	→	Selection phenotype to genotype
64		0.44		…..0111111000100…..
12		0.07		…..0111010101100…..
56		0.56		…..0001110001100…..
112		0.82		…..1010001000100…..

Selection methods

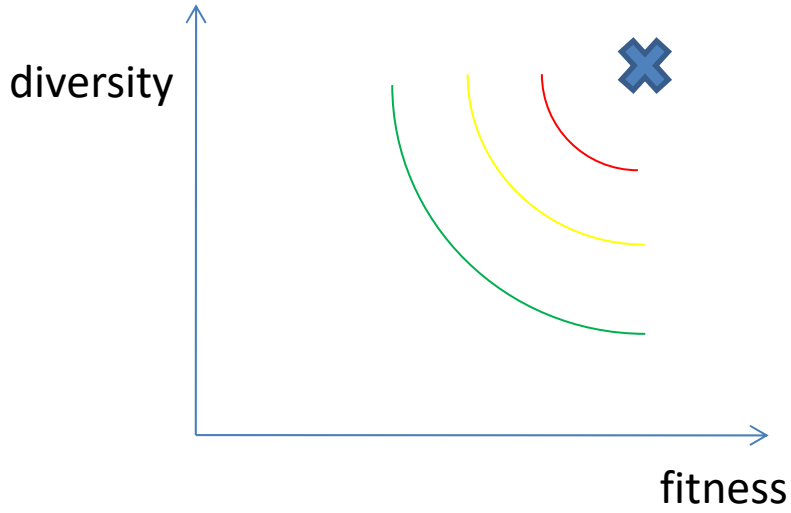
1. Based on fitness

$$P_i = \frac{f_i}{\sum_j f_j}$$

2. Rank space

$$P_1 = p_c \qquad P_2 = (1-p_c)\,p_c \qquad P_3 = (1-p_c)^2\,p_c \qquad \ldots\ldots\ldots \qquad P_{N-1} = (1-p_c)^{N-2}\,p_c \qquad P_N = (1-p_c)^{N-1}$$

3. Fitness and diversity rank



Planning problems

- S_1 , S_2 , S_3 … S_n
- S_1 , S_2 , S_3 … S_n
- S_1 , S_2 , S_3 … S_n

Rule based problems

- IF X and Y THEN …..
- IF A and B THEN …..
- IF X’ AND M THEN ….. X’- mutated X

Related resources

Readings

- Artificial Intelligence (3rd Edition), Patrick Winston, Chapter 25

Artificial Intelligence Fundamentals

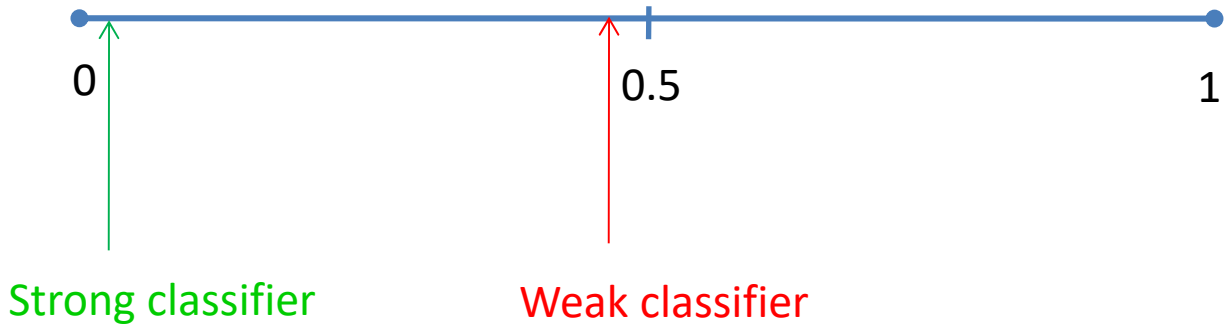
Learning: Boosting

- Binary classification – classify the elements of a given set into two groups on a given rule
- Finding the classification rule can be a difficult task
- A crowd can be smarter than the participant in the crowd?

Classifiers strong/weak

- Suppose we have a set of classifiers
  h which give as the output {-1 , +1}

- Error rate

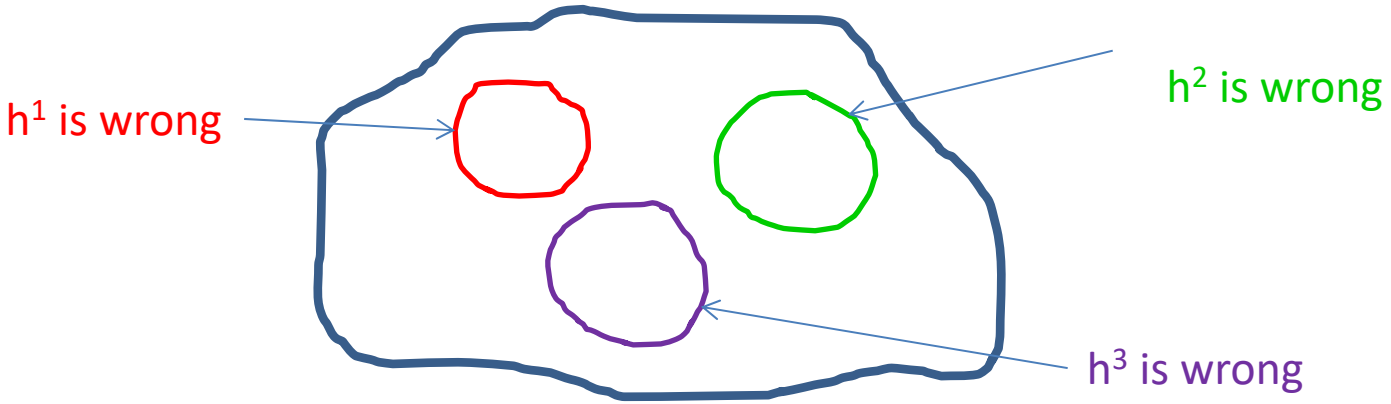


- Can we make a strong classifier by combining several of these weak classifiers and let them vote?

$$H(x) = sign\left(h^1(x) + h^2(x) + \ldots + h^n(x)\right) \quad x \text{ - is a sample}$$

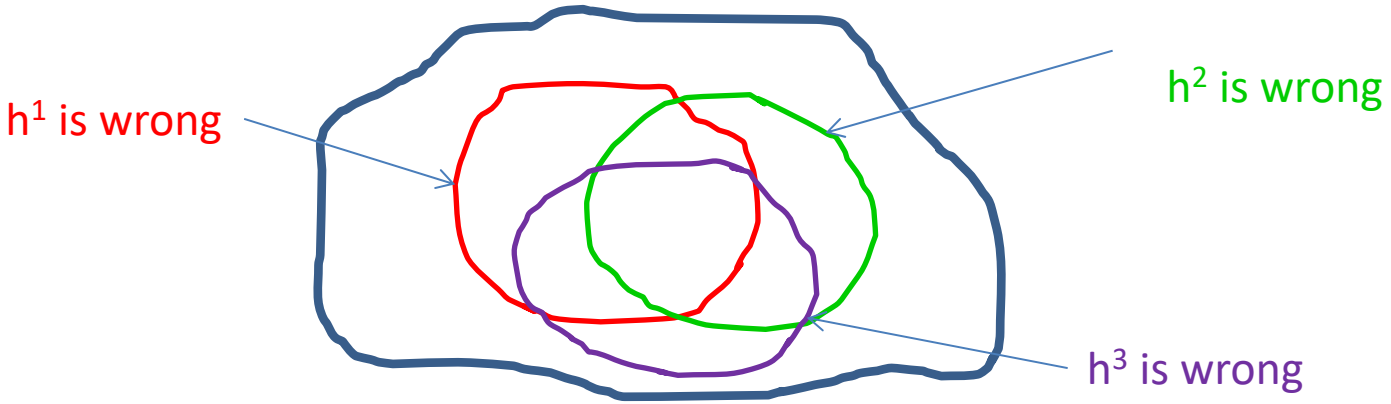
The perfect classifiers

$$H(x) = sign\left(h^1(x) + h^2(x) + h^3(x)\right)$$



- If it's look like this, always we will have 0 error

A real situation

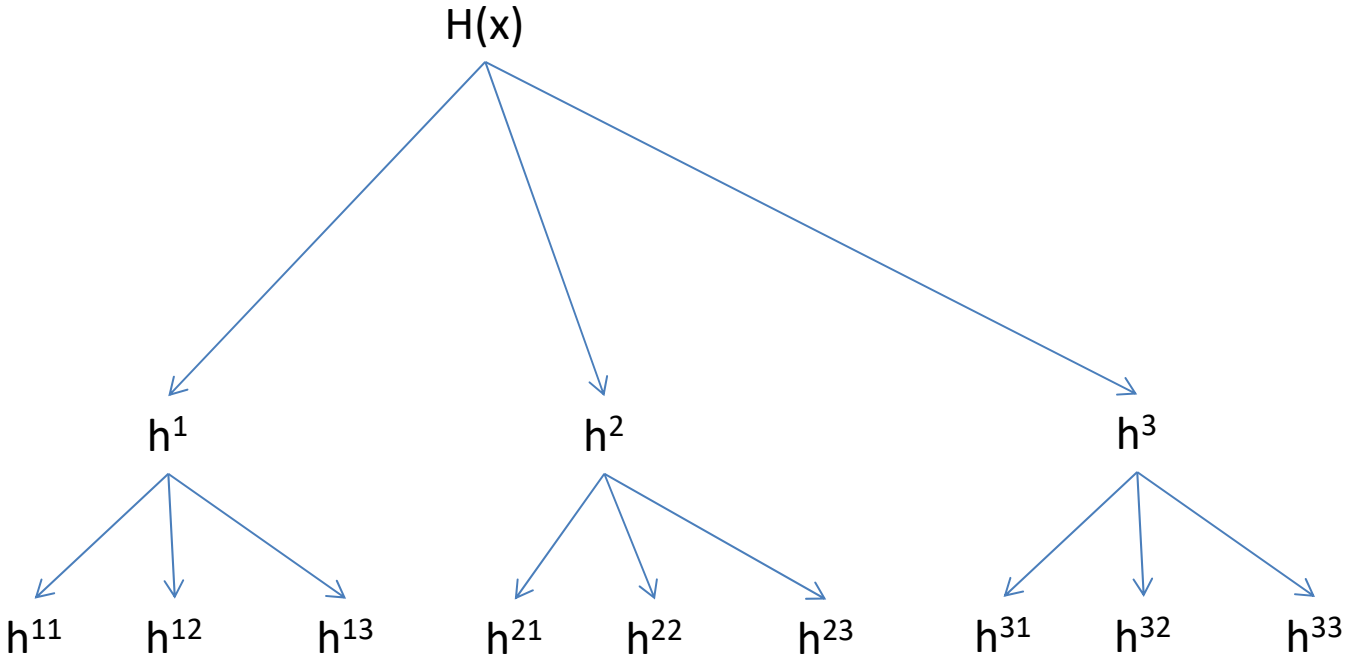


- Is the area overlapping by at least 2 classifiers sufficiently smaller than the area covers by each individual tests for wrong cases?

Idea #1

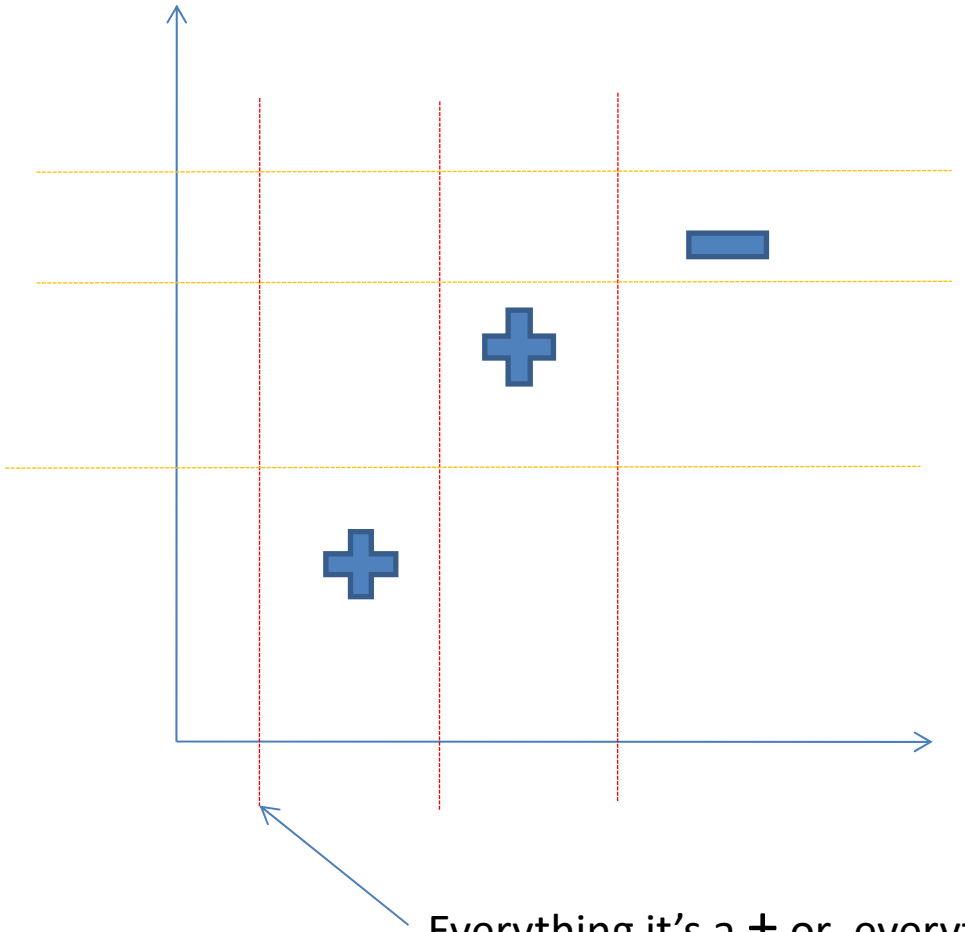
- We use undisturbed DATA to produce h^1
- We use DATA with an exaggeration of h^1 errors (disturbed set of data) to produce h^2
- We use DATA with an exaggeration of data where h^1 give a different answer than h^2 to produce h^3

Idea #2



- Get out the vote

Idea #3 – Example of classifiers



For each horizontal stump we could have 2 cases:

Up + , Down –

or

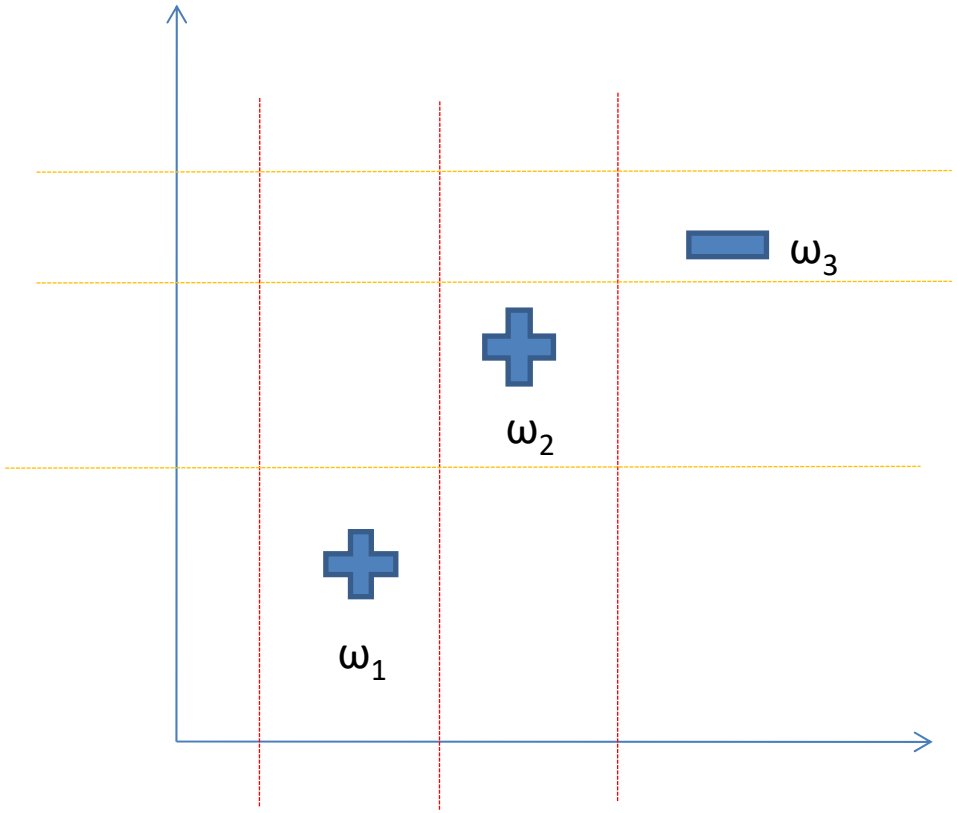
Up - , Down +

Similar for vertical stumps:

Left – Right.

Everything it's a + or everything it's a - . It's an extra test.

- Decision tree stumps – a single test
- Could be 12 decision tree stumps: for each dimension we have # of lines * 2 (we have 2 dimensions: 2*3*2=12)



$$Error = \sum_{\substack{WRONG \\ CASES}} \frac{1}{N}$$

$N -$ # of cases

In the beginning:

$$\omega_i^1 = \frac{1}{N}$$

- Weighted the samples

$$Error^t = \sum_{\substack{i-WRONG \\ CASES}} \omega_i^t$$

- Enforced a distribution

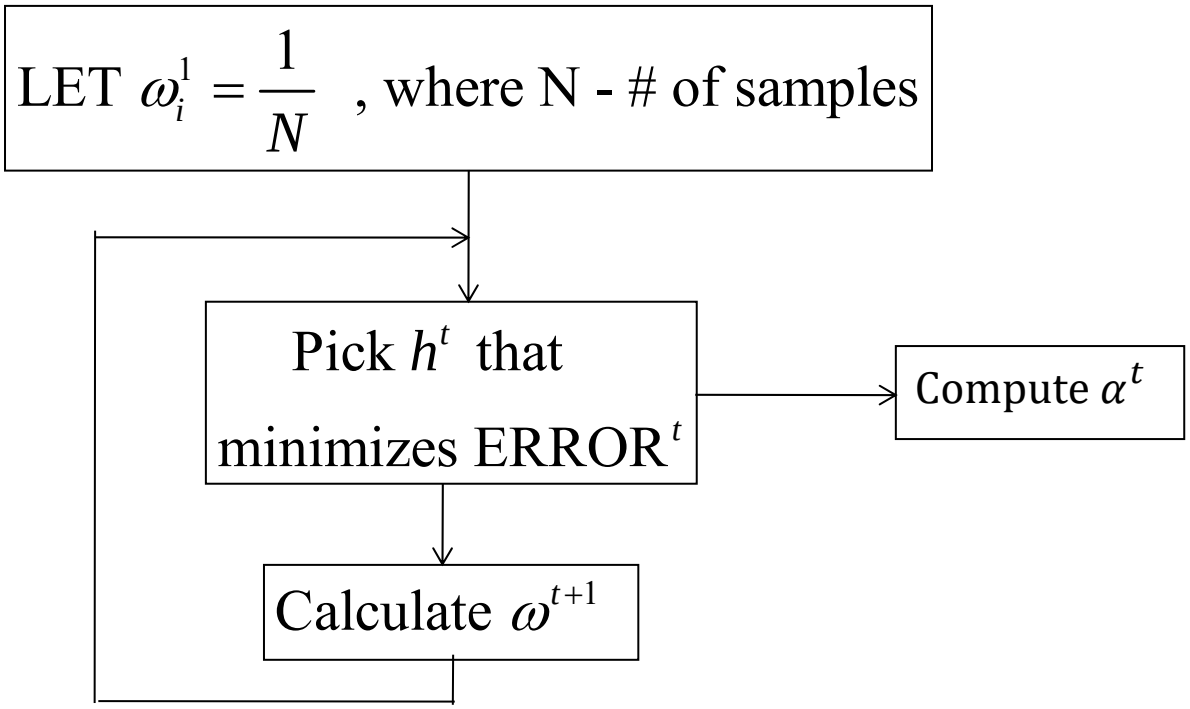
$$\sum_{\substack{ALL \\ CASES}} \omega_i^t = 1$$

Idea #4

$$H(x) = sign\left(\alpha^1 h^1(x) + \alpha^2 h^2(x) + \alpha^3 h^3(x) + ...\right)$$

- Build a classifier in multiple steps
- We don't treat equally each one on the crowd -> wisdom of weighted crowd of experts

Idea #5


LET $\omega_i^1 = \frac{1}{N}$, where N - # of samples
Pick h^t that minimizes ERROR^t
Compute α^t
Calculate ω^{t+1}

Idea #6

- Suppose that: $\omega_i^{t+1} = \frac{\omega_i^t}{Z} e^{-\alpha^t h^t(x) y(x)}$

$$h(x) = \begin{cases} +1 \text{ for samples the classifier thinks belongs to the class} \\ -1 \text{ for samples that the classifier thinks do not belong to the class} \end{cases}$$

$y(x) \in \{+1, -1\}$ - the desired output

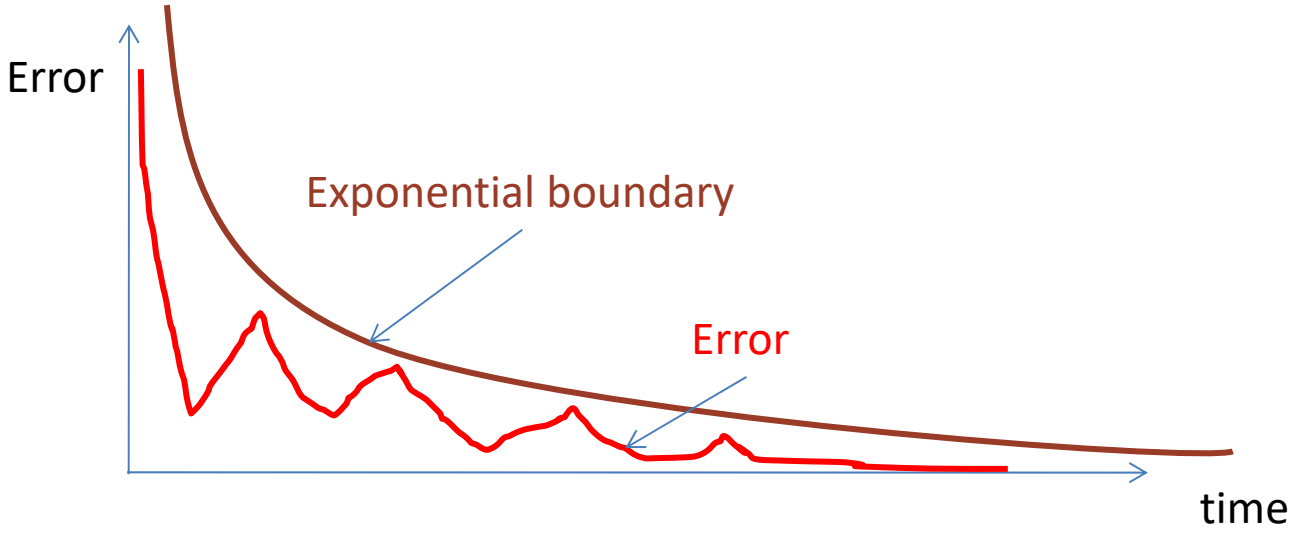
Z - the normalizer, in order to have a distribution

Minimize the error

- The error BOUND is minimized for the whole #4 if:

$$\alpha^t = \frac{1}{2}\ln\frac{1-E^t}{E^t}, \text{ where } E \text{ is the ERROR at time } t$$

- The error will be bounded by an exponential decay function
- It's guaranteed to converge on 0



Ada Boost

- You use uniform weights to start.
- For each step, you find the classifier that yields the lowest error rate for the current weights, w_i^t
- You use that best classifier, $h^t(x_i)$, to compute the error rate associated with the step, E^t
- You determine the alpha for the step, α^t from the error for the step, E^t .
- With the alpha in hand, you compute the weights for the next step, w_i^{t+1}, from the weights for the current step, w_i^t , taking care to include a normalizing factor, Z^t , so that the new weights add up to 1.
- You stop successfully when $H(x_i)$ correctly classifies all the samples, x_i; you stop unsuccessfully if you reach a point where there is no weak classifier, one with an error rate < 1/2 .

Change the weights

$$\omega_i^{t+1} = \frac{\omega_i^t}{Z} \begin{cases} \sqrt{\frac{E^t}{1-E^T}} & \text{if it's correct} \\ \sqrt{\frac{1-E^t}{E^T}} & \text{if it's wrong} \end{cases}$$

$$Z = \sqrt{\frac{E^t}{1-E^t}} \sum_{CORRECT} \omega_i^t + \sqrt{\frac{1-E^t}{E^t}} \sum_{WRONG} \omega_i^t$$

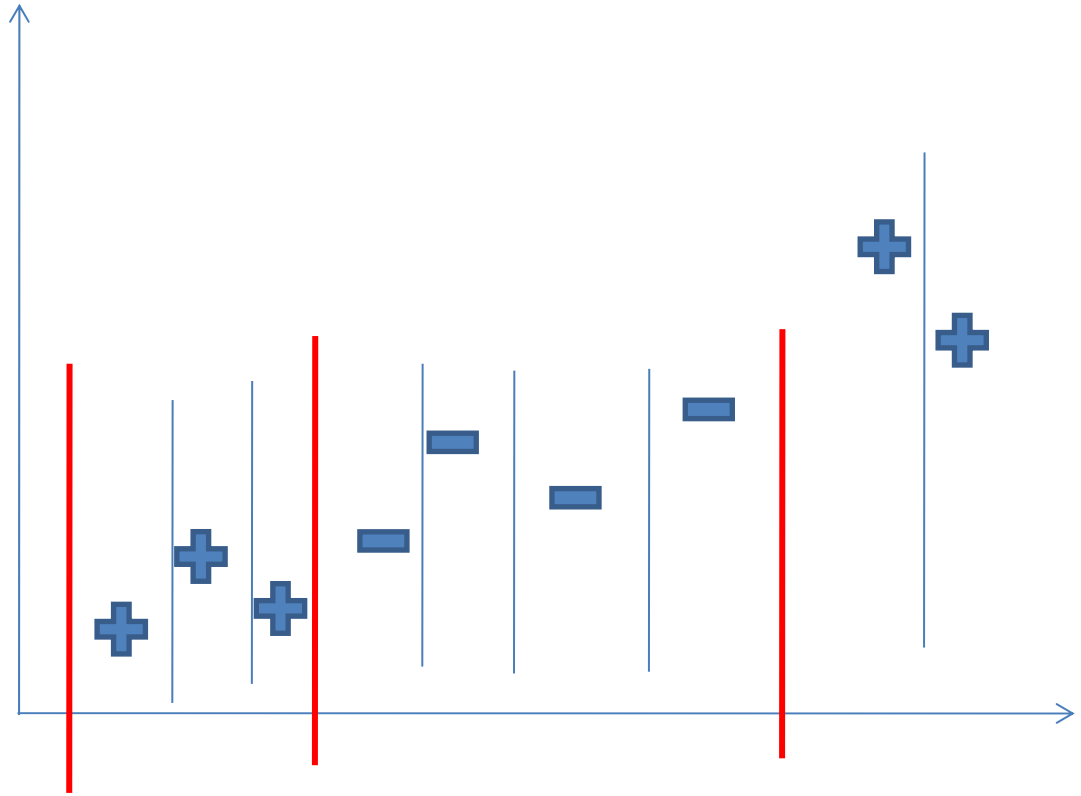
$$= \sqrt{\frac{E^t}{1-E}}(1-E^t) + \sqrt{\frac{1-E^t}{E^t}} E^t = 2\sqrt{E^t(1-E^t)}$$

$$\omega_i^{t+1} = \frac{\omega_i^t}{2} \begin{cases} \frac{1}{1-E^t} & \text{if it's correct} \\ \frac{1}{E^t} & \text{if it's wrong} \end{cases}$$

$$\sum_{CORRECT} \omega_i^{t+1} = \frac{1}{2}\frac{1}{1-E^t} \sum_{CORRECT} \omega^t = \frac{1}{2}$$

$$\sum_{WRONG} \omega_i^{t+1} = \frac{1}{2}$$

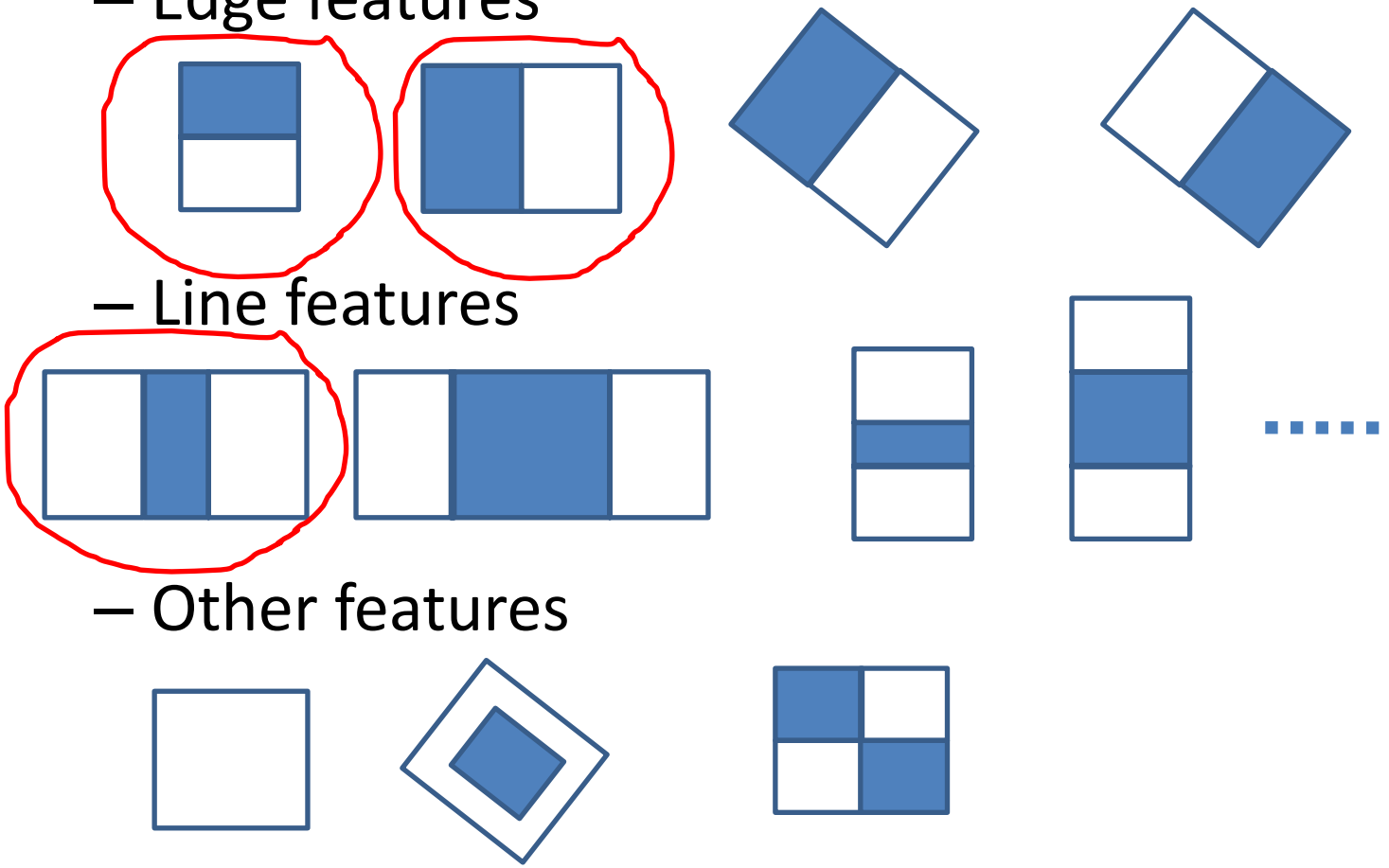
Improvements



- Tests that really matter
- Immune to overfitting

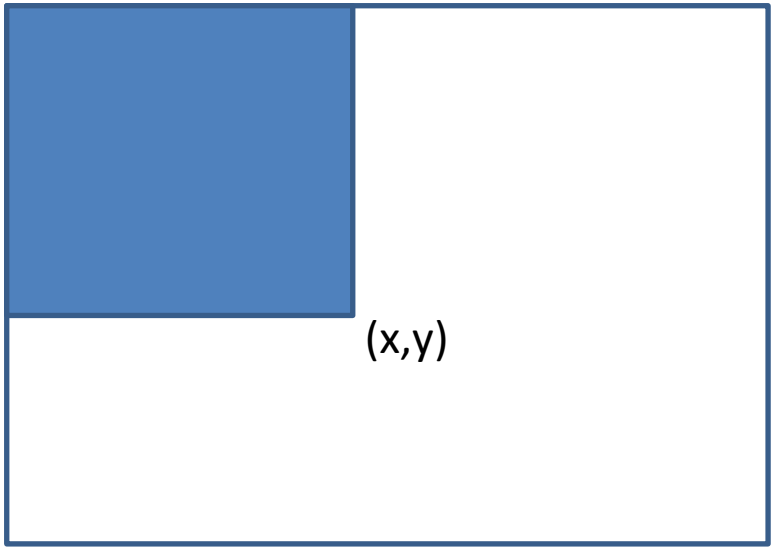
Face detection

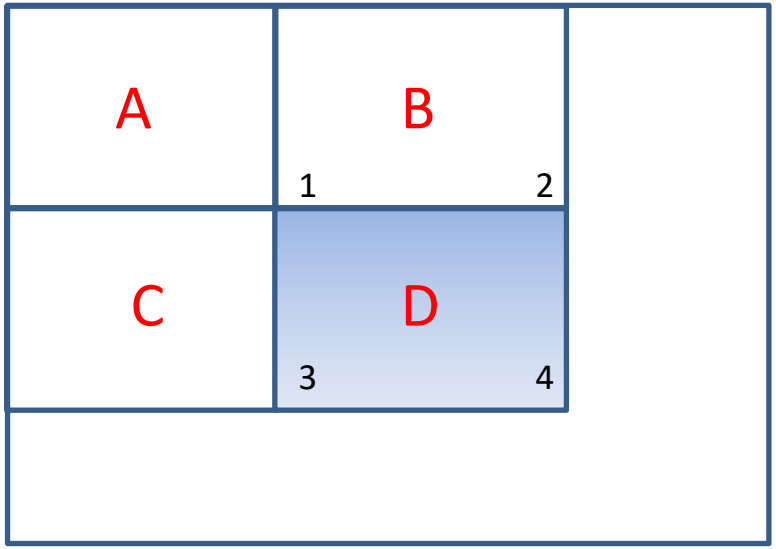
- Haar-like features
  - Edge features
  - Line features
  - Other features



Face detection

- Integral image – each pixel (x,y) is the sum of all pixels above and left of *x,y* applied to original image





$$ii(x,y) = \sum_{x' \le x, y' \le y} i(x', y'),$$

where $ii(x,y)$ – integral image

$i(x,y)$ – original image

$$v1 = ii(location\_1) = \sum_{A} i$$

$$v2 = ii(location\_2) = \sum_{A} i + \sum_{B} i$$

$$v3 = ii(location\_3) = \sum_{A} i + \sum_{C} i$$

$$v4 = ii(location\_4) = \sum_{A} i + \sum_{B} i + \sum_{C} i + \sum_{D} i$$

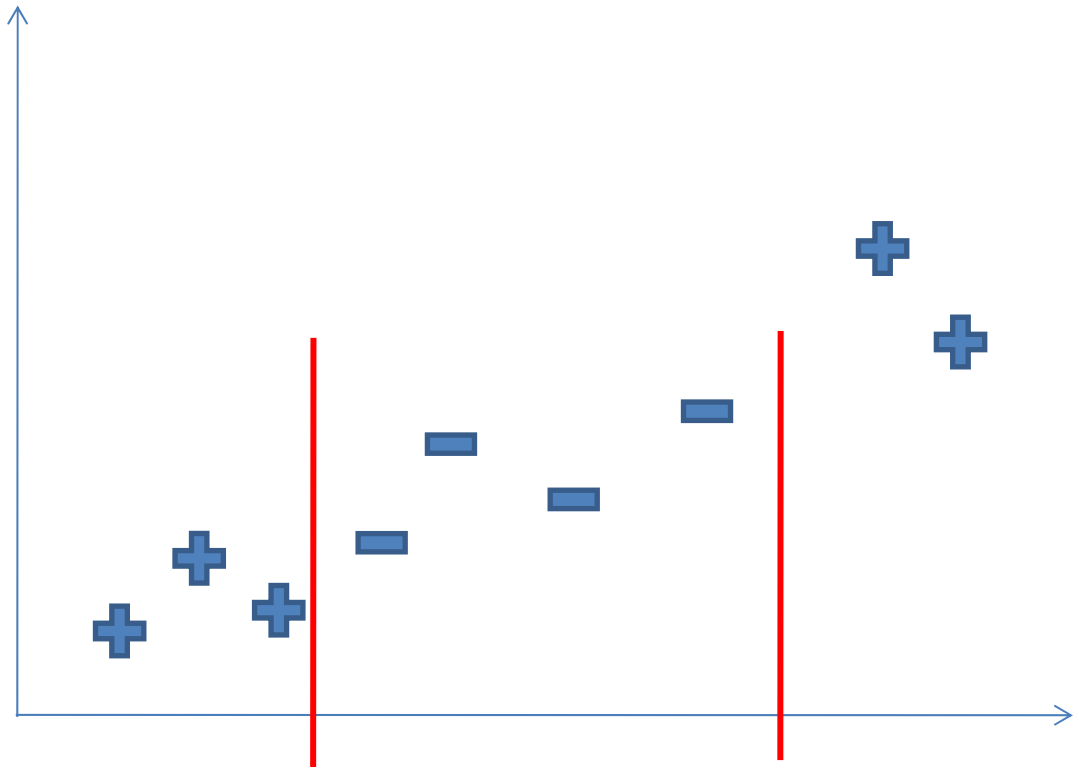
$$rect(D) = v1 + v4 - v2 - v3$$

Face detection

- For 24x24 pixels image – 162.336 features

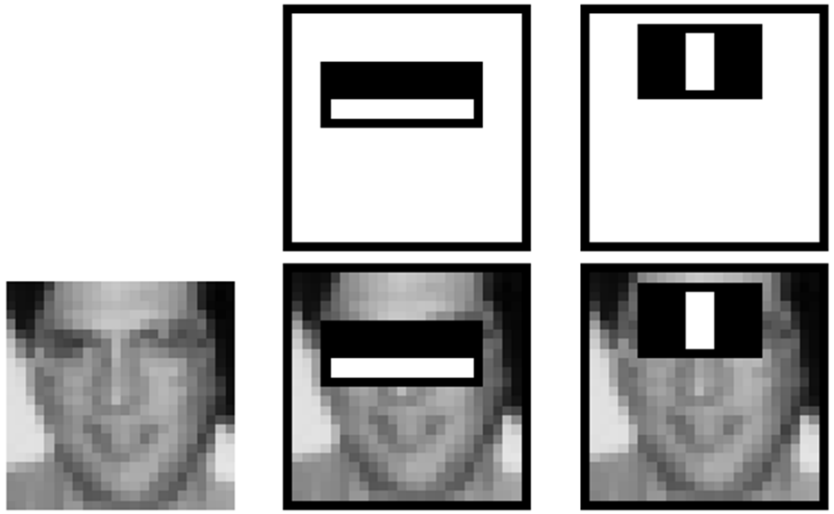
Face detection

- Choosing the threshold for each classifier



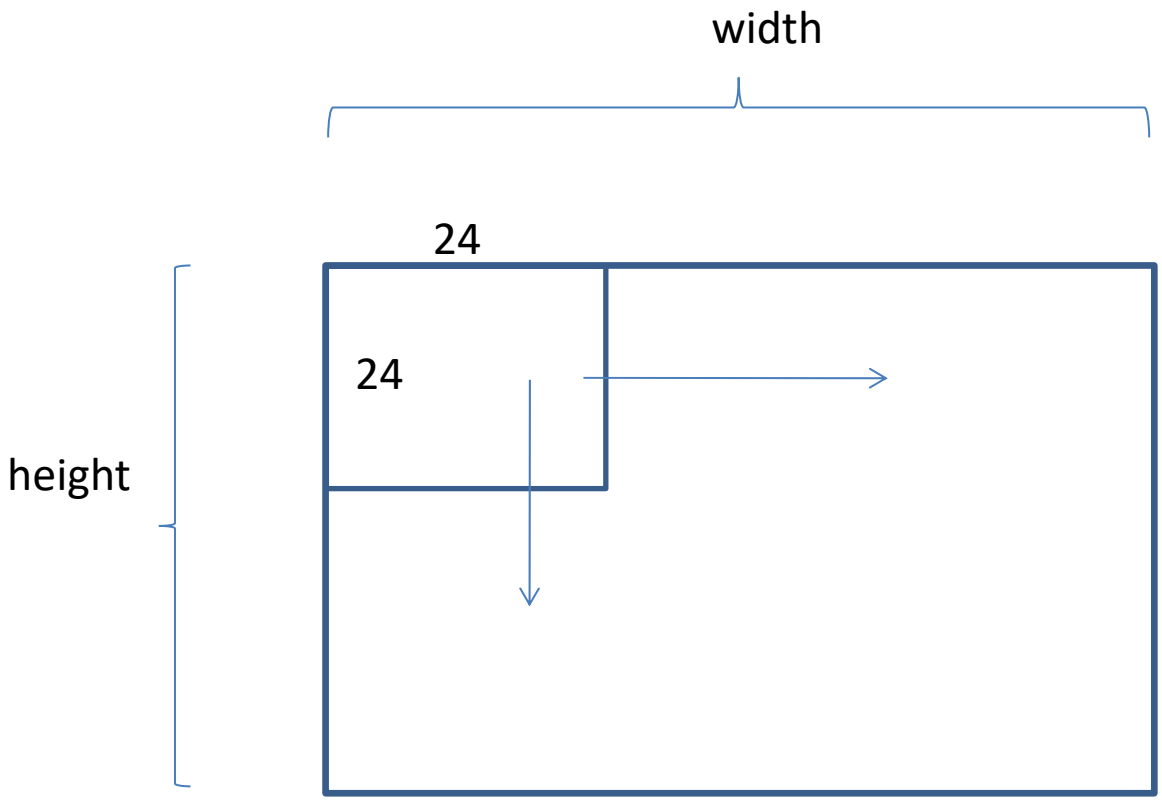
Face detection

- The first and second features selected by AdaBoost



Face detection

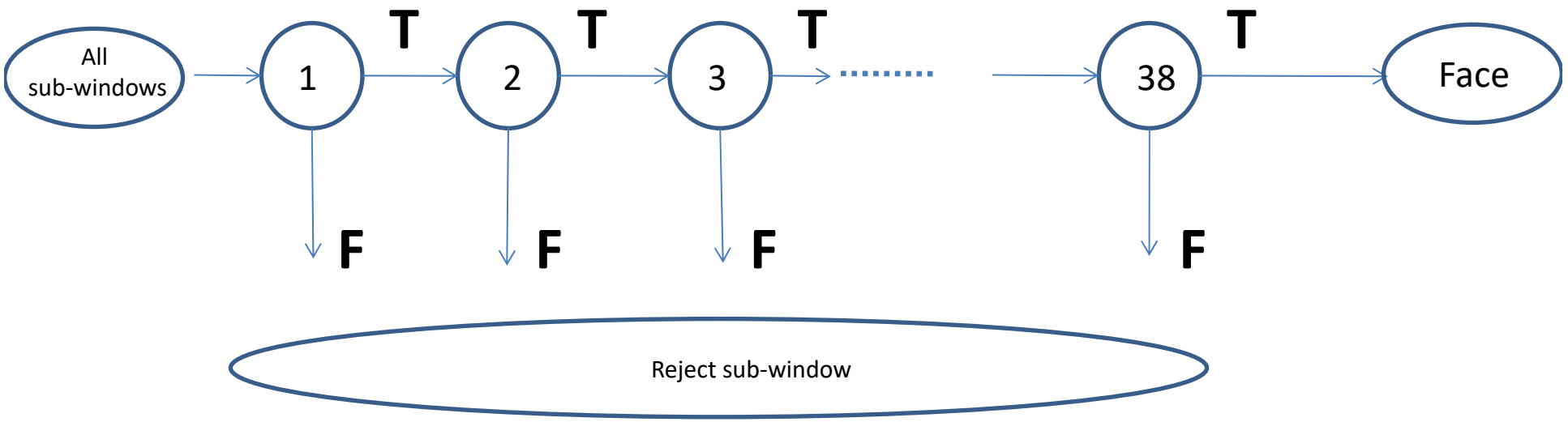
- Sub-window – 24x24 pixels



Starts from top-left corner and go to left 1 pixel at a time. When reach the end of the row, go down 1 pixel and start again from the left.

Face detection

- Cascade of classifier
  - #of features in the first 5 layers: 1, 10, 25, 25 and 50
  - total # of features in all layers - 6061



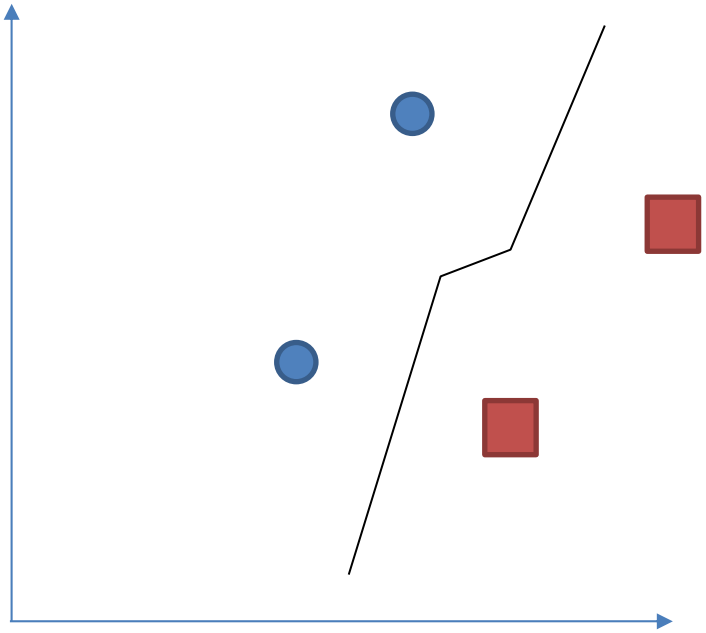
Related resources

- P. Viola, M. Jones, “Robust Real-Time Face Detection”, http://www.vision.caltech.edu/html-files/EE148-2005-Spring/pprs/viola04ijcv.pdf

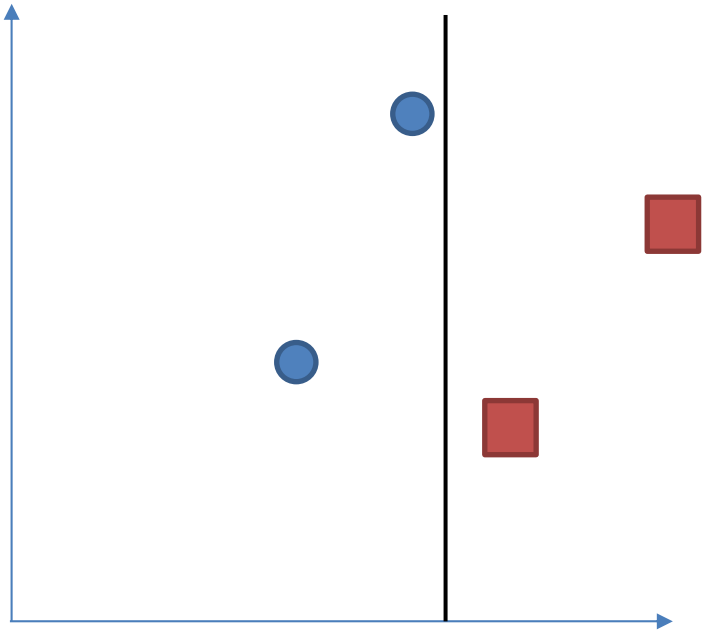
Artificial Intelligence Fundamentals

Learning: SVM

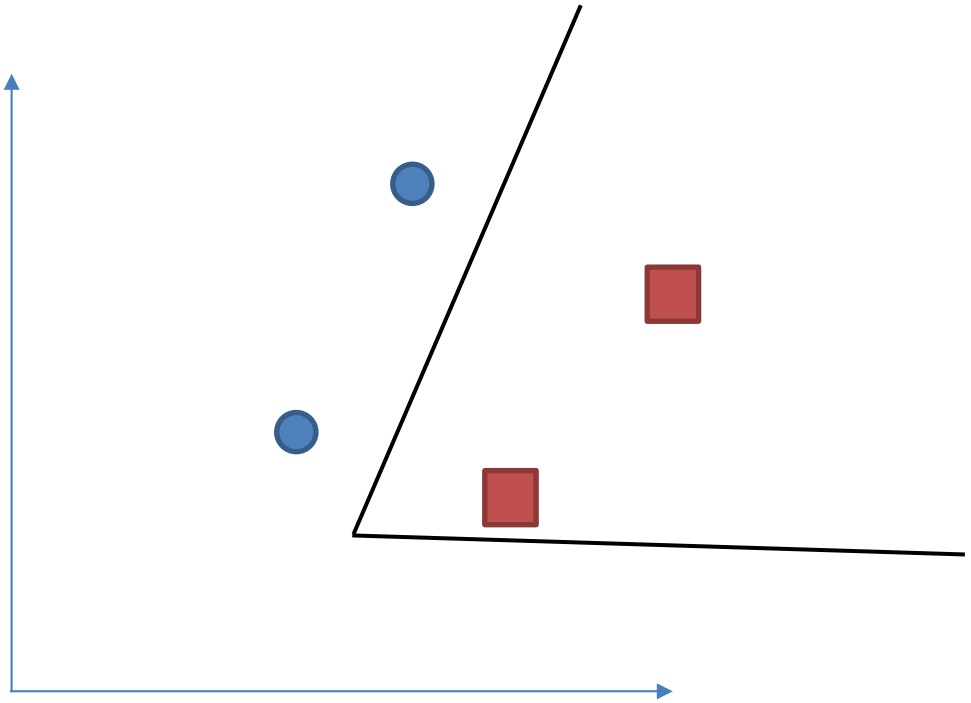
Separate 2 classes - kNN



Separate 2 classes – Decision tree

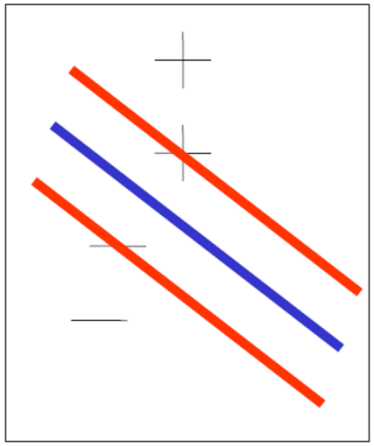


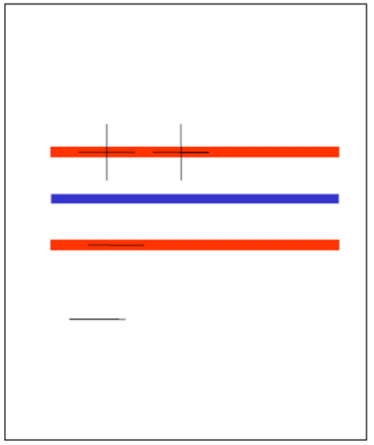
Separate 2 classes – NN



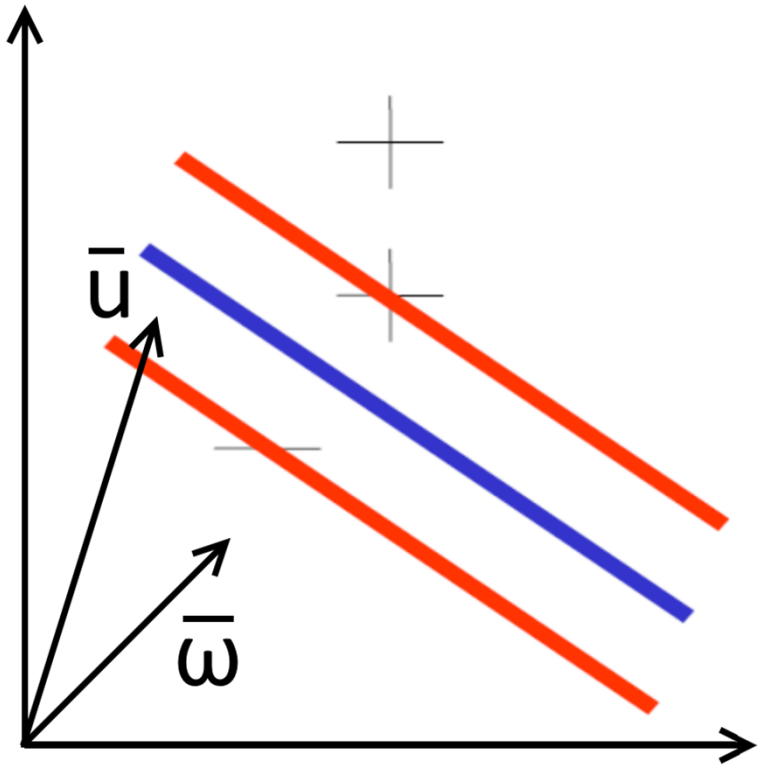
Widest street approach

- Divide a space with decision boundaries (NN, DT, kNN)
- A very sophisticated idea
- Vladimir Vapnik and Alexey Chervonenkis (1963)
- Separating the + from the – as wide as possible





Step 1



- Widest space that separate the + from –
- $\overline{\omega}$ - A vector perpendicular to the median line of that space (street)
- $\overline{u}$ - An unknown example
- Is on the right side of the street?
- Project the $\overline{u}$ on the $\overline{\omega}$
- DECISION RULE : we have a positive example if:

$$\overline{\omega} \bullet \overline{u} \geq c \; or$$
$$\overline{\omega} \bullet \overline{u} + b \geq 0$$

Step 2

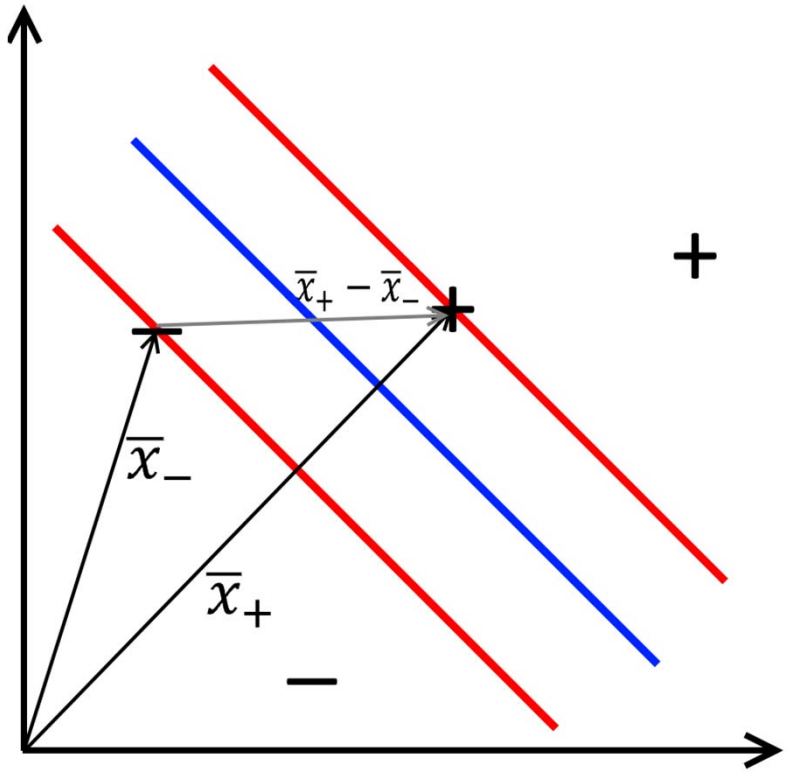
$$\bar{\omega} \cdot \overline{x_{i_+}} + b \geq 1$$
$$\bar{\omega} \cdot \overline{x_{i_-}} + b \leq -1$$

$$y_i(\bar{\omega} \cdot \bar{x_i} + b) \geq 1$$
$$y_i(\bar{\omega} \cdot \bar{x_i} + b) \geq 1$$

- Introducing a new variable for mathematical convenient
- y_i such that $y_i = +1$ for positive samples and -1 for negative samples

$$y_i(\bar{\omega} \cdot \bar{x_i} + b) - 1 \geq 0$$
$$y_i(\bar{\omega} \cdot \bar{x_i} + b) - 1 = 0 \text{ for } x_i \text{on the boundary}$$

Step 3 – width of the street



Width of the street is:

$$(\overline{x_+} - \overline{x_-}) \cdot \frac{\bar{\omega}}{\|\omega\|} =$$

$$\frac{1 - b + 1 + b}{\|\omega\|} = \frac{2}{\|\omega\|}$$

- Maximize that expression $\frac{2}{\|\omega\|}$, maximize $\frac{1}{\|\omega\|}$,

  minimize $\|\omega\|$, minimize $\frac{1}{2}\|\omega\|^2$

Step 4 – development

- Minimize with the help of Lagrange multipliers
- $L = \frac{1}{2}\|\overline{\omega}\|^2 - \sum \alpha_i [y_i(\overline{\omega} \cdot \overline{x_i} + b) - 1]$
- Derivatives and set it to 0

$$\frac{\partial L}{\partial \overline{\omega}} = \overline{\omega} - \sum_i \alpha_i y_i \overline{x_i} \qquad \overline{\omega} = \sum_i \alpha_i y_i \overline{x_i}$$

(Linear combination of the inputs)

$$\frac{\partial L}{\partial b} = -\sum_i \alpha_i y_i \qquad \sum_i \alpha_i y_i = 0$$

- $L = \frac{1}{2}\left(\sum_i \alpha_i y_i \overline{x_i}\right) \cdot \left(\sum_j \alpha_j y_j \overline{x_j}\right) - \sum_i \alpha_i y_i \overline{x_i} \cdot \left(\sum_j \alpha_j y_j \overline{x_j}\right) - \sum_i \alpha_i y_i b + \sum_i \alpha_i$

- maximum of the following expression

$$L = \sum_i \alpha_i - \frac{1}{2}\sum_i \sum_j \alpha_i \alpha_j y_i y_j \overline{x_i} \cdot \overline{x_j}$$

- optimization depends only on the dot product of pairs of samples

Step 5 – Switch to another perspective

$$\sum \alpha_i y_i \overline{x_i} \cdot \bar{u} + b \geq 0 \text{ then } +$$

- Decision rule depends only on the dot product of samples vectors and unknown
- ϕ - a transformation that will take us from the space we're in into a space where things are more convenient

$$\Phi(\overline{x_i}) \cdot \Phi\left(\overline{x_j}\right) \text{ to maximize}$$

- Find a transformation K - *kernel*

$$K\left(\overline{x_i}, \overline{x_j}\right) = \Phi(\overline{x_i}) \cdot \Phi\left(\overline{x_j}\right)$$

- I don’t need to know the transformation into another space, just to know the K – the dot product of transformations into another space

$$(\bar{u} \cdot \bar{v} + 1)^n - polinomial$$

$$e^{-\frac{(\bar{u}-\bar{v})}{\sigma}} - gaussian$$

Conclusions

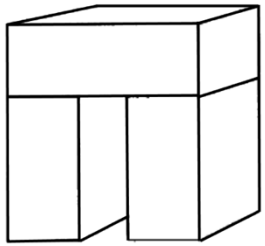
- Produces global solutions
- Kernel functions – convex
- Immune against local minima
- Not immune against overfitting

Artificial Intelligence Fundamentals

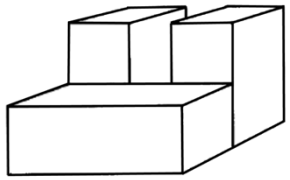
Learning: Near misses, Felicity Conditions

A different type of learning

- Learning by analyzing the differences that appear in a sequence of observations
- Learning in a human-like way from a single example in one shot
- Learning something definite from every example
- How can that happen ?

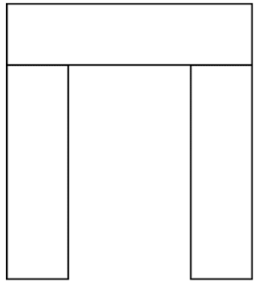


- Arch
- Got a general idea
- What's important ?

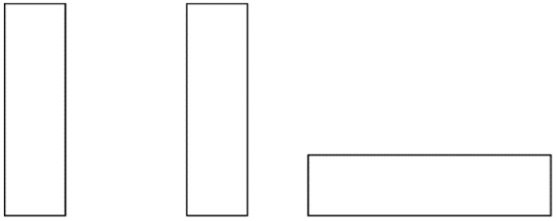


- Not an arch
- Learn something very different

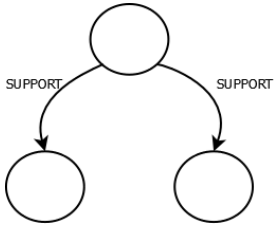
Step 1 – Require link heuristic (specialization)

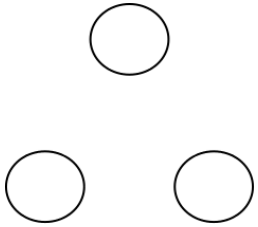


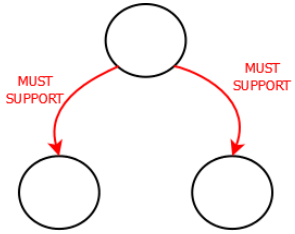
- Initial model
- Is the starting point



- Near miss

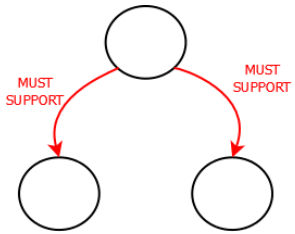




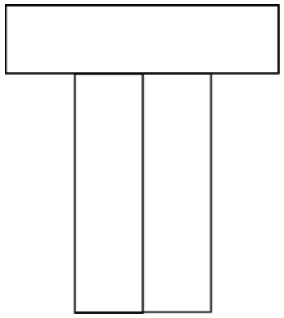


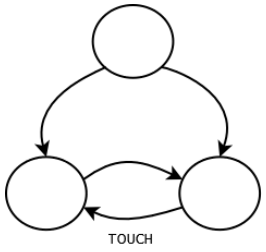
- only 1 difference
- Support relations are important

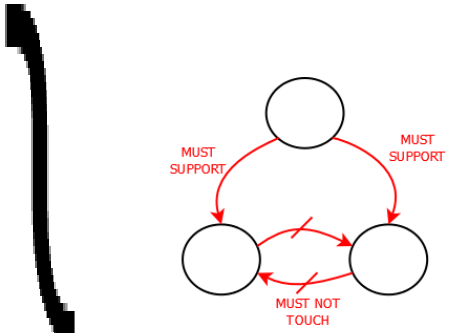
Step 2 – Forbid link heuristic (specialization)



- Evolving model



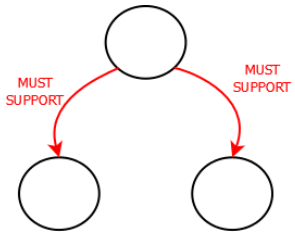




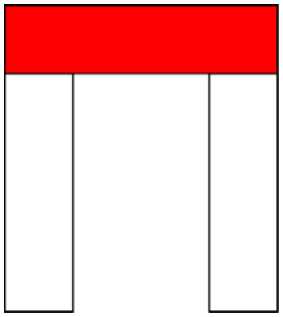
- only 1 difference
- Not touch relations are important

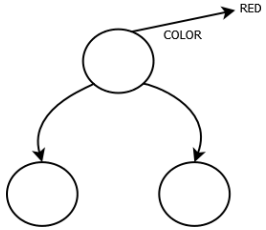
- Near miss

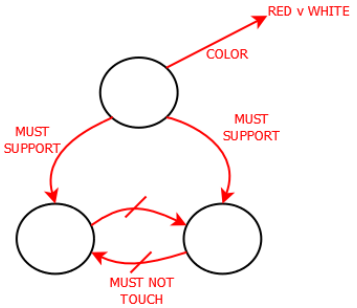
Step 3 – Extend set heuristic (generalization)



- Evolving model
- Color of the top is white



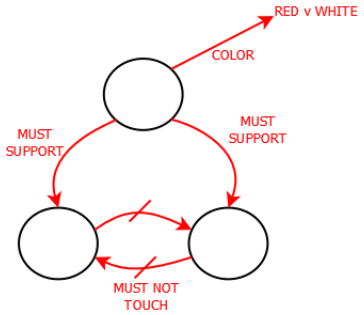




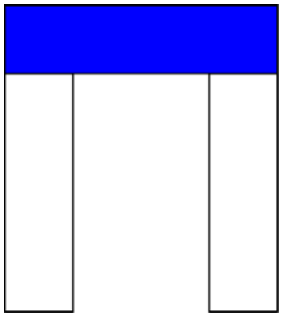
- only 1 difference
- The color relation is important

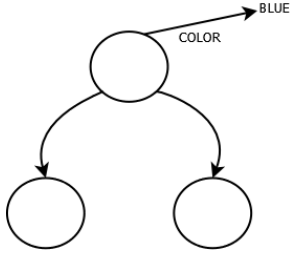
- Example

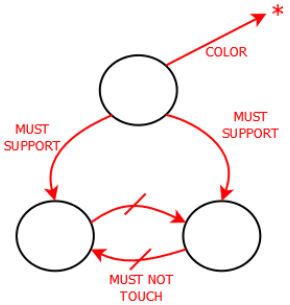
Step 4 – Drop link heuristic (generalization)



- Evolving model



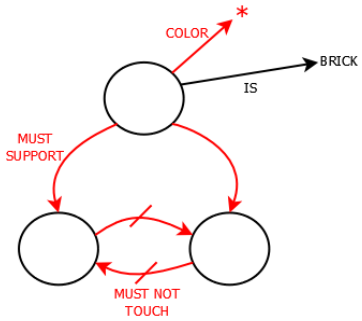




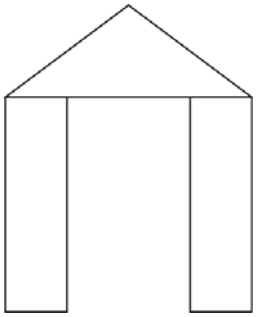
- only 1 difference
- The color relation – anything can be here

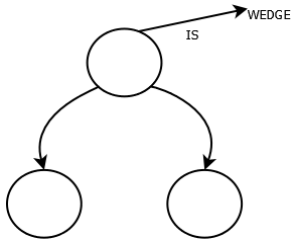
- Example

Step 5 – Climb tree heuristic (generalization)

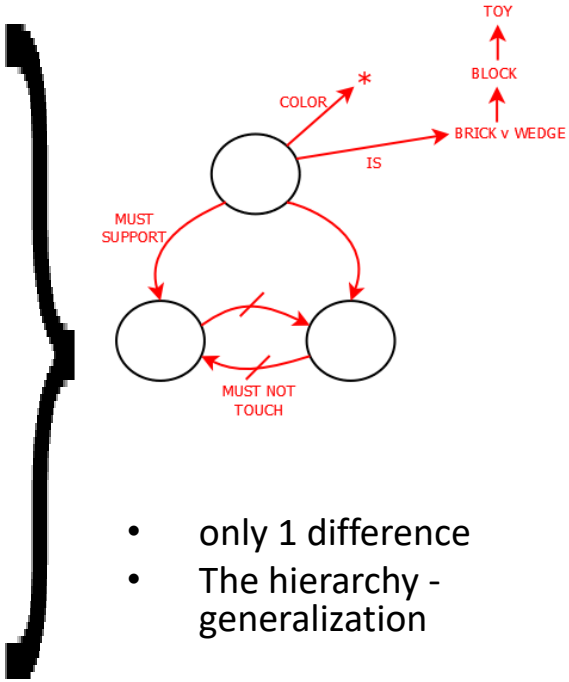


- Evolving model





- Example

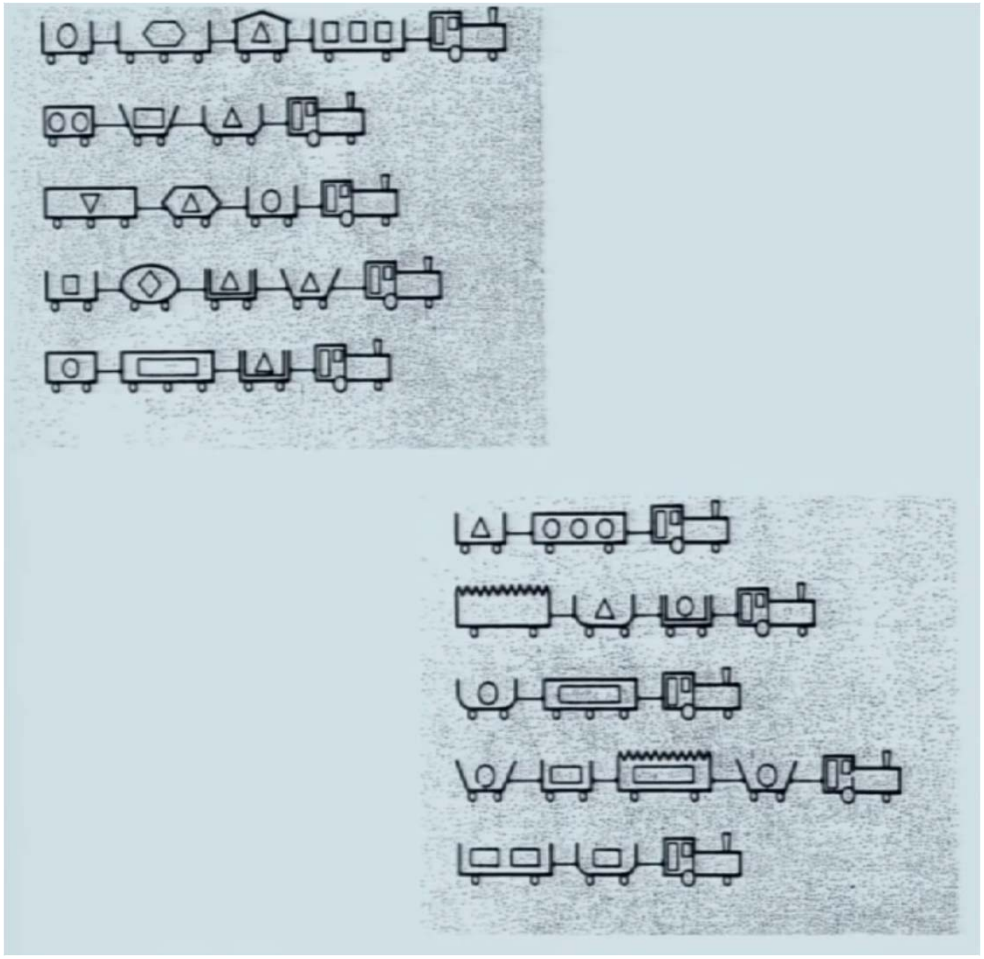


- only 1 difference
- The hierarchy - generalization

Deploying the heuristics

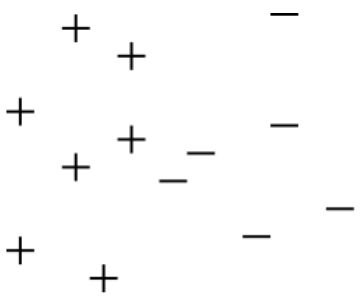
- Examples are presented one at a time by a teacher who's eager for the student to learn
  - Combine generalization with specialization
  - Require a teacher
  - Need to do with a human (don't have much memory)
- If the examples are presented all at once:
  - Pick one of the positive examples to work (SEED) – it's an exactly description of a particular thing
  - Search for one heuristic that loosens the description so that it covers more of the positives (expand the description)
  - It cover 2 negative examples
  - Continue until eliminate all negative examples and keep only the positive ones
  - Beam search
  - Can be done in batches
  - Computer is very good at this (lots of memory)

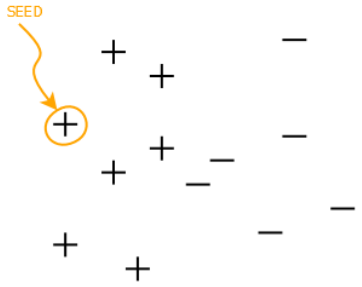
Distinguish the top from bottom

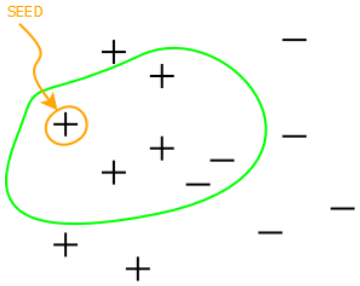


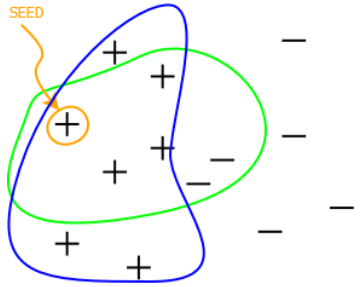
The top trains all have a short car with a closed top.

All at once deploying





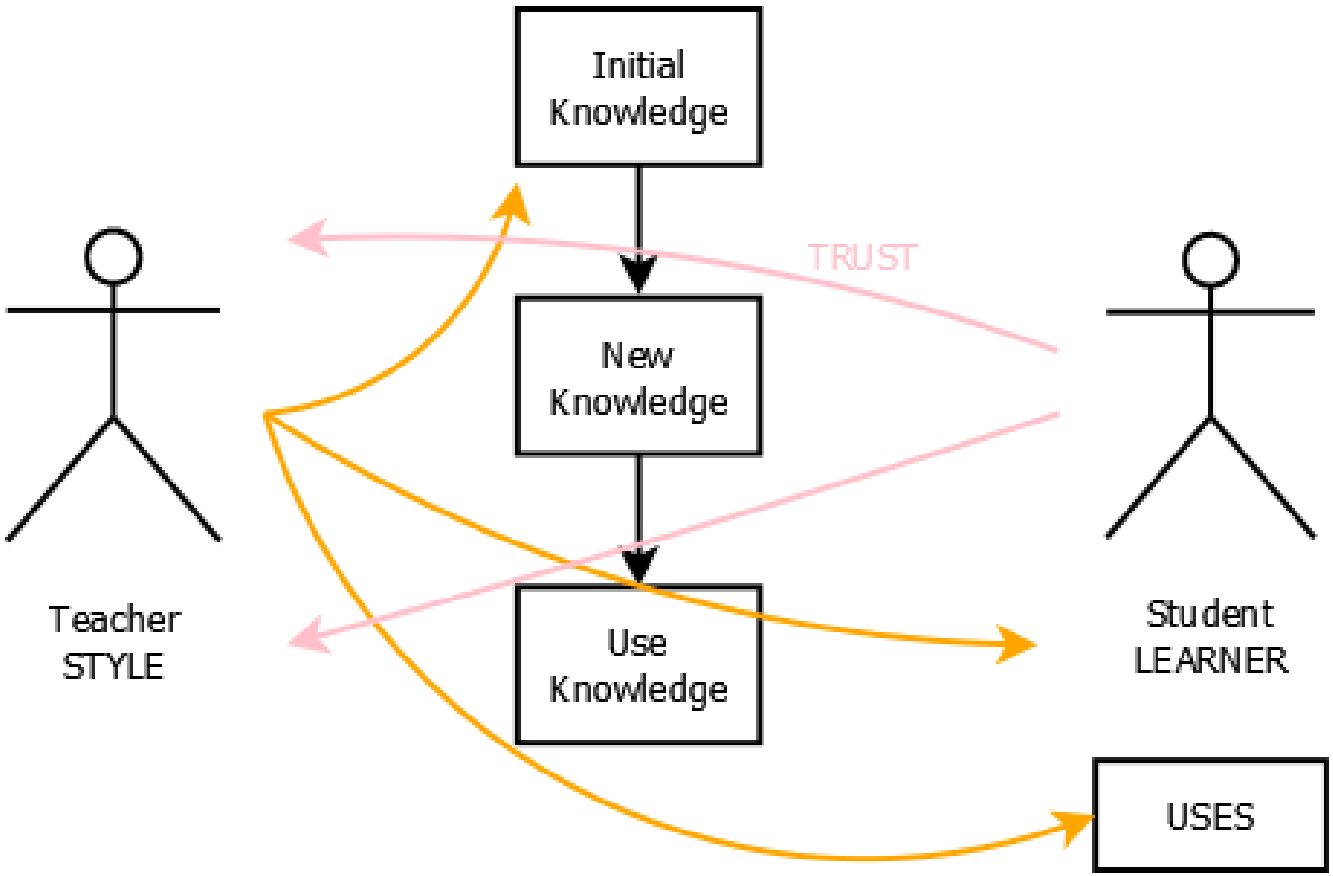




Beam search.

Felicity conditions

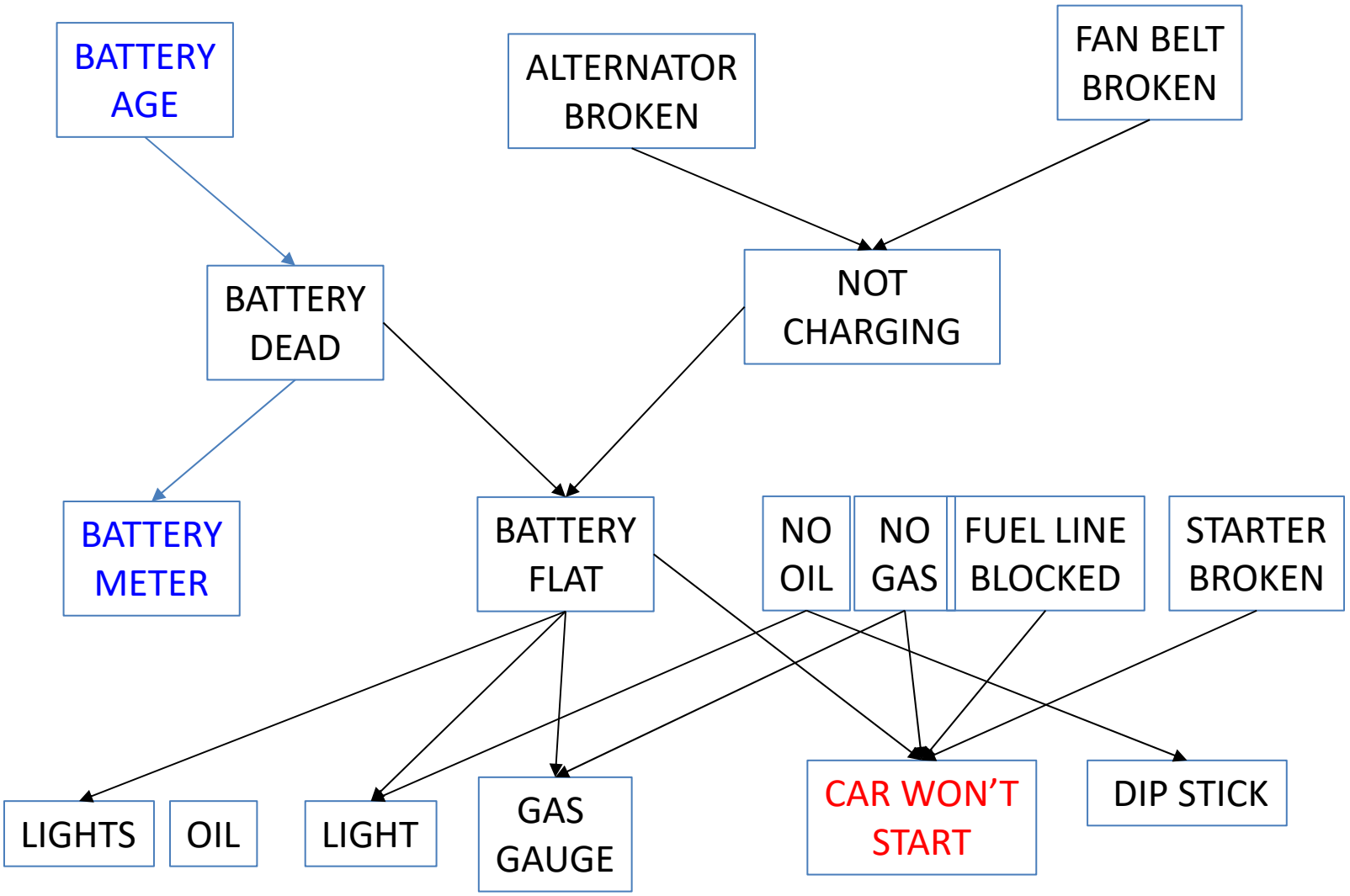
- Require a teacher, a student and covenants that hold between them
- Transform an initial state of knowledge into a new state of knowledge -> the student is smarter
- For learning – one side has to know something about the other side:
  - Teacher
    - Initial state of knowledge
    - The way that the student learns
    - The computational capacity & the uses of knowledges
  - Learner
    - Trust


Initial
Knowledge
TRUST
New
Knowledge
Use
Knowledge
Teacher
STYLE
Student
LEARNER
USES

Artificial Intelligence Fundamentals

Learning: Bayes Network

Example – Bayes Network



Can we diagnose the problem?

- Bayes Network
  - Composed of nodes
  - Nodes correspond to events
  - You might or might not know the events (random variables)
  - Nodes are linked by arcs
  - Arcs suggest that a child node is influenced (in a probabilistic way) by its parent
  - 16 variables
  - The space is 2^{16} possible values
- The Bayes Network is a compact representation of a distribution over this very large joint probability distribution of all of these variables
  - Once we specify it, we can observe (e.g., the car won't start and the lights, etc.) and compute probability (like the alternator is broken)

Bayes Network

Used extensively in almost all fields of smart computer systems

diagnostics

Google

finance

prediction

robotics

machine learning

Building blocks of more advanced AI techniques

Particle Filters

Hidden Markov Models

MDP’s and POMDP’s

Kalman Filters

many others

Probabilities

- Used to express uncertainty
- Coin
  - Probability for heads is 0.5 $P(H) = \frac{1}{2}$
  - Probability for tails is $P(T) = \frac{1}{2}$
  - $P(T) + P(H) = 1$
  - Suppose $P(T) = \frac{1}{4}$, $P(H) = \frac{3}{4}$
  - Suppose $P(H) = \frac{1}{2}$, the probability having three heads into a row will be: $P(H, H, H) = \frac{1}{2} \cdot \frac{1}{2} \cdot \frac{1}{2} = \frac{1}{8}$, the coin flips are independent

Probabilities

- Flip the coin 4 times
  - X_i=result of i-*th* coin flip $X_i = \{H, T\}, P_i(H) = \frac{1}{2}$
  - $P(X_1 = X_2 = X_3 = X_4) =?$
  - 1/16 + 1/16 = 1/8

  - $P(X_1X_2X_3X_4\ contains\ at\ least\ 3H) =?$
  - 1/16+1/16+1/16+1/16+1/16=5/16

Probabilities

- Basic
  - $0 \leq P(A) \leq 1$
  - $P(True) = 1$ and $P(False\ ) = 0$
- Complementary probability (for binary event)
  - $P(A) = p \Rightarrow P(\neg A) = 1 - p$
- Independence, X independent of Y $(X \perp Y)$
  - $X \perp Y : P(X)P(Y) = P(X, Y)$
    - P(X),P(Y) – marginals
    - P(X,Y) – joint probability
  - $X \perp Y : P(X|Y) = P(X)$

Dependence

- $P(X_1 = H) = \frac{1}{2}, \begin{cases} H: P(X_2 = H|X_1 = H) = 0.9 \\ T: P(X_2 = T|X_1 = T) = 0.8 \end{cases}$
- $P(X_2 = H)$=?
- $P(X_2 = H) = P(X_2 = H|X_1 = H) \cdot P(X_1 = H) + P(X_2 = H|X_1 = T) \cdot P(X_1 = T) = 0.9 \cdot 0.5 + 0.2 \cdot 0.5 = 0.45 + 0.1 = 0.55$

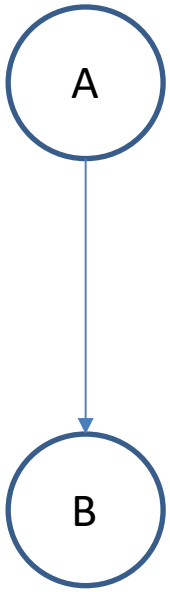
- Total probability:
  - $P(Y) = \sum_i P(Y|X = i) \cdot P(X = i)$
- Negation of probabilities
  - $P(\neg X|Y) = 1 - P(X|Y)$
- Joint probability
  - $P(X, Y) = P(X|Y) \cdot P(Y)$
- Chain rule
  - $P(X_n, X_{n-1}, \dots, X_1) = \prod_{i=n}^{1} P(X_i|X_{i-1}, \dots, X_1) = P(X_n|X_{n-1}, X_{n-2}, \dots, X_1)P(X_{n-1}|X_{n-2}, \dots, X_1) \dots P(X_2|X_1)P(X_1)$

Bayes Rule

- $P(A|B) = \frac{P(B|A)\cdot P(A)}{P(B)}$
- Cancer example:
  - $P(C) = 0.01\,, P(\neg C) = 0.99$
  - $P(+|C) = 0.9\ ,\ P(-|C) = 0.1$
  - $P(+|\neg C) = 0.2\ ,\ P(-|\neg C) = 0.8$
  - $P(C|+) = \frac{P(+|C)\cdot P(C)}{P(+)} = \frac{P(+|C)\cdot P(C)}{P(+|C)\cdot P(C)+P(+|\neg C)\cdot P(\neg C)}$

$$= \frac{0.9 * 0.01}{0.9 * 0.01 + 0.2 * 0.99} = 0.0434$$

Bayes Rule – graphical representation



P(A)

P(B|A)
P(B| ¬A)

- B – observable but A not observable
- Diagnostic Reasoning: P(A|B) or P(A| ¬B)

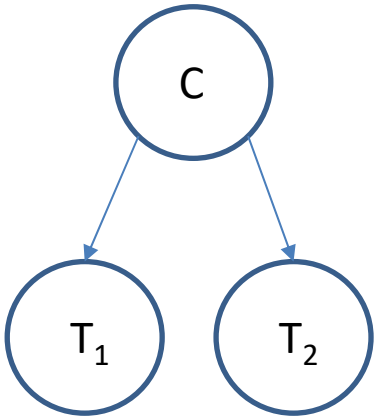
Bayes Rule trick

- $P(A|B) = \frac{P(B|A) \cdot P(A)}{P(B)}$, what happens if I don't know the P(B) ?
- $P(\neg A|B) = \frac{P(B|\neg A) \cdot P(\neg A)}{P(B)}$
- The normalizer P(B) is identical
- $P(A|B) + P(\neg A|B) = 1$
- $P'(A|B) = P(B|A) \cdot P(A)$
- $P'(\neg A|B) = P(B|\neg A) \cdot P(\neg A)$
- $P(A|B) = \eta P'(A|B)$
- $P(\neg A|B) = \eta P'(\neg A|B)$
- $\eta = \frac{1}{P\prime(A|B) + P\prime(\neg A|B)}$

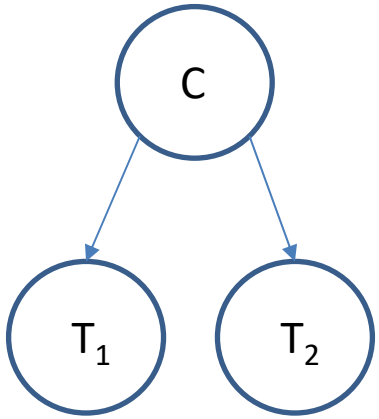
Bayes Rules – cancer example

- $P(C) = 0.01\,, P(\neg C) = 0.99$
- $P(+|C) = 0.9\ ,\ P(-|C) = 0.1$
- $P(+|\neg C) = 0.2\ ,\ P(-|\neg C) = 0.8$
- $P(C|T_1 = +, T_2 = +) = P(C|++) = ?$

	Prior	+	+	P’	P(C|++)
C	0.01	0.9	0.9	0.0081	0.1698
¬C	0.99	0.2	0.2	0.0396	
				0.0477	

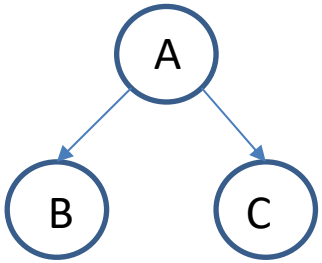


Conditional Independence



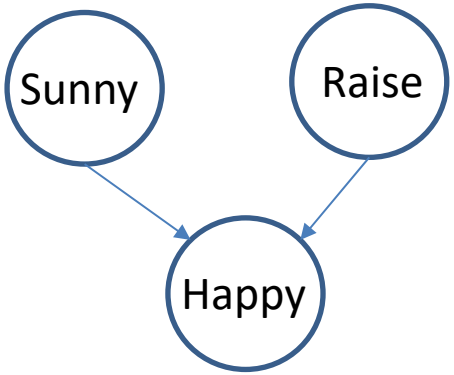
- T_1 and T_2 are conditionally independent
- Knowing anything about T_1 would not help us make a statement about T_2
- $P(T_2|C,T_1) = P(T_2|C)$
- The independence only holds true if we know C (see the diagram)

Conditional Independence



- Given A, $B \perp C$ are independent , written as $B \perp C \mid A$
- B and C are not independent if we don't know A
- $P(T_2 = +|T_1 = +)$=?
- $P(T_2 = +|T_1 = +) = P(+_2|+_1, C)P(C|+_1) + P(+_2|+_1, \neg C)P(\neg C|+_1) = P(+_2|C)P(C|+_1) + P(+_2|\neg C)P(\neg C|+_1)$=0.9*0.043+0.2*0.957=0.2301

Different type of Bayes Network



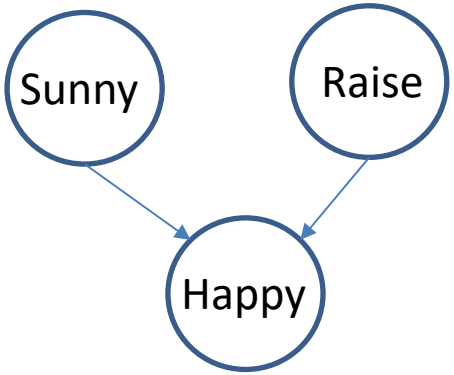
- P(S)=0.7 P(R)=0.01

- P(H|S,R)=1
- P(H| ¬S,R)=0.9
- P(H|S, ¬ R)=0.7
- P(H| ¬S, ¬ R)=0.1

- P(R|S)=?
- P(R|S)=P(R)=0.01

- P(R|H,S)=?
- $P(R|H,S) = \frac{P(H|R,S)\cdot P(R|S)}{P(H|S)} = \frac{P(H|R,S)\cdot P(R)}{P(H|S,R)\cdot P(R)+P(H|S,\neg R)\cdot P(\neg R)} = 0.0142$
- $P(R|H) = \frac{P(H|R)\cdot P(R)}{P(H)} = \frac{[P(H|R,S)\cdot P(S)+P(H|R,\neg S)\cdot P(\neg S)]\cdot P(R)}{P(H|R,S)\cdot P(R,S)+P(H|\neg R,S)\cdot P(\neg R,S)+P(H|R,\neg S)\cdot P(R,\neg S)+P(H|\neg R,\neg S)\cdot P(\neg R,\neg S)} = 0.0185,\ \ P(R,S) = P(R)\cdot P(S), \ldots$
- $P(R|H,\neg S) = \frac{P(H|R,\neg S)\cdot P(R|\neg S)}{P(H|\neg S)} = \frac{P(H|R,\neg S)\cdot P(R)}{P(H|R,\neg S)\cdot P(R)+P(H|\neg R,\neg S)\cdot P(\neg R)} = 0.0833$

Different type of Bayes Network

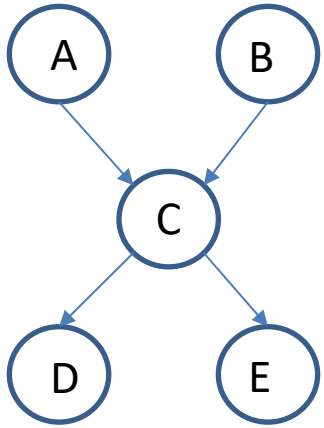


- $P(R|H,S) = 0.0142$
- $P(R|S) = P(R) = 0.01$
- $P(R|H,\neg S) = 0.0833$

- $R \perp S$ if we don't know anything about H
- $not(R \perp S)$ if we know about H
- Knowledge of H makes 2 variables that previously were independent, non-independent
- Independence does not imply conditional independence

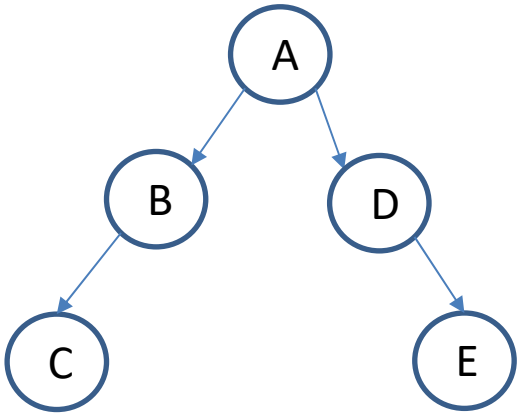
Bayes Networks

- Bayes networks define probability distribution over graphs or random variables

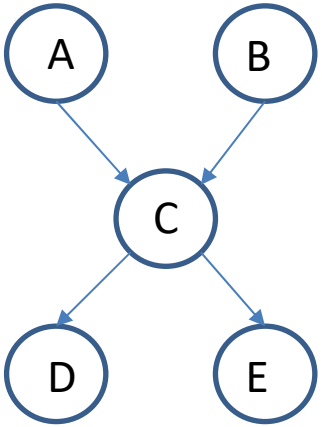


- Bayes network defines the distribution over those 5 random variables
- The Bayes network is defined by probability distributions that are inherent to each individual node
- For node A and B, we have a distribution P(A) and P(B), A and B have no incoming arcs
- For node C we have a distribution P(C|A,B)
- For node D and E we have P(D|C) and P(E|C)
- P(A,B,C,D,E)=P(A)P(B)P(C|A,B)P(D|C)P(E|C)
- Whereas the joint distribution over any 5 variables requires 2^5-1 =31, but the Bayes network requires only 10 variables (1+1+4+2+2=10)
- QUIZ: How many parameters do we need to specify the network defined on the Slide 2 (naïve joint over 16 variables is 2^{16}-1?
- QUIZ ANSWER: 47

D-SEPARATION



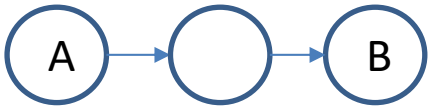
- $C \perp A$ (False)
- $C \perp A \mid B$ (True)
- $C \perp D$ (False)
- $C \perp D \mid A$ (True)
- $E \perp C \mid D$ (True)

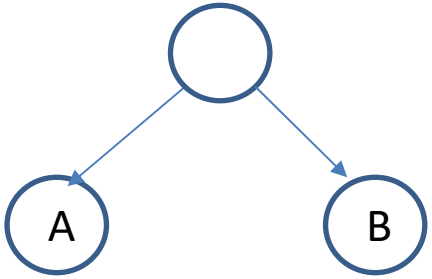


- $A \perp E$ (False)
- $A \perp E \mid B$ (False)
- $A \perp E \mid C$ (True)
- $A \perp B$ (True)
- $A \perp B \mid C$ (False)

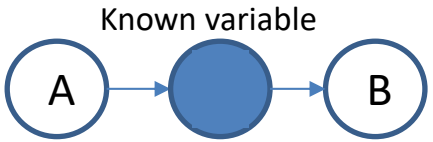
D-SEPARATION

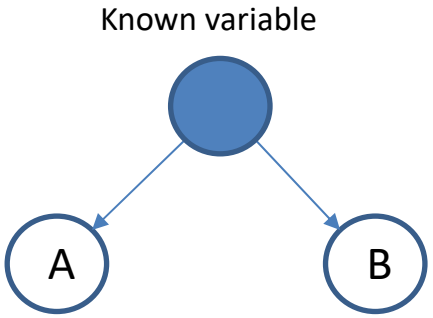
Active triplets
(render variables dependent $\text{not}(A \perp B)$)





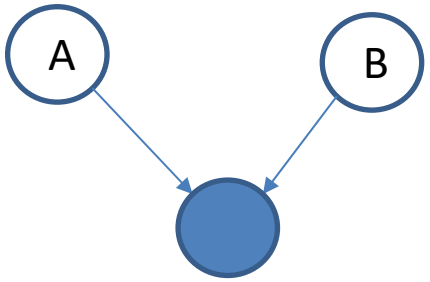
Inactive triplets
(render variables independent $A \perp B$)



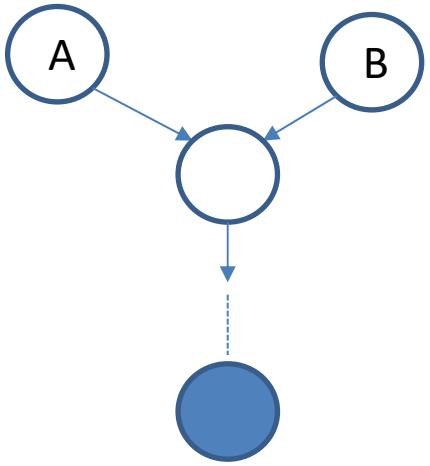


D-SEPARATION

Active triplets
(render variables dependent $\text{not}(A \perp B)$)

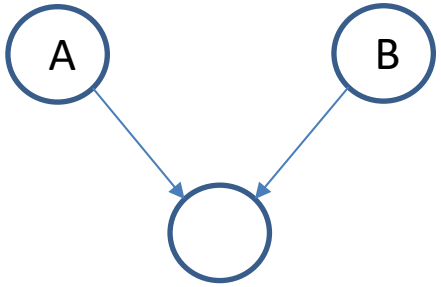


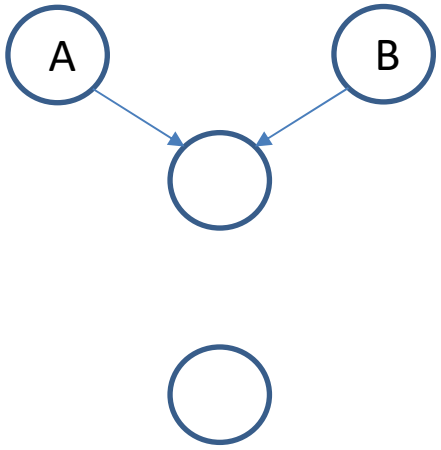
Known variable



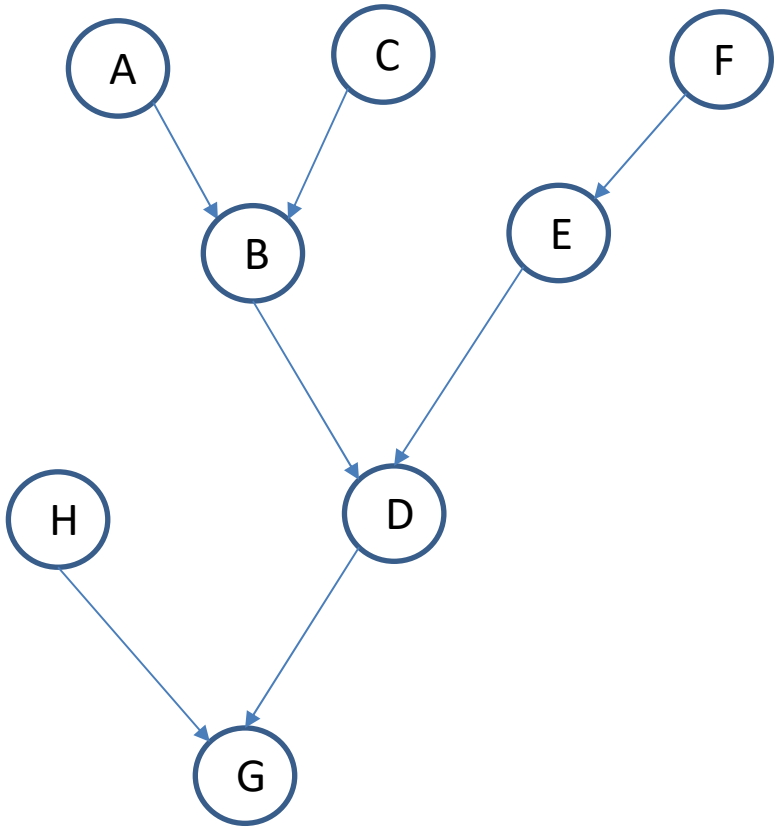
Known variable

Inactive triplets
(render variables independent $A \perp B$)





D-SEPARATION



- $F \perp A$ (True)
- $F \perp A \mid D$ (False)
- $F \perp A \mid G$ (False)
- $F \perp A \mid H$ (True)