Describing Imperfect-Information Games

Extended Game Description Language GDL-II

- `role(r)` means that `r` is a role (i.e. a player) in the game
- `init(f)` means that `f` is true in the initial position (state)
- `true(f)` means that `f` is true in the current state
- `does(r,a)` means that role `r` does action `a` in the current state
- `next(f)` means that `f` is true in the next state
- `legal(r,a)` means that it is legal for `r` to play `a` in the current state
- `goal(r,v)` means that `r` gets goal value `v` in the current state
- `terminal` means that the current state is a terminal state
- `distinct(s,t)` means that terms `s` and `t` are syntactically different
- `random` is a player that chooses its moves randomly
- `sees(r,p)` means that role `r` perceives `p` in the next state

The Monty Hall Game

State Representation

closed(2)

closed(3)

chosen(3)

car(2)

step(3)

Monty Hall: Vocabulary

- Object constants

`candidate`	Player
`noop,`	
`switch`	Moves

- Functions

`closed(number),`	
`chosen(number),`	
`car(number),`	
`step(number)`	Fluents
`hide_car(number),`	
`choose(number),`	
`open_door(number)`	Moves

Players and Initial State

```
role(candidate)
role(random)

init(closed(1))
init(closed(2))
init(closed(3))
init(step(1))
```

Move Generator

```
% Monty
legal(random,hide_car(D))  <= true(step(1)) ∧
                              true(closed(D))

legal(random,open_door(D)) <= true(step(2)) ∧
                              true(closed(D)) ∧
                              ¬true(car(D)) ∧
                              ¬true(chosen(D))

legal(random,noop)         <= true(step(3))


% Player
legal(candidate,choose(D)) <= true(step(1)) ∧
                              true(closed(D))
legal(candidate,noop)      <= true(step(2))
legal(candidate,noop)      <= true(step(3))
legal(candidate,switch)    <= true(step(3))
```

Physics

```
next(car(D))     <= does(random,hide_car(D))
next(car(D))     <= true(car(D))

next(closed(D)) <= true(closed(D)) ∧
                   ¬does(random,open_door(D))
```

```
next(chosen(D)) <= does(candidate,choose(D))
next(chosen(D)) <= true(chosen(D)) ∧
                   ¬does(candidate,switch)
next(chosen(D)) <= does(candidate,switch) ∧
                   true(closed(D)) ∧
                   ¬true(chosen(D))
```

```
next(step(2))   <= true(step(1))
next(step(3))   <= true(step(2))
next(step(4))   <= true(step(3))
```

Player's Percept, Termination, Goal

```
sees(candidate,D) <= does(random,open_door(D))
```

```
terminal <= true(step(4))


goal(candidate,100) <= true(chosen(D)) ∧
                       true(car(D))
goal(candidate,  0) <= true(chosen(D)) ∧
                       ¬true(car(D))
```

Perfect- vs. Imperfect-Information Games

The execution model for GDL and perfect-information games ensures that

- all players know the complete rules
- all players know the initial position
- all moves are deterministic
- all players are immediately informed about each other's moves

The execution model for GDL-II and perfect-information games ensures that

- all players know the complete rules
- all players know the initial position
- all moves are deterministic
- `random` chooses moves randomly
- players are not automatically informed about each other's moves; their percepts are determined according to `sees`

The General Game Model

An *n*-player, imperfect-information game is a structure with components:

$\{r_1, \ldots, r_n\}$ – players

S – set of states

$A_1, \ldots, A_n, A_{n+1}$ – *n*+1 sets of actions, one for each player plus one for `random`

$P_1, \ldots, P_n$ – *n* sets of percepts, one for each player

$l_1, \ldots, l_n, l_{n+1}$ – where $l_i \subseteq A_i \times S$, the legality relations

$u: S \times A_1 \times \ldots \times A_n \times A_{n+1} \rightarrow S$ – update function

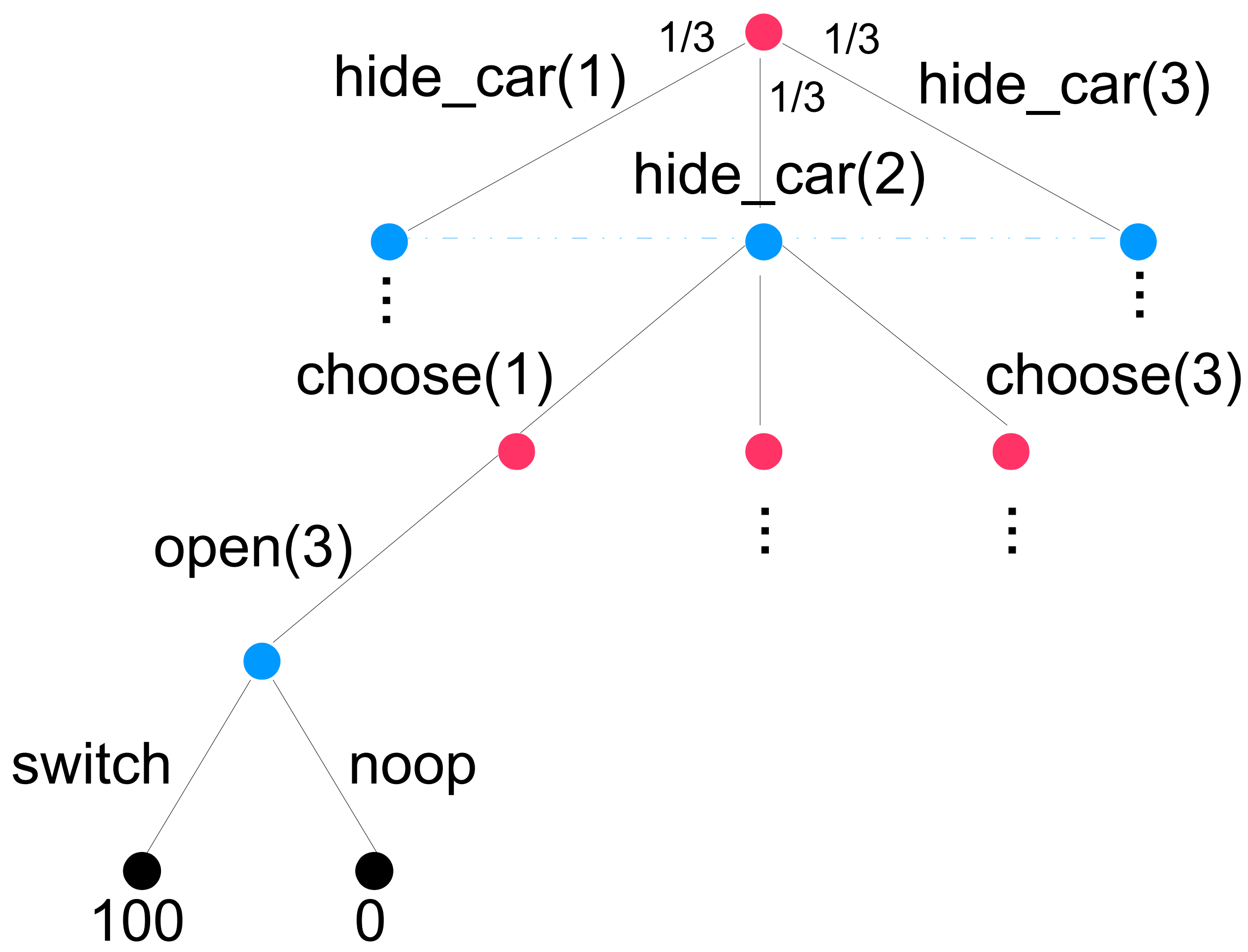
$\mathfrak{I}_i: S \times A_1 \times \ldots \times A_n \times A_{n+1} \rightarrow 2^{P_i}$ – information relation, one for each player

$s_1 \in S$ – initial game state

$t \subseteq S$ – the terminal states

$g_1, \ldots, g_n$ – where $g_i \subseteq S \times \mathbb{N}$, the goal relations

Games in Extensive Form



Other Examples: Kriegspiel

Standard chess in GDL-II requires to add

```
sees(white,M) <= does(black,M)
sees(black,M) <= does(white,M)
```

Omitting these rules gives you Kriegspiel (cf. Slide 10)

To play Kriegspiel effectively, players need to be informed whenever they attempt a move that is invalid in the current position:

```
sees(R,badMoveTryAgain) <= does(R,M) ∧ ¬validMove(M)
sees(black,yourMoveNow) <= does(white,M) ∧ validMove(M)
sees(white,yourMoveNow) <= does(black,M) ∧ validMove(M)
```

Other Examples: Communication, Negotiation

```
% player P makes a private offer

legal(P,offer(Q,trade(X,Y))) <= true(has(P,X)) ∧
                                true(has(Q,Y))

% players see all offers they get

sees(Q,offer_by(P,O)) <= does(P,offer(Q,O))
```

Syntactic Restrictions

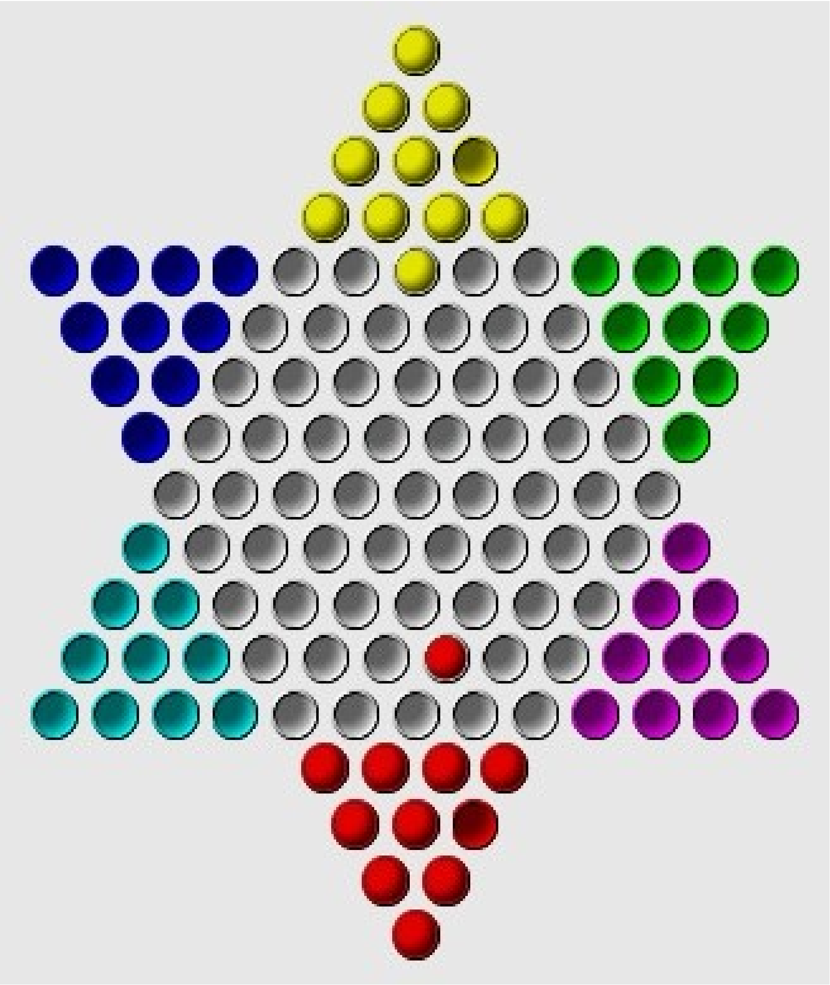
GDL-II has the same syntactic restrictions as GDL (safety, stratification, recursion restriction).

The keywords can only be used as follows.

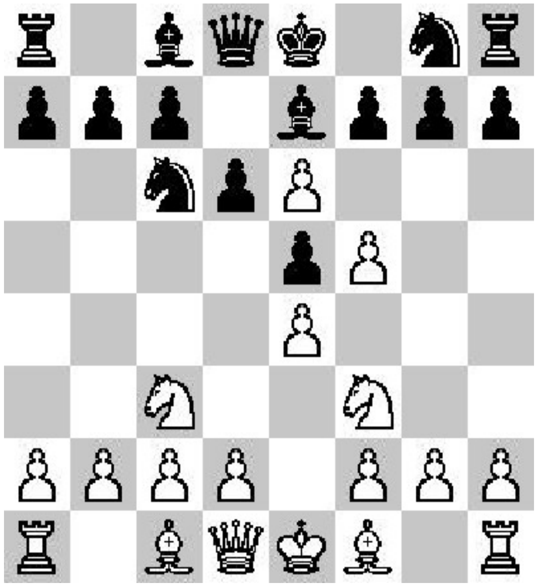
1. `role` as head of clause only appears in facts (i.e., clauses with empty body)
2. `random` only appears as first argument in `role, legal, does`
3. `init` only appears as head of clauses and does not depend on any of `true`, `legal`, `does`, `next`, `sees`, `terminal`, `goal`
4. `true` only appears in bodies of clauses
5. `does` only appears in clause bodies, and none of `legal`, `terminal`, `goal` depends on `does`
6. `next` only appears as head of clauses
7. `sees` only appears as head of clauses

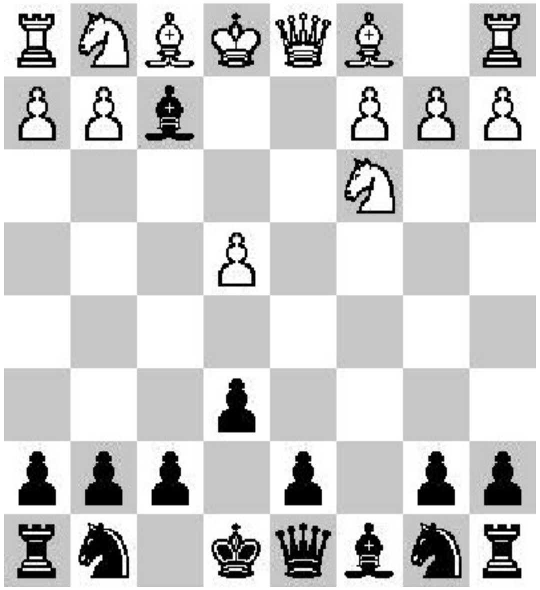
Game Theory

Example (1)

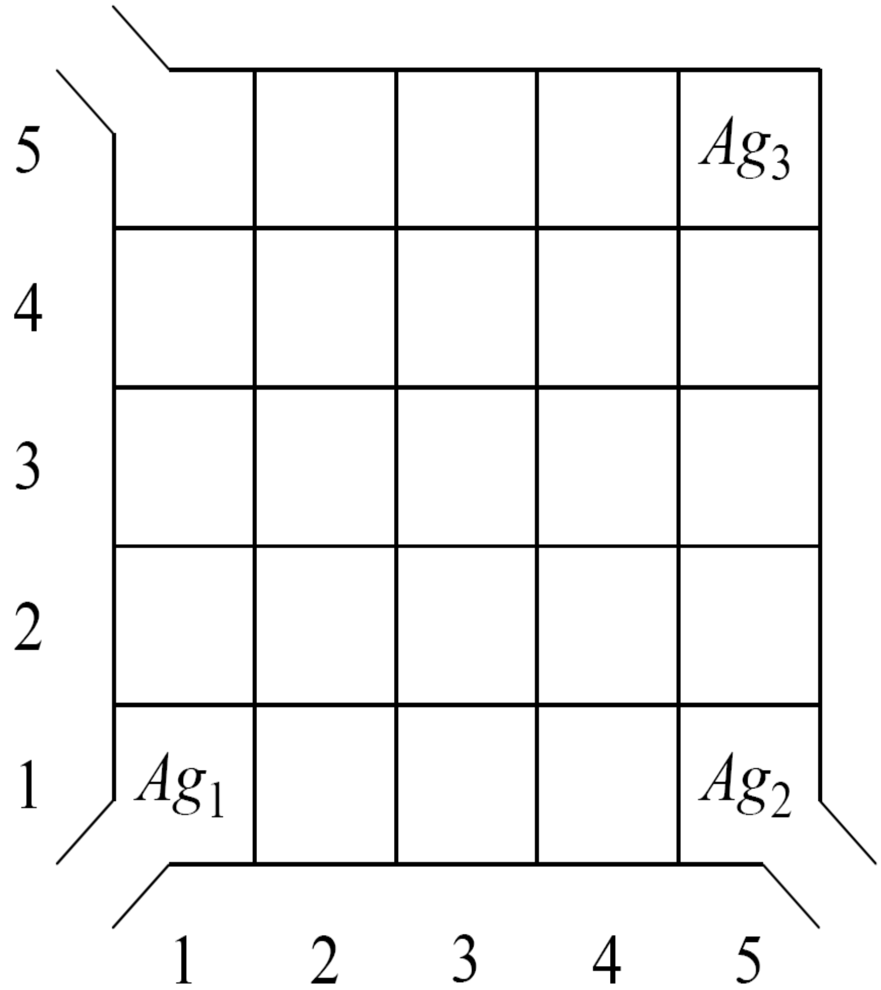


Example (2)





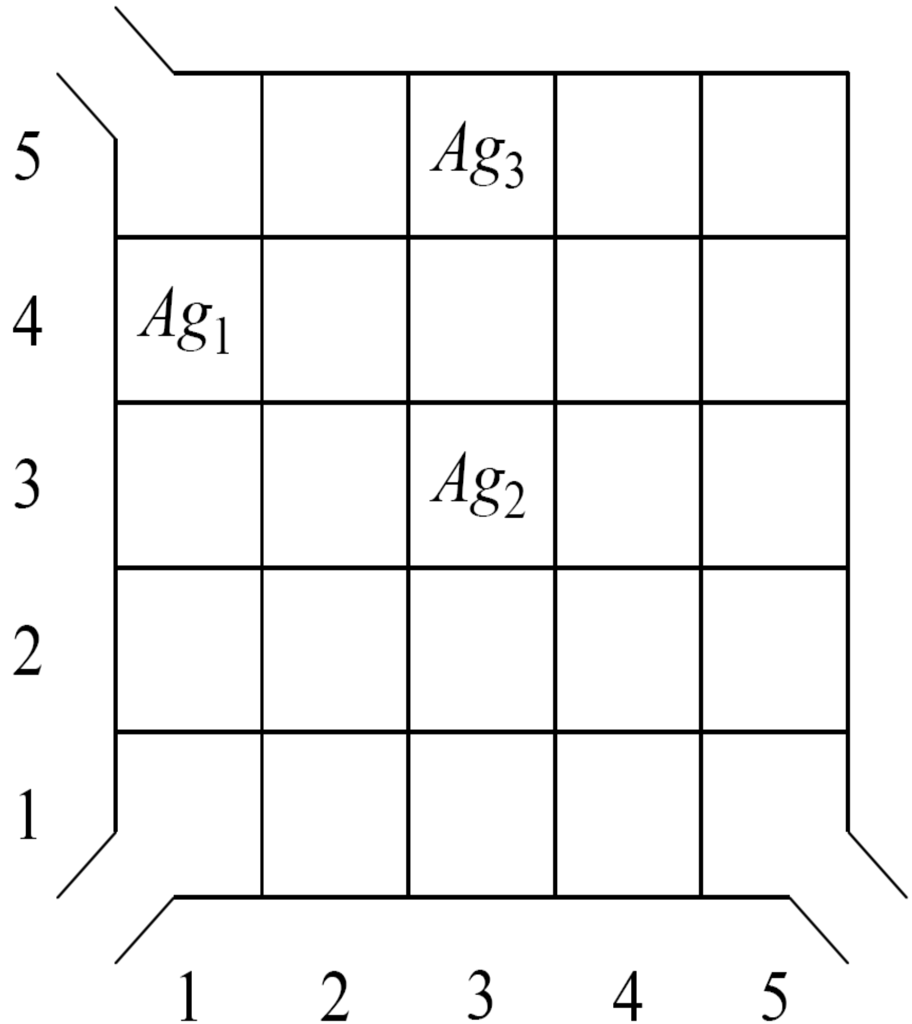
Example (3) "Agent Battle"



Competition and Cooperation

- The “pathological” assumption (e.g., if a bi-partite state transition graph is used to build the Minimax-search tree) says that opponents choose the joint move that is most harmful for us.
- This is usually too pessimistic for non-zerosum games or games with $n > 2$ players. *Rational* opponents choose the move that's best for them rather than the one that's worst for us.

Example



Game Theory gives the answer

Strategies

Game model:

S – set of states
$A_1, \ldots, A_n$ – n sets of actions, one for each player
$l_1, \ldots, l_n$ – where $l_i \subseteq A_i \times S$, the legality relations
$g_1, \ldots, g_n$ – where $g_i \subseteq S \times \mathbb{N}$, the goal relations

A strategy x_i for player i maps every state to a legal move for i

$x_i : S \rightarrow A_i$ (such that $(x_i(S),S) \in l_i$)

(Note that even for Chess the number of different strategies is finite. But they outnumber the atoms in the universe.)

Strategies for Agent-Battle

Example strategy for Ag_1:

$\{At(Ag_1,1,1), At(Ag_2,5,1), At(Ag_3,5,5)\} \rightarrow$ Go(East)

$\{At(Ag_1,1,1), At(Ag_2,5,1), At(Ag_3,4,5)\} \rightarrow$ Go(North)

⋮

$\{At(Ag_1,1,4), At(Ag_2,3,3), At(Ag_3,3,5)\} \rightarrow$ Go(North)

⋮

Similar for Ag_2, Ag_3

Towards the Normal Form of Games

Each *n*-tuple of strategies directly determines the outcome of a match.

Example:

Start with 7 coins. Players A and B take turn in removing one or two coins. Whoever takes the last coin wins.

Strategy Player A:

$\{7 \rightarrow 2, 6 \rightarrow 2, 5 \rightarrow 2, 4 \rightarrow 1, 3 \rightarrow 1, 2 \rightarrow 2, 1 \rightarrow 1\}$

Strategy Player B:

$\{7 \rightarrow 2, 6 \rightarrow 1, 5 \rightarrow 2, 4 \rightarrow 1, 3 \rightarrow 2, 2 \rightarrow 2, 1 \rightarrow 1\}$

Outcome: (0, 100)

Games in Normal Form

An *n-player game in normal form* is an *n*+1-tuple

$$\Gamma = (X_1, \ldots, X_n, u)$$

where X_i is the set of strategies for player *i* and

$$u = (u_1, \ldots, u_n) : \bigtimes_{i=1}^{n} X_i \rightarrow \mathbb{N}^i$$

are the utilities of the players for each *n*-tuple of strategies.

Roshambo

2-player games are often depicted as matrices

	Rock	Scissors	Paper
Rock	50 50	0 100	100 0
Scissors	100 0	50 50	0 100
Paper	0 100	100 0	50 50

2-Finger-Morra

	1 Finger	2 Fingers
1 Finger	30 / 70	90 / 10
2 Fingers	90 / 10	0 / 100

Battle of the Sexes

	Ballgame	Opera
Ballgame	3 4	2 2
Opera	1 1	4 3

Prisoner's Dilemma

	Cooperate	Defect
Cooperate	3 3	4 1
Defect	1 4	2 2

Equilibria

Let $\Gamma = (X_1, ..., X_n, u)$ be an n-player game.

$(x_1^*, ..., x_n^*) \in X_1 \times ... \times X_n$ equilibrium

if for all $i = 1, ..., n$ and all $x_i \in X_i$

$$u_i(x_1^*, ..., x_{i-1}^*, x_i, x_{i+1}^*, ..., x_n^*) \leq u_i(x_1^*, ..., x_n^*)$$

An equilibrium is a tuple of optimal strategies: No player has a reason to deviate from his strategy, given the opponents' strategies.

Full Cooperation

	a	b
a	4 4	1 2
b	2 3	3 1

Battle of the Sexes

	Ballgame	Opera
Ballgame	4, 3	2, 2
Opera	1, 1	3, 4

(Note that the outcome for both players is bad if they choose to play different equilibria.)

Cooperation

	a	b
a	4, 4	2, 2
b	1, 1	3, 3

(Note that the concept of an equilibrium doesn't suffice to achieve the best possible outcome for both players.)

Prisoner's Dilemma

	Cooperate	Defect
Cooperate	3 3	4 1
Defect	1 4	2 2

(Note that the outcome which is better for both players isn't even an equilibrium!)

Dominance

A strategy $x \in X_i$ dominates a strategy $y \in X_i$ if

$$u_i(x_1, \ldots, x_{i-1}, x, x_{i+1}, \ldots, x_n) \geq u_i(x_1, \ldots, x_{i-1}, y, x_{i+1}, \ldots, x_n)$$

for all $(x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_n) \in X_1 \times \ldots \times X_{i-1} \times X_{i+1} \times \ldots \times X_n$.

A strategy $x \in X_i$ strongly dominates a strategy $y \in X_i$ if

x dominates y and y does not dominate x.

Assume that opponents are rational: They don't choose a strongly dominated strategy.

Removing Strongly Dominated Strategies

	a	b
a	4, 4	2, 1
~~b~~	~~3, 2~~	~~1, 3~~

Iterated Dominance

Let a zero-sum game be given by

	a	b	c	d	e
a	10	7	6	9	8
b	10	4	6	9	5
c	9	7	9	8	8
d	2	6	4	3	7

Iterated Dominance (2)

	a	b	c	d	e
a	10	7	6	9	8
~~b~~	~~10~~	~~4~~	~~6~~	~~9~~	~~5~~
c	9	7	9	8	8
~~d~~	~~2~~	~~6~~	~~4~~	~~3~~	~~7~~

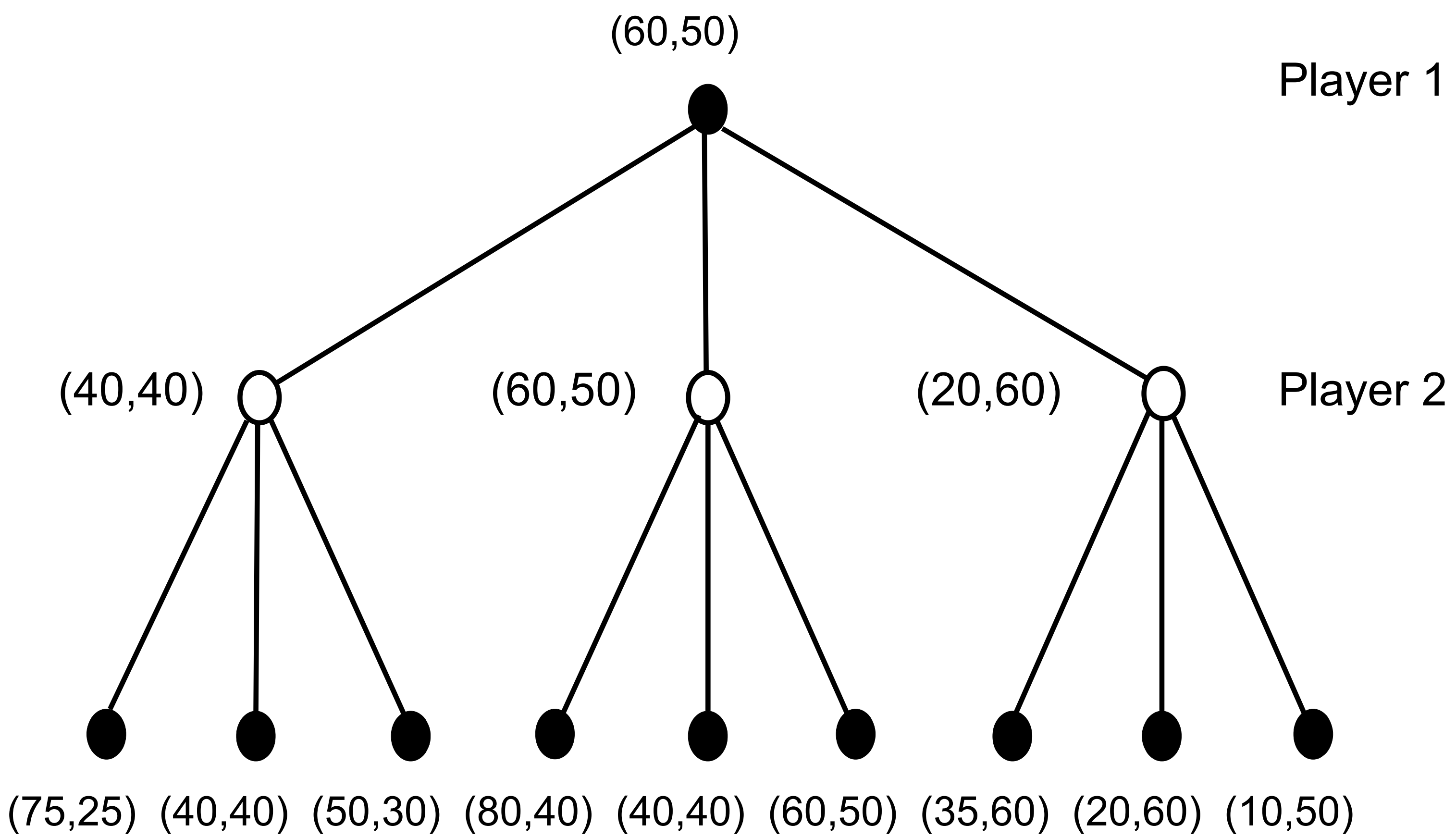
Iterated Dominance (3)

	a	b	c	d	e
a	10	7	6	9	8
c	9	7	9	8	8

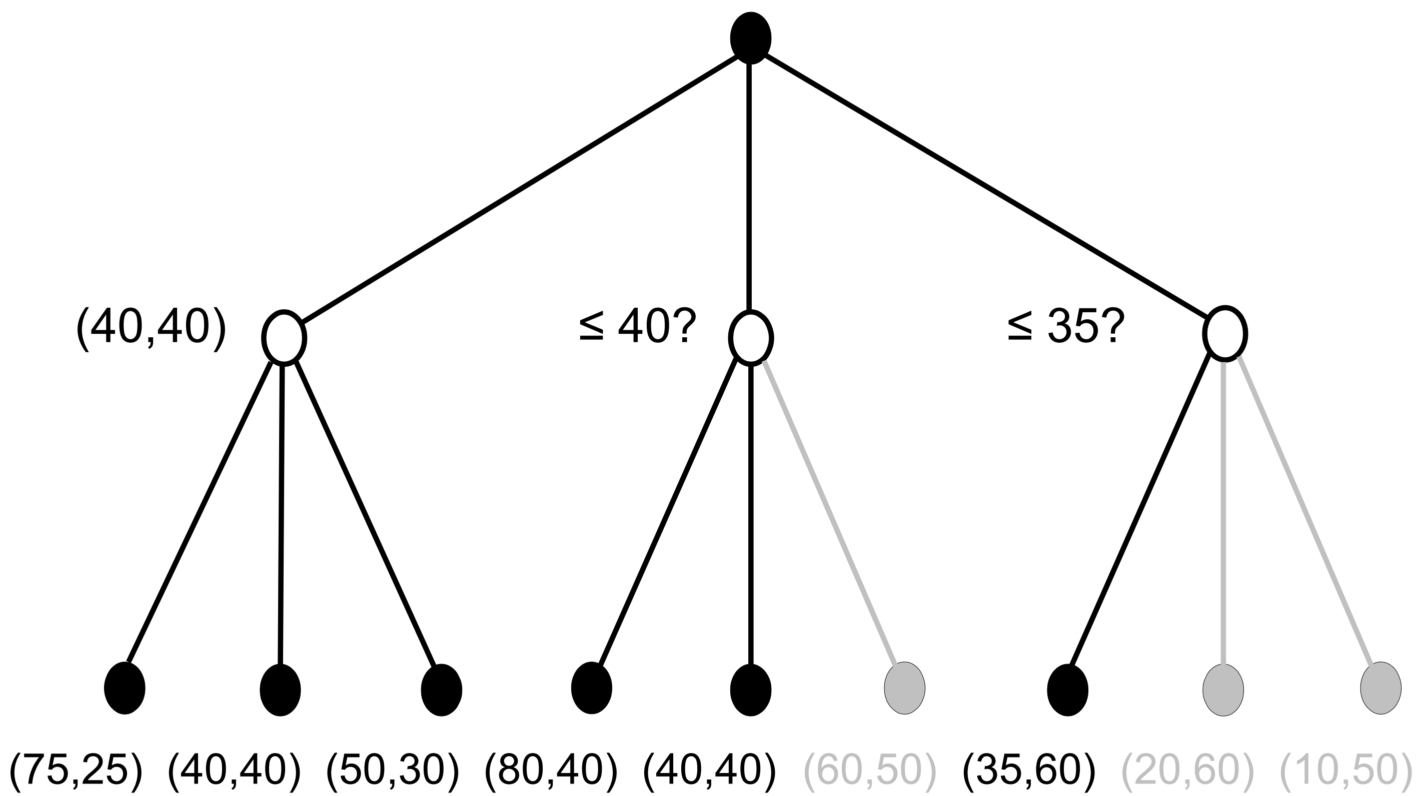
Iterated Dominance (4)

	b	c
a	7	6
c	7	9

Game Tree Search with Dominance



α-β-Principle Does Not Apply



The Need to Randomise: Roshambo

	Rock	Scissors	Paper
Rock	50 / 50	100 / 0	0 / 100
Scissors	0 / 100	50 / 50	100 / 0
Paper	100 / 0	0 / 100	50 / 50

This game has no equilibrium

2-Finger-Morra

	1 Finger	2 Fingers
1 Finger	30 70	90 10
2 Fingers	90 10	0 100

This game, too, has no equilibrium

Mixed Strategies

Let $(X_1, \ldots, X_n, u)$ be an n-player game, then its mixed extension is

$$\Gamma = (\mathrm{P}_1, \ldots, \mathrm{P}_n, (e_1, \ldots, e_n))$$

where for each $i=1, \ldots, n$

$$\mathrm{P}_i = \{\mathrm{p}_i\text{: } \mathrm{p}_i \text{ probability measure over } X_i\}$$

and for each $(p_1, \ldots, p_n) \in \mathrm{P}_1 \times \ldots \times \mathrm{P}_n$

$$e_i(p_1, \ldots, p_n) = \sum_{x_1 \in X_1} \ldots \sum_{x_n \in X_n} u_i(x_1, \ldots, x_n) * p_1(x_1) * \ldots * p_n(x_n)$$

Nash's Theorem.

Every mixed extension of an n-player game has at least one equilibrium.

Roshambo

	Rock	Scissors	Paper
Rock	50, 50	100, 0	0, 100
Scissors	0, 100	50, 50	100, 0
Paper	100, 0	0, 100	50, 50

The unique equilibrium is

$$\left(\left(\frac{1}{3},\frac{1}{3},\frac{1}{3}\right),\left(\frac{1}{3},\frac{1}{3},\frac{1}{3}\right)\right)$$

2-Finger-Morra

	1 Finger	2 Fingers
1 Finger	70, 30	10, 90
2 Fingers	10, 90	100, 0

The unique equilibrium is

$$(p_1^*, p_2^*) = \left(\left(\frac{3}{5}, \frac{2}{5}\right), \left(\frac{3}{5}, \frac{2}{5}\right)\right)$$

with $e_1(p_1^*, p_2^*) = 46$ and $e_2(p_1^*, p_2^*) = 54$

Iterated Row Dominance for Mixed Strategies

Let a zero-sum game be given by

	a	b	c
a	10	0	8
b	6	4	4
c	2	8	7

Then $p_1 = \left(\frac{1}{2}, 0, \frac{1}{2}\right)$ strongly dominates $p_1' = (0,1,0)$.

Hence, for all $(p_a', p_b', p_c') \in P_1$ with $p_b' > 0$ there exists a dominating strategy $(p_a, 0, p_c) \in P_1$.

Iterated Row Dominance for Mixed Strategies

	a	b	c
a	10	0	8
~~b~~	~~6~~	~~4~~	~~4~~
c	2	8	7

Now $p_2 = \left(\frac{1}{2}, \frac{1}{2}, 0\right)$ dominates $p_2' = (0,0,1)$.

Iterated Row Dominance for Mixed Strategies

	a	b	c
a	10	0	8
c	2	8	7

The unique equilibrium is $\left(\left(\frac{3}{8}, 0, \frac{5}{8}\right), \left(\frac{1}{2}, \frac{1}{2}, 0\right)\right)$.

Further Reading

If you're interested in topics for a project/ thesis/... on Logic-Based Agents or General Game Playing, go to:

http://cgi.cse.unsw.edu.au/~mit

Further Reading

- www.general-game-playing.de
- games.stanford.edu/competition/misc/aaai.pdf
- www.ru.is/faculty/hif/papers/cadiaplayer_aaai08.pdf
- cgi.cse.unsw.edu.au/~mit/Papers/AAAI10a.pdf