Lecture 7:
Extensive-form game

Yi, Yung (이융)
KAIST, Electrical Engineering
http://lanada.kaist.ac.kr
yiyung@kaist.edu

Contents

- So far, “Simultaneous play”
- What happens if a game is played sequentially
  - One player can see what other player chooses, and then decides on its strategy
- Equilibrium?
- What other issues?

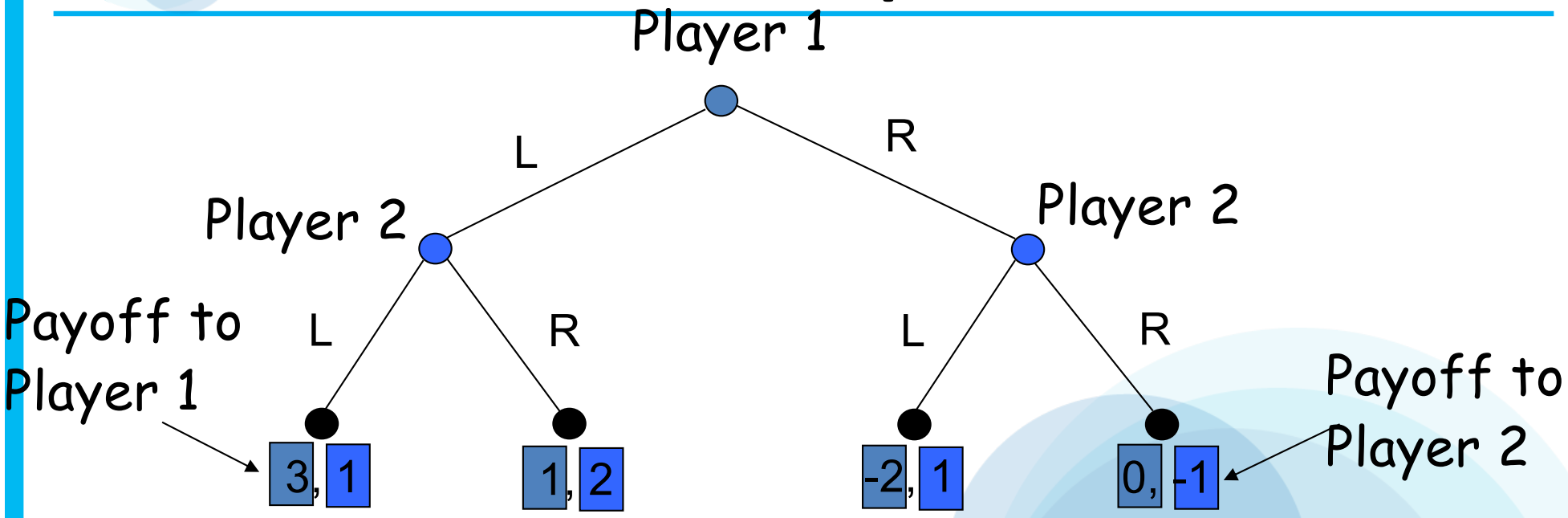
Sequential Game

- A major class of “dynamic games”, where players take their de cisions in a certain predefined order
- Role of information at each stage: very important
  - Perfect/imperfect information
- Thus, distinguish between action vs. strategy
  - Example: if an individual has to decide what to do in the evening, and the options are camping or staying at home;
  - Strategy: “If the weather report predicts dry weather for the evening, then I will go out camping; otherwise, I will stay at home”
  - Action: After knowing about the weather, the individual would take an action

Game Trees (Extensive form)

- Extensive-form (i.e., tree)
  - Most useful representation of sequential games
  - Discrete strategy space
- Game represented as a tree
  - each non-leaf node represents a decision point for some player
  - edges represent available choices
- Can be converted to matrix game (Normal form)
  - “plan of action” must be chosen beforehand

Game Trees Example



- Strategy set for Player 1: {L, R}
- Strategy for Player 2:

what to do when P1 plays L

what to do when P1 plays R

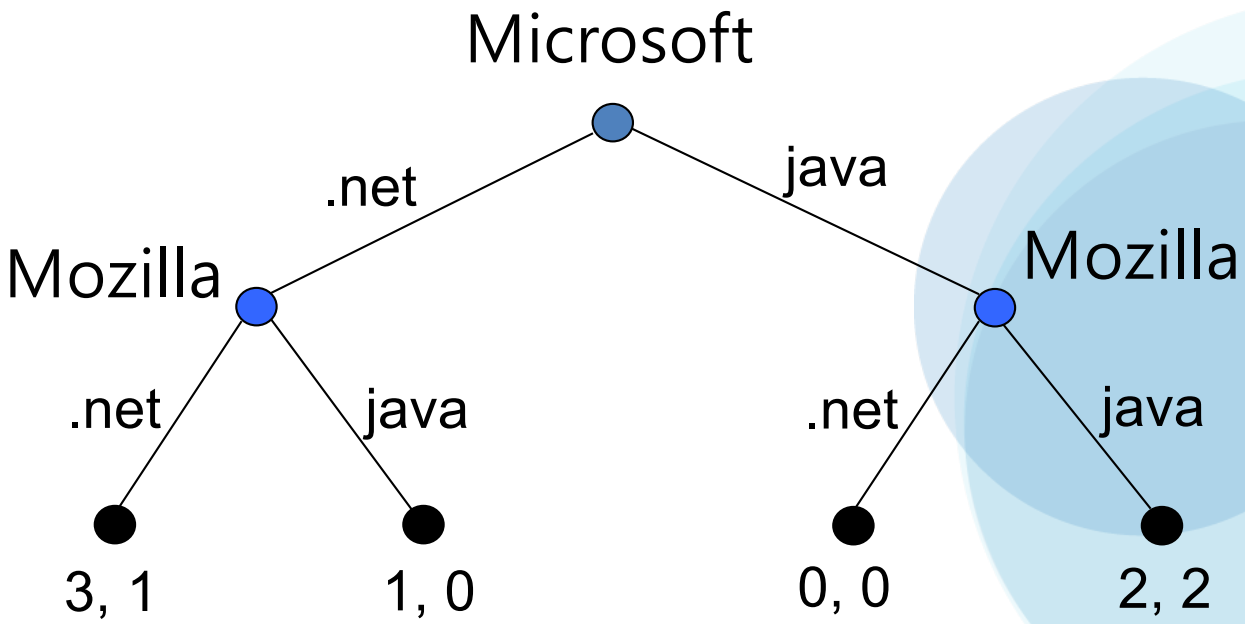
- Strategy set for Player 2: {LL, LR, RL, RR}

More Formal Extensive Game Definition

- An extensive form game
  - a finite set *N* of players
  - a finite height game tree
  - payoff function $u_i(s)$ for each player $i \in N$
    - where *S* is a leaf node of game tree
- Game tree: set of nodes and edges
  - each non-leaf node represents a decision point for some player
  - edges represent available choices
- Perfect information
  - all players have full knowledge of game history

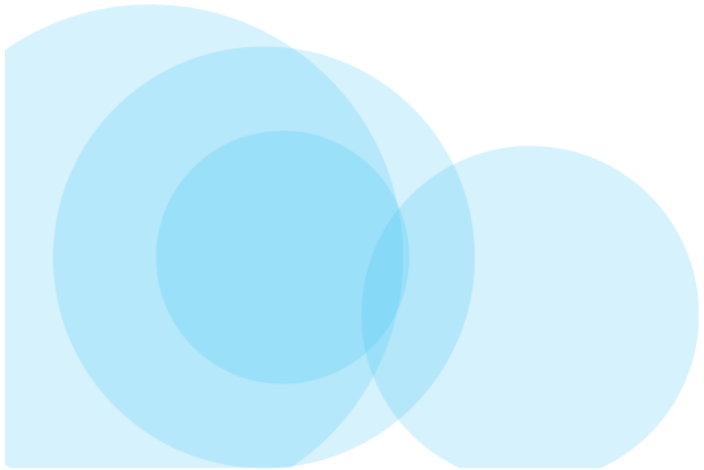
Game Tree Example

- Microsoft and Mozilla are deciding on adopting new browser technology (.net or java)
  - Microsoft moves first, then Mozilla makes its move

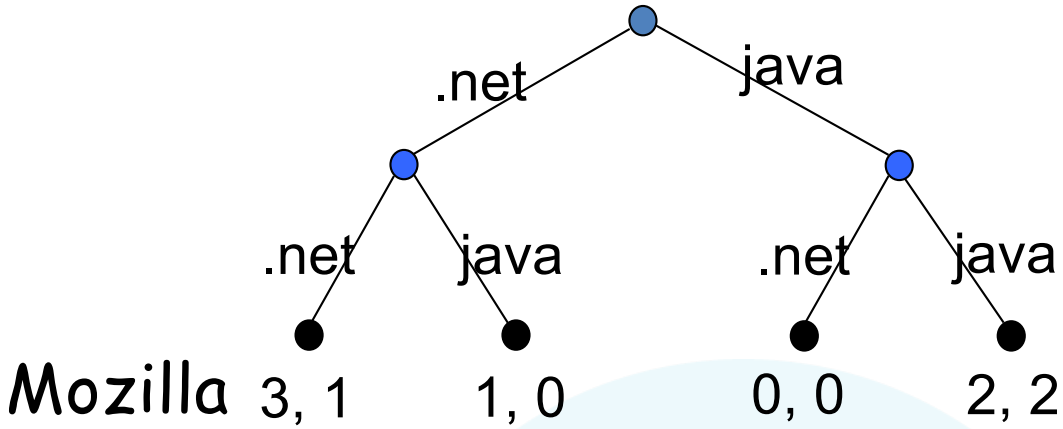


- Non-zero sum game
  - what are the NEP?

Can we look at an extensive-form game from its associated normal form game?



Converting to Matrix Game

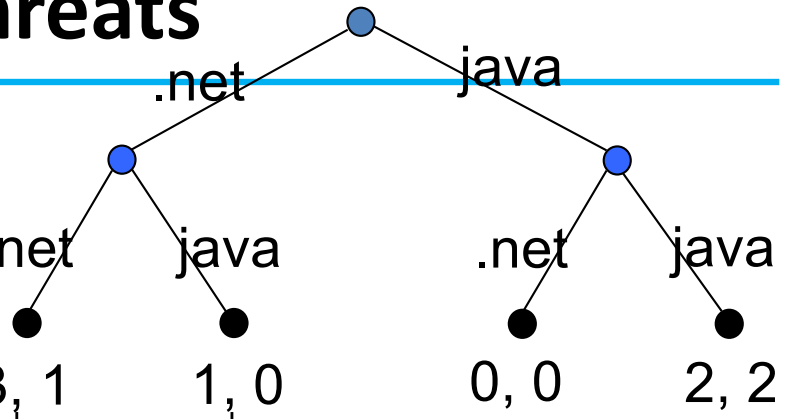


Microsoft

	.net, .net	.net, java	java, .net	java, java
.net	3, 1	3, 1	1, 0	1, 0
java	0, 0	2, 2	0, 0	2, 2

- Every game in extensive form can be converted into a normal form
  - exponential growth in number of strategies

NEP and Incredible Threats



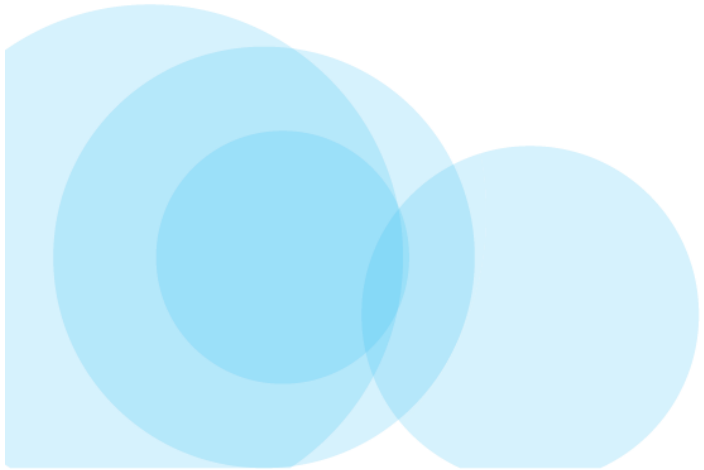
Microsoft \ Mozilla	.net, .net	.net, java	java, .net	java, java
.net	3, 1	3, 1	1, 0	1, 0
java	0, 0	2, 2	0, 0	2, 2

NEP

incredible threat

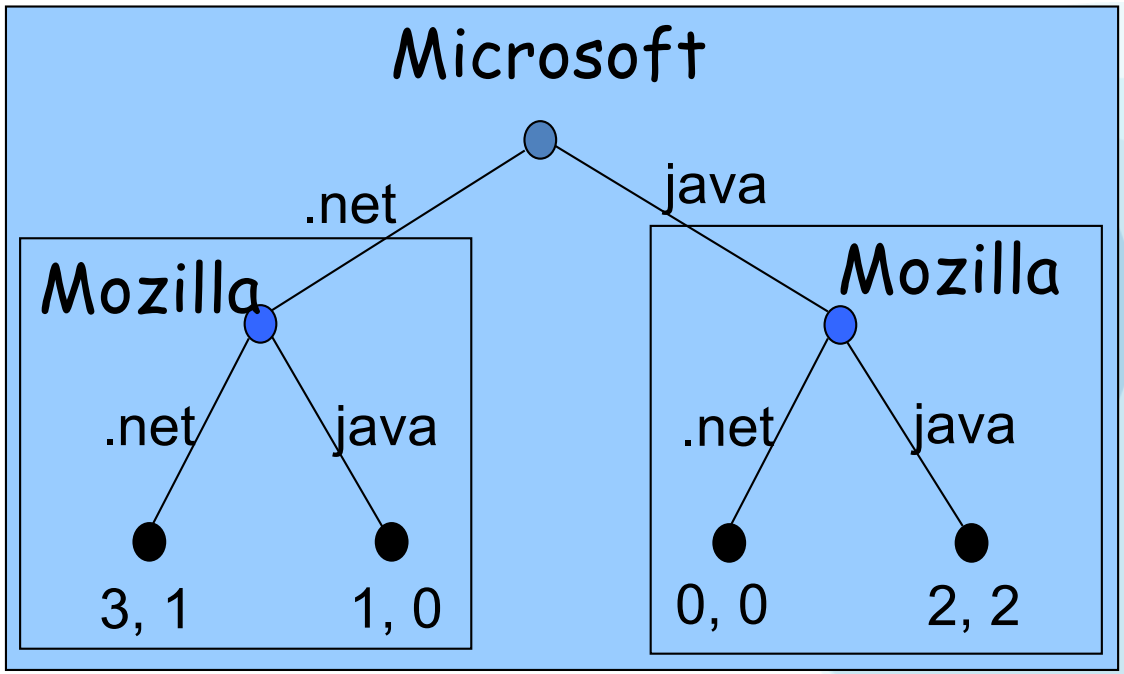
- Play “java no matter what” is not credible for Mozilla
  - if Microsoft plays .net then .net is better for Mozilla than java

How should I characterize “real equilibriums” in extensive-form games?



The Subgame Concept

- Def: a subgame is any subtree of the original game that also defines a proper game
  - includes all descendents of non-leaf root node



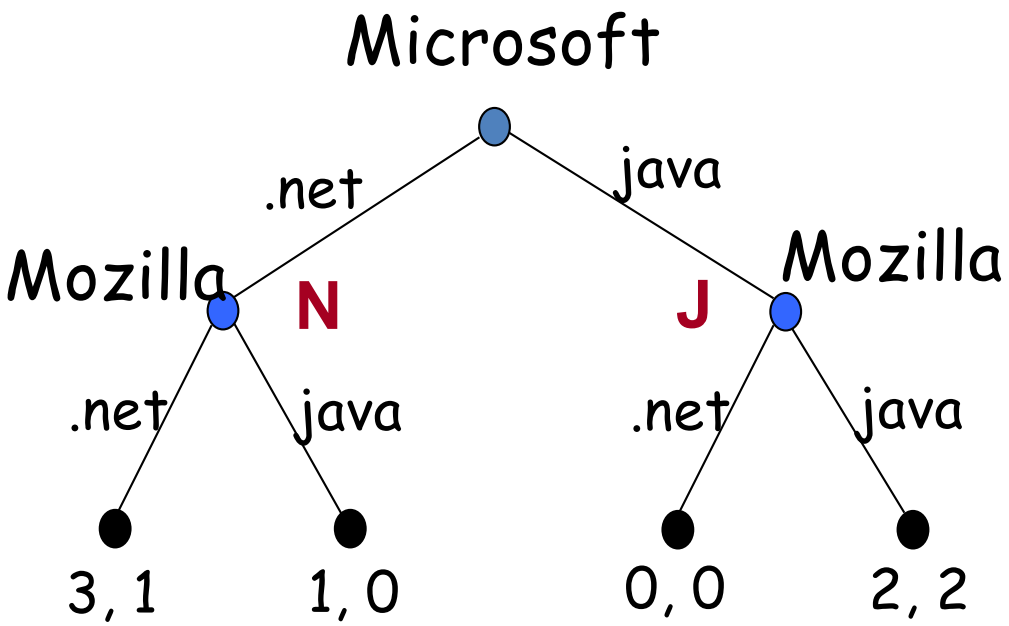
- 3 subtrees
  - full tree, left tree, right tree

Subgame Perfect Nash Equilibrium

- Def: a NEP is *subgame perfect* if its restriction to *every* subgame is also a NEP of the subgame

- Thr: Every extensive form game has at least one subgame perferct Nash equilibrium
  - Kuhn's theorem, based on backward induction

Subgame Perfect Nash Equilibrium



- (N, NN) is not a NEP when restricted to the subgame starting at **J**
- (J, JJ) is not a NEP when restricted to the subgame starting at **N**
- (N, NJ) is a subgame perfect Nash equilibrium

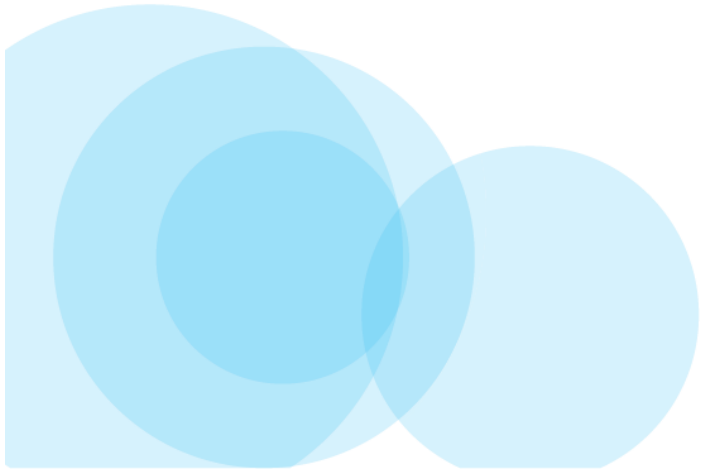
Mozilla

MS	NN	NJ	JN	JJ
N	3,1	3,1	1,0	1,0
J	0,0	2,2	0,0	2,2

Subgame Perfect NEP

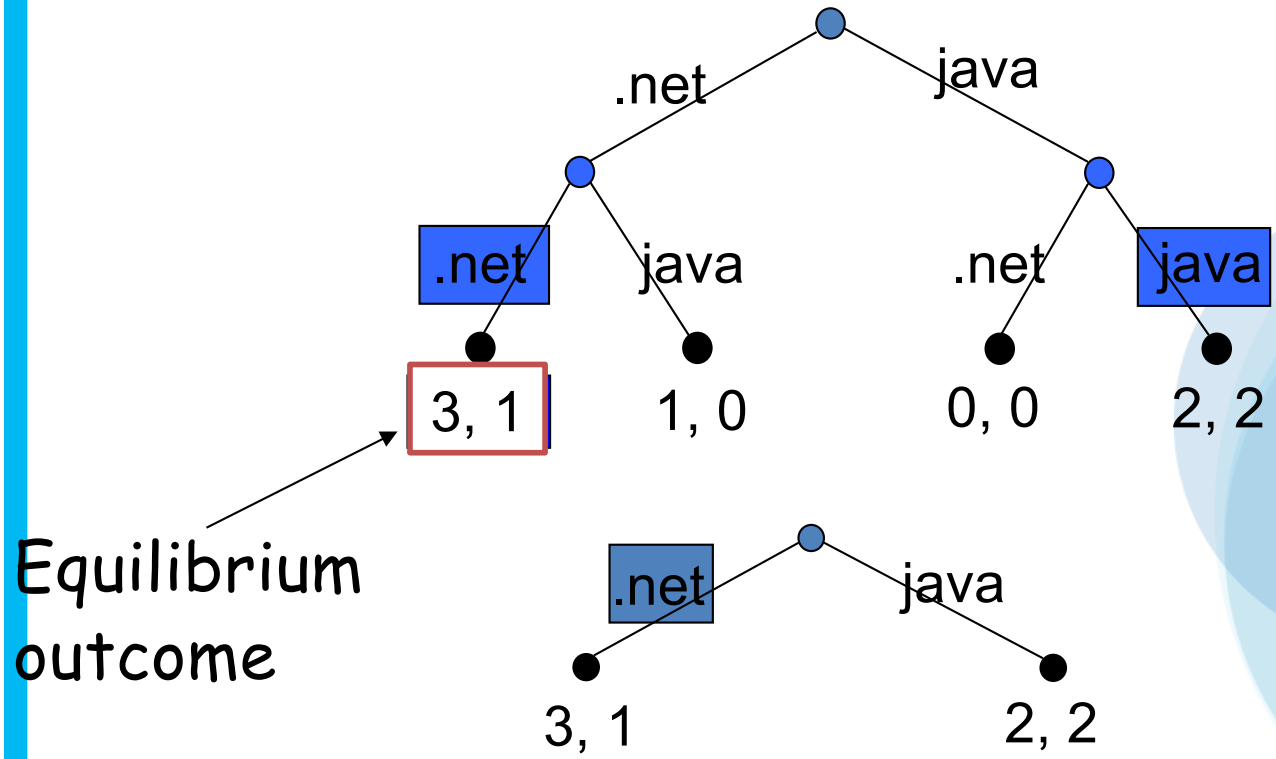
Not subgame Perfect NEP

What is a good algorithm to find equilibria in extensive-form game?



Solving the Game (Backward Induction)

- Starting from terminal nodes
  - move up the game tree making best choice



Best strategy for Mozilla: .net, java (follow Microsoft)

Best strategy for Microsoft: .net

- Single NEP
  - Microsoft → .net, Mozilla → .net, java

Backward Induction on Game Trees

- Kuhn's Thr: Backward induction always leads to a saddle point (on games with perfect information)
  - Saddle point = subgame perfect NEP
  - game value at equilibrium is unique (for zero-sum games)
- In general, multiple NEPs are possible after backward induction
  - cases with no strict preference over payoffs
- Effective mechanism to remove "bad" NEP
  - incredible threats

Summary