Beyond Minimax: Nonzero-Sum Game Tree Search with Knowledge Oriented Players

Tian Sang & Jiun-Hung Chen

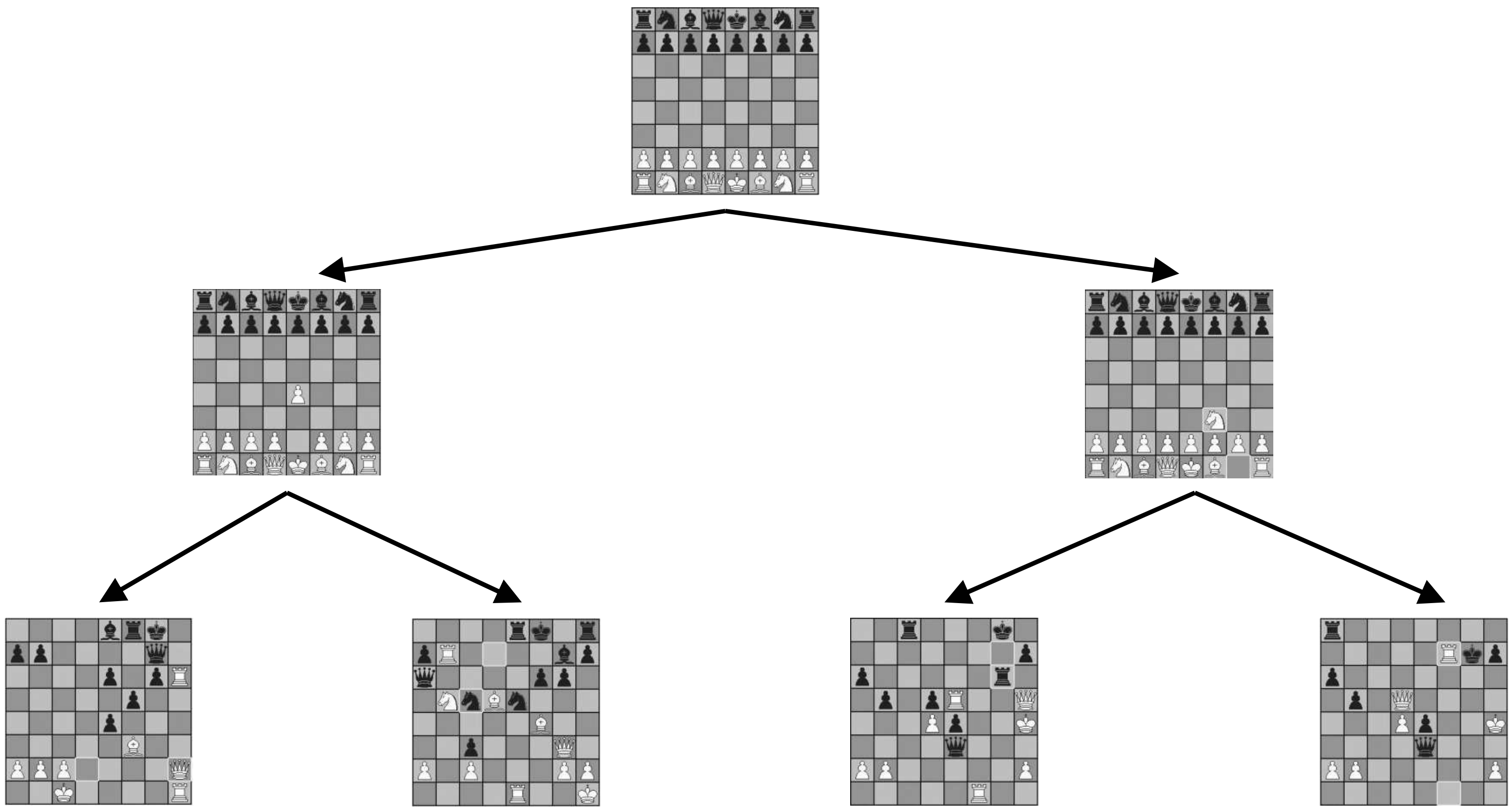
Department of Computer Science & Engineering
University of Washington

AAAI Metareasoning Workshop, July, 2008

Acknowledgement

- Our best appreciation to Paul Beame, professor in CSE, University of Washington, who gave us a great deal of constructive comments on this work
- Thanks to the anonymous reviewers for Metareasoning, who helped us modify the paper

Game Tree



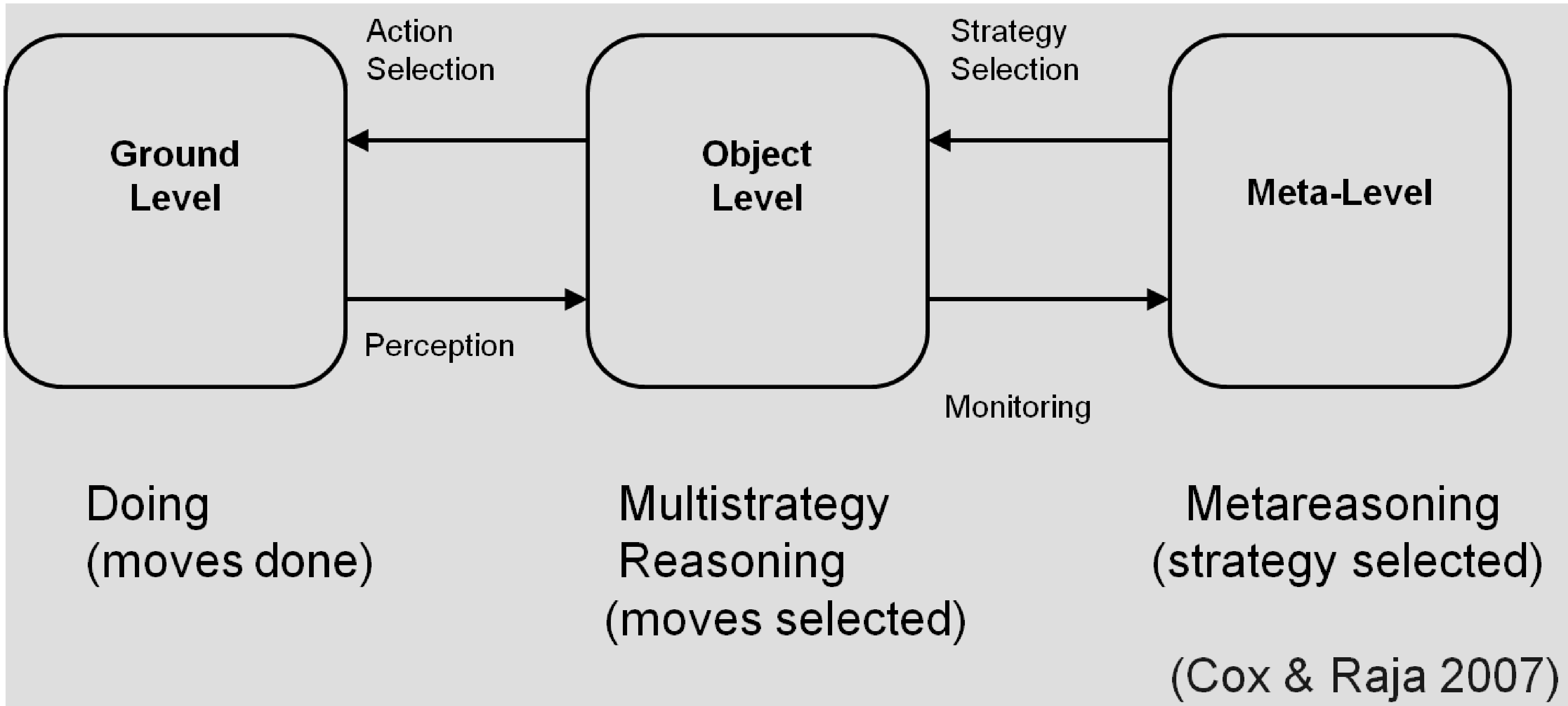
Scenario

- ►NOT Nash-equilibrium-style matrix games
- ►Finite-horizon one-time game with two selfish players moving alternatively, like chess
- ►A player has complete or incomplete knowledge about the opponent, including knowledge about the opponent's knowledge
- ►Max (min) player tries to maximize (minimize) his own outcome, based on his knowledge

Overview

- Zero-sum game vs. Nonzero-sum game
- Why Minimax?
- Opponent model and M* search
  - Requires complete knowledge about opponent
- Our framework for Knowledge-Oriented Players (KOP)
  - Capable of modeling incomplete knowledge
- New M^h & Hybrid Minimax-M^h algorithms
  - Guaranteed better outcome than Minimax
- Why is communication important?

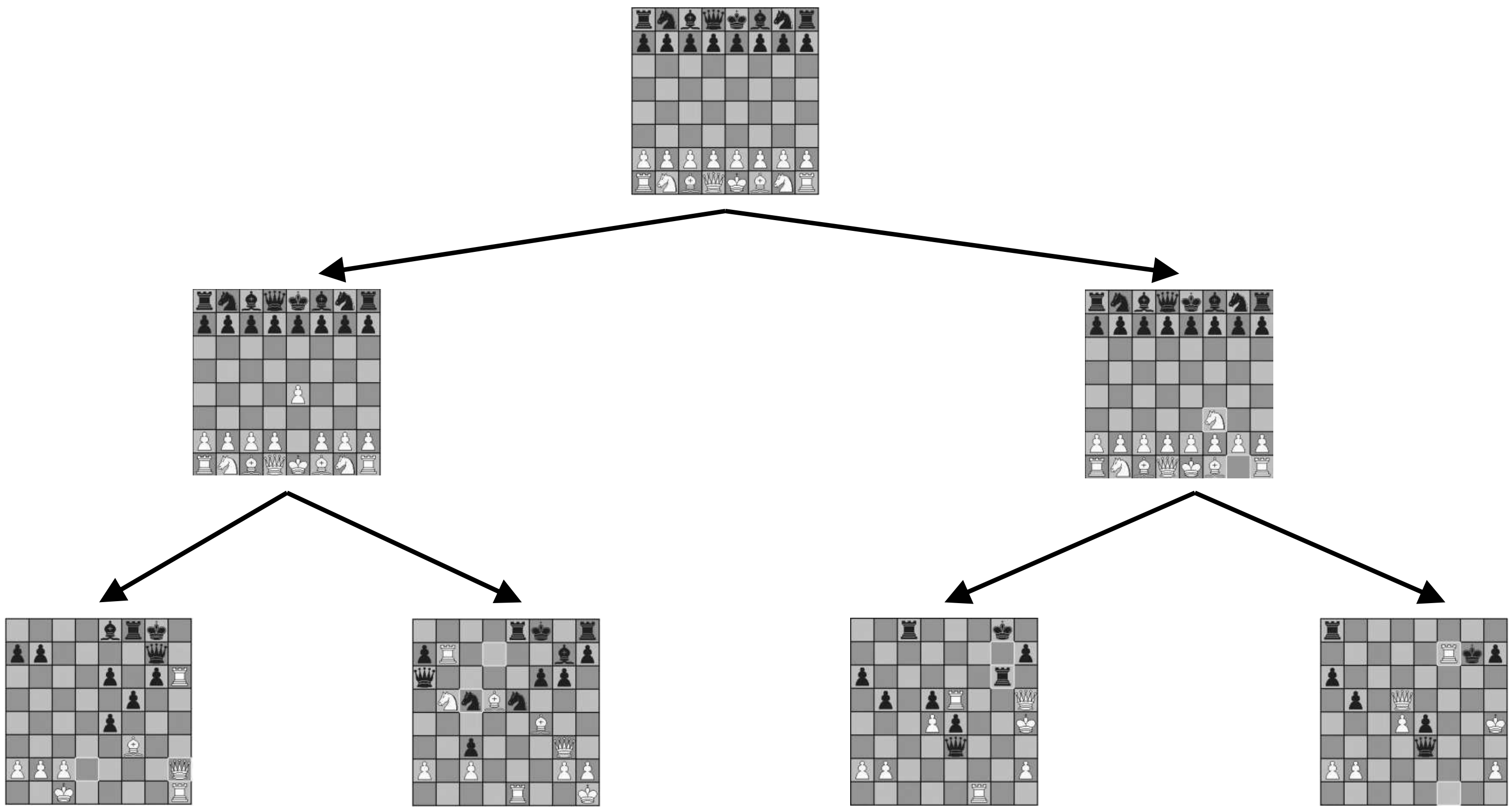
Metareasoning



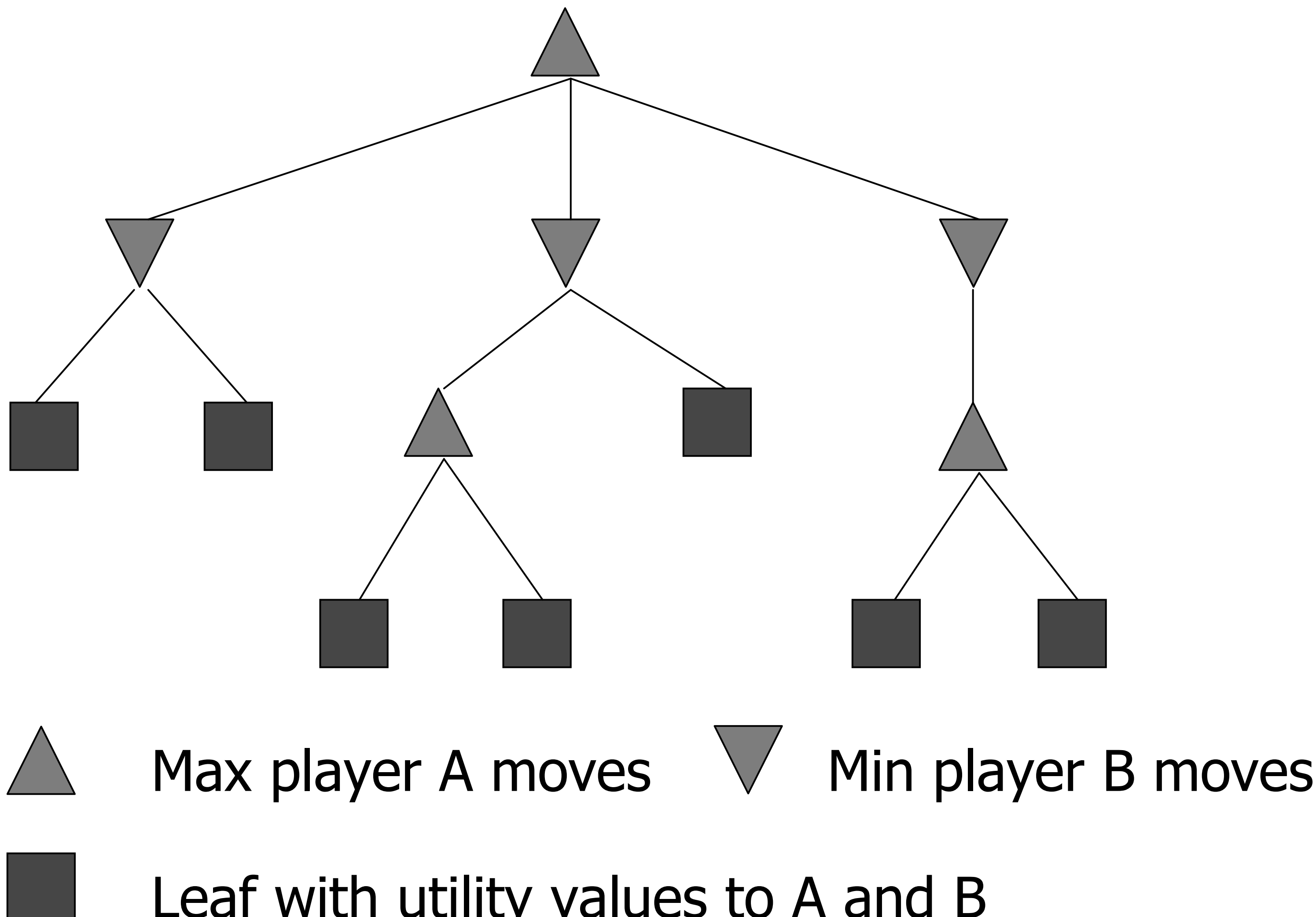
Metareasoning

- Player A versus player B
- A thinks: in order to make an optimal move, I should consider B's responses to each possible move. Since B may know that I would consider this, I should consider that B would consider that I would consider B's responses…
  - Simulating myself and the opponent
- You are simulating your opponent, while your opponent may be simulating you
  - The results can be unpredictable

Game Tree



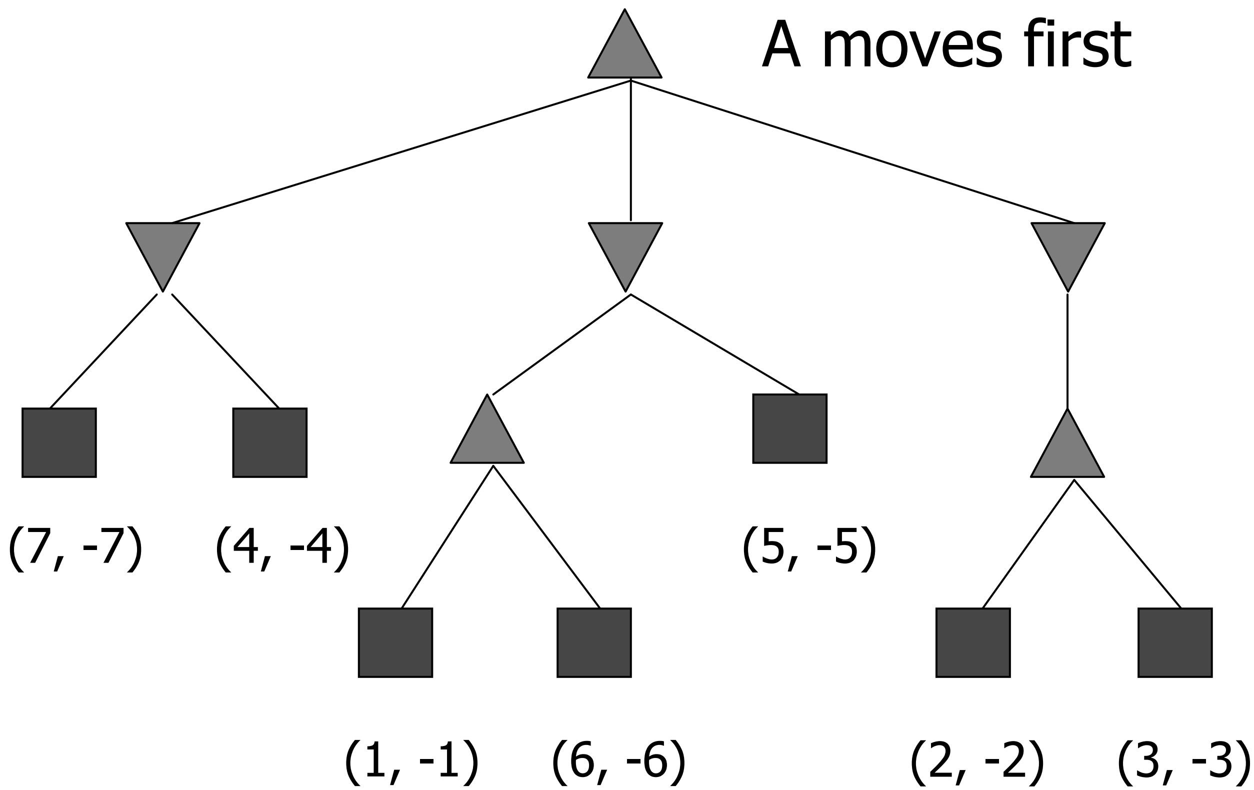
Abstract Game Tree



Game Tree and Strategies

- A leaf L in a game tree t has value(L), a pair of values for each player respectively
- An internal node N in t has two fields
  - next(N): pointer to one of N's children
  - value(N): same as value(next(N))
- A strategy is an algorithm that assigns values to next(N) and value(N) of every node N
  - Given strategies of both players and a game tree, the outcome is decided: just following the next pointers alternatively to reach a leaf

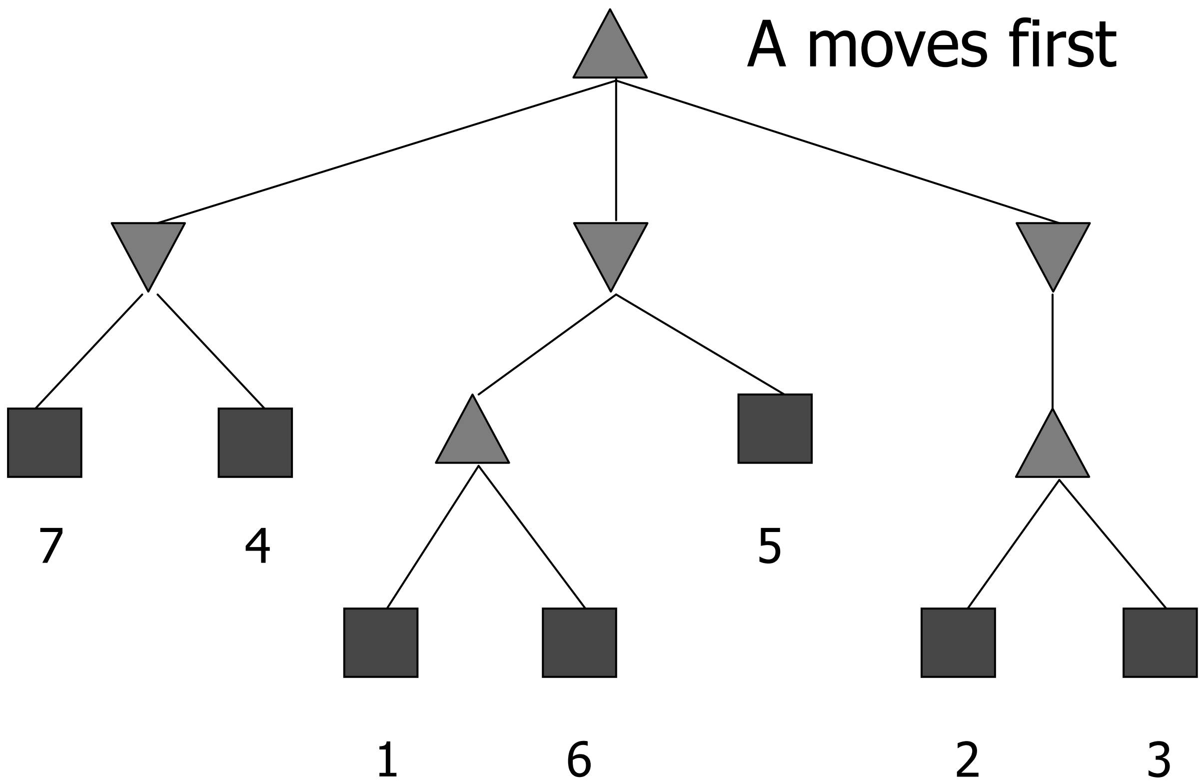
Zero-sum Game Tree



Goal: A or B maximizes the outcome by E_A or E_B

Zero-sum: $E_A(x) + E_B(x) = 0$, where x is a leaf

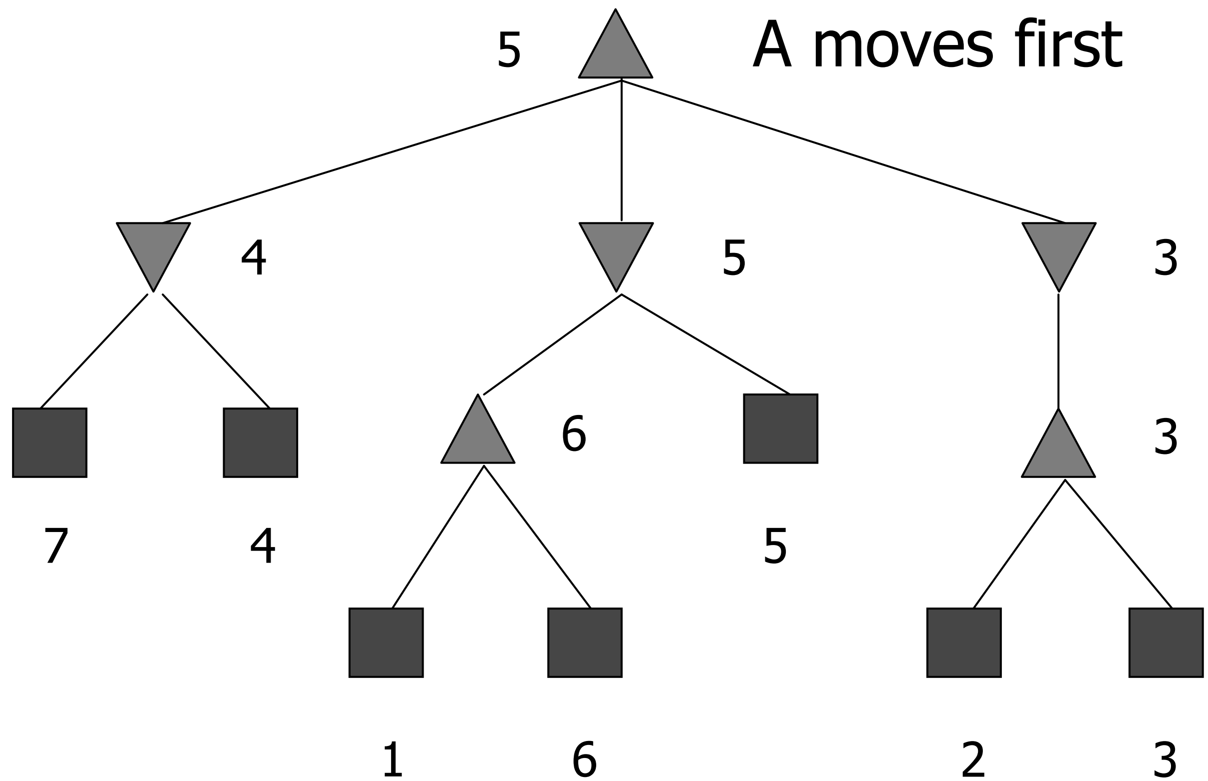
Zero-sum Game Tree



Alternatively, only one function is needed.

Goal: A maximizes the outcome while B minimizes it

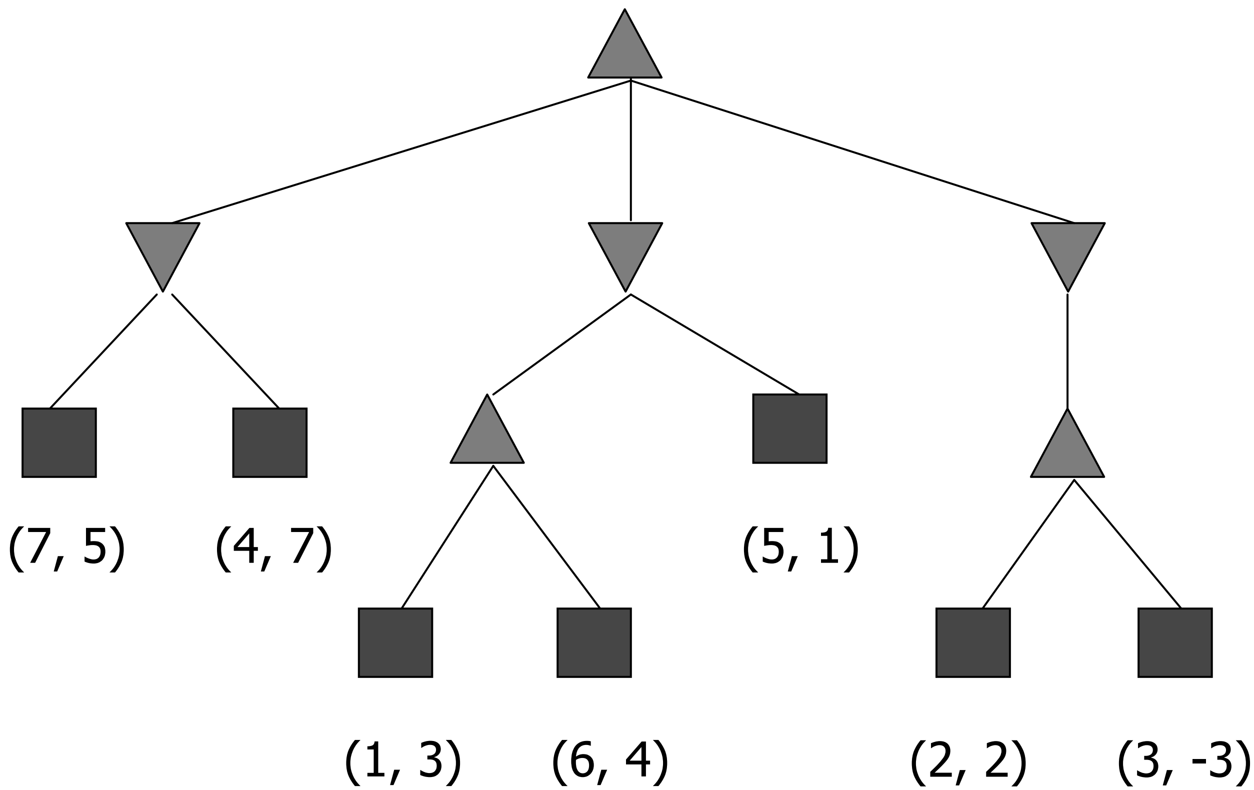
Minimax for Zero-sum Game



Minimax assumes that the perfect opponent will choose the worst move for a player (Shannon 1950)

Minimax yields the outcome of 5

Nonzero-sum Game Tree



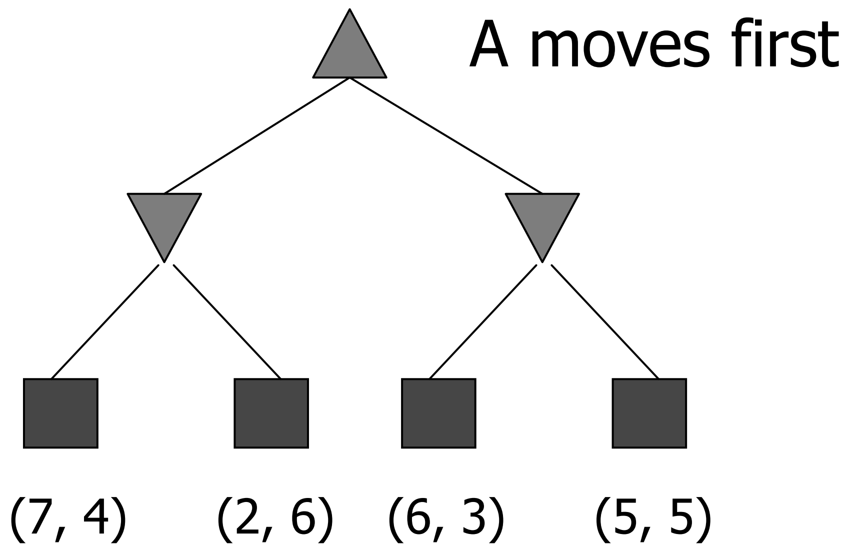
Goal: A or B maximizes the outcome by E_A or E_B

Nonzero-sum: E_A and E_B may not be correlated

Opponent Model (OM)

- Opponent Model (Carmel & Markovitch 1993; Iida et al. 1993) is the knowledge about the opponent
- OM assumes
  1) A knows B's "secret" E_B
  2) A knows that B does not know A's "secret" E_A
  3) A knows that B assumes A uses Minimax to maximize E_B
- Given 1-3, OM search is optimal for A
- Problem: too strong assumptions
  - 2) means knowledge asymmetry
  - 3) means that B may have a wrong assumption on A, possibly inconsistent with B's knowledge

An Example of OM Search



- If A uses Minimax, A will move right
  - Expecting to get (5, 5), actually will get (6, 3)
- If A uses OM search, A will move left
  - Reaching (7, 4), instead of (2, 6) which B does not prefer, better than Minimax for A!

M* Search

- Extending OM recursively: both players have OM of each other (Carmel & Markovitch 1993)
  - M^0 is Minimax and M^1 is OM search
- M* assumes
  1) A's OM strictly contains B's OM
  2) There is a sequence of simulations: A's M^d, B's M^{d-1}, A's M^{d-2}, B's M^{d-3} …, (A's or B's) M^0
- Given 1) and 2), M* search is optimal for A
- Problem: too strong and inconsistent assumptions
  - B has a wrong OM of A
  - Outcome for B is NOT guaranteed!
    - Why would B even use M^{d-1}?

General Knowledge Scenario

- ►Players' knowledge may not subsume each other: M* search does not apply
  - For example, consider the case where (1) A knows EB and B knows EA and (2) They do not know each other knows
- ►Players' knowledge can be incomplete but must be correct
  - Players should not assume something wrong
- ►We need a more powerful model for reasoning about knowledge and strategies

Knowledge Oriented Players (KOP)

- Each player has the same model
  - Unlike in OM search, A and B have different models
- A player $P = (E_P, S_P, K(P))$
  - E_P : evaluation function defined on all leaves
  - S_P : strategy
  - $K(P) = \{\varphi | K_P\varphi\}$ ($K_P\varphi$ read as "P knows φ")
  - K(P): knowledge of P, represented by a maximum consistent set of formulas in the S5n axiom system

The $S5_n$ Axiom System

- A1. All tautologies of the propositional logic
- A2. $(K_P\varphi \wedge (K_P\varphi \Rightarrow \psi)) \Rightarrow K_P\psi$ (deduction)
- A3. $K_P\varphi \Rightarrow \varphi$ (the knowledge axiom)
- A4. $\neg K_P\varphi \Rightarrow K_P\neg K_P\varphi$ (positive introspection)
- A5. $K_P\varphi \Rightarrow K_PK_P\varphi$ (negative introspection)
- More details in (Halpern & Moses 1992)

Atoms and Common Knowledge

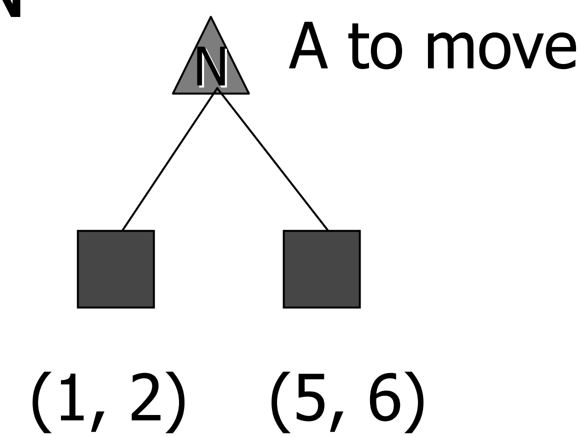
- E_P, S_P, are atoms
  - Either fully known or unknown by a player
  - $K_P E_P$ and $K_P S_P$ always true
- A formula φ is common knowledge iff everybody knows φ, everybody knows that everybody knows φ, ...
  - Common knowledge is treated as tautologies

Levels of Knowledge

Level d	A's knowledge f(A, d)	B's knowledge f(B, d)
0	$K_A E_A$	$K_B E_B$
1	$K_A E_A$, $K_A E_B$	$K_B E_B$, $K_B E_A$
2	$K_A E_A$, $K_A E_B$, $K_A K_B E_A$	$K_B E_B$, $K_B E_A$, $K_B K_A E_B$
…	…	…

K-decide

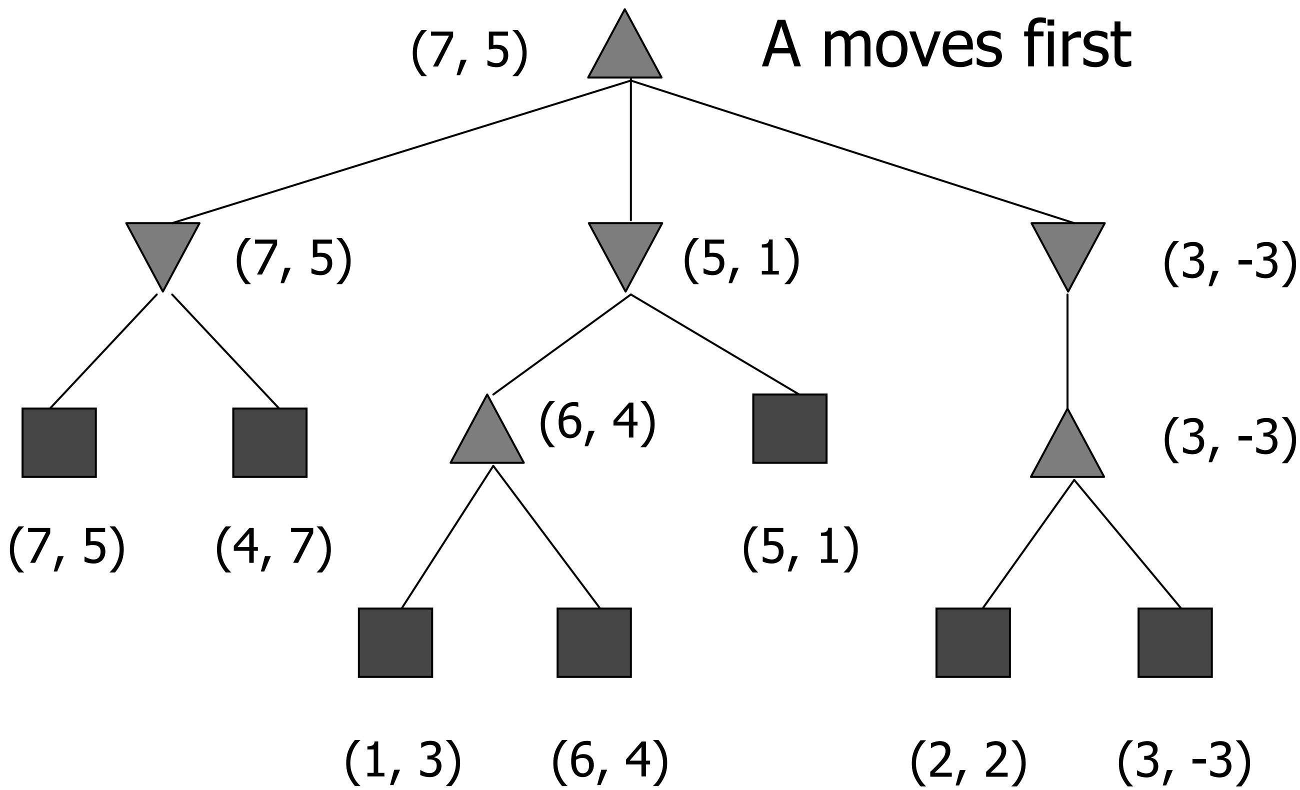
- A node n is K-decided iff it is a leaf or both A and B agree on next(n) and value(n), according to K(A) and K(B)
- For example, whether node N is K-decided depends on if B knows E_A
- There is an algorithm to compute K-decided nodes



K-decide (to skip)

- Theorem. Given a node N in a game tree t, if $f(A,d) \subseteq K(A) \wedge f(B,d) \subseteq K(B) \wedge height(N) \leq d$ then N is K-decided
- Selfish players must use strategies that conform to K-decided nodes
- The Decide algorithm determines whether a node in a tree is K-decided
- When all the nodes in a tree are decided, we get a simple M^h algorithm

The M^h Algorithm



- A (or B) truthfully maximizes (or minimizes) E_A (or E_B)
- The values can be computed from bottom-up
- Final outcome is 7 for A, compared to 5 by Minimax

M^h vs. M^*

- Theorem. M^h is equivalent to M^* on all game trees only when M^* has an infinite number of simulations
- The expected outcome by M^h is guaranteed as least as good as that by Minimax
- If M^* has any wrong simulation (e.g., when a players' knowledge does not subsume the opponent's), the outcome is unpredictable, possibly worse than Minimax

M^h and SPE

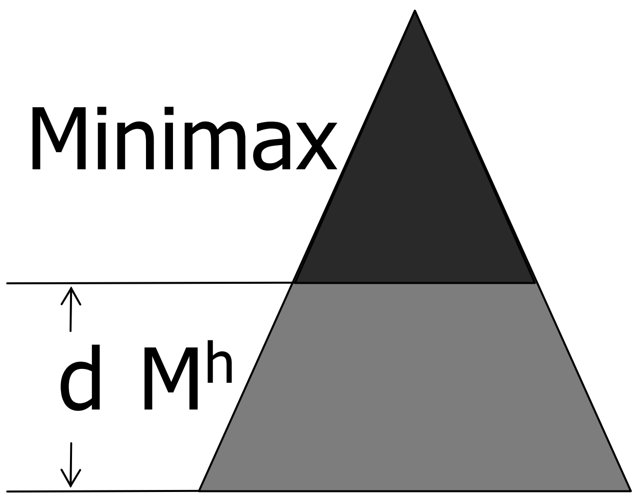
- Theorem. M^h is mutually enforced, that is, you have to use it if your opponent uses it.
  - Similar to the property of Subgame Perfect Equilibrium (SPE) in game theory (Kuhn 1953)
  - However, SPE requires perfect information; in our cases, knowledge is often incomplete
- Theorem. Given t where A moves first, if $f(A, height(t)-1) \subseteq K(A)$ then M^h is the optimal strategy for A
- Common knowledge on E_A and E_B suffices

When Knowledge is Insufficient

- To be optimal, M^h requires "sufficient" knowledge, i.e., $f(A, height(t)-1) \subseteq K(A)$
- When knowledge is insufficient, the outcome of using M^h is NOT guaranteed
- However, at least we can use Minimax to guarantee the worst case outcome in any nonzero-sum game
  - Can we do better?

The Hybrid Minimax-M^h Algorithm

- Idea: Use M^h on K-decided nodes, and use Minimax on other nodes to guarantee the worst case outcome
- With d levels of knowledge, the hybrid algorithm applies Minimax to the nodes with height greater than d, and applies M^h to the nodes with height at most d



- The algorithm takes a parameter d: HM^0M^d

The Hybrid Minimax-M^h Algorithm

- Theorem. Given t where A moves first, if $f(A,d) \subseteq K(A)$ then the HM^0M^d Algorithm is at least as good as Minimax (M^0) alone for A
- The hybrid HM^0M^d algorithm works for "levels of knowledge"; however, the expected outcome by HM^0M^d is at most as good as that by M^h
  - "Sufficient" knowledge wanted for M^h

Knowledge Update via Communication

- New knowledge can be obtained from communication between players
- For example, initially A knows E_A and E_B, and B only knows E_B. Now A announces E_B to B, and then B knows that A knows E_B
- Goal: updated knowledge forces both players to use M^h
- Interesting implication: a player has no incentive to hide his "secret"

Two communication schemes

1. A announces both E_A and E_B to B, so that E_A and E_B become common knowledge
   - Requiring $d \geq 1 \wedge f(A,d) \subseteq K(A)$ and Public Announcement Logic with relativized common knowledge (PAL-RC) (van Benthem et al. 2005)
2. A announces that A is using M^h
   - Requiring $d \geq 2 \wedge f(A,d) \subseteq K(A)$
   - A knows that B will be able to verify A's M^h, which forces B to use M^h as well

Conclusion

- A new framework of Knowledge Oriented Players
  - Incomplete knowledge, unusual in game theory
- New algorithms for guaranteed better outcome than Minimax
  - M^h for sufficient knowledge
  - Hybrid Minimax-M^h for insufficient knowledge
- Communication schemes for knowledge update: forcing the optimal strategy

Future Work

- ►Knowledge with uncertainty
- ►Partial knowledge of evaluation functions
- ►Other constraints between knowledge and strategies
- ►Relation to mixed strategy in game theory