Introduction to Artificial Intelligence (G51IAI)

Dr Rong Qu

Game Playing

Garry Kasparov and Deep Blue. © 1997, GM Gabriel Schwartzman's Chess Camera, courtesy IBM.

Game Playing

- Up till now we have assumed the situation is not going to change whilst we search
  - Shortest route between two towns
  - The same goal board of 8-puzzle, n-Queen
- Game playing is not like this
  - Not sure of the state after your opponent move
  - Goals of your opponent is to prevent your goal, and vice versa

Game Playing

- In these two hours
  - Brief history of game playing in AI
  - Important techniques in AI game playing
    - Minimax
    - Alpha beta pruning

Game Playing

- Game Playing has been studied for a long time
  - Babbage (1791-1871)
    - Analytical machine
    - tic-tac-toe
  - Turing (1912-1954)
    - Chess playing program
  - Within 10 years a computer will be a chess champion
    - Herbert Simon, 1957

Game Playing

- Why study game playing in AI

  - Games are intelligent activities
  - It is very easy to measure success or failure
  - Do not require large amounts of knowledge
  - They were thought to be solvable by straightforward search from the starting state to a winning position

Game Playing - Checkers

- Arthur Samuel
  - 1952 – first checker program, written for an IBM 701
  - 1954 - Re-wrote for an IBM 704
    - 10,000 words of main memory

Game Playing - Checkers

- Arthur Samuel
  - Added a learning mechanism that ***learnt*** its own evaluation function by playing against itself
  - After a few days it could beat its creator
  - And compete on equal terms with strong human players

Game Playing - Checkers

- Jonathon Schaeffer – Chinook, 1996
  - In 1992 Chinook won the US Open
  - Plays a perfect end game by means of a database
  - And challenged for the world championship
    - http://www.cs.ualberta.ca/~chinook/

Game Playing - Checkers

- Jonathon Schaeffer – Chinook, 1996
  - Dr Marion Tinsley
    - World championship for over 40 years, only losing three games in all that time
    - Against Chinook he suffered his fourth and fifth defeat
    - But ultimately won 21.5 to 18.5

Game Playing - Checkers

- Jonathon Schaeffer – Chinook, 1996
  - Dr Marion Tinsley
    - In August 1994 there was a re-match but Marion Tinsley withdrew for health reasons
    - Chinook became the official world champion

Game Playing - Checkers

- Jonathon Schaeffer – Chinook, 1996
  - Uses Alpha-Beta search
  - Did not include any learning mechanism
  - Schaeffer claimed Chinook was rated at 2814
  - The best human players are rated at 2632 and 2625

Game Playing - Checkers

- Chellapilla and Fogel – 2000
  - “Learnt” how to play a good game of checkers
  - The program used a *population* of games with the best competing for *survival*
  - Learning was done using a *neural network* with the synapses being changed by an *evolutionary strategy*
    - Input: current board position
    - Output: a value used in minimax search

Game Playing - Checkers

- Chellapilla and Fogel – 2000
  - During the training period the program is given
    - no information other than whether it won or lost (it is not even told by how much)
    - No strategy and no database of opening and ending positions
  - The best program beats a commercial application 6-0
  - The program was presented at CEC 2000 (San Diego) and prize remain unclaimed

Game Playing - Chess

No computer can play even an amateur-level game of chess

Hubert Dreyfus, 1960’s

Game Playing - Chess

- Shannon - March 9th 1949 - New York
  - Size of search space (10^{120} - average of 40 moves)
    - 10^{120} > number of atoms in the universe
    - 200 million positions/second = 10^{100} years to evaluate all possible games
      - Age of universe = 10^{10}
  - Searching to depth = 40, at one state per microsecond, it would take 10^{90} years to make its first move

Game Playing - Chess

- 1957 – AI pioneers Newell and Simon predicted that a computer would be chess champion within ten years
- Simon: “I was a little far-sighted with chess, but there was no way to do it with machines that were as slow as the ones way back then”
- 1958 - First computer to play chess was an IBM 704
  - about one millionth capacity of deep blue

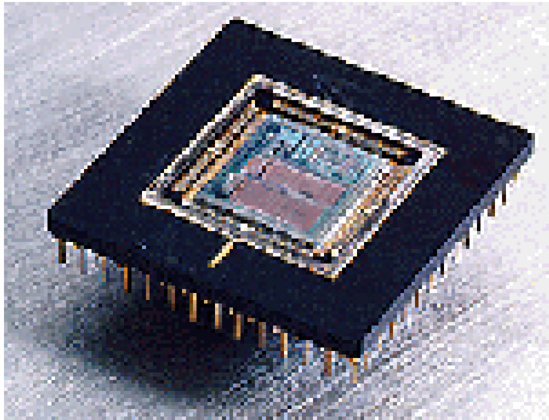
Game Playing - Chess

- 1967 : Mac Hack competed successfully in human tournaments
- 1983 : “Belle” attained expert status from the United States Chess Federation
- Mid 80’s : Scientists at Carnegie Mellon University started work on what was to become Deep Blue
  - Sun workstation, 50K positions per second
  - Project moved to IBM in 1989

Game Playing - Chess

- May 11th 1997, Gary Kasparov lost a six match game to deep blue, IBM Research
  - 3.5 to 2.5
  - Two wins for deep blue, one win for Kasparov and three draws

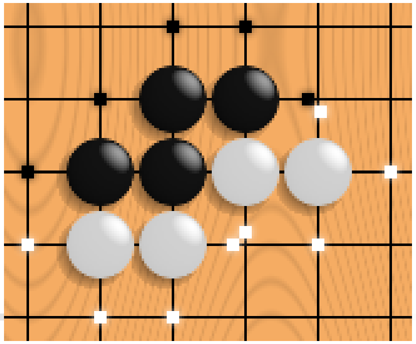
(http://www.research.ibm.com/deepblue/meet/html/d.3.html)



Game Playing - Chess

- Still receives a lot of research interests
- Computer program to "learn" how to play chess, rather than being "told" how it should play
- Research on game playing at School of CS, Nottingham

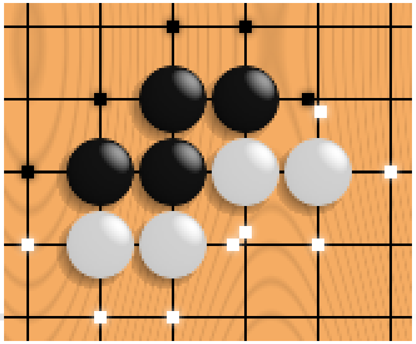
Game Playing – Go*



- A significant challenge to computer programmers, not yet much helped by fast computation
- Search methods successful for chess and checkers do not work for Go, due to many qualities of the game
  - Larger area of the board (five times the chess board)
  - New piece appears every move - progressively more complex

*wikipedia: http://en.wikipedia.org/wiki/Go_(game)

Game Playing – Go*



- A significant challenge to computer programmers, not yet much helped by fast computation
- Search methods successful for chess and checkers do not work for Go, due to many qualities of the game
  - A material advantage in Go may just mean that short-term gain has been given priority
  - Very high degree of pattern recognition involved in human capacity to play well
  - …

*wikipedia: http://en.wikipedia.org/wiki/Go_(game)

Game Playing

- Other games in research
  - Poker
  - Othello
  - …
- Previous third year projects
  - Chess
  - Poker
  - Blackjack
  - …

Game Playing - Minimax

- Game Playing
  - An opponent tries to thwart your every move
- 1944 - John von Neumann outlined a search method (*Minimax*)
  - *maximise* your position whilst *minimising* your opponent's

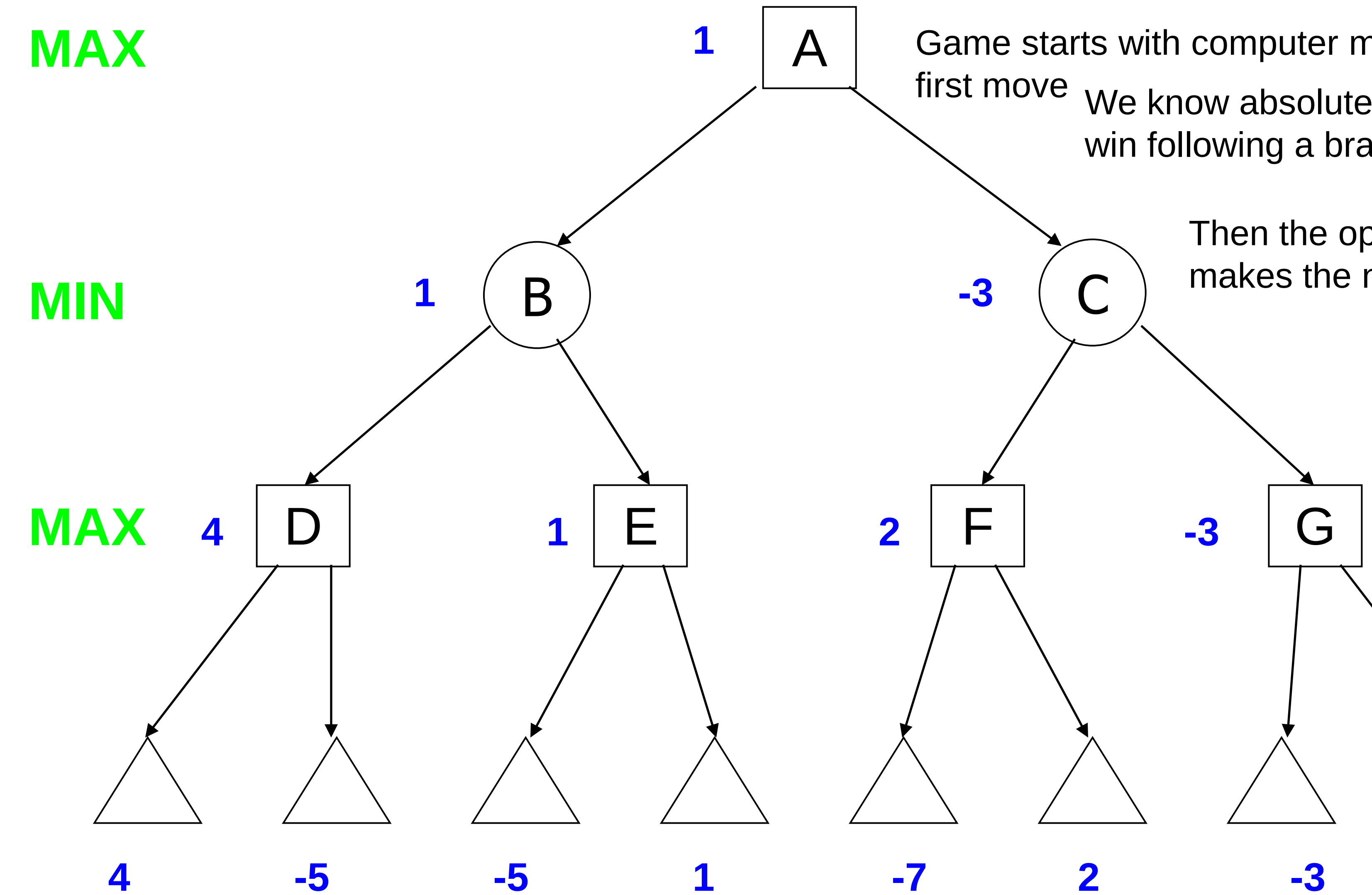
Game Playing - Minimax

- In order to implement we need a method of measuring how good a position is
  - Often called a *utility function*
  - Initially this will be a value that describes our position exactly

Assume we can generate the full search tree

Of course for larger problem it's not possible to draw the entire tree

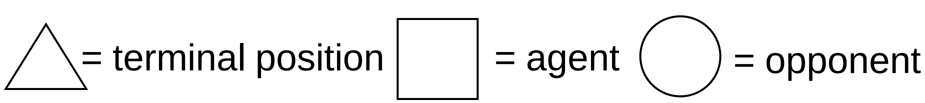
The idea is computer wants to force the opponent to lose, and maximise its own chance of winning



Game starts with computer making the first move

We know absolutely who will win following a branch

Then the opponent makes the next move



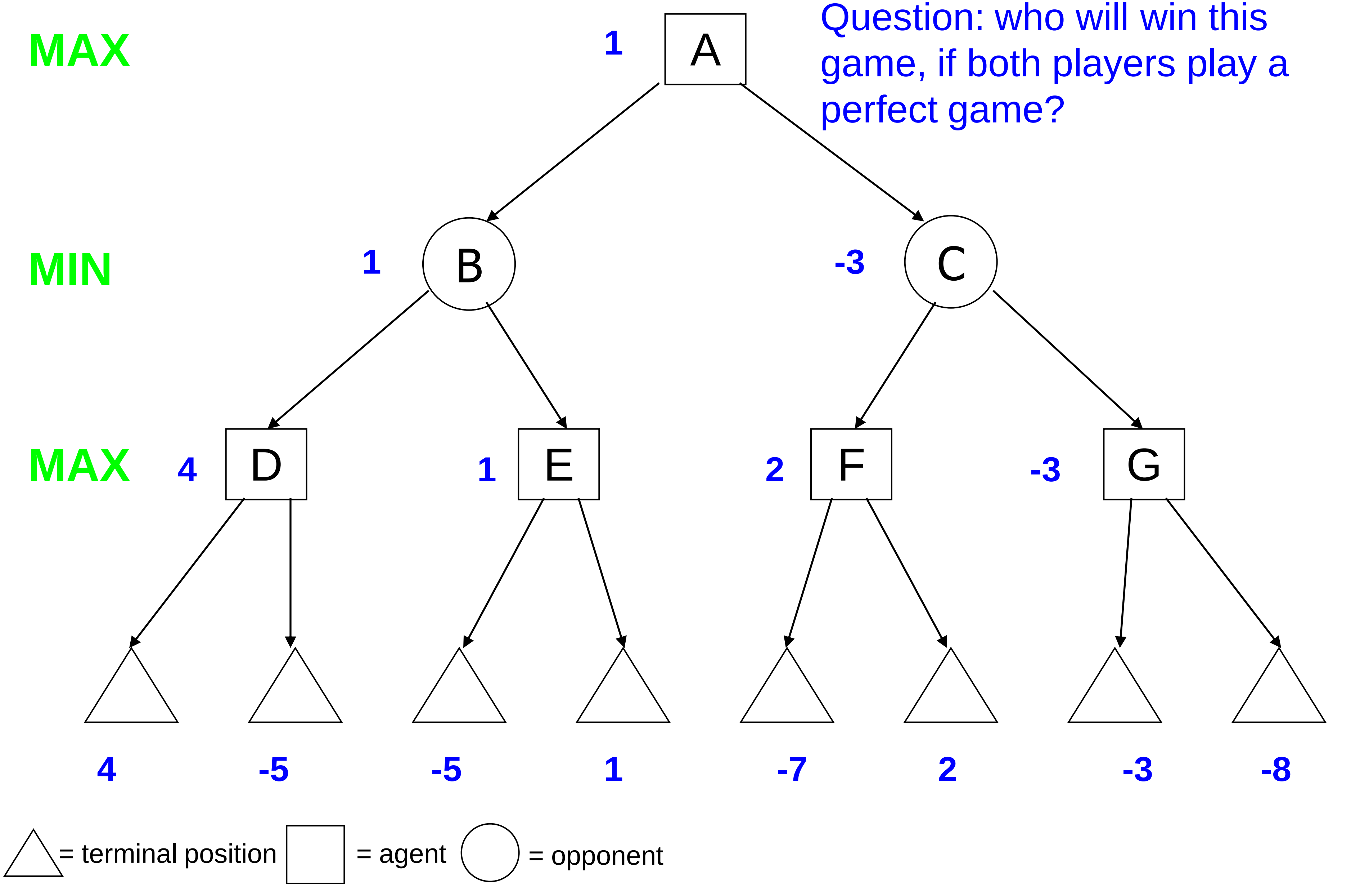
Values are propagated back up through the tree based on whose turn it is and whether they are trying to maximise or minimise at the point

Now we can decide who win the game

Assume positive: computer wins

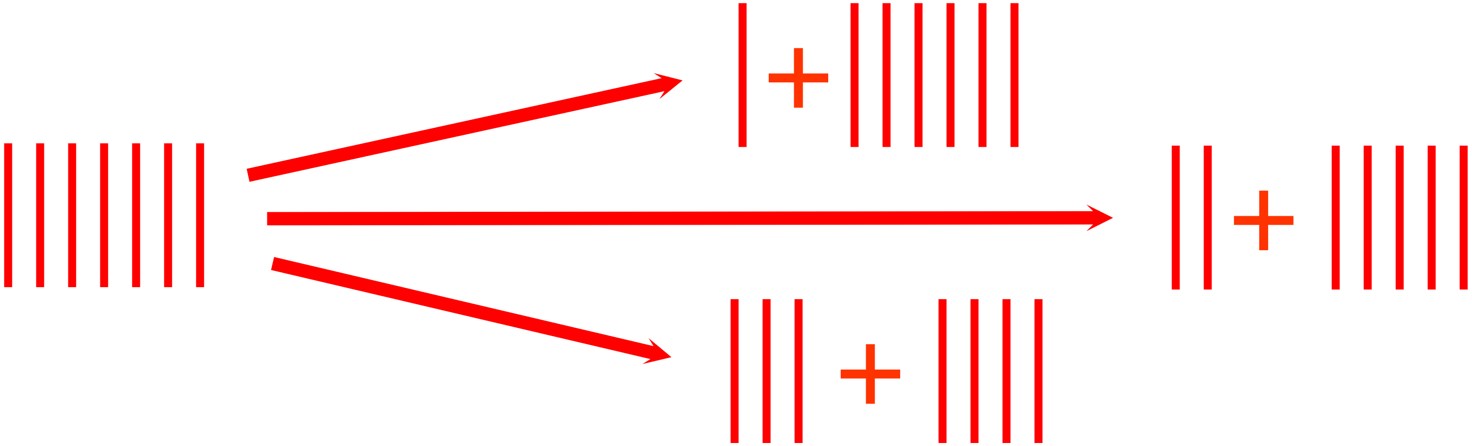
Now the computer is able to play a perfect game. At each move it'll move to a state of the highest value.

Question: who will win this game, if both players play a perfect game?



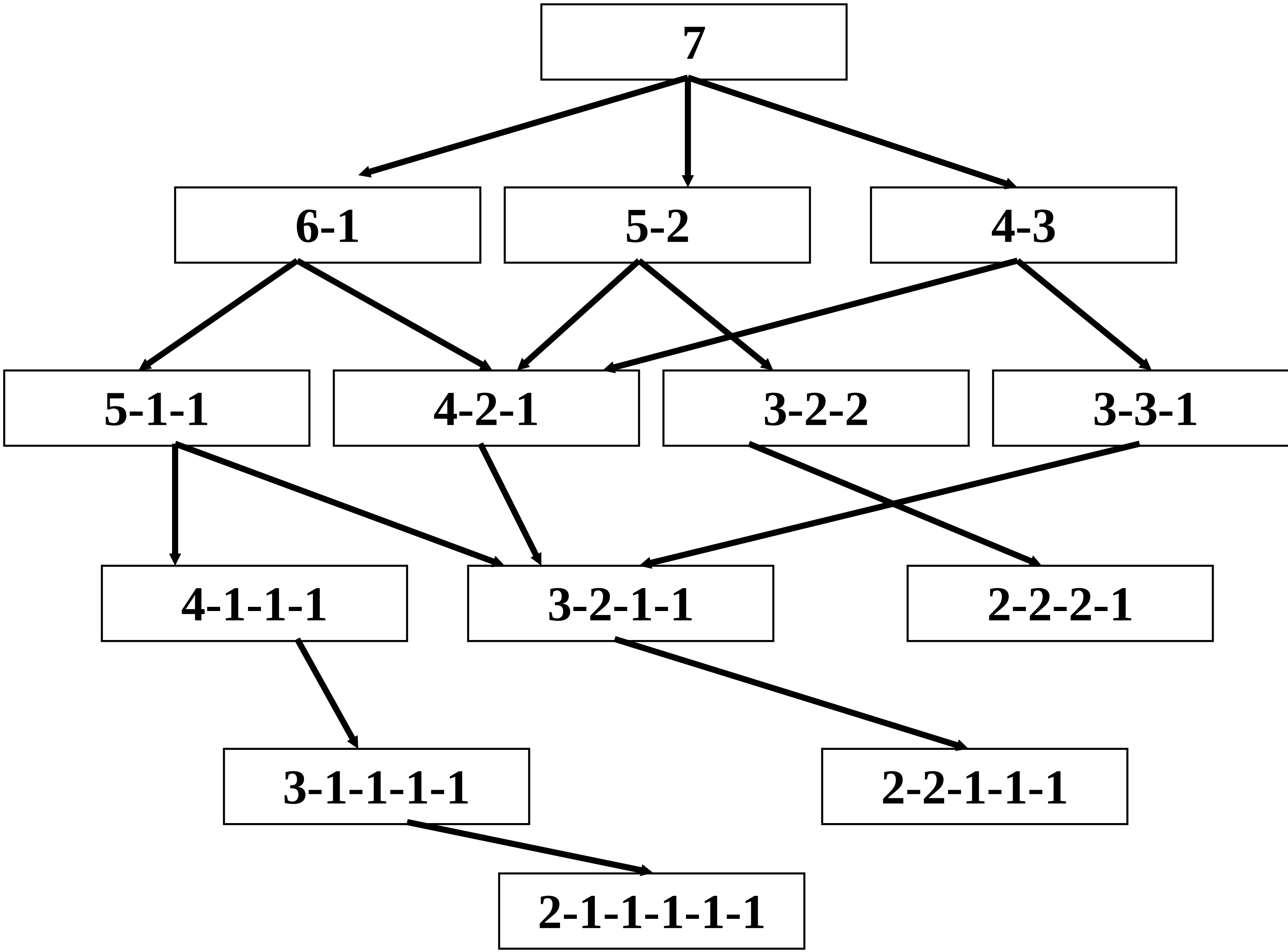
Game Playing - Minimax

- Nim
  - Start with a pile of tokens
  - At each move the player must divide the tokens into two non-empty, non-equal piles



Game Playing - Minimax

- Nim
  - Starting with 7 tokens, draw the complete search tree
  - At each move the player must divide the tokens into two non-empty, non-equal piles


7
6-1
5-2
4-3
5-1-1
4-2-1
3-2-2
3-3-1
4-1-1-1
3-2-1-1
2-2-2-1
3-1-1-1-1
2-2-1-1-1
2-1-1-1-1-1

Game Playing - Minimax

- Conventionally, in discussion of minimax, have two players “MAX” and “MIN”
- The utility function is taken to be the utility for MAX
- Larger values are better for “MAX”

Game Playing - Minimax

- Assuming MIN plays first, complete the MIN/MAX tree
- Assume that a utility function of
  - 0 = a win for MIN
  - 1 = a win for MAX

Game Playing - Minimax

- Player MAX is going to take the best move available
  - Will select the next state to be the one with the highest utility

- Hence, value of a MAX node is the MAXIMUM of the values of the next possible states
  - i.e. the maximum of its children in the search tree

Game Playing - Minimax

- Player MIN is going to take the best move available for MIN i.e. the worst available for MAX
  - Will select the next state to be the one with the lowest utility
  - higher utility values are better for MAX and so worse for MIN

- Hence, value of a MIN node is the MINIMUM of the values of the next possible states
  - i.e. the minimum of its children in the search tree

Game Playing - Minimax

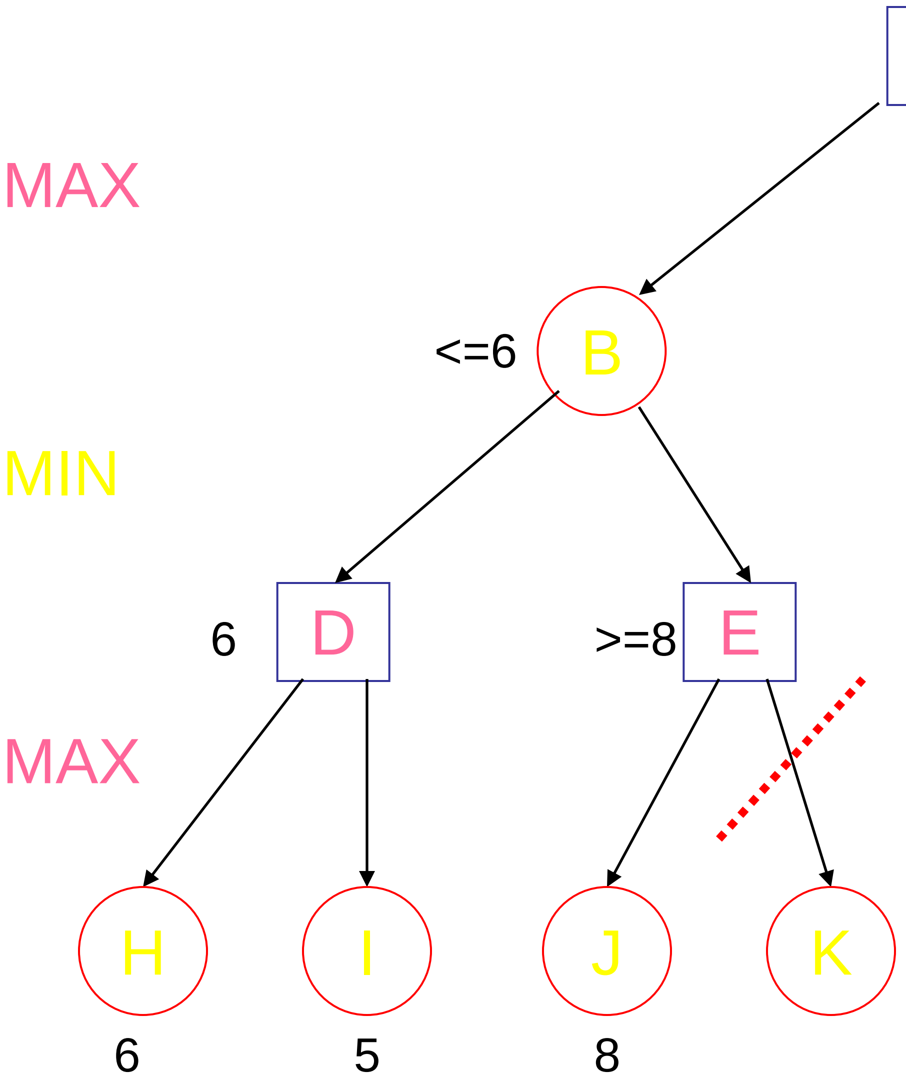
- A “MAX” move takes the best move for MAX
  - so takes the MAX utility of the children
- A “MIN” move takes the best for min
  - hence the worst for MAX
  - so takes the MIN utility of the children
- Games alternate in play between MIN and MAX

Game Playing - Minimax

- Efficiency of the search
  - Game trees are very big
  - Evaluation of positions is time-consuming
- How can we reduce the number of nodes to be evaluated?
  - "alpha-beta search"

Game Playing - Minimax

- At each node
  - Decide a value which reflects our position of winning from the point
  - Heuristic function
    - Possibility of winning
    - Different from that in A* for search problem, which estimate how close we are to the goal



STOP! What else can you deduce now!?

On discovering util(D) = 6

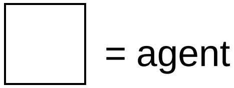
we know that util(B) <= 6

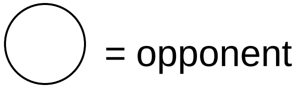
On discovering util(J) = 8

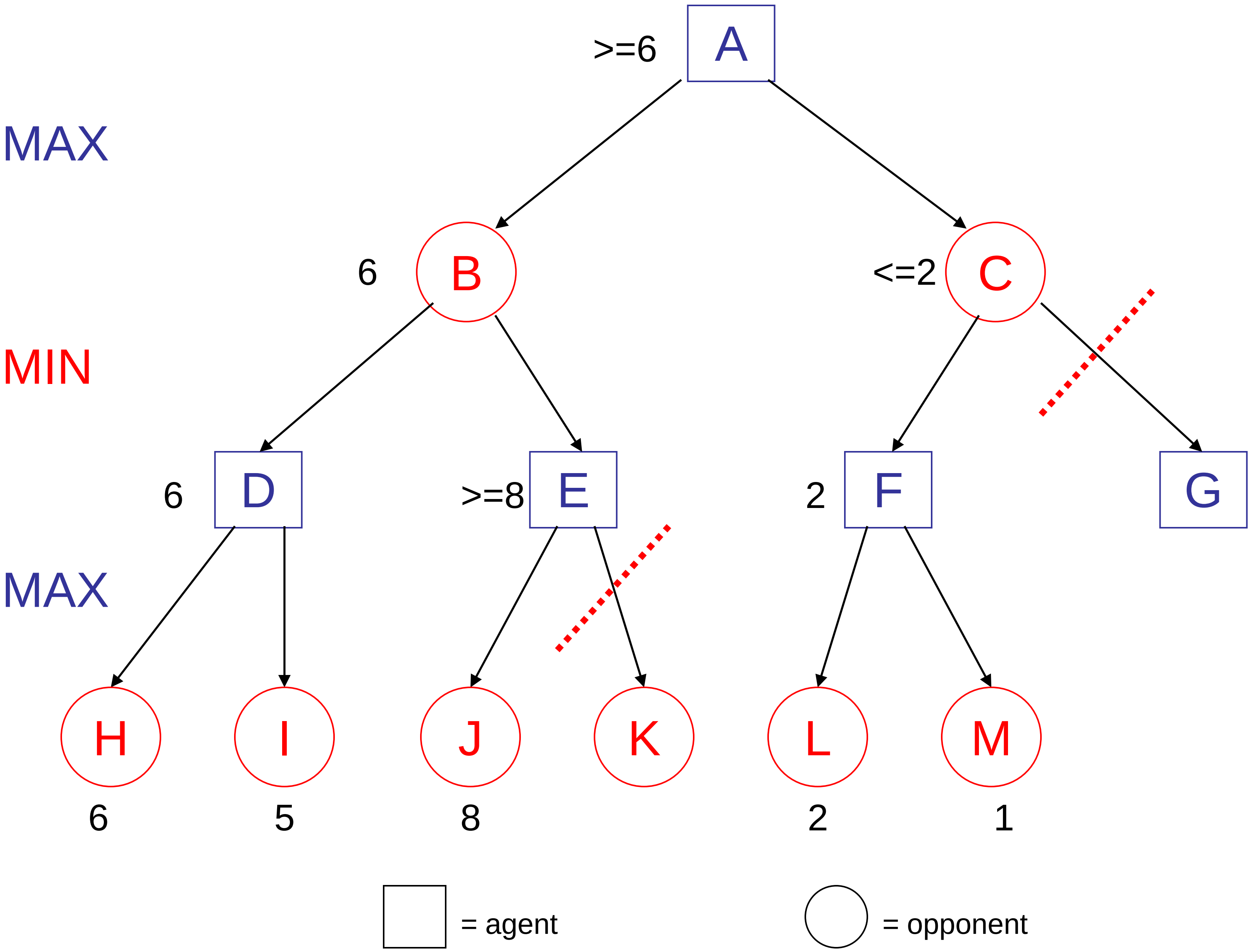
we know that util(E) >= 8

Can stop expansion of E as best play will not go via E

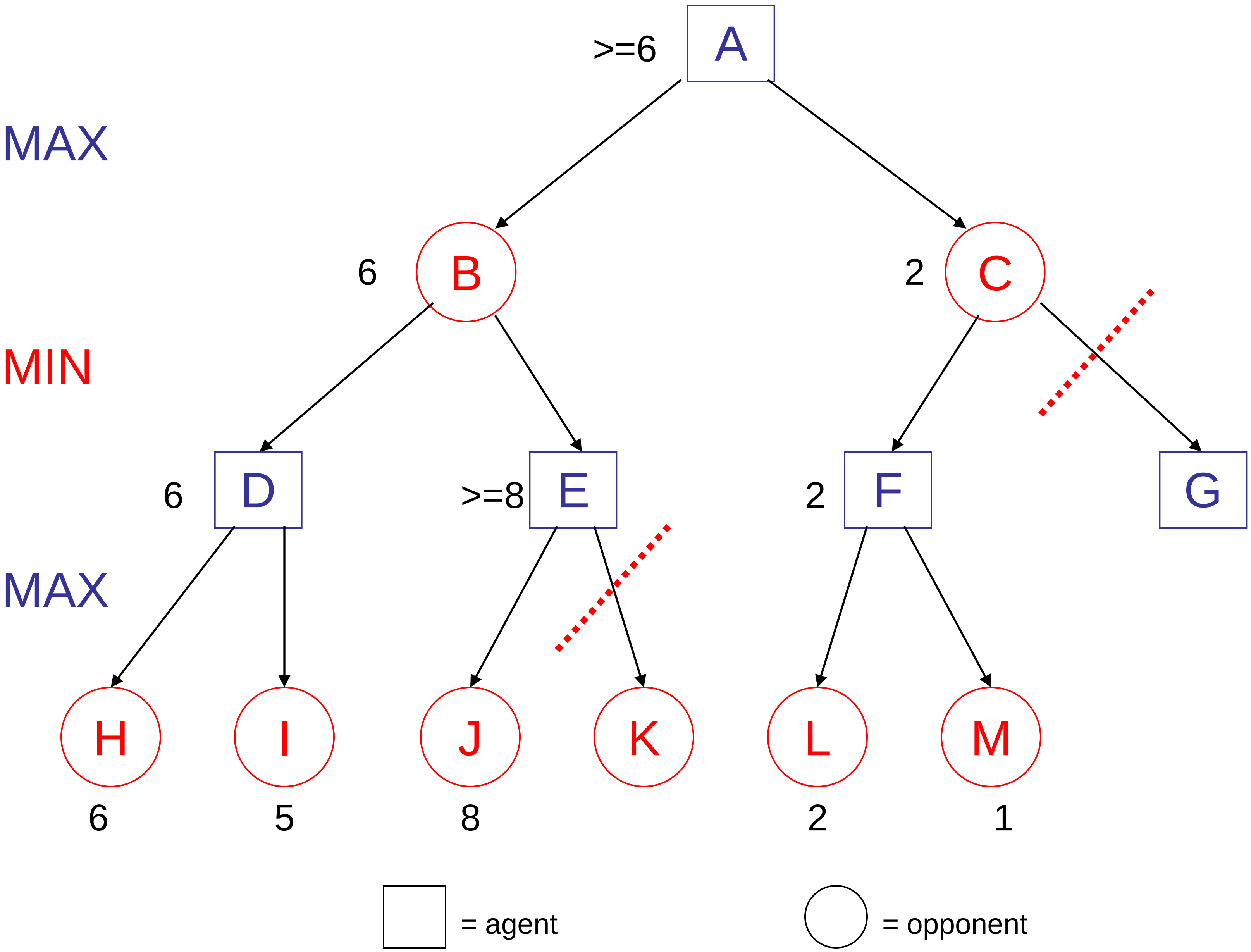
Value of K is irrelevant – prune it!





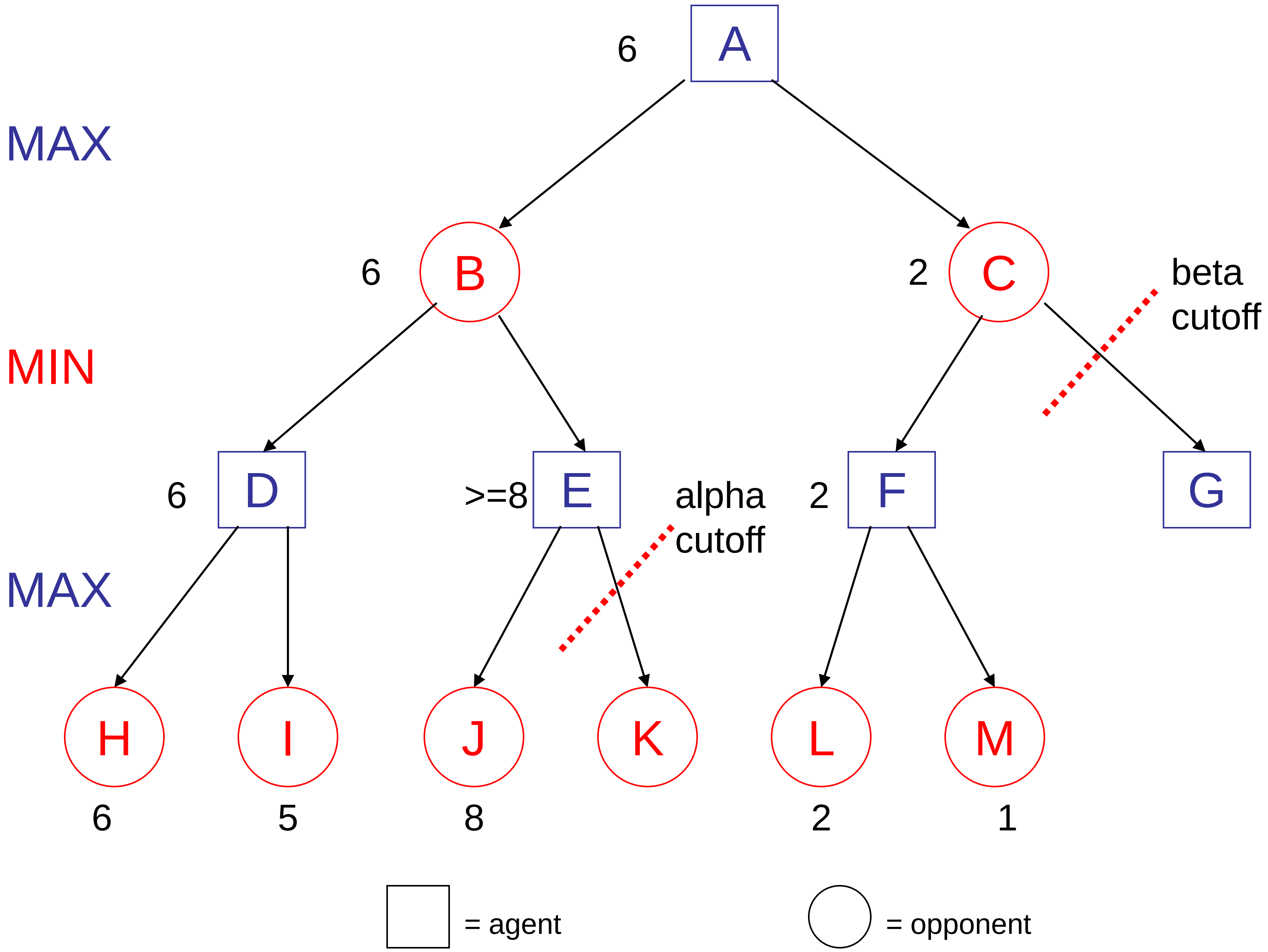


MAX
MIN
MAX
>=6
A
6
B
<=2
C
6
D
>=8
E
2
F
G
H
I
J
K
L
M
6
5
8
2
1
= agent
= opponent



>=6
A
MAX
6
B
2
C
MIN
6
D
>=8
E
2
F
G
MAX
H
I
J
K
L
M
6
5
8
2
1
= agent
= opponent

Alpha-beta Pruning



Game Playing - Alpha-beta Pruning

- If we were doing Breadth-First Search, would you still be able to prune nodes in this fashion?
- NO! Because the pruning on node D is made by evaluating the tree underneath D
- This form of pruning relies on doing a Depth-First search

Game Playing - Alpha-beta Pruning

- To maximise pruning we want to first expand those children that are best for the parent
  - cannot know which ones are really best
  - use heuristics for the “best-first” ordering
- If this is done well then alpha-beta search can effectively double the depth of search tree that is searchable in a given time
  - Effectively reduces the branching factor in chess from about 30 to about 8
  - This is an enormous improvement!

Game Playing - Alpha-beta Pruning

- The pruning was based on using the results of the “DFS so far” to deduce upper and lower bounds on the values of nodes

- Conventionally these bounds are stored in terms of two parameters
  - alpha α
  - beta β

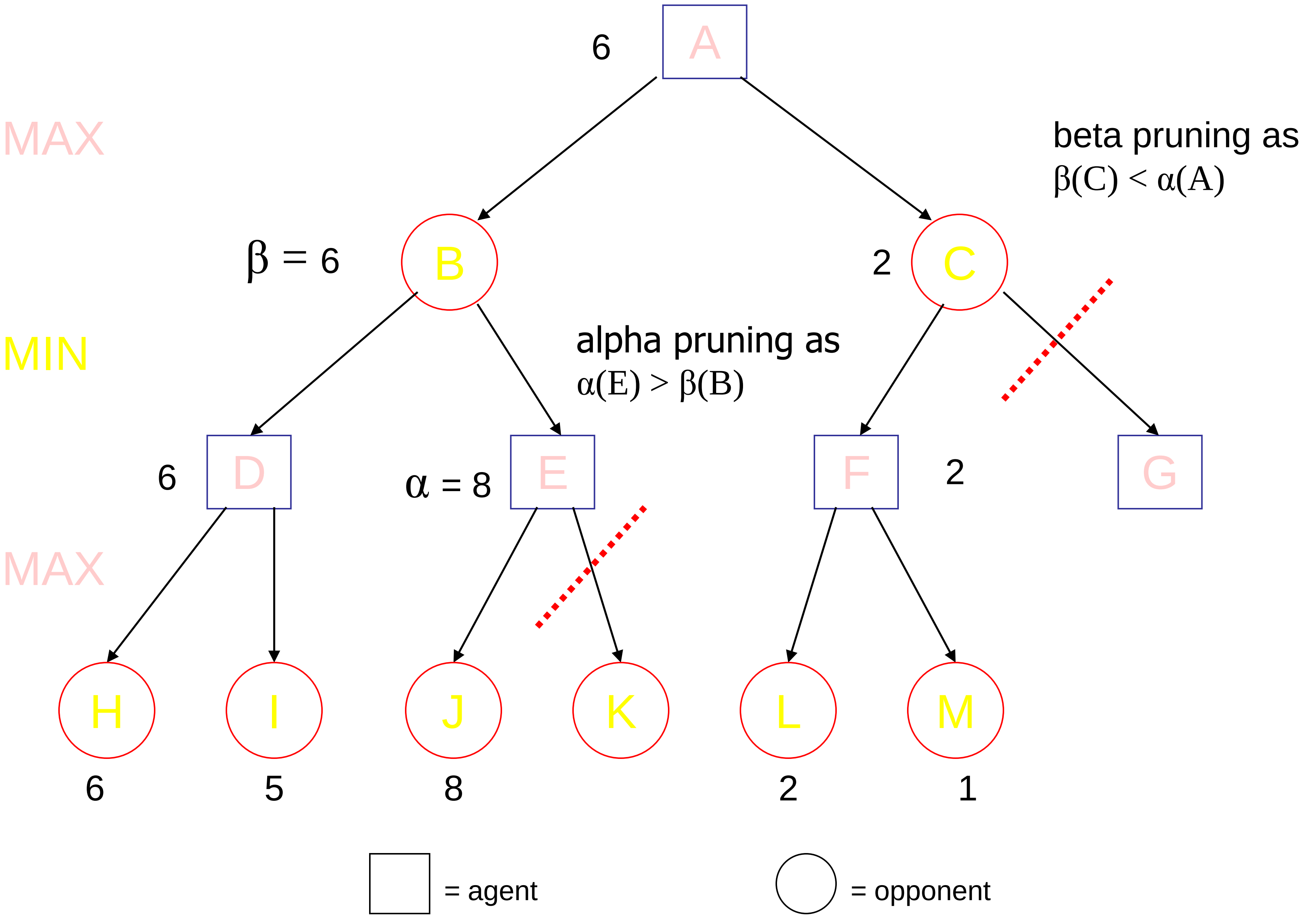
Game Playing - Alpha-beta Pruning

- α values are stored with each MAX node
- each MAX node is given a value of alpha that is the current best lower-bound on its final value
  - initially is $-\infty$ to represent that nothing is known
  - as we do the search then α at a node can increase, but it can never decrease – it always gets better for MAX

Game Playing - Alpha-beta Pruning

- β values are stored with each MIN node
- each MIN node is given a value of beta that is the current best upper-bound on its final value
  - initially is $+\infty$ to represent that nothing is known
  - as we do the search then β at a node can decrease, but it can never increase – it always gets better for MIN

Alpha-beta Pruning



Game Playing - classification

- So far have only considered games such as chess, checkers, and nim

- These games are:

1. Fully observable
   - Both players have full and perfect information about the current state of the game
2. Deterministic
   - There is no element of chance
   - The outcome of making a sequence of moves is entirely determined by the sequence itself

Game Playing - classification

- Fully vs. Partially Observable
  - Some games are only partially observable
  - Players do not have access to the full "state of the game"
  - e.g. card games – you typically cannot see all of your opponents cards

Game Playing - classification

- Deterministic vs. Stochastic
  - In many games there is some element of chance
  - E.g. Backgammon – throw dice in order to move

You are expected to be aware of these simple classifications

Summary – game playing

- History
  - Checkers
  - Chess
  - Go
- Techniques
  - Minimax
  - Alpha-beta pruning
- Game classifications