OTEVŘENÁ INFORMATIKA

Solving Extensive-Form Games

Branislav Bošanský

AE4M36MAS, Fall 2015

Extensive-Form Games

Perfect-Information Games

Perfect-Information Games with Chance

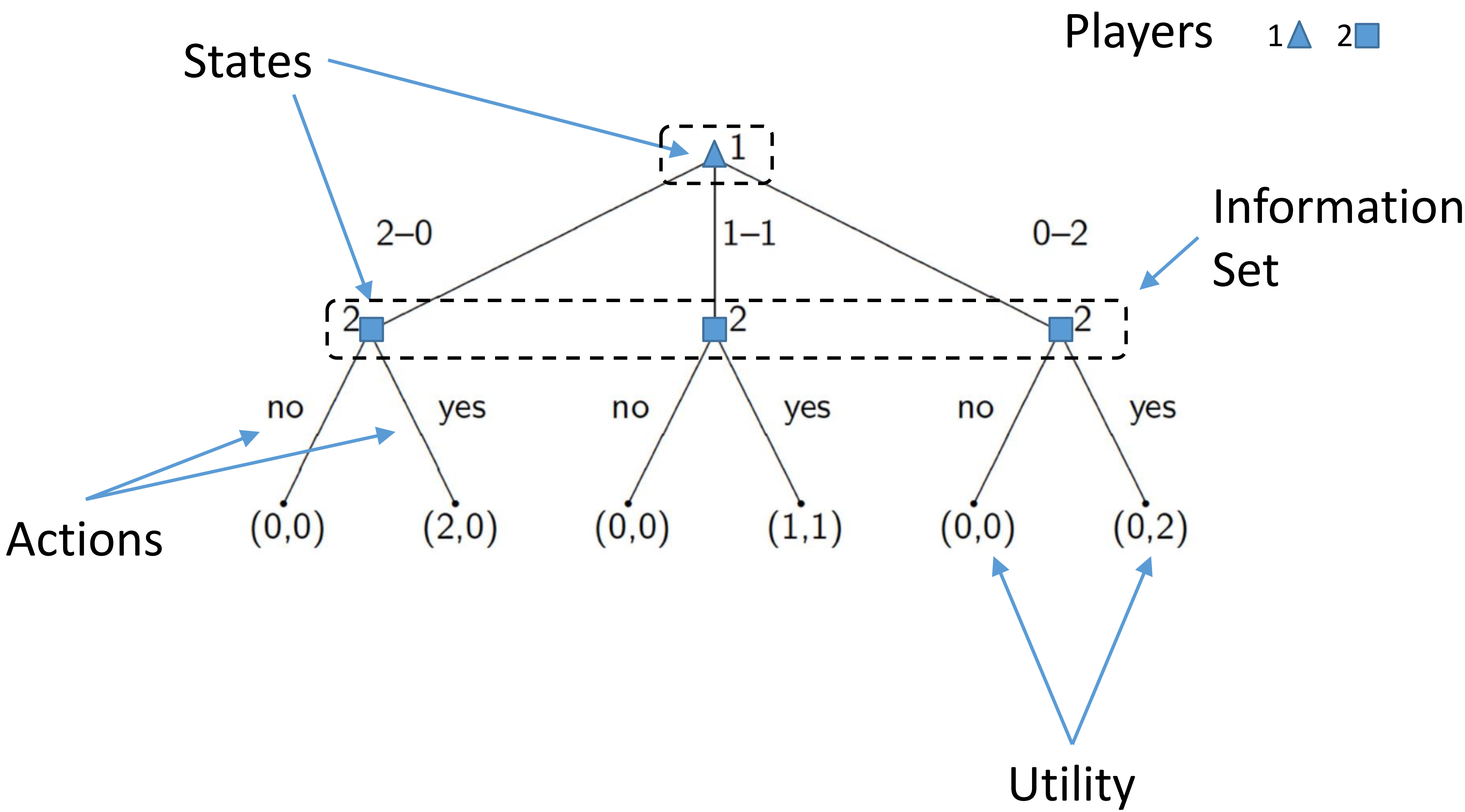
Imperfect-Information Games

Solving Zero-Sum Games

Solving General-Sum Games

Approximate Solutions to Large Zero-Sum Games

Imperfect Information EFGs



Solving II Zero-Sum EFGs with perfect recall

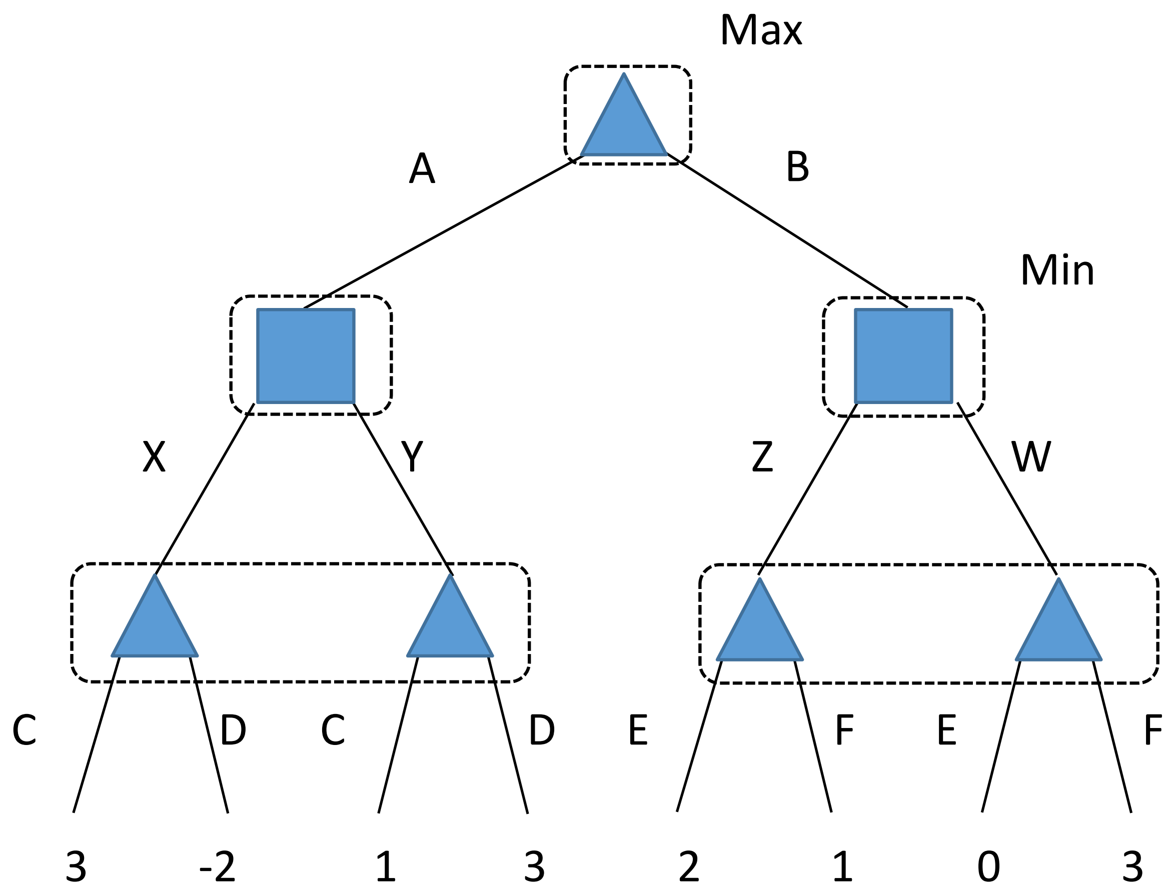
Exact algorithms:

- Why backward induction does not work?
- Transformation to the normal form
- Using the sequence form (Koller et al. 1996, von Stengel 1996)
- Iterative extensions (a.k.a. double-oracle algorithms (McMahan et al. 2006)) of the sequence form (Bosansky et al. 2014)

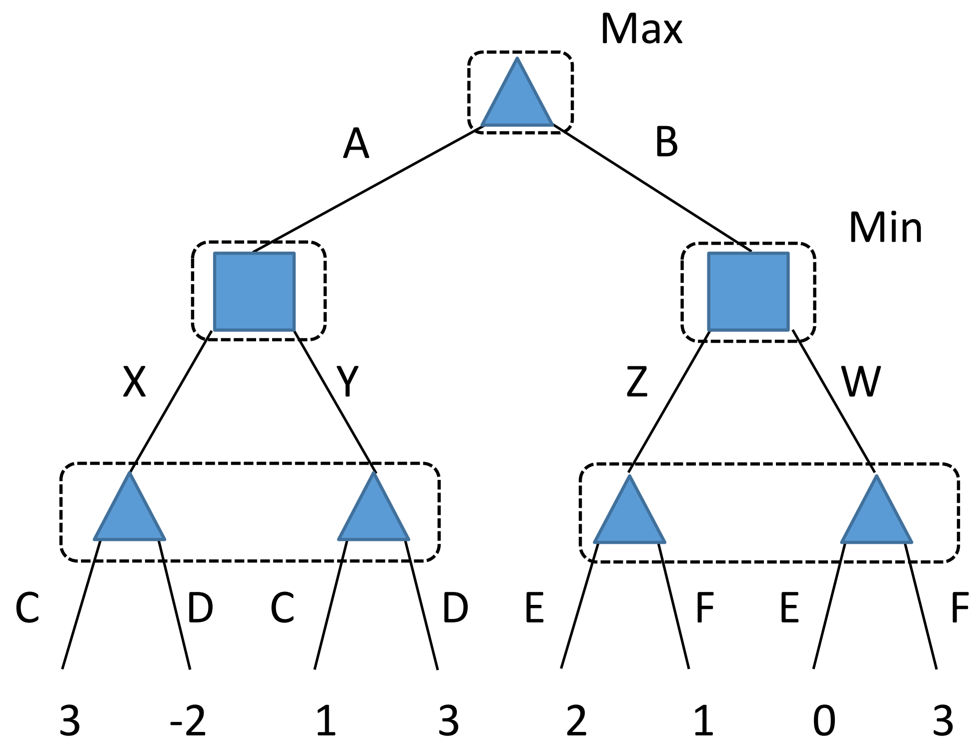
Approximate algorithms:

- Counterfactual Regret Minimization (Zinkevich et al. 2008, Lanctot et al. 2009, Bowling et al. 2015)
- Excessive Gap Technique (Hoda et al. 2010, Waugh et al. 2015)

Imperfect Information Zero-Sum EFG

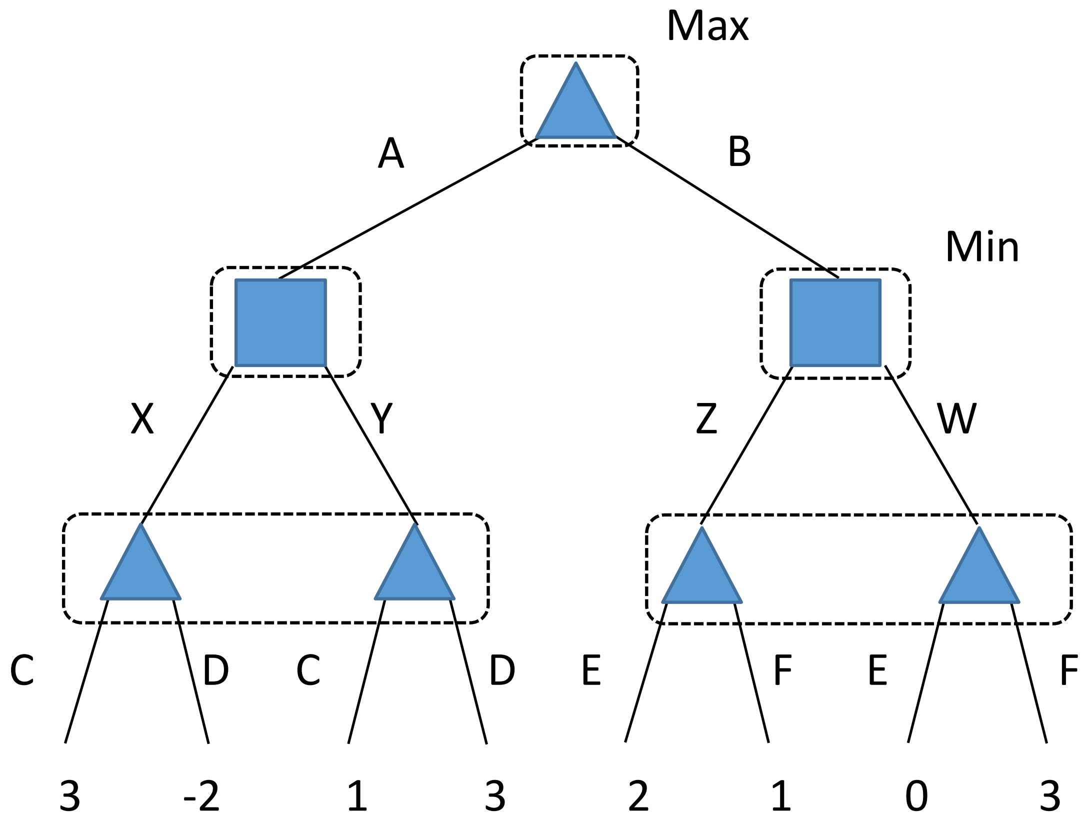


Imperfect Information Zero-Sum EFG



	XZ	XW	YZ	YW
ACE	3	3	1	1
ACF	3	3	1	1
ADE	-2	-2	3	3
ADF	-2	-2	3	3
BCE	2	0	2	0
BCF	1	3	1	3
BDE	2	0	2	0
BDF	1	3	1	3

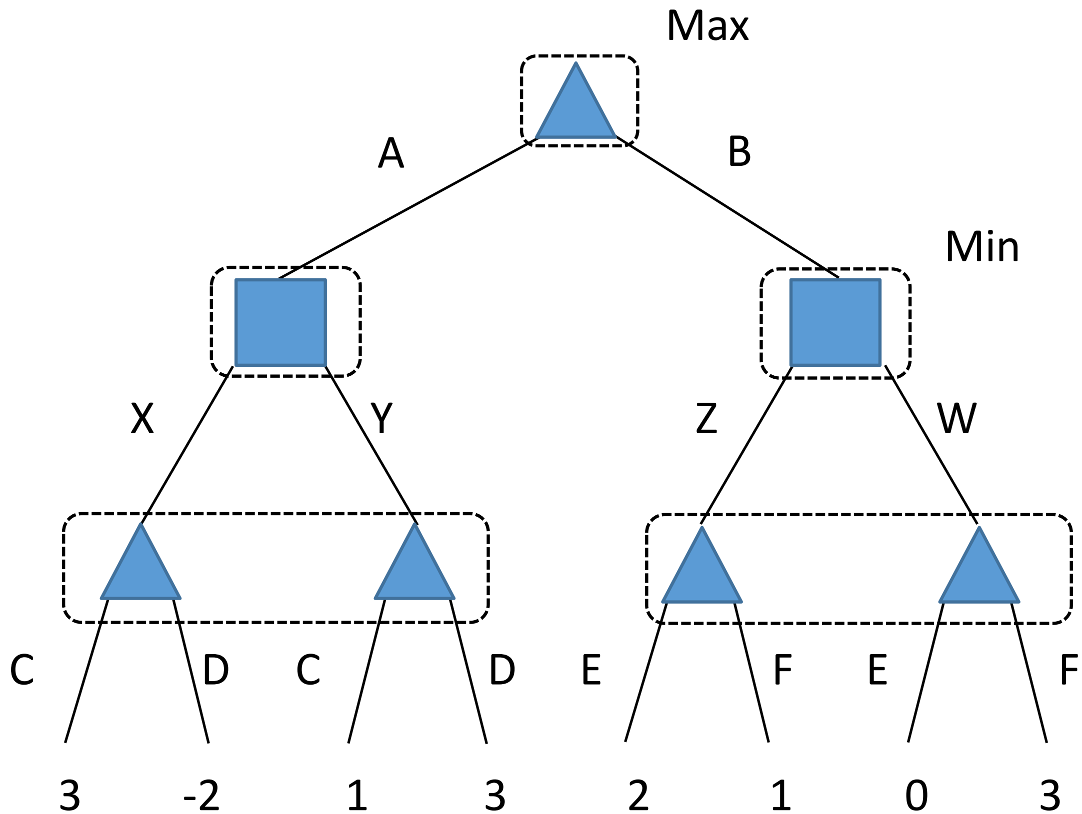
II EFGs - Sequences



Triangle ($\boldsymbol{\Sigma}_1$)	Box ($\boldsymbol{\Sigma}_2$)
∅	∅
A	X
B	Y
AC	Z
AD	W
BE	
BF	

- alternative representation of strategies
- $\sigma_i \in \Sigma_i$
- we use $\sigma_i a$ to denote executing an action a after the sequence σ_i

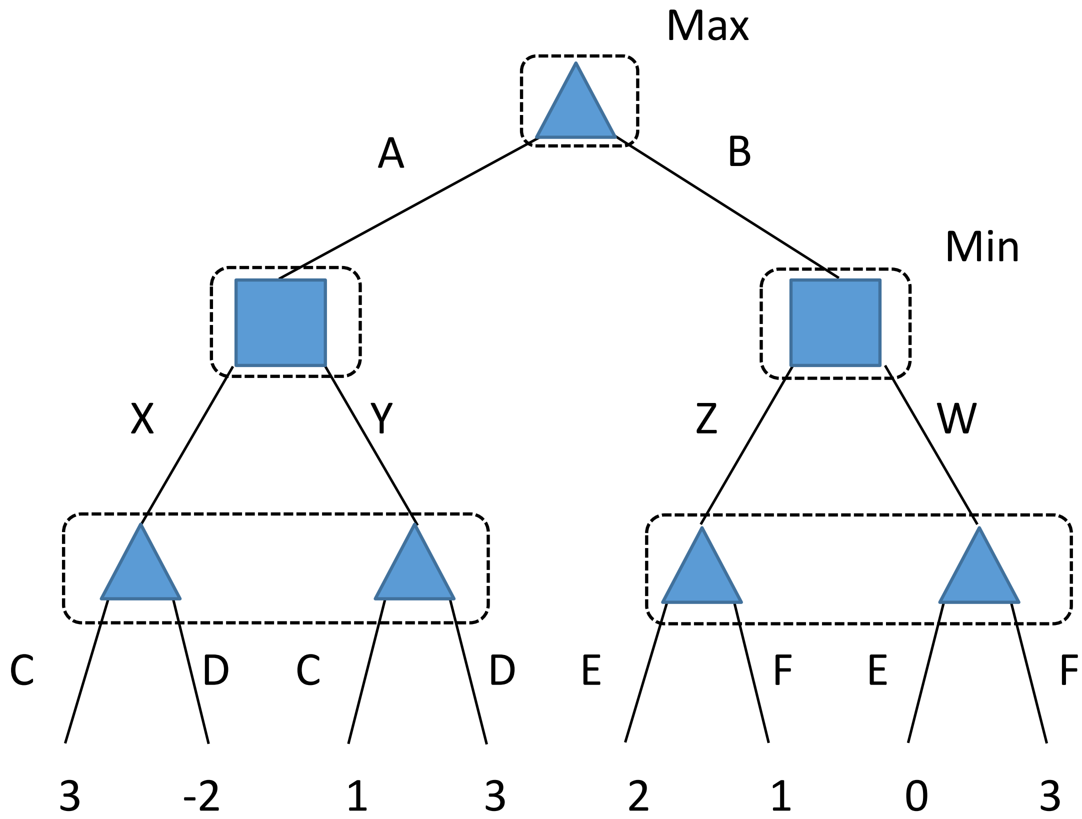
II EFGs - Sequences



Triangle (Σ_1)	**Box (Σ_2)**
∅	∅
A	X
B	Y
AC	Z
AD	W
BE	
BF	

- extension of the utility function *g*
  - $g_i\colon \Sigma_1 \times \Sigma_2 \rightarrow \mathbb{R}$
- sequentially execute actions of the players
  - stop at either:
    - leaf - $z \in Z$ $g_i(\sigma_1, \sigma_2) = u_i(z)$
    - there is no applicable action $g_i(\sigma_1, \sigma_2) = 0$

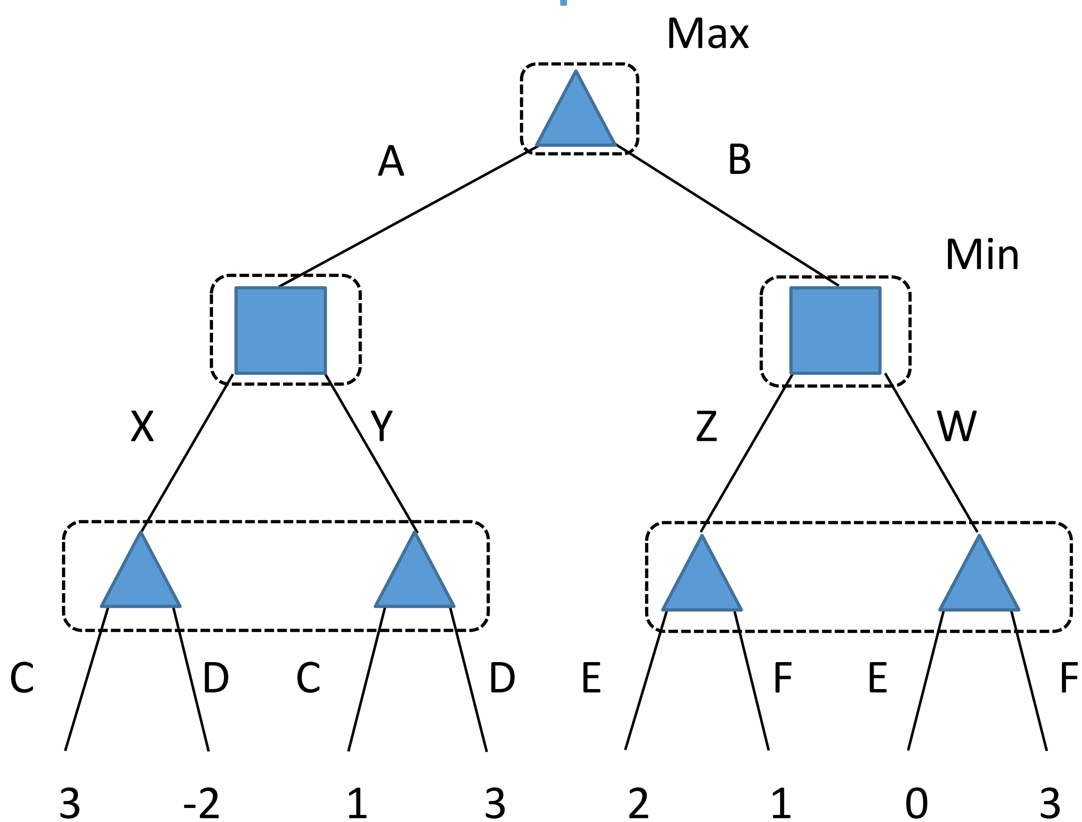
II EFGs - Sequences



Triangle (Σ_1)	Box (Σ_2)
∅	∅
A	X
B	Y
AC	Z
AD	W
BE	
BF	

- In EFGs with chance nodes
  - g corresponds to an expected utility of **all reachable leafs** (Z')
  - $g(\sigma_1, \sigma_2) = \sum_{z \in Z'} u_i(z)\gamma(z)$
    where γ is the probability of Nature playing a sequence of actions reaching leaf $z \in Z'$

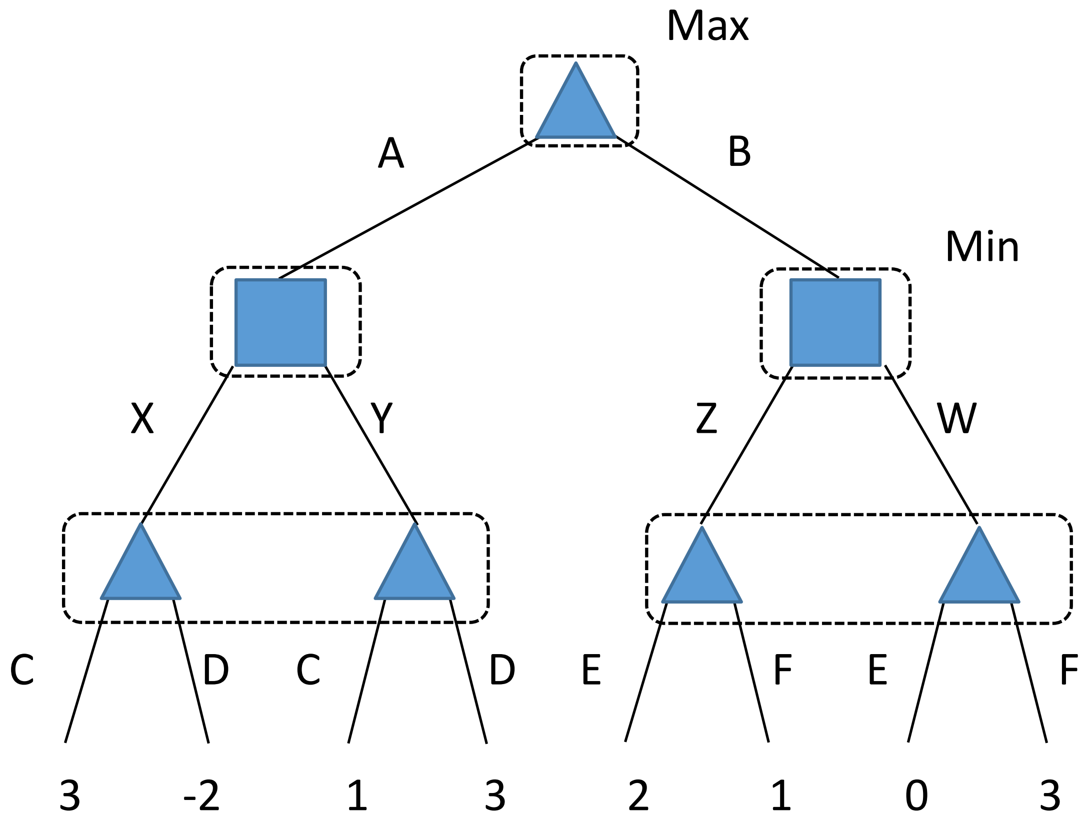
II EFGs - Sequences



Triangle (Σ_1)	Box (Σ_2)
∅	∅
A	X
B	Y
AC	Z
AD	W
BE	
BF	

- Examples
  - $g_1(\emptyset, W) = 0$
  - $g_1(AC, W) = 0$
  - $g_1(BF, W) = 3$
  - …

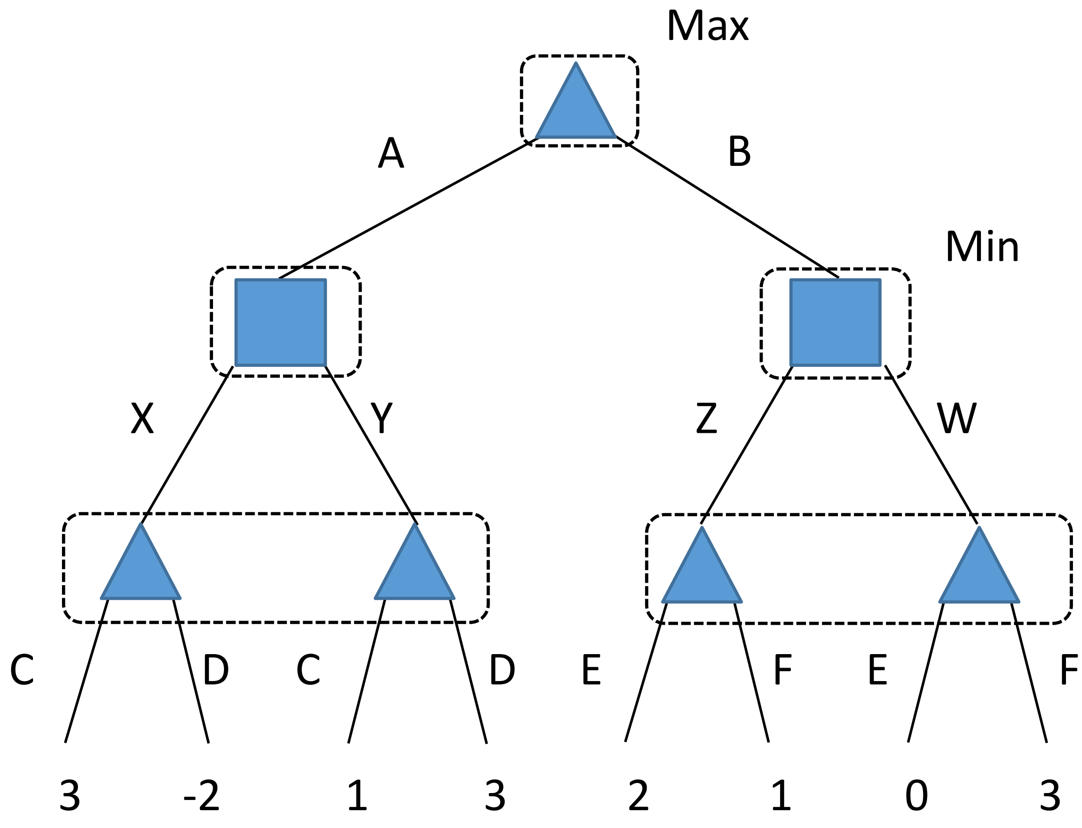
II EFGs – Realization Plans



Triangle (Σ_1)	**Box (Σ_2)**
∅	∅
A	X
B	Y
AC	Z
AD	W
BE	
BF	

- behavioral strategies represented **as realization plans**
  - probabilities over sequences of actions
  - assuming the opponent allows us to play the actions from the sequence

II EFGs – Realization Plans



Triangle (Σ_1)	Box (Σ_2)
Ø	Ø
A	X
B	Y
AC	Z
AD	W
BE	
BF	

- $r_1(\emptyset) = 1$
- $r_1(A) + r_1(B) = r_1(\emptyset)$
- $r_1(AC) + r_1(AD) = r_1(A)$
- $r_1(BE) + r_1(BF) = r_1(B)$

- $r_2(\emptyset) = 1$
- $r_2(X) + r_2(Y) = r_2(\emptyset)$
- $r_2(Z) + r_2(W) = r_2(\emptyset)$

- network-flow perspective

II EFGs – Sequence Form LP

- NE of a zero-sum game can be found by solving sequence form LP
  - finding the best realization plan r_1 against a best-responding player 2
  - $\mathcal{I}(\sigma)$ – information set, in which the last action of sequence σ was executed
  - $seq(I)$ – sequence leading to an information set I
  - v_I - expected utility in an information set

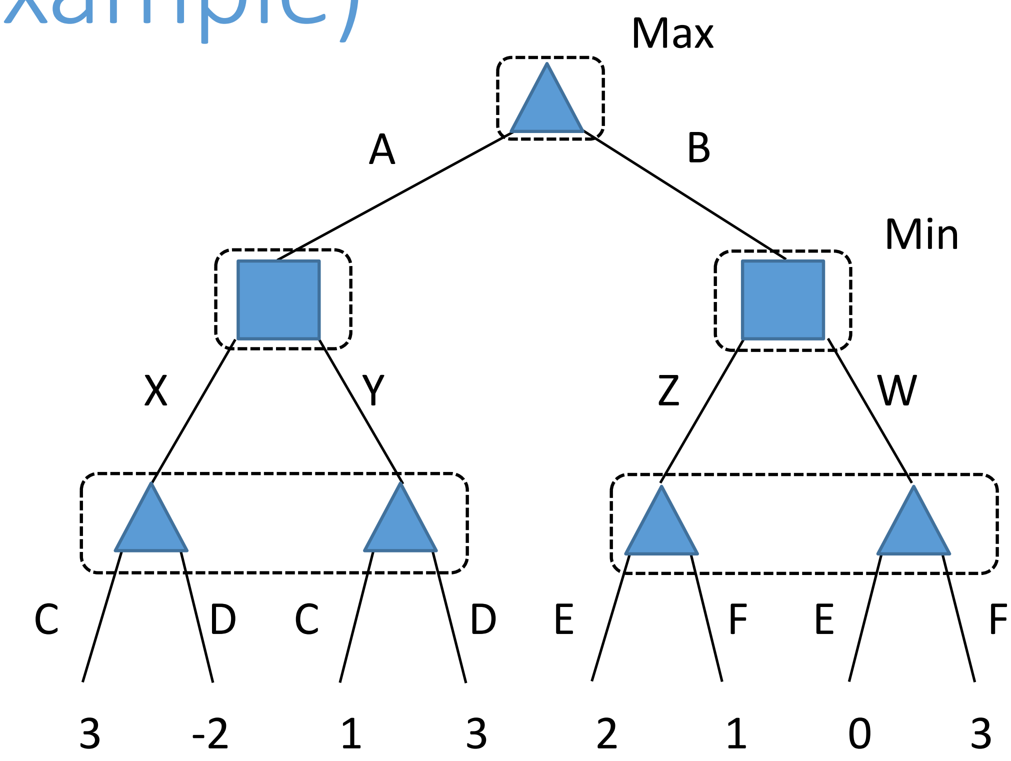
$$\max_{\mathrm{r}_1, v} v_{\mathcal{I}(\emptyset)}$$

$$r_1(\emptyset) = 1, 0 \le r_1(\sigma) \le 1 \qquad \forall \sigma \in \Sigma_1$$

$$r_1(\sigma) = \sum_{a \in \chi(I_{1,k})} r_1(\sigma a) \qquad \forall I_{1,k} \in I_1, \sigma = seq(I_{1,k})$$

$$v_{\mathcal{I}(\sigma_2)} \le \sum_{I_{2,j} | seq(I_{2,j}) = \sigma_2} v_{I_{2,j}} + \sum_{\sigma_1 \in \Sigma_1} g_1(\sigma_1, \sigma_2) r_1(\sigma_1) \quad \forall \sigma_2 \in \Sigma_2$$

Sequence Form LP (example)



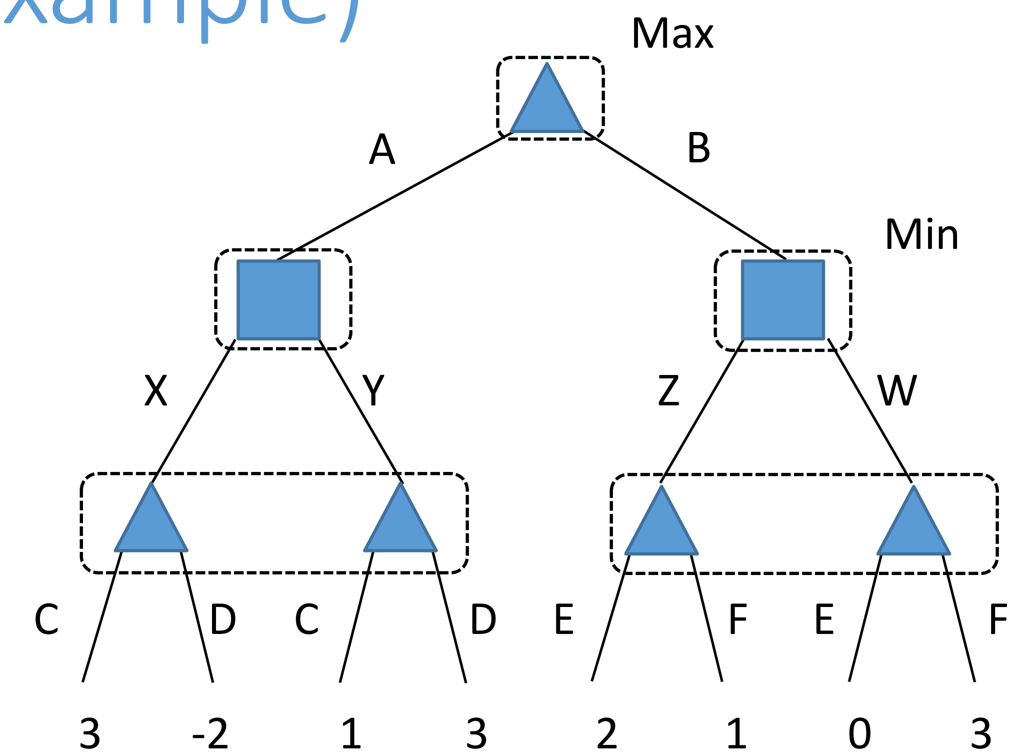
$$\max_{r_1, v} v_{\mathcal{J}(X)} + v_{\mathcal{J}(Z)}$$

$$r_1(\emptyset) = 1, r_1(A) + r_1(B) = r_1(\emptyset),$$
$$r_1(A) = r_1(AC) + r_1(AD)$$
$$r_1(B) = r_1(BE) + r_1(BF)$$

$$v_{\mathcal{J}(X)} \le 0 + g(AC, X) \cdot r_1(AC) + g(AD, X) \cdot r_1(AD)$$
$$v_{\mathcal{J}(Y)} \le 0 + g(AC, Y) \cdot r_1(AC) + g(AD, Y) \cdot r_1(AD)$$
$$v_{\mathcal{J}(Z)} \le 0 + g(BE, Z) \cdot r_1(BE) + g(BF, Z) \cdot r_1(BF)$$
$$v_{\mathcal{J}(W)} \le 0 + g(BE, W) \cdot r_1(BE) + g(BF, W) \cdot r_1(BF)$$

- note that $\mathcal{J}(X) = \mathcal{J}(Y)$ and $\mathcal{J}(Z) = \mathcal{J}(W)$

Sequence Form LP (example)



$$\min_{r_2, v} v_{\mathcal{I}(A)}$$

$$r_2(\emptyset) = 1, r_2(X) + r_2(Y) = r_2(\emptyset),$$
$$r_2(Z) + r_2(W) = r_2(\emptyset)$$

$$v_{\mathcal{I}(A)} \geq v_{\mathcal{I}(AC)}, v_{\mathcal{I}(B)} \geq v_{\mathcal{I}(BD)}$$
$$v_{\mathcal{I}(AC)} \geq g(AC, X) \cdot r_2(X) + g(AC, Y) \cdot r_2(Y)$$
$$v_{\mathcal{I}(AD)} \geq g(AD, X) \cdot r_2(X) + g(AD, Y) \cdot r_2(Y)$$
$$v_{\mathcal{I}(BE)} \geq g(BE, Z) \cdot r_2(Z) + g(BE, W) \cdot r_2(W)$$
$$v_{\mathcal{I}(BF)} \geq g(BF, Z) \cdot r_2(Z) + g(BF, W) \cdot r_2(W)$$

- note that $\mathcal{I}(A) = \mathcal{I}(B), \mathcal{I}(AC) = \mathcal{I}(AD),$ and $\mathcal{I}(BE) = \mathcal{I}(BF)$

Extensive-Form Games

Perfect-Information Games

Perfect-Information Games with Chance

Imperfect-Information Games

Solving Zero-Sum Games

Solving General-Sum Games

Approximate Solutions to Large Zero-Sum Games

General Sum EFGs – Sequence Form LCP

- NE of a general-sum game can be found by solving a sequence form LCP (linear complementarity problem)
  - satisfiability program
  - realization plans for both players
  - connection between realization plans and best responses via complementarity constraints
  - best-response inequalities are rewritten using slack variables

$$r_i(\emptyset) = 1, 0 \le r_i(\sigma_i) \le 1 \qquad \forall i \in N, \forall \sigma_i \in \Sigma_i$$

$$r_i(\sigma_i) = \sum_{\{a \in \chi(I_{i,j})\}} r_i(\sigma_i a) \qquad \forall i \in N, \forall I_{i,j} \in I_i, \sigma_i = seq(I_{i,j})$$

$$v_{\mathcal{I}(\sigma_i)} = s_{\sigma_i} + \sum_{\{I_{i,j}: seq(I_{i,j}) = \sigma_i\}} v_{I_{i,j}} + \sum_{\sigma_{-i} \in \Sigma_{-i}} g_i(\sigma_i, \sigma_{-i}) r_i(\sigma_i) \quad \forall i \in N, \forall \sigma_i \in \Sigma_i$$

$$r_i(\sigma_i) \cdot s_{\sigma_i} = 0 \qquad \forall i \in N, \forall \sigma_i \in \Sigma_i$$

$$0 \le s_{\sigma_i} \qquad \forall i \in N, \forall \sigma_i \in \Sigma_i$$

General Sum EFGs – practical algorithms

- computing one (any) NE
  - Lemke algorithm

- computing some specific NE
  - e.g., maximizing welfare, maximizing utility for some player, …
  - MILP reformulations (Sandholm et al. 2005, Audet et al. 2009)
    - complementarity constraints can be replaced by using a binary variable that represents whether a sequence is used in a strategy with a non-zero probability
    - big-M notation
  - poor performance (10^4 nodes) using state-of-the-art MILP solvers (e.g., IBM CPLEX, …)

Extensive-Form Games

- Perfect-Information Games
- Perfect-Information Games with Chance
- Imperfect-Information Games
- Solving Zero-Sum Games
- Solving General-Sum Games
- Approximate Solutions to Large Zero-Sum Games

Approximate algorithms - CFR

- we can learn the best strategy to play
- learning is done via repeated self-play
- under certain conditions we approximate the optimal (NE) strategy

- we restrict to zero-sum games
- no-regret learning
- construct the complete game tree
    - in each iteration traverse through the game tree and adapt the strategy in each information set according to the learning rule
    - this learning rule minimizes the (counterfactual) regret
    - the algorithm minimizes the overall regret in the game
    - the average strategy converges to the optimal strategy

Regret and Counterfactual Regret

- player i's regret for not playing an action a_i' against the opponent's action a_{-i}

$$u_i(a'_i, a_{-i}) - u_i(a_i, a_{-i})$$

- in extensive-form games we need to evaluate the value for each action in an information set (**counterfactual value**):

$$v_i(s, I) = \sum\nolimits_{z \in Z_I} \pi_{-i}^{s}(z[I]) \pi_i^s(z|z[I]) u_i(z)$$

  - Z_I are the leafs reachable from I
  - $z[I]$ is the history prefix of z in I
  - $\pi_i^s(\mathrm{h})$ is the probability of player i reaching node h following strategy s

Regret and Counterfactual Regret

- counterfactual value for one deviation in information set I; strategy s is altered in information set I by playing action a: $v_i(s_{I\to a}, I)$
- at a time step t, the algorithm computes **counterfactual regret** for current strategy

$$r_i^t(I,a) = v_i(s_{I\to a}^t, I) - v_i(s^t, I)$$

- the algorithm calculates the **cumulative regret**

$$R_i^T(I,a) = \sum_{t=1}^{T} r_i^t(I,a), \qquad R_i^{T,+}(I,a) = \max\{R_i^T(I,a), 0\}$$

- strategy for new iteration is selected using **regret matching**

$$s_i^{t+1}(I,a) = \begin{cases} \dfrac{R_i^{T,+}(I,a)}{\sum_{a'\in\chi(I)} R_i^{T,+}(I,a')} & \text{if the denominator is positive} \\[2ex] \dfrac{1}{|\chi(I)|} & \text{otherwise} \end{cases}$$

Regret and Counterfactual Regret

- average cumulative regret converges to zero with iterations

$$\bar{R}_i^T \leq \frac{\Delta_{i,u}|I_i|\sqrt{\max_k |\chi(I_{i,k})|}}{\sqrt{T}}$$

- average strategy converges to optimal strategy
- many additional improvements (sampling, MC versions, …)
- for details see PhD thesis by Marc Lanctot (2013)

- modification of CFR (CFR+) was used to solve two-player limit poker (Bowling et al. 2015)
  - uses only positive updates of regret
  - instead of the average strategy the algorithm uses the immediate (or current) strategy
  - the immediate strategy does not (provably) converge to NE

Comparison SQF vs. CFR

SQF (and iterative variants)

- the leading exact algorithm
- suffers from memory requirements
- memory is reduced with double-oracle variants
- these work best for games with small support

CFR

- leading algorithm in practice
- memory-efficient
- robust and applicable in more general settings
- average strategy converges slowly

Open Questions in EFGs

- very active and challenging sub-field of computational game theory
  - When does the current strategy in CFR+ converge in zero-sum EFGs?
  - What is the expected number of iterations of double-oracle algorithms?
  - How to solve games with imperfect recall?
  - What is the optimal strategy to use in general-sum EFGs? (opponent modeling)
  - …