Research Project Proposal:
Abstractions in Extensive-Form Games

Jacopo Pio Gargano
jacopopio.gargano@mail.polimi.it
CSE Track

POLITECNICO
MILANO 1863

HONOURS PROGRAMME
HP-SR
in Information Technology

Recreational games

Chess

Recreational games

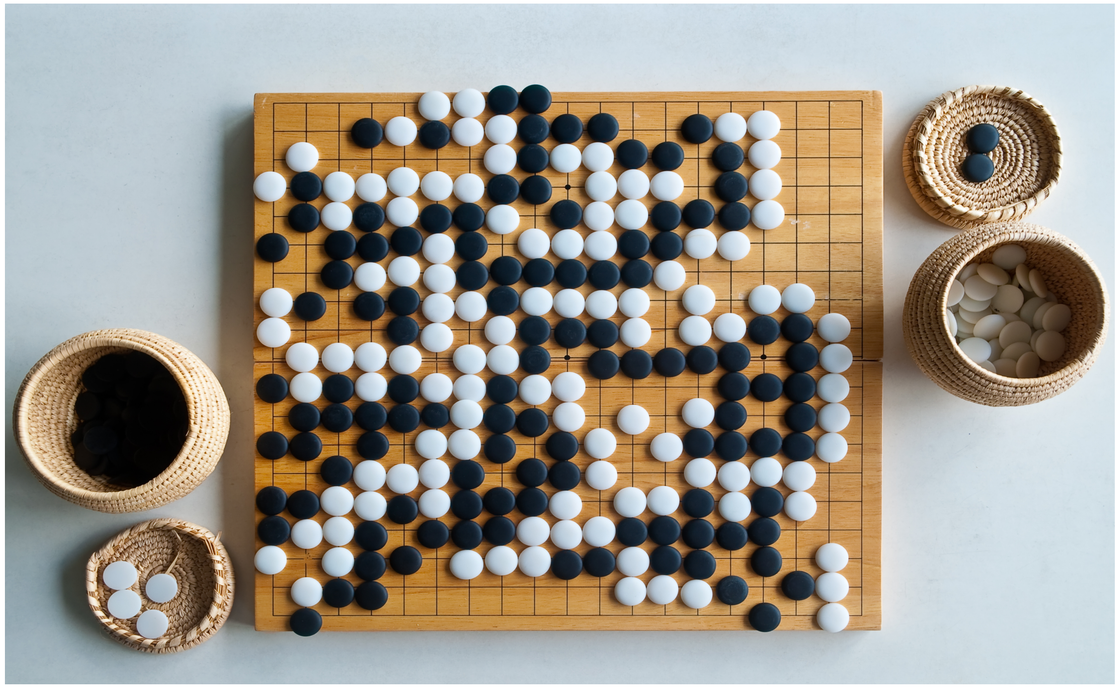
Chess

1996

IBM

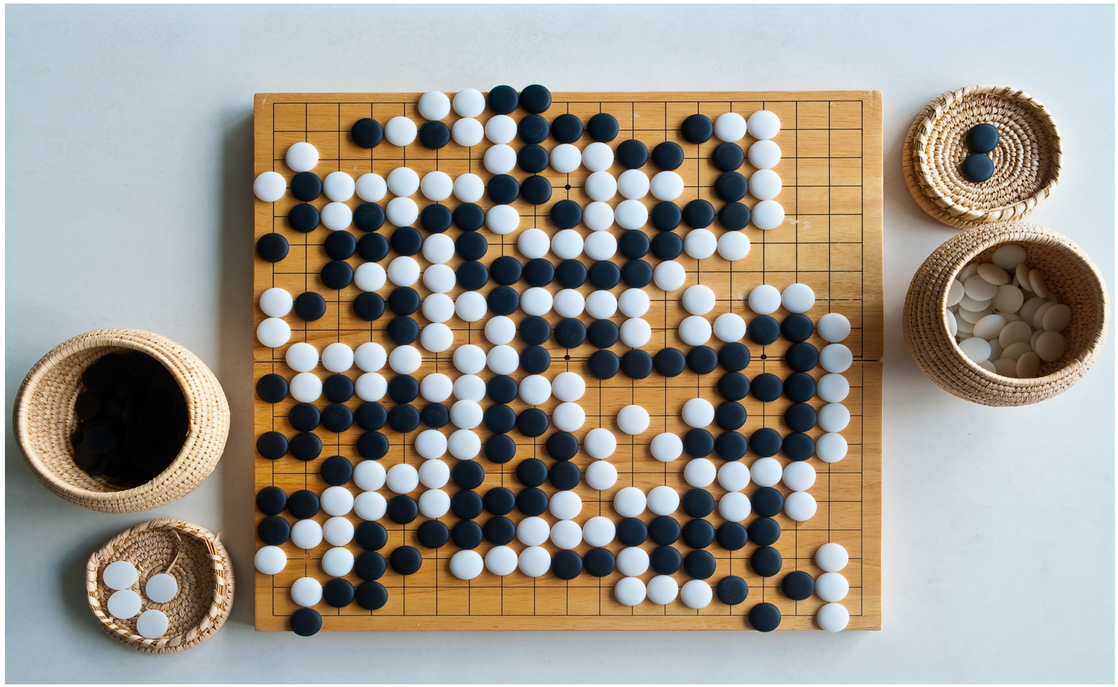
MiniMax with alpha-beta pruning search

Recreational games



Go

Recreational games



Go

2015

Google DeepMind

Monte Carlo tree search
Deep neural networks
Reinforcement Learning - self-play

Physical security

ARMOR - LAX airport

PROTECT - New York City port

Physical security

ARMOR - LAX airport

PROTECT - New York City port

USC University of Southern California

Attacker VS Defender
Limited resources
Find optimal resource allocation

Physical security

Wildlife poaching

Forest protection

Physical security

Wildlife poaching

Forest protection

USC University of Southern California

Reserves are huge
Predict poachers positions
Find optimal resource allocation

Car racing

Defender

Attacker

Military

Attacker

Defender

Military

Attacker

Defender

Dogfighting

Game theory

- Theoretical framework for strategic interaction
- Mathematical models and algorithms (Algorithmic Game Theory)
- Conflict and cooperation
- Intelligent rational decision-makers
- Decisions influencing agents' welfare

Game theory

A *game* is a process consisting in:

- a set of players
- an initial situation
- rules that players must follow
- all possible final situations - outcomes
- the preferences of all the players - utilities

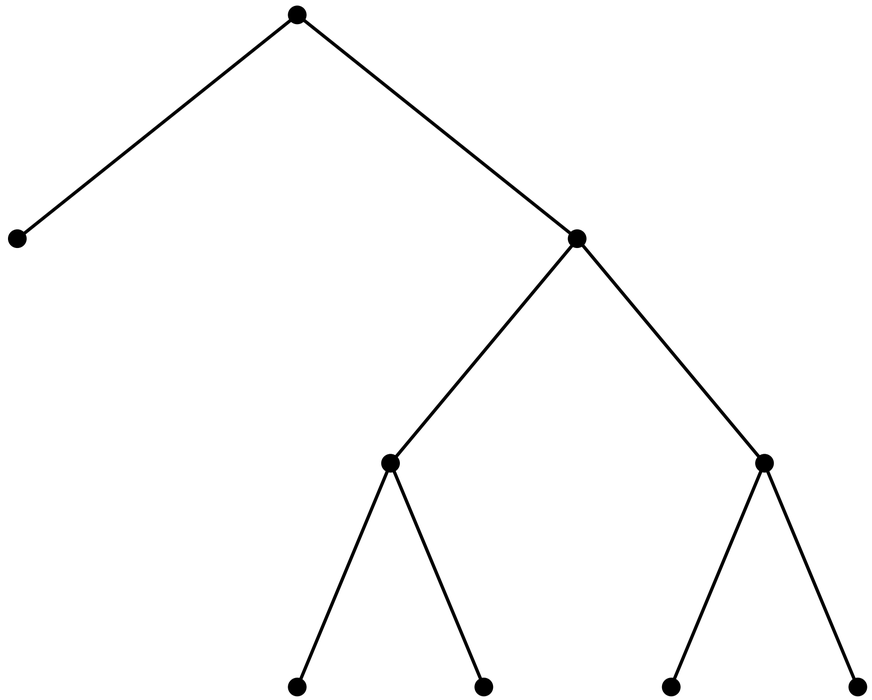
Game tree representation

Player 1

Player 2

Players Terminal nodes Decision nodes Actions Utilities

Game tree representation

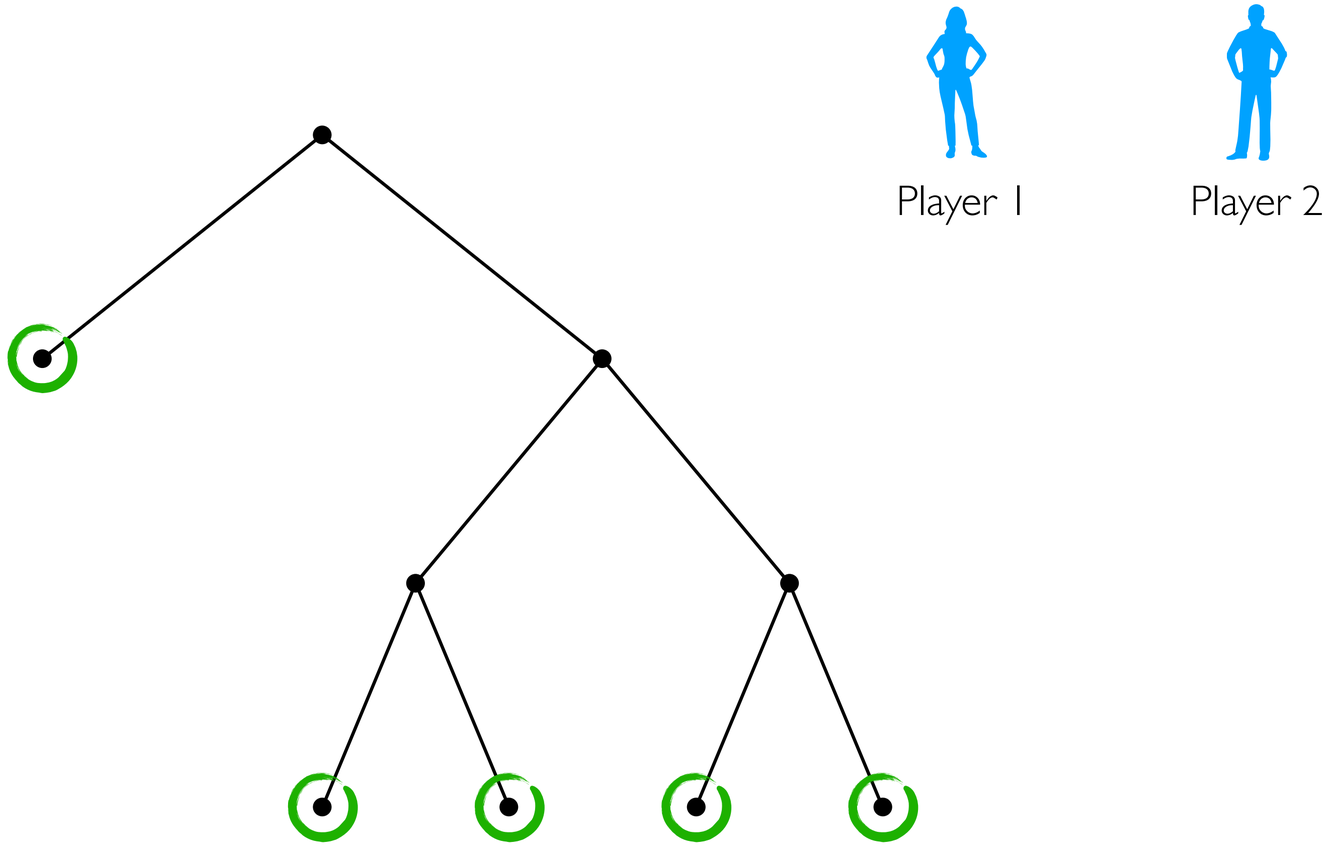


Player 1

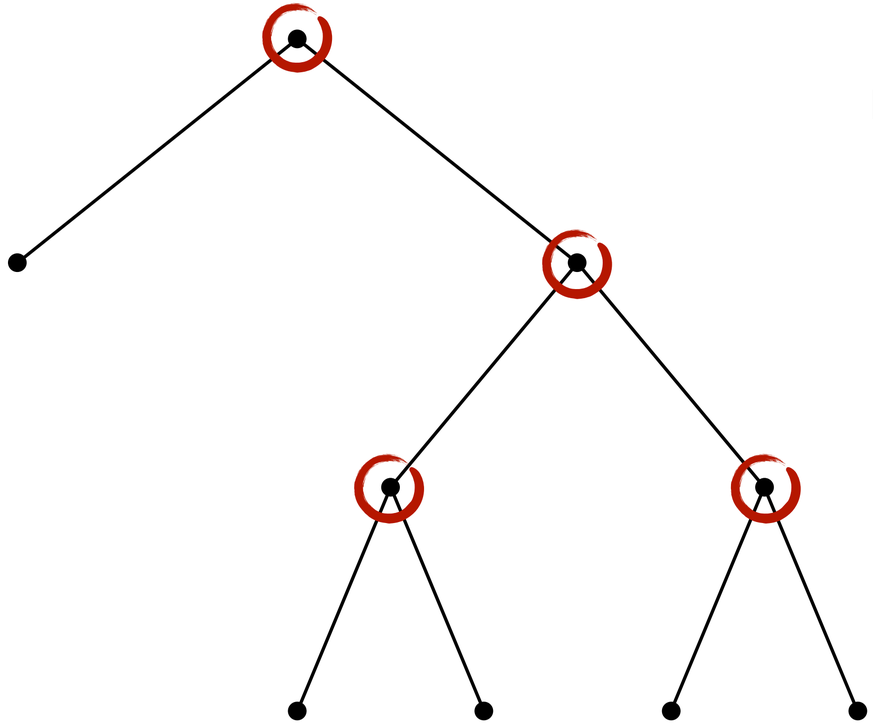
Player 2

Players Terminal nodes Decision nodes Actions Utilities

Game tree representation



Game tree representation

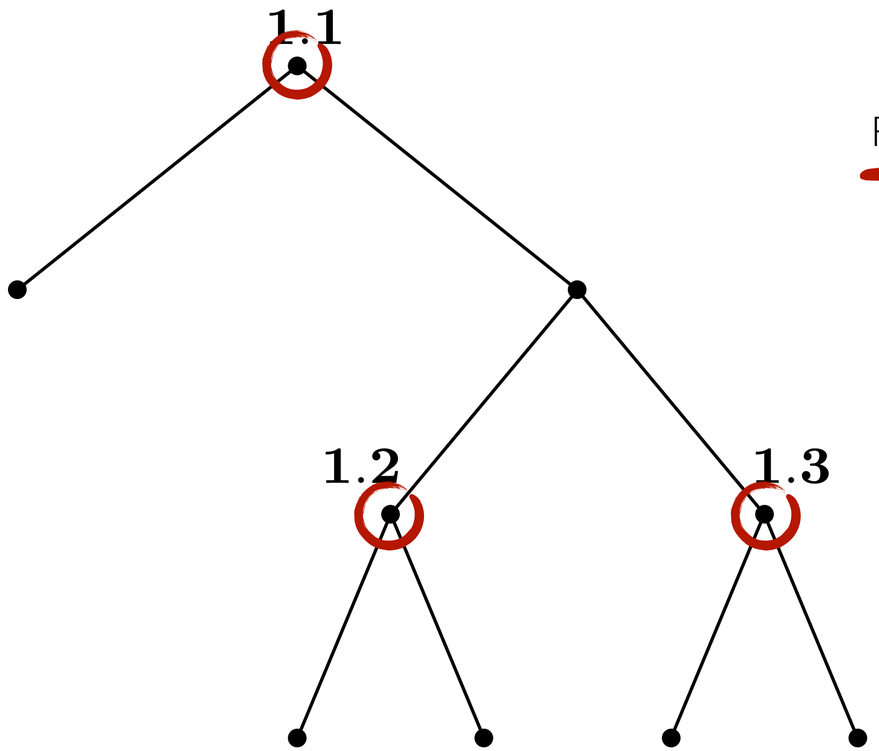


Player 1

Player 2

Players Terminal nodes Decision nodes Actions Utilities

Game tree representation

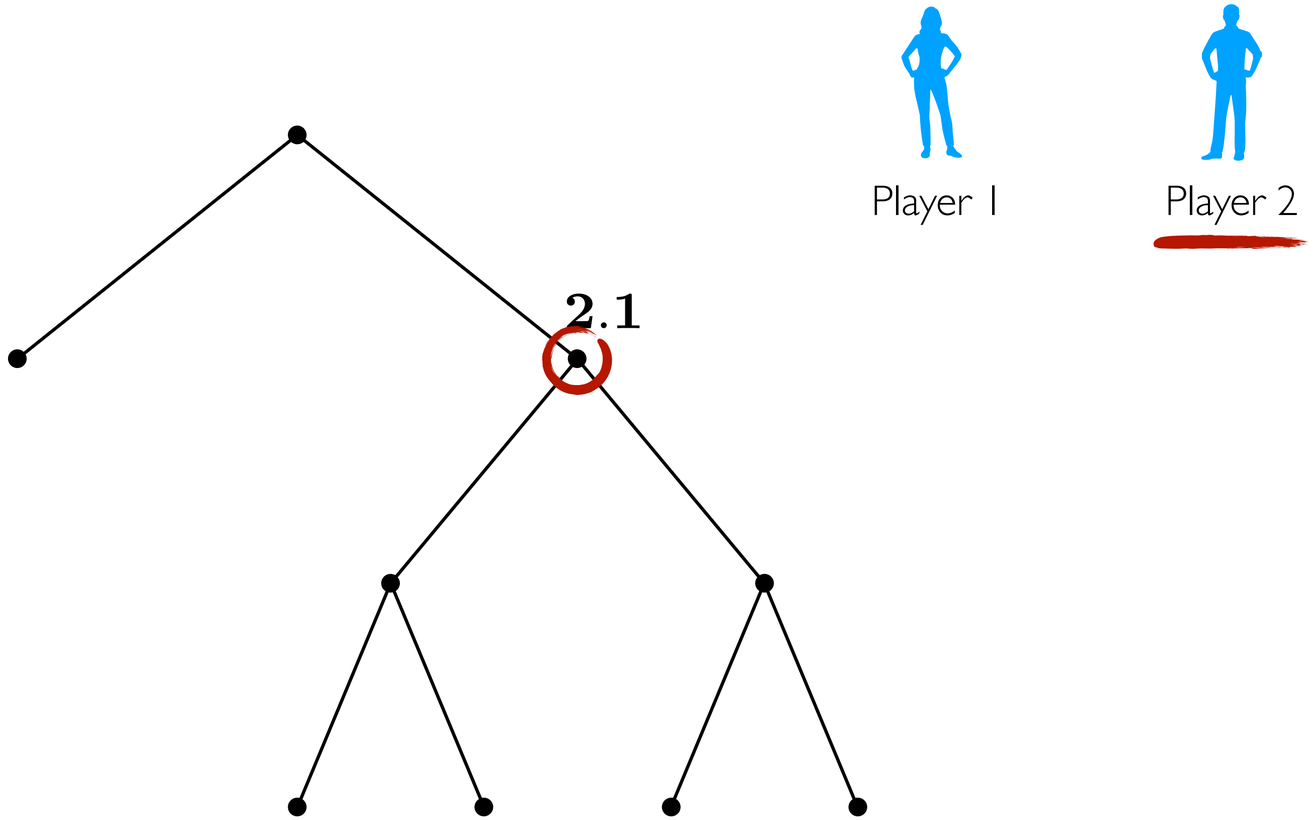


Player 1

Player 2

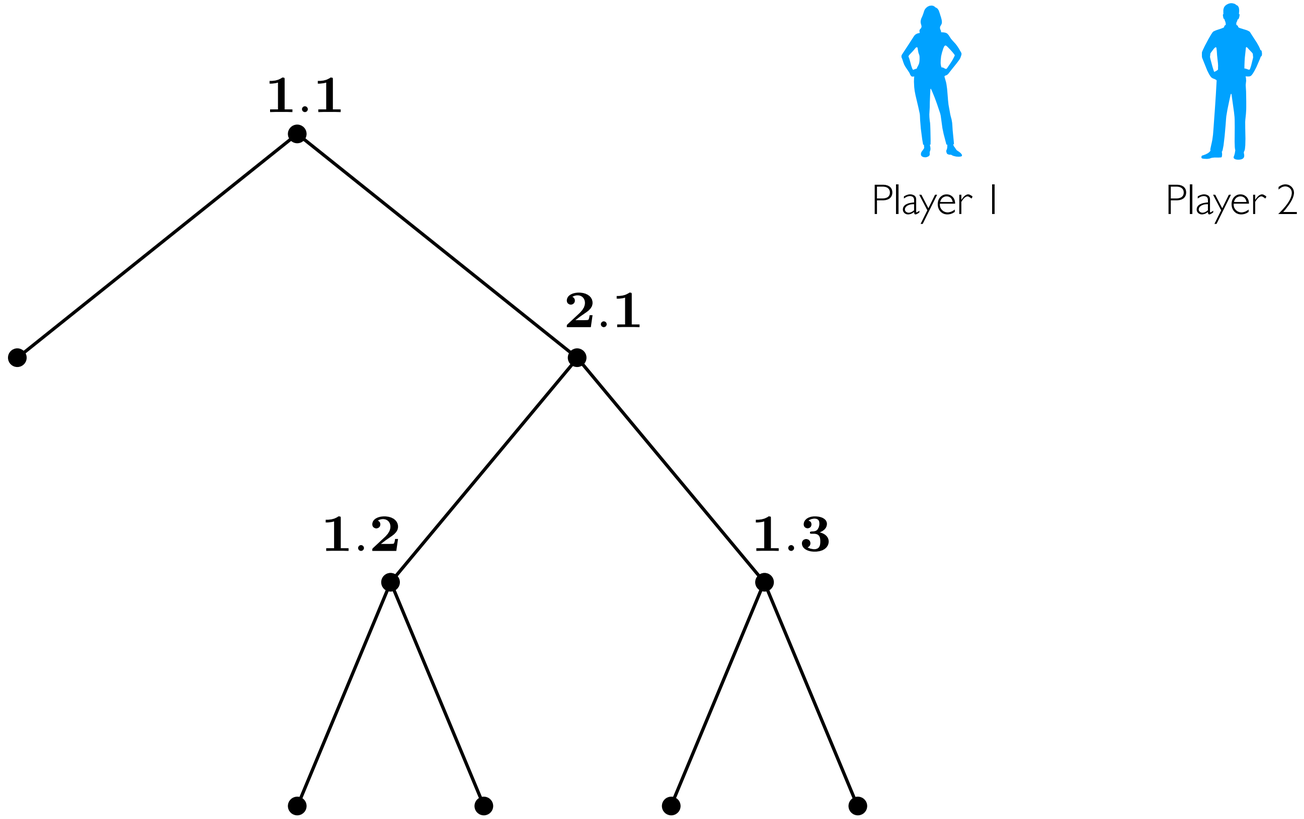
Players Terminal nodes Decision nodes Actions Utilities

Game tree representation

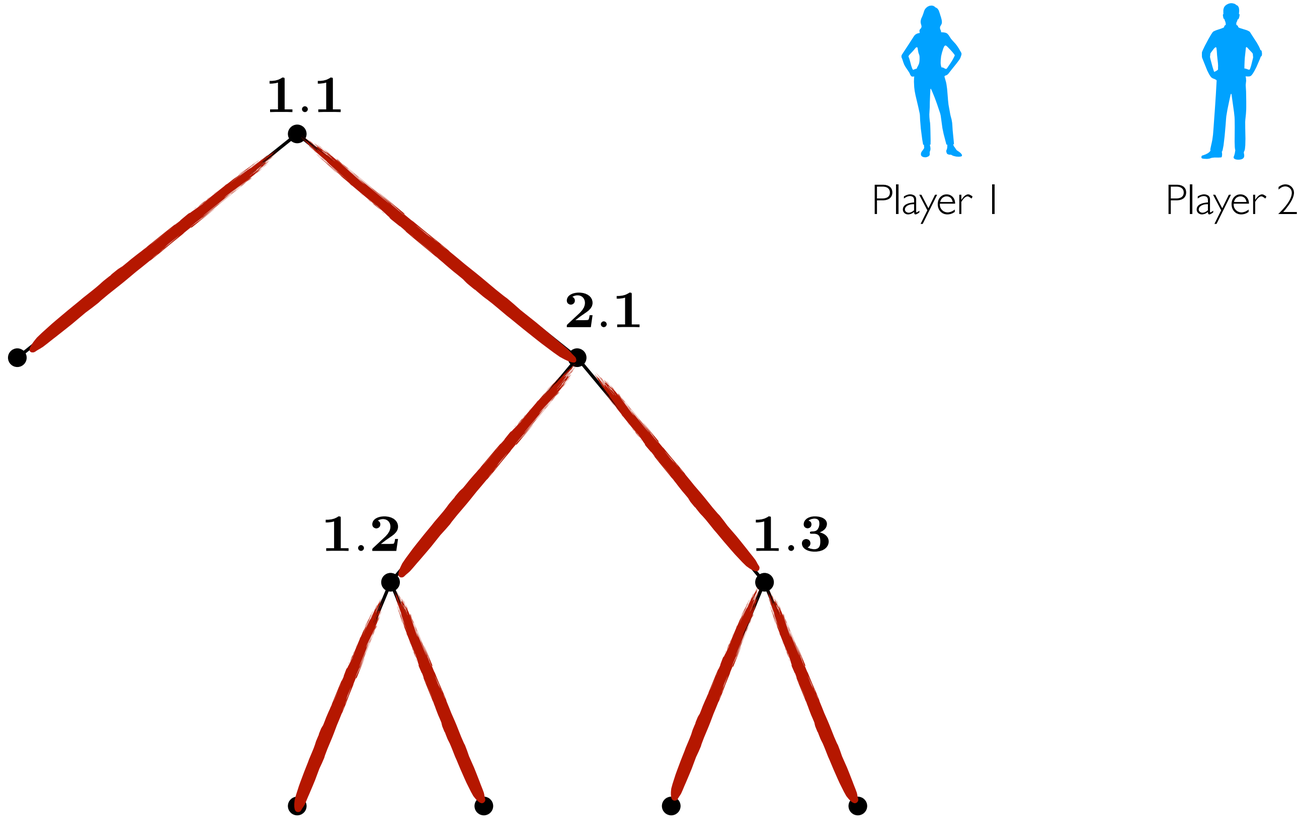


Players Terminal nodes Decision nodes Actions Utilities

Game tree representation

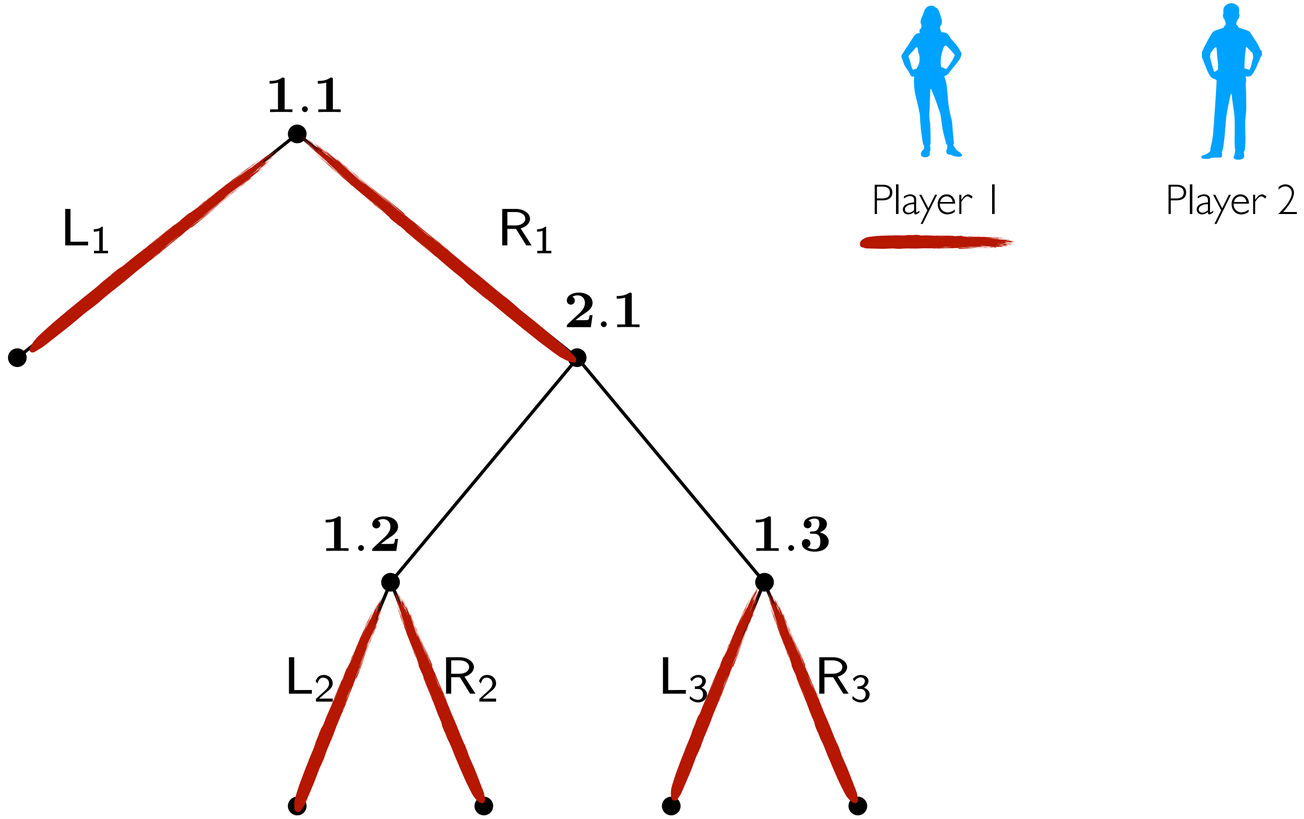


Game tree representation



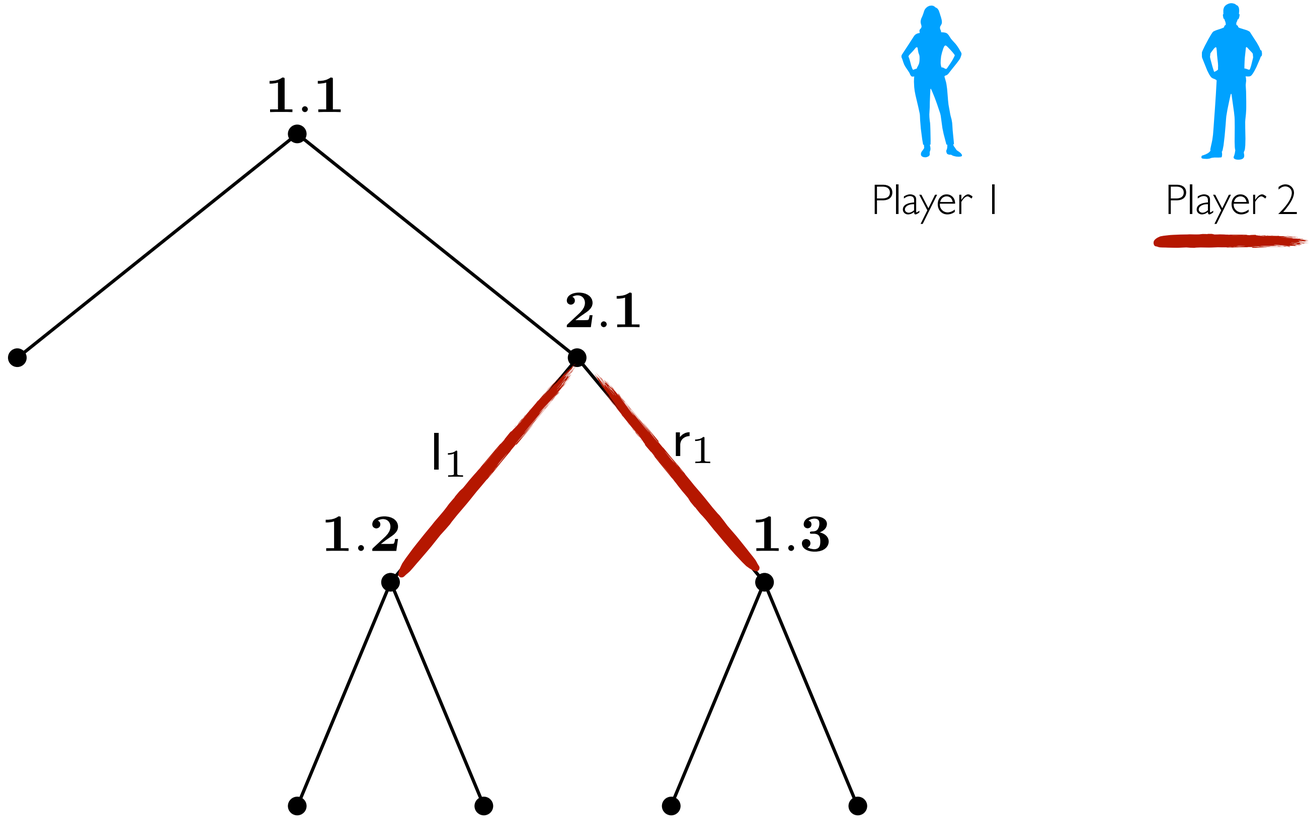
Players Terminal nodes Decision nodes Actions Utilities

Game tree representation

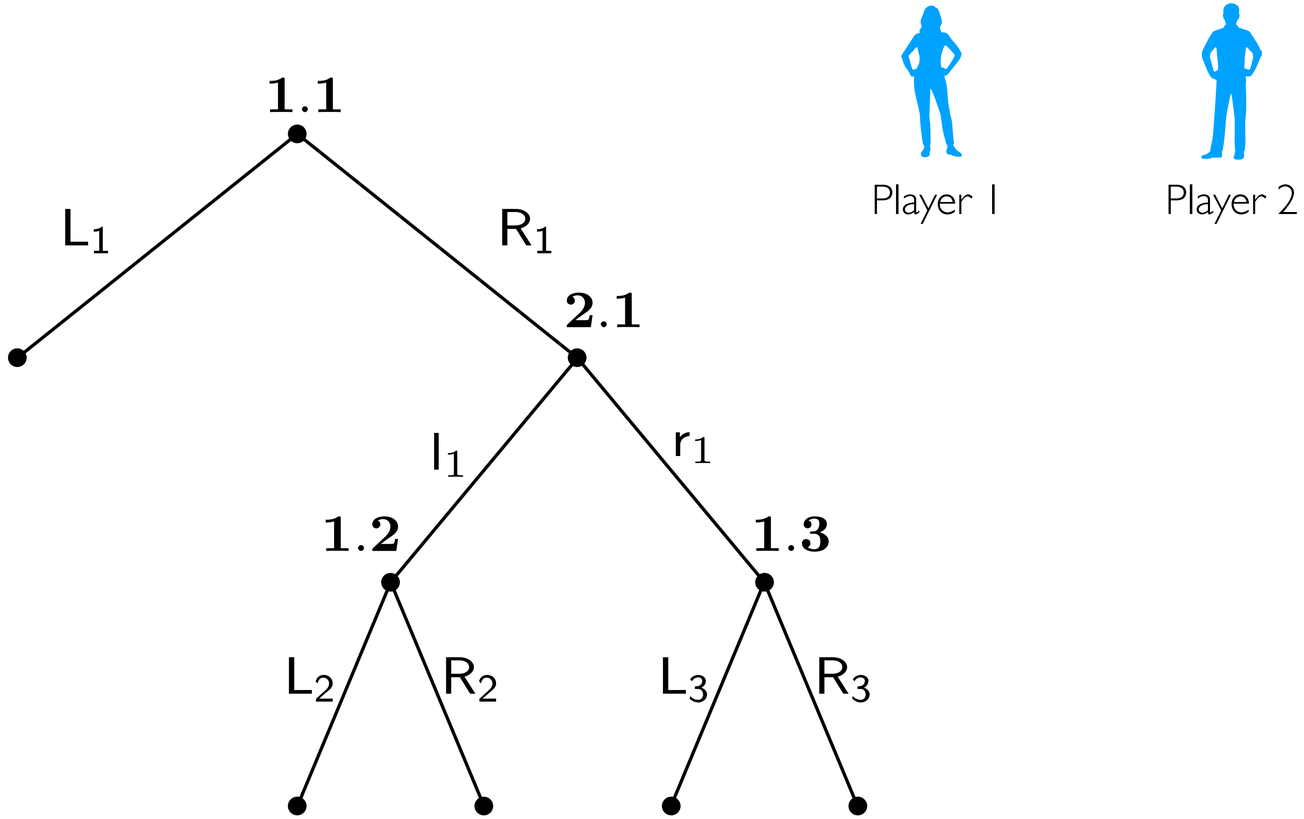


Players Terminal nodes Decision nodes Actions Utilities

Game tree representation

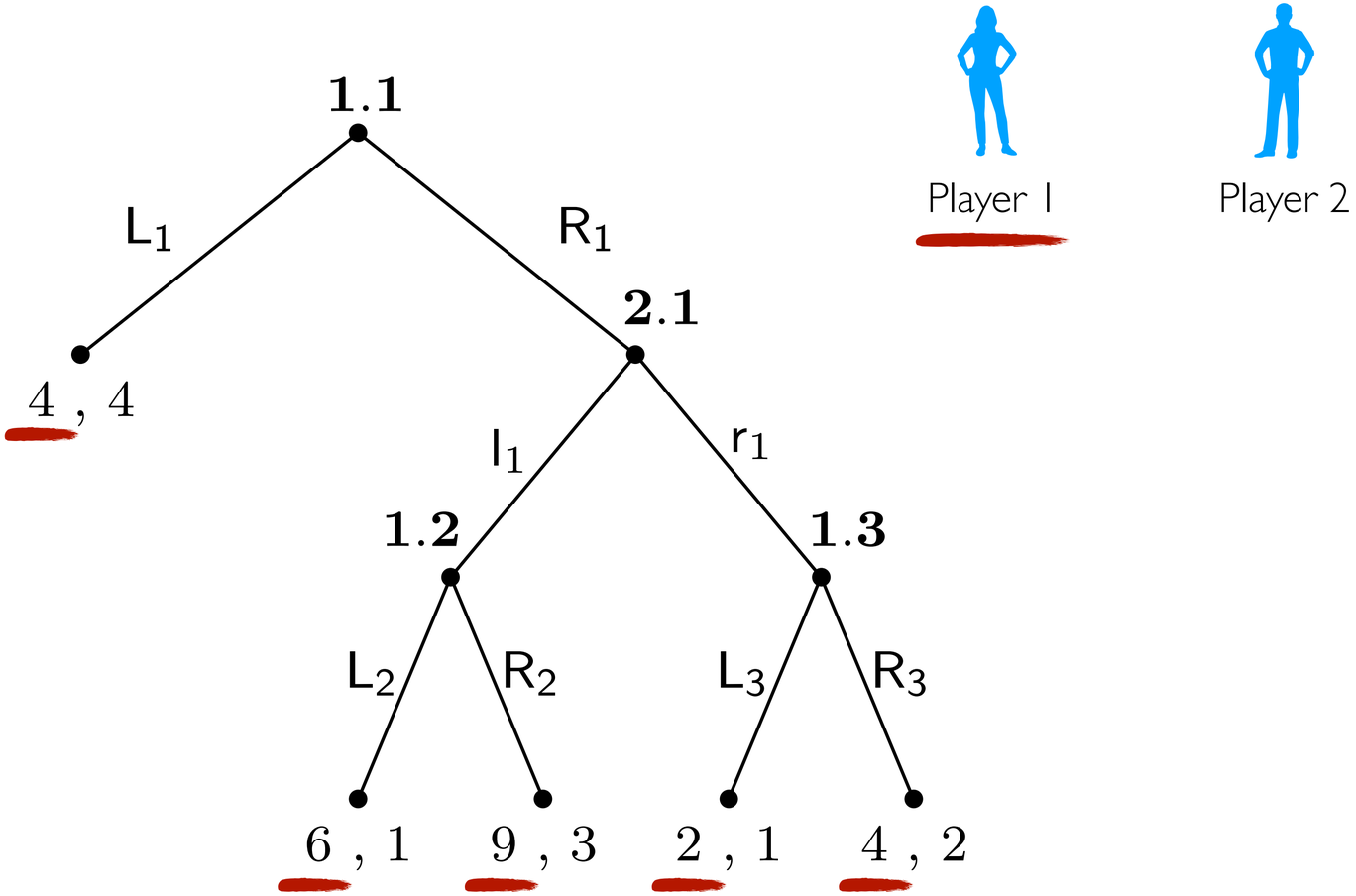


Game tree representation



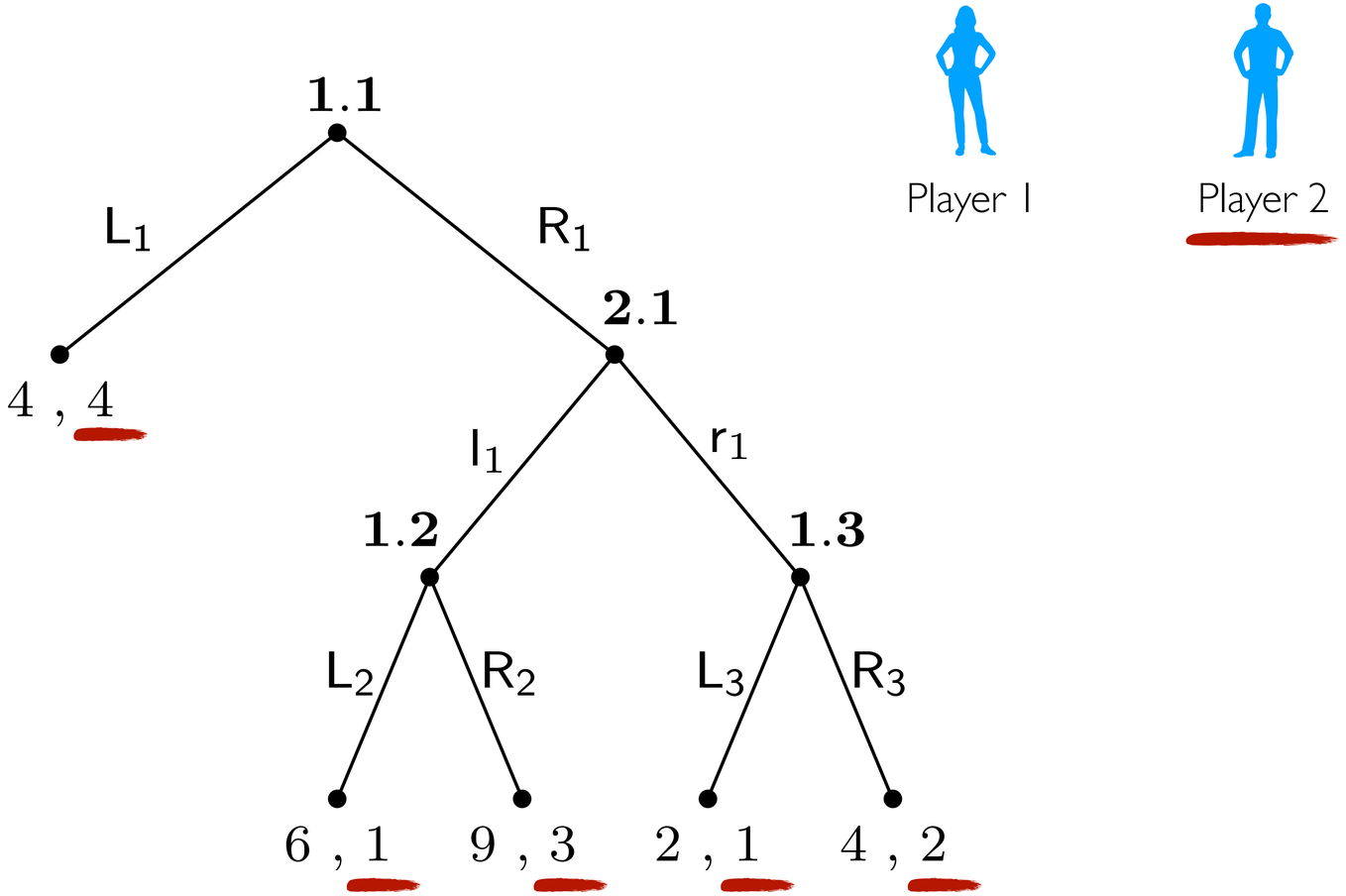
Players Terminal nodes Decision nodes Actions Utilities

Game tree representation



Players Terminal nodes Decision nodes Actions Utilities

Game tree representation



Players Terminal nodes Decision nodes Actions Utilities

Formal model

$$(N, A, V, T, \iota, \rho, \chi, U)$$

Formal model

$$(N, A, V, T, \iota, \rho, \chi, U)$$

Set of players $\{1, 2\}$
(Nature can be a player)

Formal model

$$(N, A, V, T, \iota, \rho, \chi, U)$$

Set of players {1, 2}
(Nature can be a player)

Set of actions $\{L_1, R_1, l_1, r_1, L_2, R_2, \ldots\}$

Formal model

$$(N, A, V, T, \iota, \rho, \chi, U)$$

Set of players {1, 2}
(Nature can be a player)

Set of actions {L_1, R_1, l_1, r_1, L_2, R_2, …}

Set of decision nodes {1.1, 1.2, 1.3, 2.1}

Formal model

$$(N, A, V, T, \iota, \rho, \chi, U)$$

Set of players {1, 2}
(Nature can be a player)

Set of actions $\{L_1, R_1, l_1, r_1, L_2, R_2, \ldots\}$

Set of decision nodes {1.1, 1.2, 1.3, 2.1}

Set of terminal nodes $\{t_1, t_2, t_3, t_4, t_5\}$

Formal model

$$(N, A, V, T, \iota, \rho, \chi, U)$$

Set of players {1, 2}
(Nature can be a player)

Set of actions $\{L_1, R_1, l_1, r_1, L_2, R_2, \ldots\}$

Set of decision nodes {1.1, 1.2, 1.3, 2.1}

Set of terminal nodes $\{t_1, t_2, t_3, t_4, t_5\}$

Player function: $V \longrightarrow N$

Formal model

$$(N, A, V, T, \iota, \rho, \chi, U)$$

Set of players {1, 2}
(Nature can be a player)

Set of actions $\{L_1, R_1, l_1, r_1, L_2, R_2, \ldots\}$

Set of decision nodes {1.1, 1.2, 1.3, 2.1}

Set of terminal nodes $\{t_1, t_2, t_3, t_4, t_5\}$

Player function: $V \longrightarrow N$

Next node function: $V \times A \longrightarrow V \cup T$

Formal model

$$(N, A, V, T, \iota, \rho, \chi, U)$$

Set of players {1, 2}
(Nature can be a player)

Set of actions $\{L_1, R_1, l_1, r_1, L_2, R_2, \ldots\}$

Set of decision nodes {1.1, 1.2, 1.3, 2.1}

Set of terminal nodes $\{t_1, t_2, t_3, t_4, t_5\}$

Player function: $V \longrightarrow N$

Next node function: $V \times A \longrightarrow V \cup T$

Action function: $V \longrightarrow 2^A$

Formal model

$$(N, A, V, T, \iota, \rho, \chi, U)$$

Set of players {1, 2}
(Nature can be a player)

Set of actions $\{L_1, R_1, l_1, r_1, L_2, R_2, \ldots\}$

Set of decision nodes {1.1, 1.2, 1.3, 2.1}

Set of terminal nodes $\{t_1, t_2, t_3, t_4, t_5\}$

Player function: $V \longrightarrow N$

Next node function: $V \times A \longrightarrow V \cup T$

Action function: $V \longrightarrow 2^A$

Set of utility functions: $\{U_1, U_2\}$

Perfect vs imperfect information

Perfect information

Perfect vs imperfect information

Perfect information

Imperfect information

Perfect vs imperfect information

Perfect information

Imperfect information

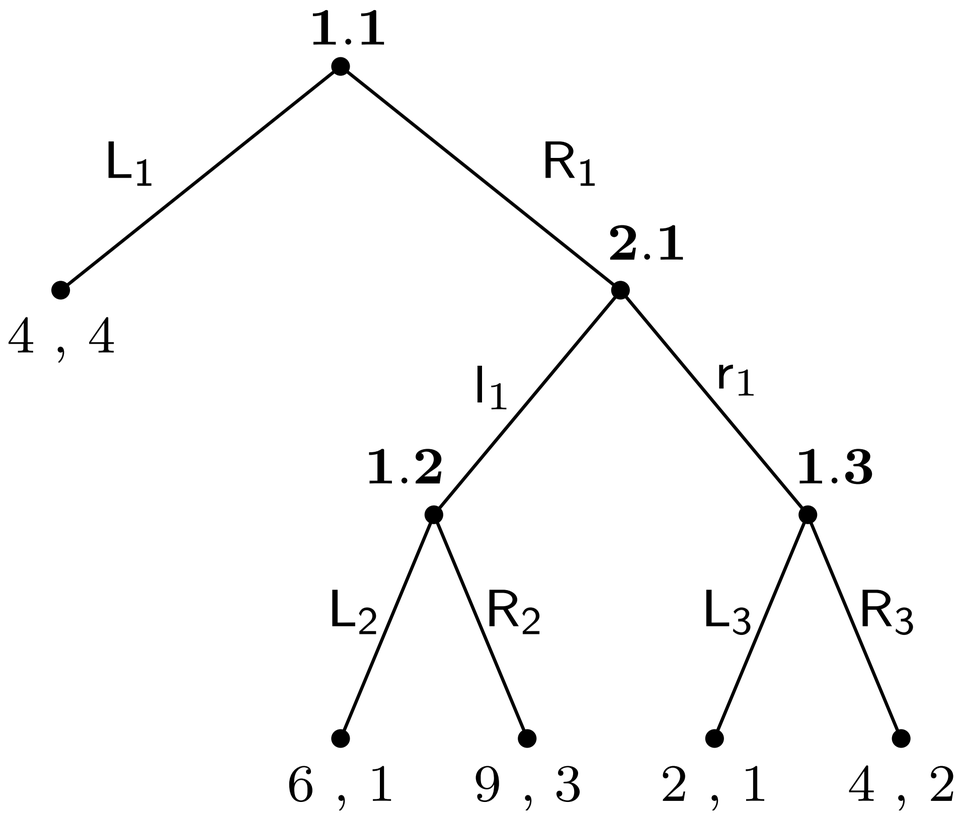
Information sets

Strategies

A function associating to each information set a probability distribution over the available actions

Strategies

A function associating to each information set a probability distribution over the available actions

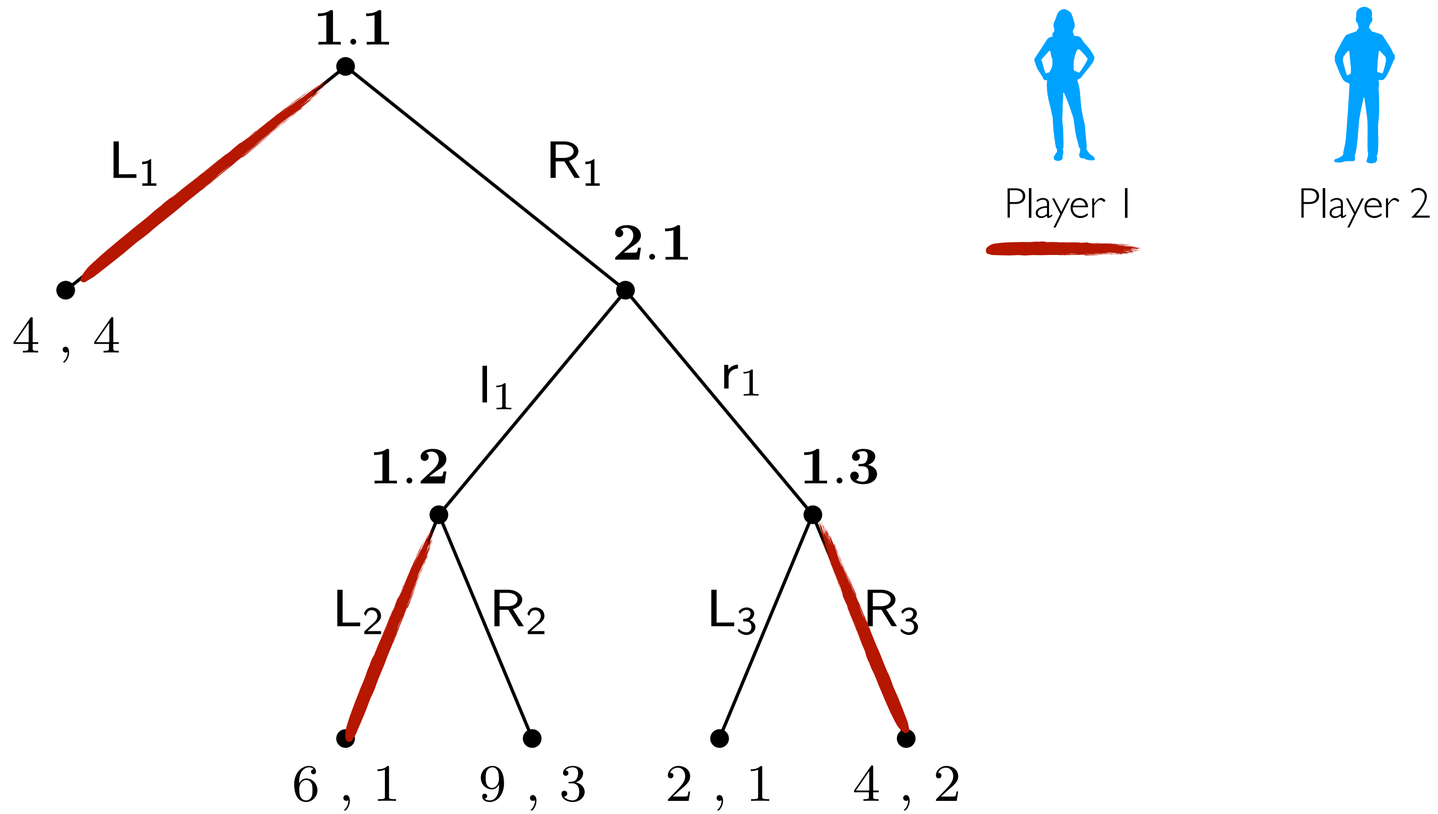


Player 1

Player 2

Strategies

A function associating to each information set a probability distribution over the available actions



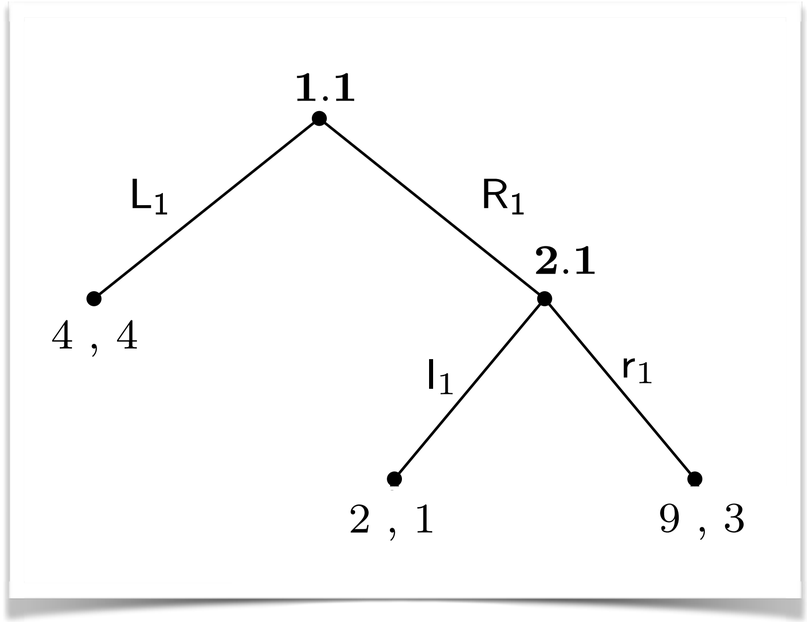
Nash equilibrium

A joint combination of strategies, stable w.r.t. unilateral deviations of a single player

Nash equilibrium

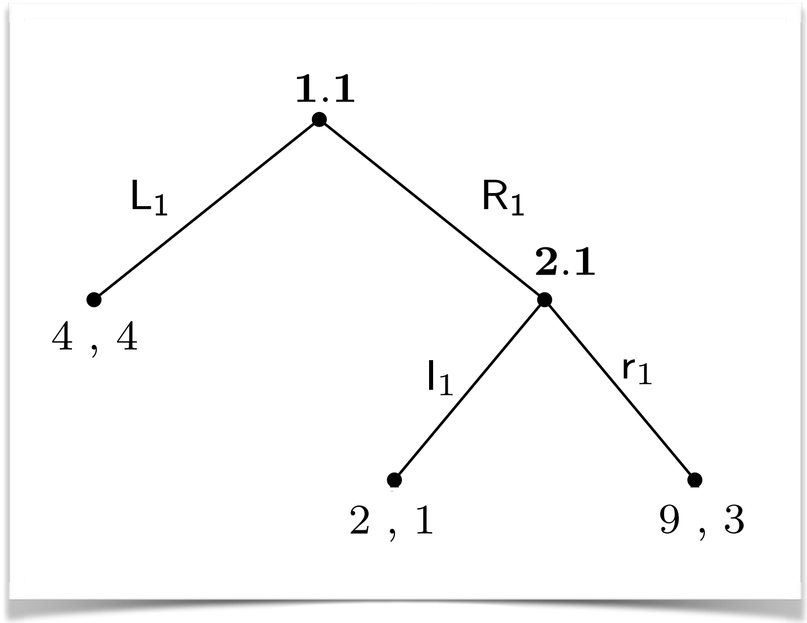
A joint combination of strategies, stable w.r.t. unilateral deviations of a single player

Player 1 Player 2



Nash equilibrium

A joint combination of strategies, stable w.r.t. unilateral deviations of a single player



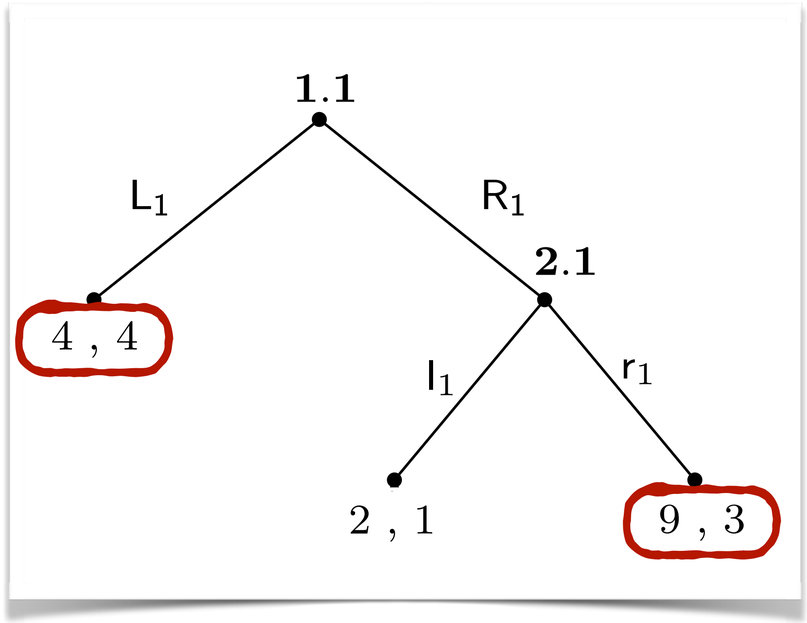
Player 2

Player I

	l_1	**r_1**
L_1	4 , 4	4 , 4
R_1	2 , 1	9 , 3

Nash equilibrium

A joint combination of strategies, stable w.r.t. unilateral deviations of a single player



Player 2

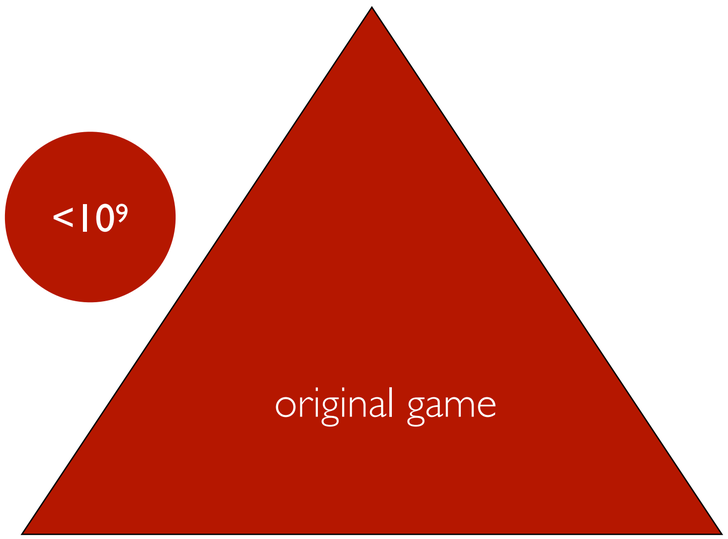
Player 1

	l_1	r_1
L_1	4 , 4	4 , 4
R_1	2 , 1	9 , 3

Nash equilibrium

- Mixed strategy NE
- Every n-player finite game has at least one Nash Equilibrium profile in mixed strategies
- epsilon-Nash Equilibrium

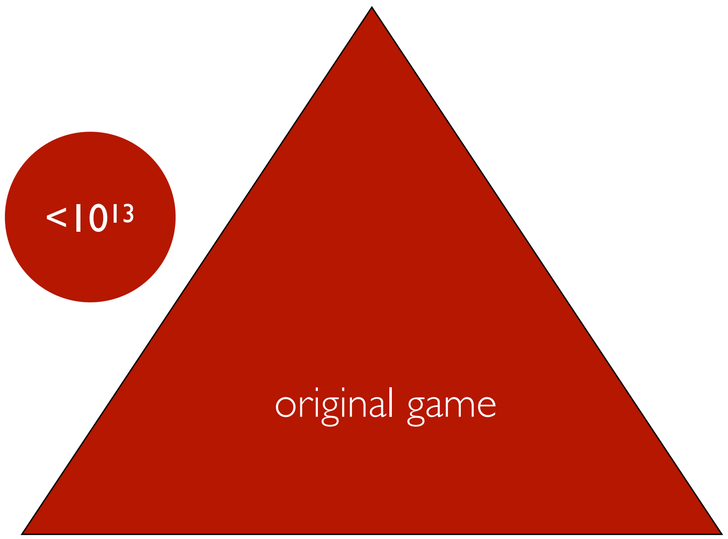
State-of-the-art game solving



Linear programming

(Shi, Littman, 2000)
(Billings et al., 2003)
(Gilpin, Sandholm, 2007)

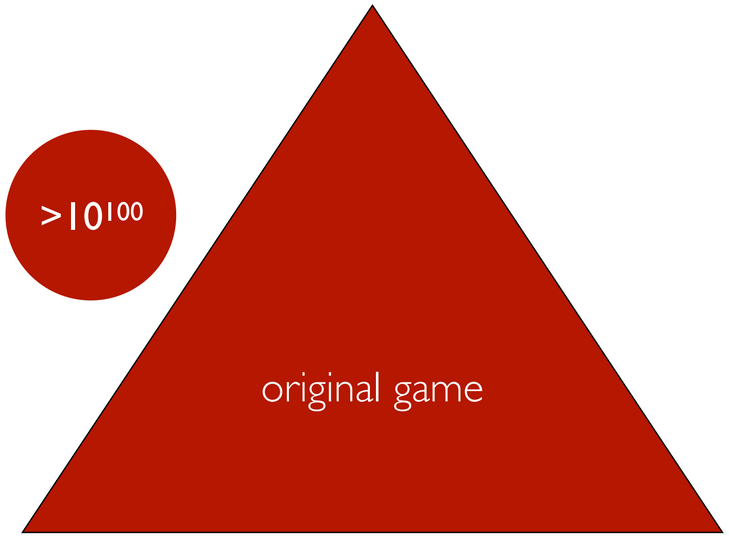
State-of-the-art game solving



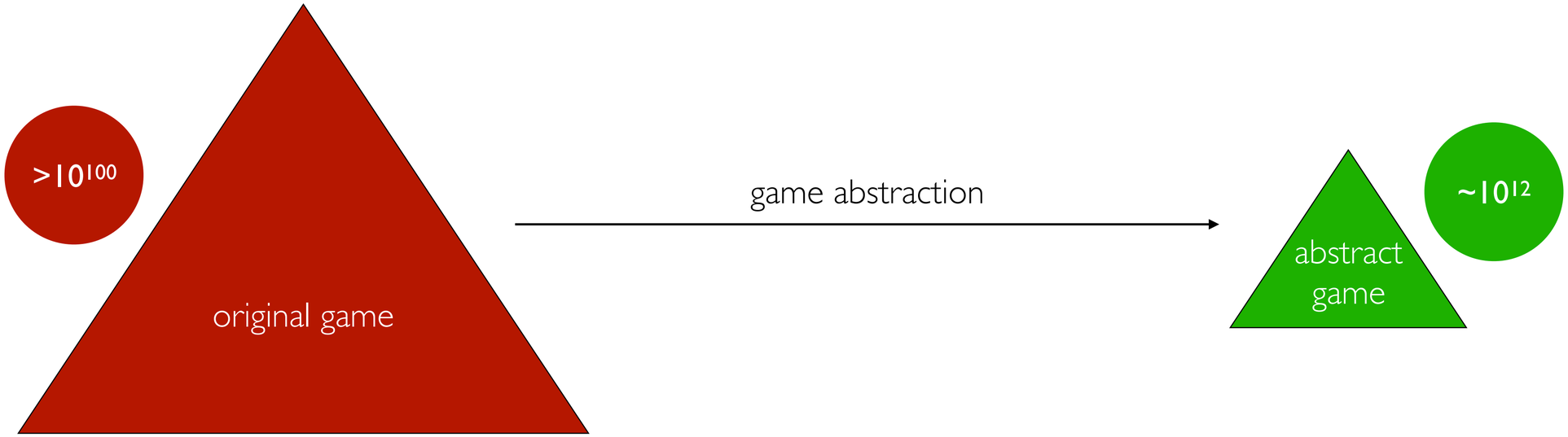
No-regret learning

Zinkevich et al., 2008

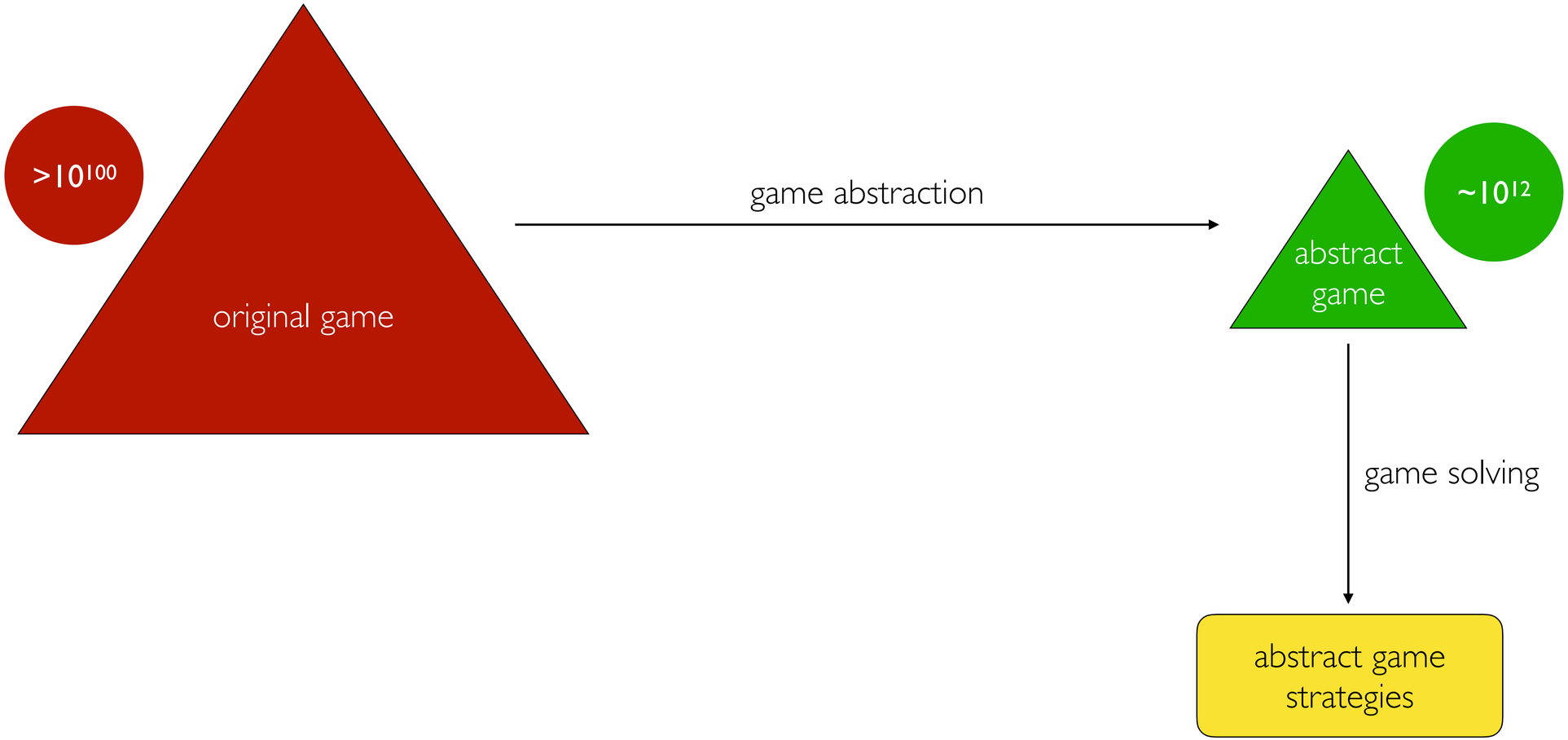
State-of-the-art huge-game solving: *preplay*



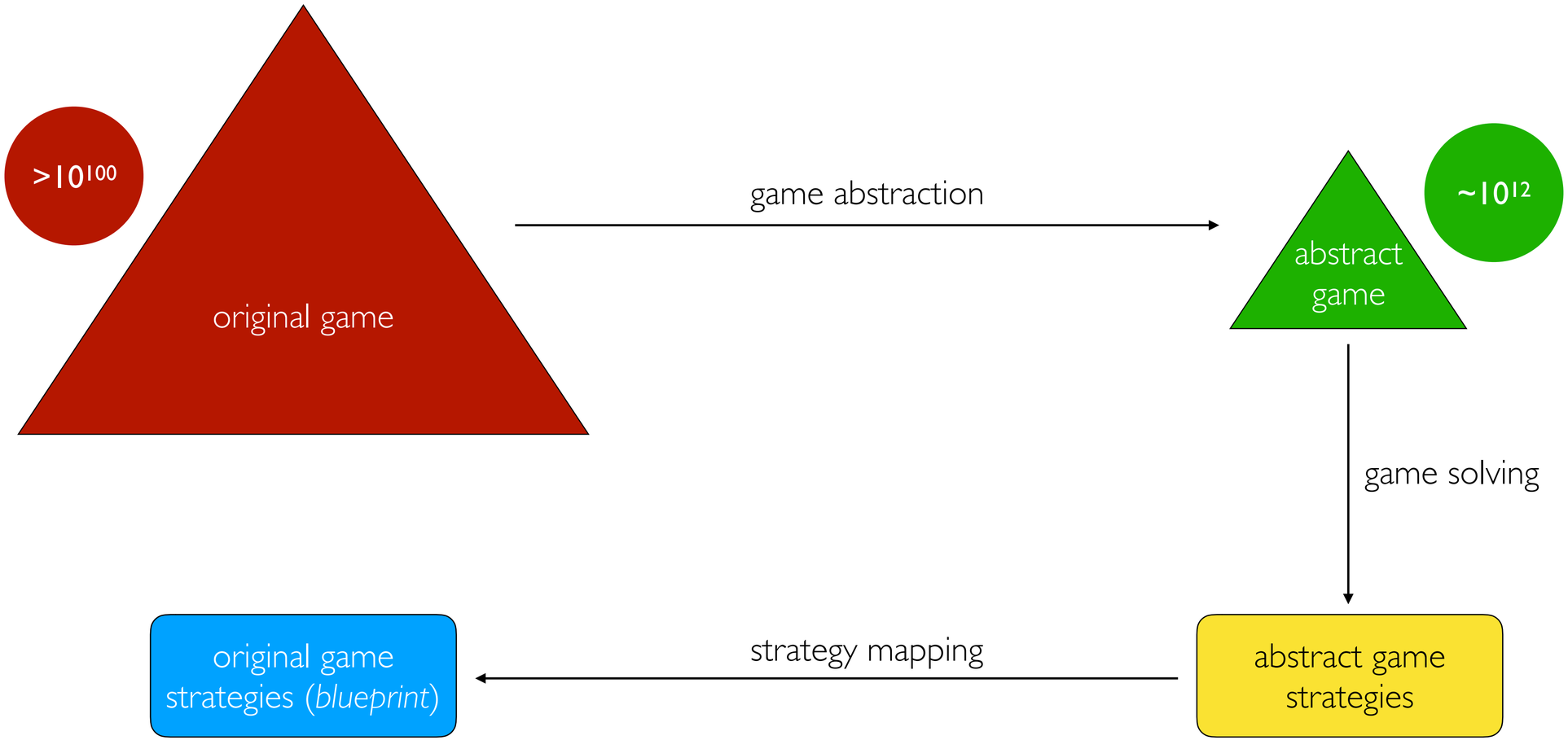
State-of-the-art huge-game solving: *preplay*



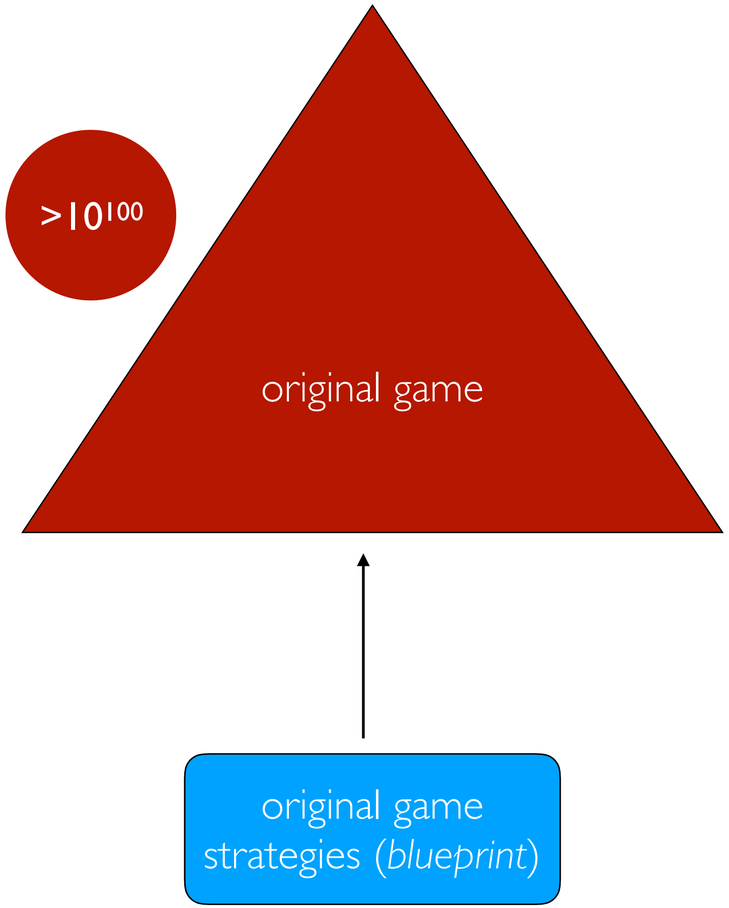
State-of-the-art huge-game solving: *preplay*



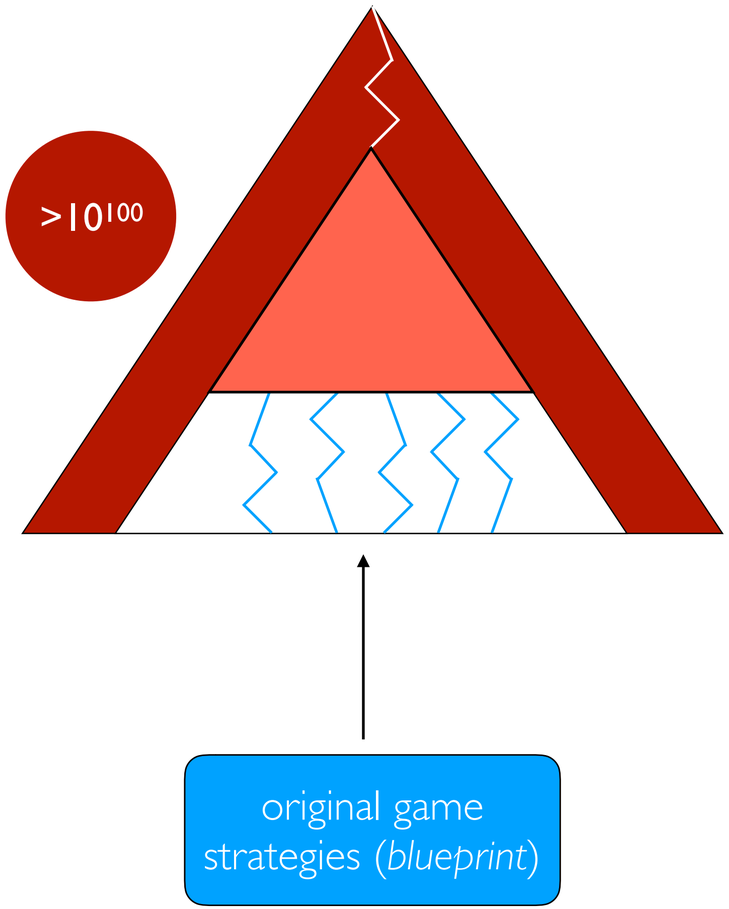
State-of-the-art huge-game solving: *preplay*



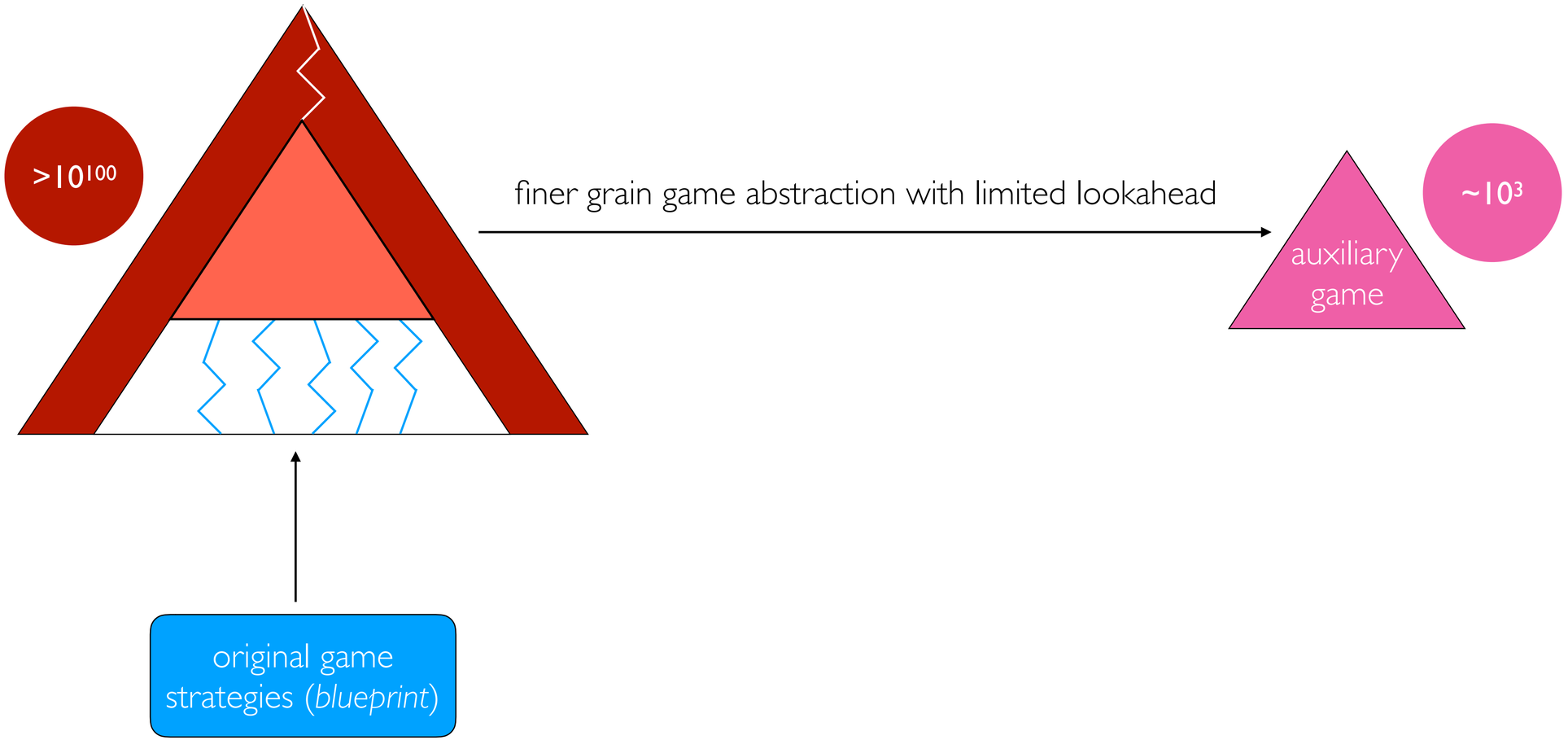
State-of-the-art huge-game solving: *play*



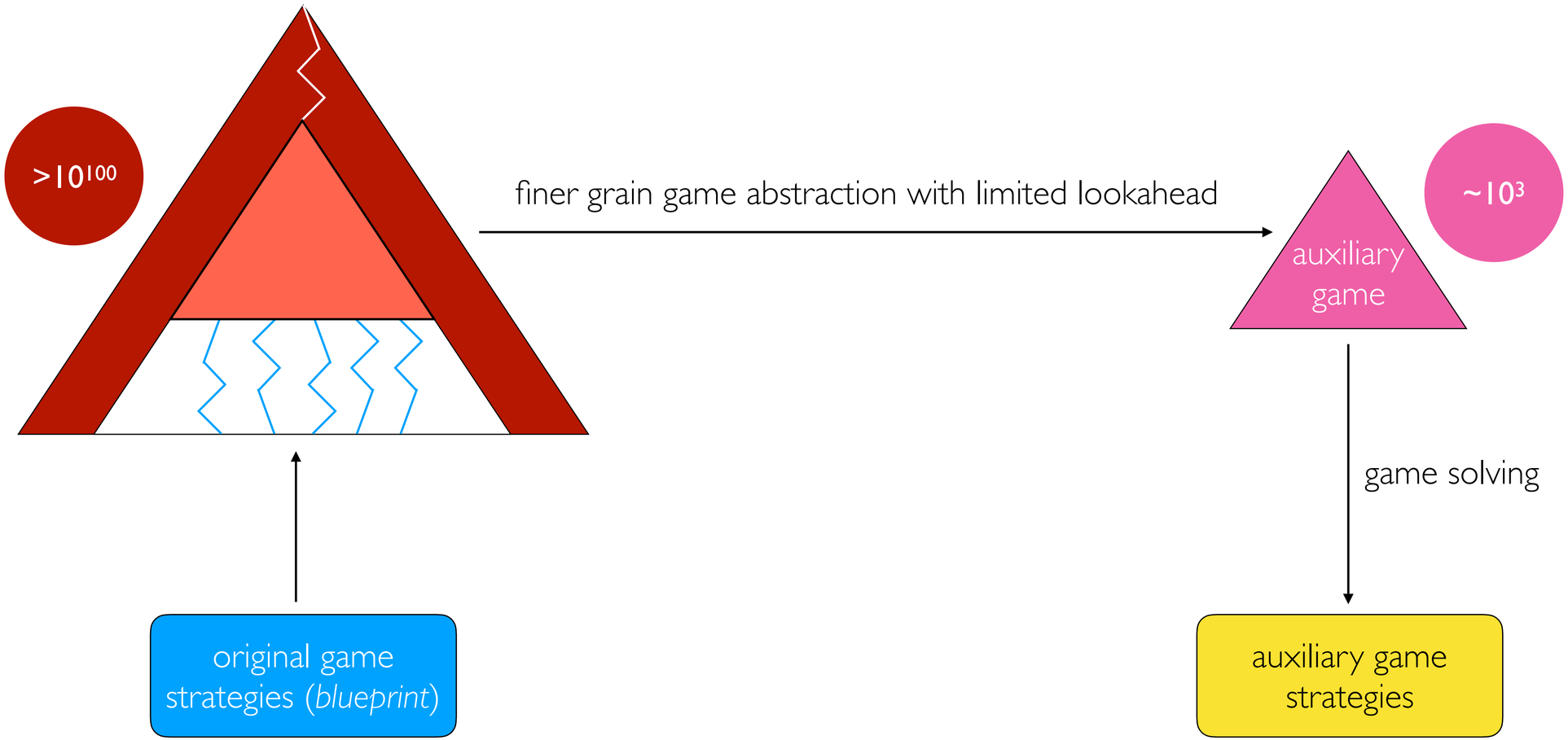
State-of-the-art huge-game solving: *play*



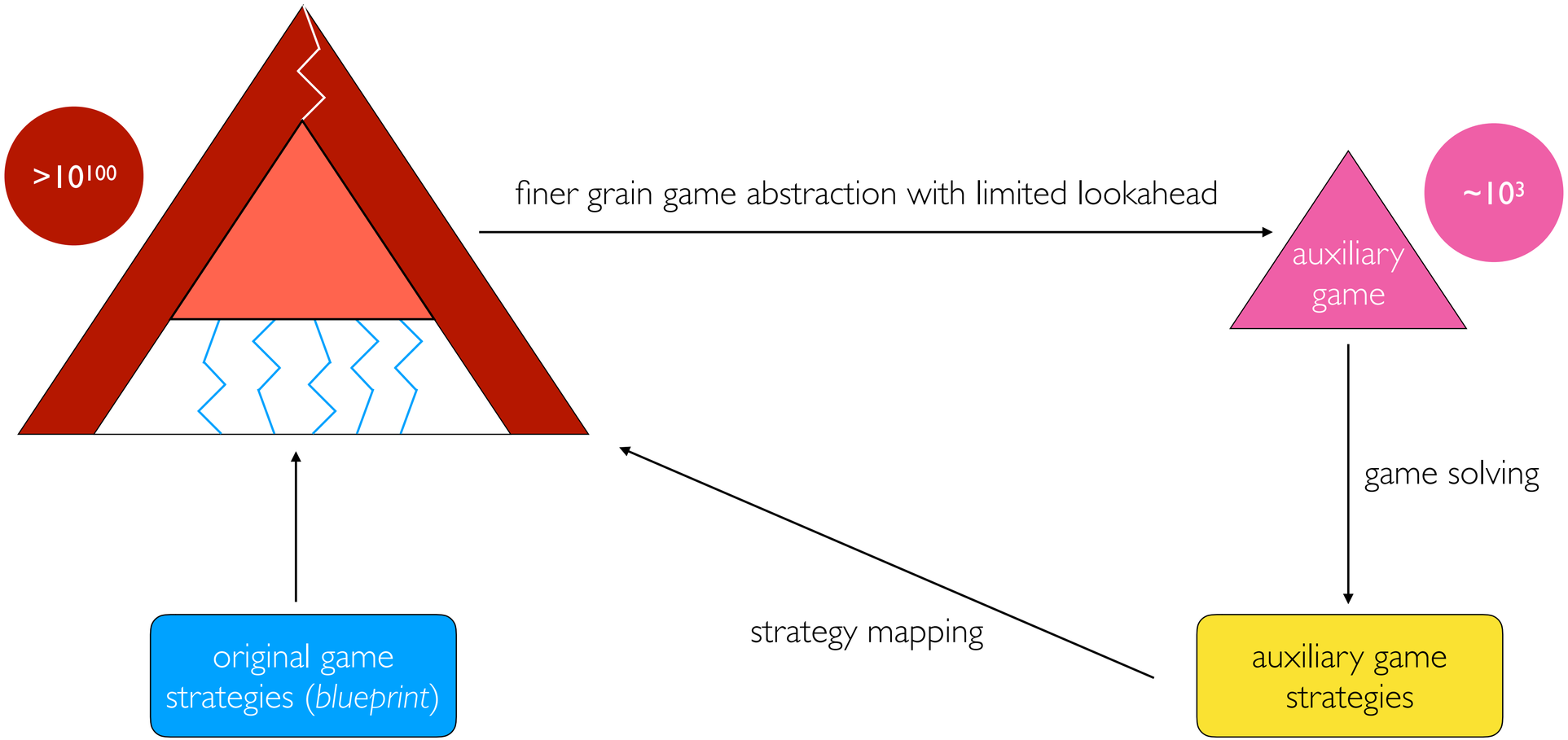
State-of-the-art huge-game solving: *play*



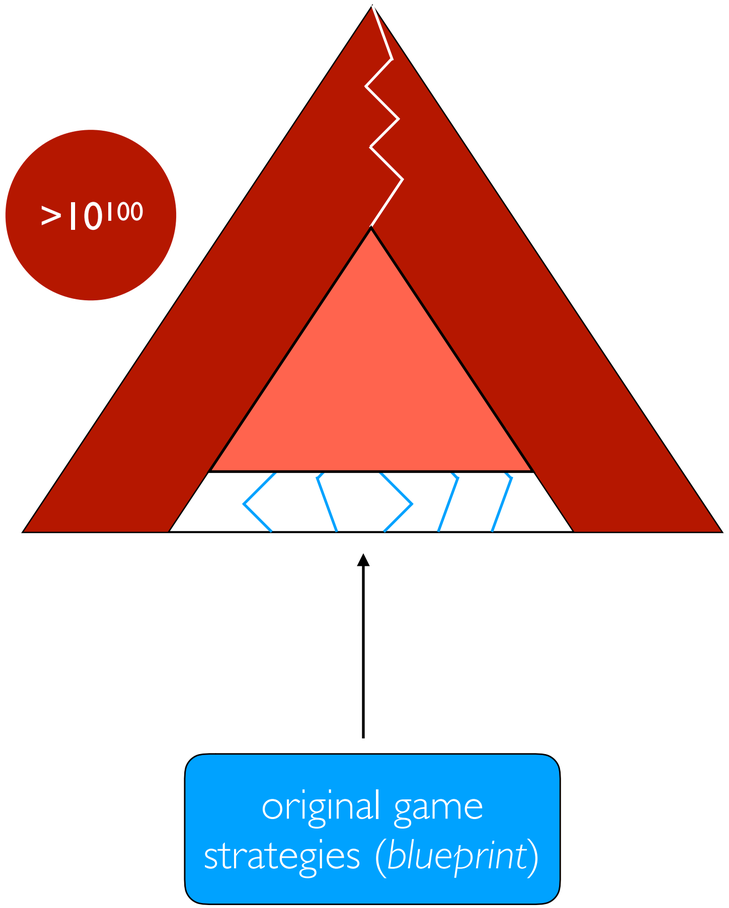
State-of-the-art huge-game solving: *play*



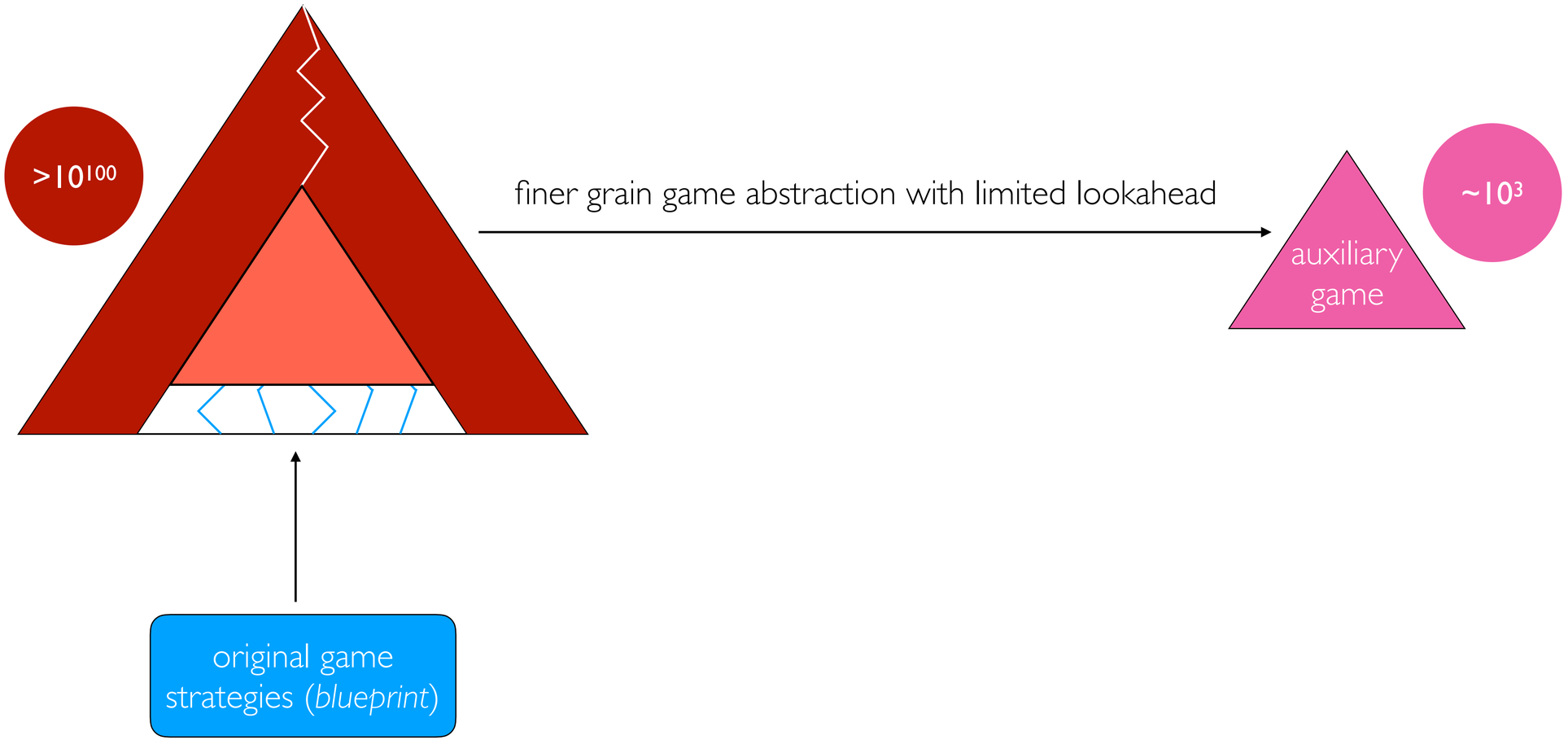
State-of-the-art huge-game solving: *play*



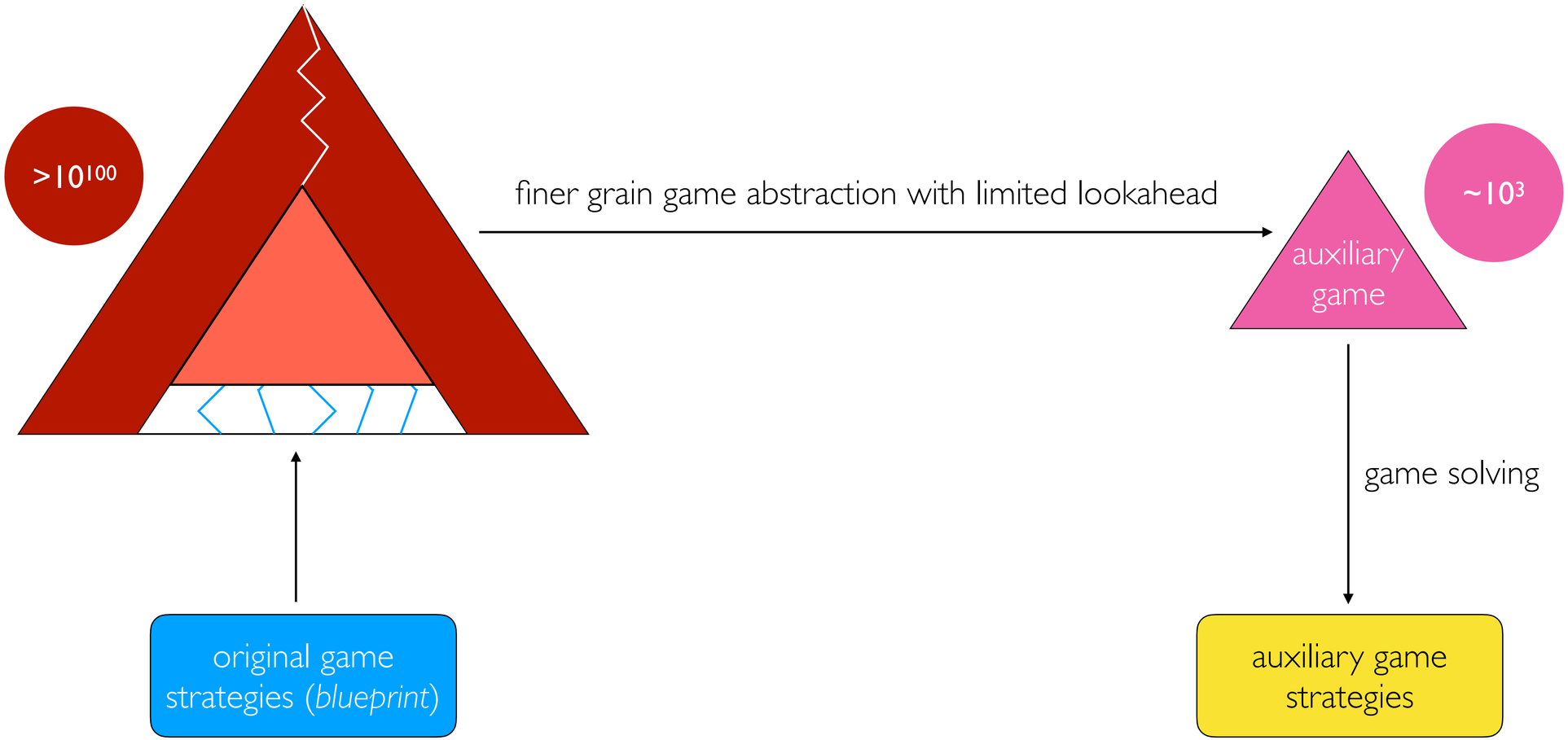
State-of-the-art huge-game solving: *play*



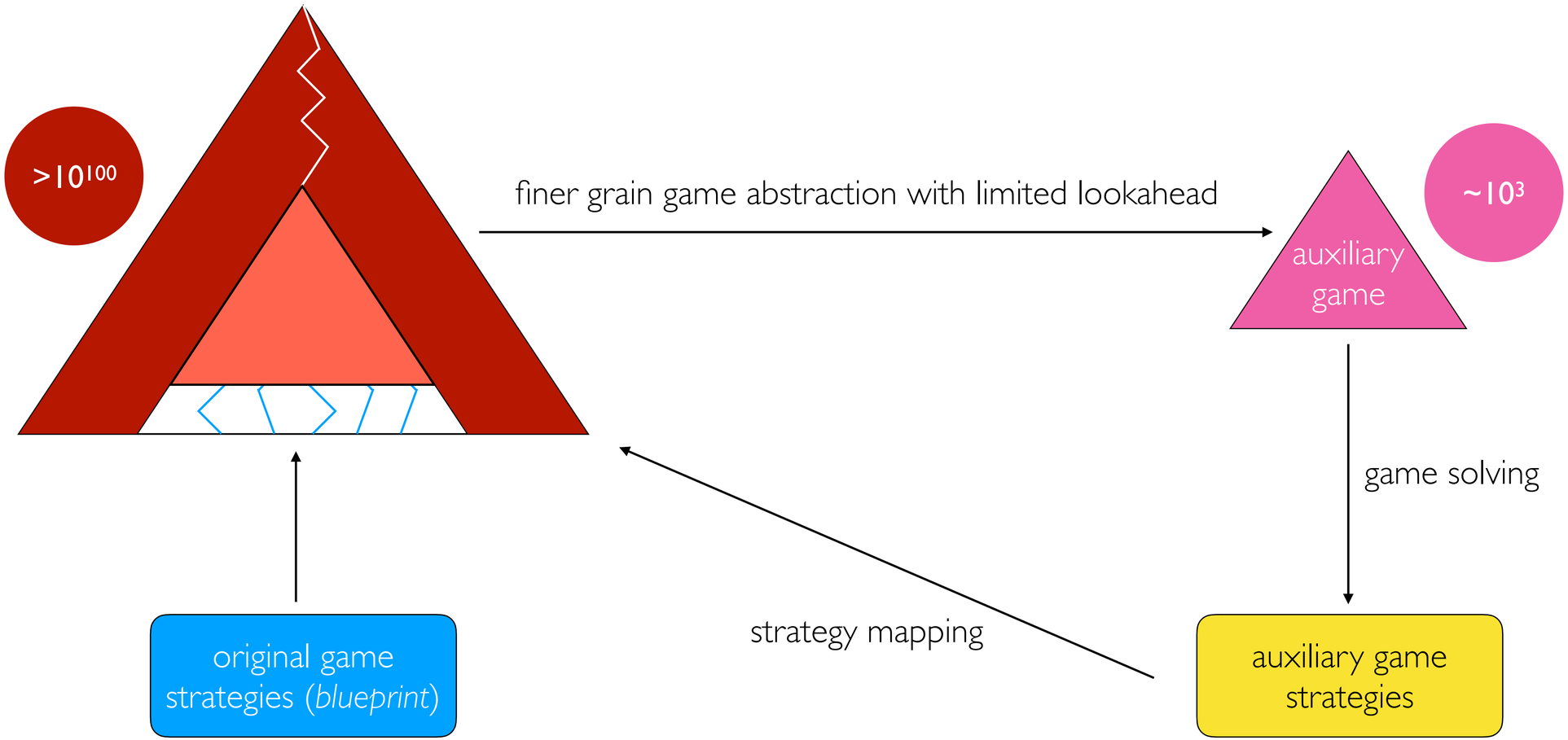
State-of-the-art huge-game solving: *play*



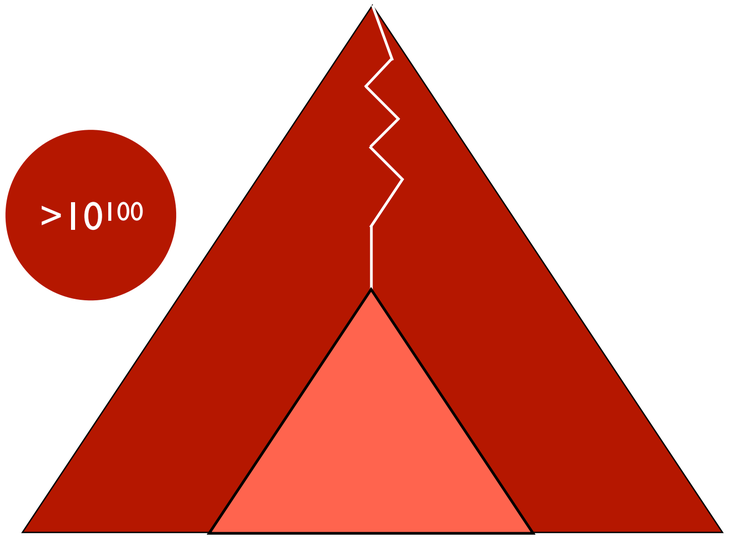
State-of-the-art huge-game solving: *play*



State-of-the-art huge-game solving: *play*

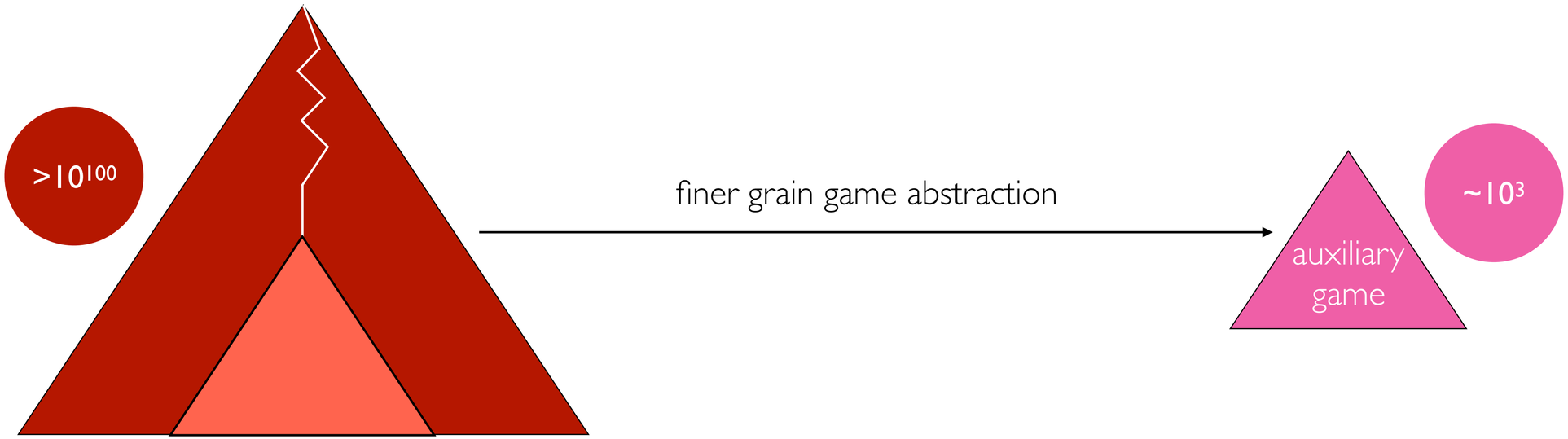


State-of-the-art huge-game solving: *play*



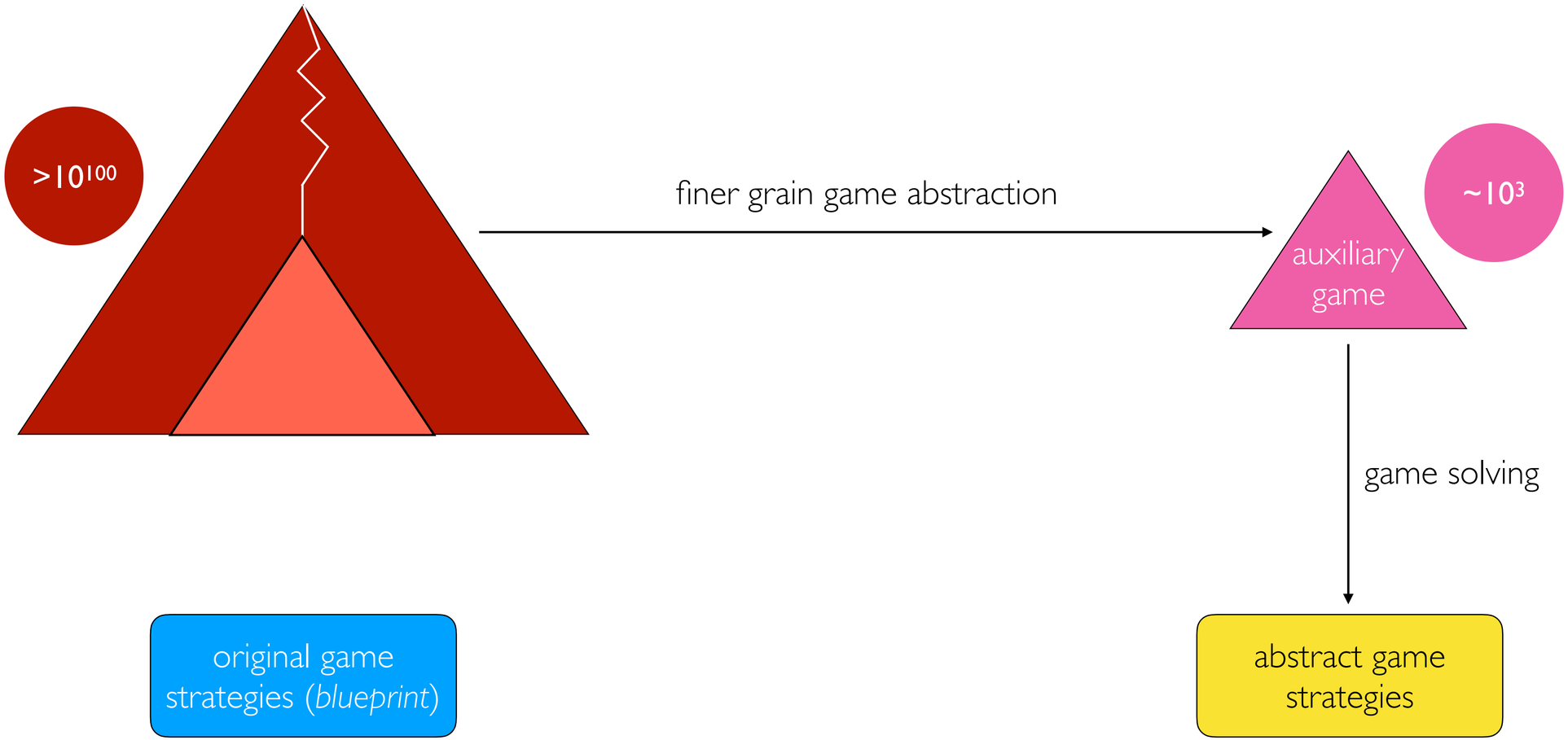
original game
strategies (*blueprint*)

State-of-the-art huge-game solving: *play*

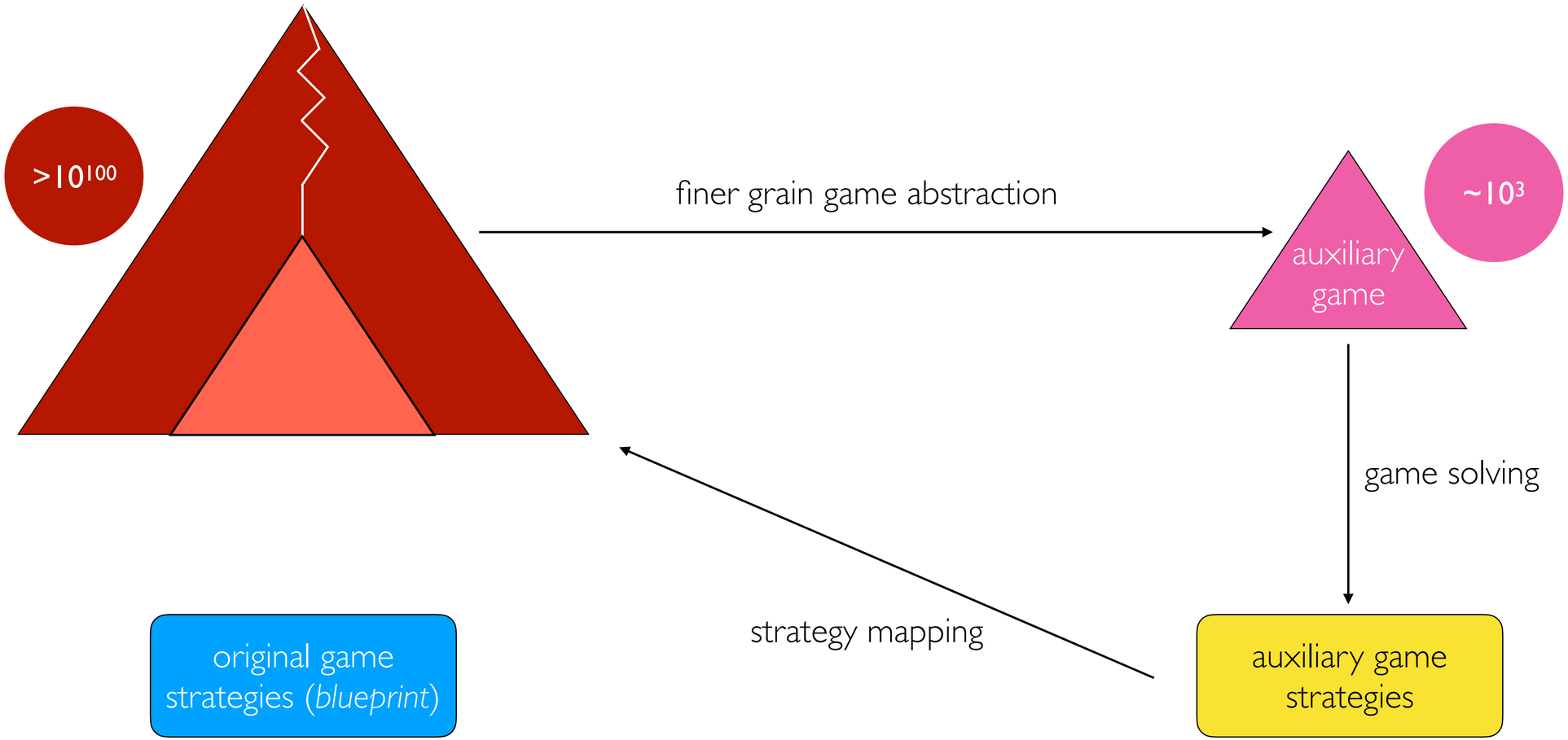


original game strategies (*blueprint*)

State-of-the-art huge-game solving: *play*



State-of-the-art huge-game solving: *play*



Game solving

- Optimization problem definition
- Counter Factual Regret minimization plus (CFR+) (Zinkevich et al., *Counter Factual Regret minimization in games with incomplete information*, NeurIPS, 2008)
- Monte Carlo CFR+ (Lanctot et al., *Monte Carlo sampling for regret minimization in extensive gaems*, NeurIPS, 2009)

Abstractions

- Smaller version of the game capturing the most essential properties of the real domain

- Abstracted game solution provides a useful approximation of the optimal strategy

Abstractions

- Smaller version of the game capturing the most essential properties of the real domain
- Abstracted game solution provides a useful approximation of the optimal strategy

- Lossless information abstractions (Gilpin, Sandoholm, 2007)
- Lossy information abstractions (Gilpin, Sandholm, 2007)

Abstractions

- Information abstractions
  - Linear programming and bucketing
  - Expectation-based and potential-aware abstractions (Gilpin, Sandholm, 2007)
- Action abstractions
  - Actions discretization and game refinement (Brown, Sandholm, 2015)
- Simulation-based abstractions (Tuyls et al. 2018), (Viqueira et al., 2019)

Beyond abstractions

Libratus, 2017

Carnegie Mellon University

Game Abstraction + MCCFR self-play
Nested subgame solving
Self-improvement

Beyond abstractions

Libratus, 2017

Pluribus, 2019

Carnegie Mellon University

Game Abstraction + MCCFR self-play
Nested subgame solving
Self-improvement

facebook Artificial Intelligence

Extended to 6 players

Simulation-based abstractions

- Bottom-up approach based on *data* (artificial learning)
- Game *traces* (observed vs generated)
- Query an *oracle* for noisy payoff given a strategy
- *Model-free*

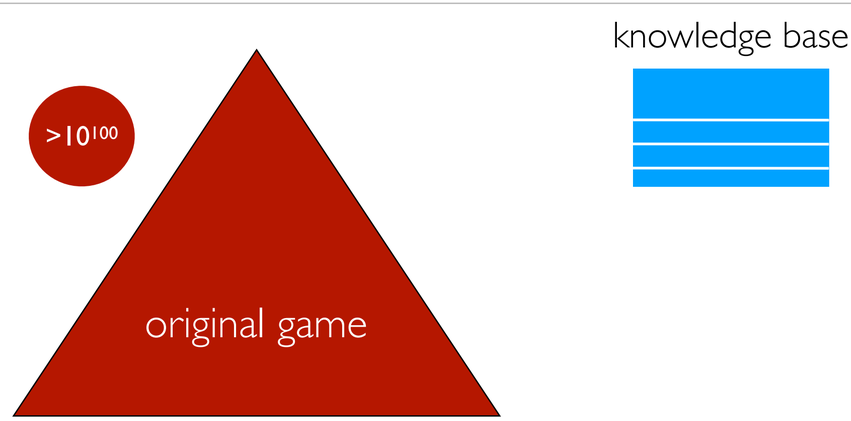
The problem

- Real-world games and strategic scenarios are *too large* to be represented
- No clear *domain-independent* abstraction approach was presented to solve these games
- Poker as the main reference application

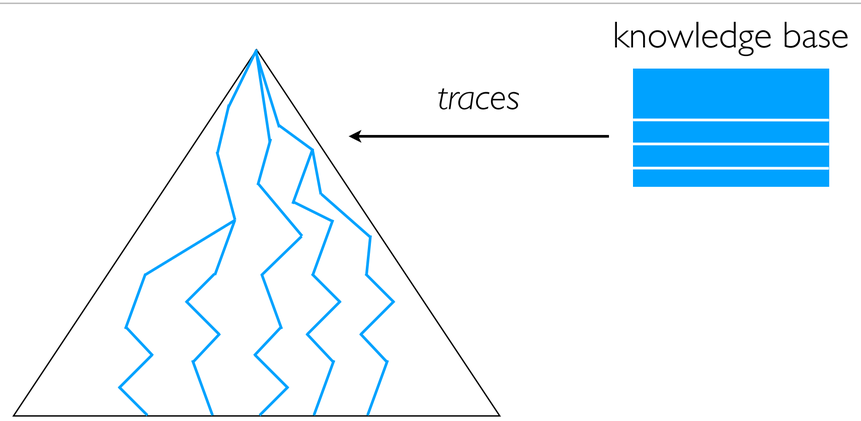
Our goal

Develop a *bottom-up model-free* abstraction approach, supported by *theoretical guarantees,* able to find *mixed strategy* Nash equilibria in *any* extensive-form game in a *simulation-based* fashion.

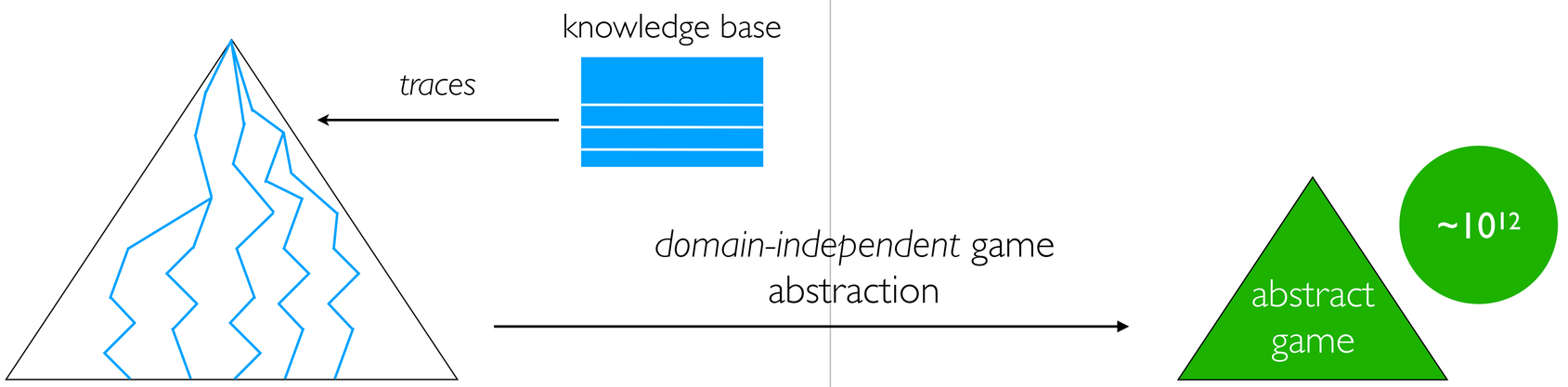
Idea

	Original game	Abstract game
Game		
Strategy		

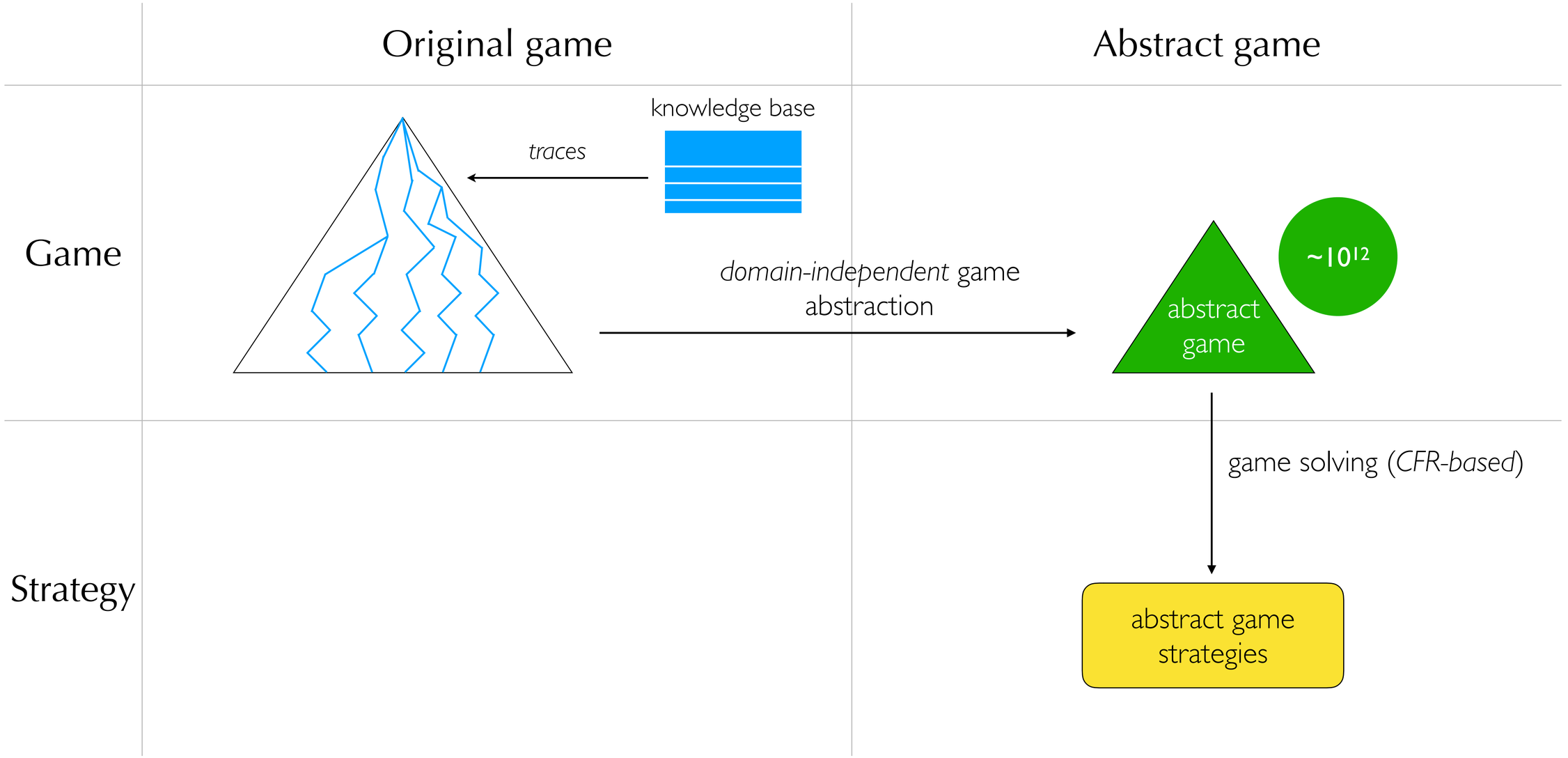
Idea

	Original game	Abstract game
Game		
Strategy		

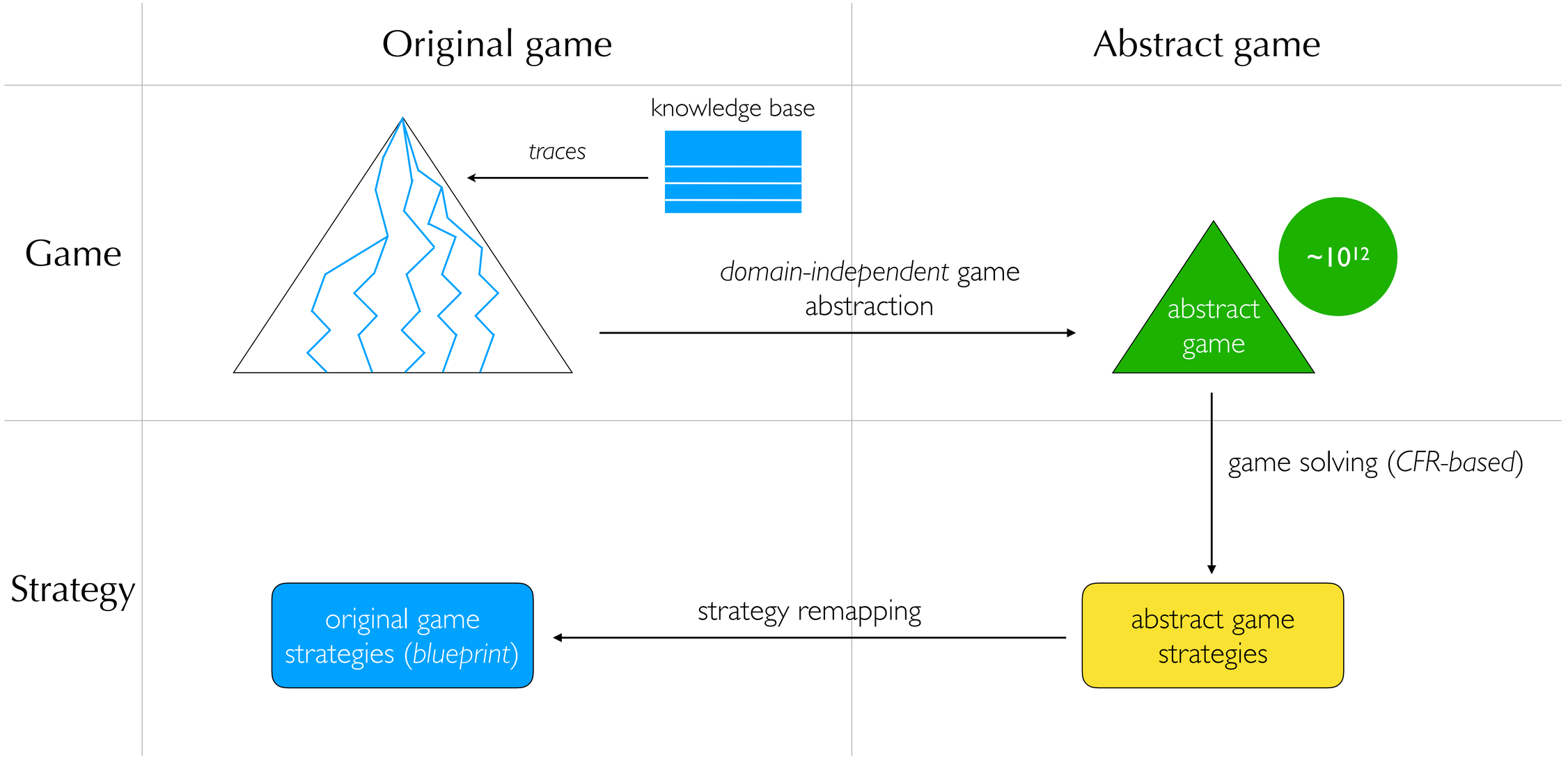
Idea

	Original game	Abstract game
Game		
Strategy		

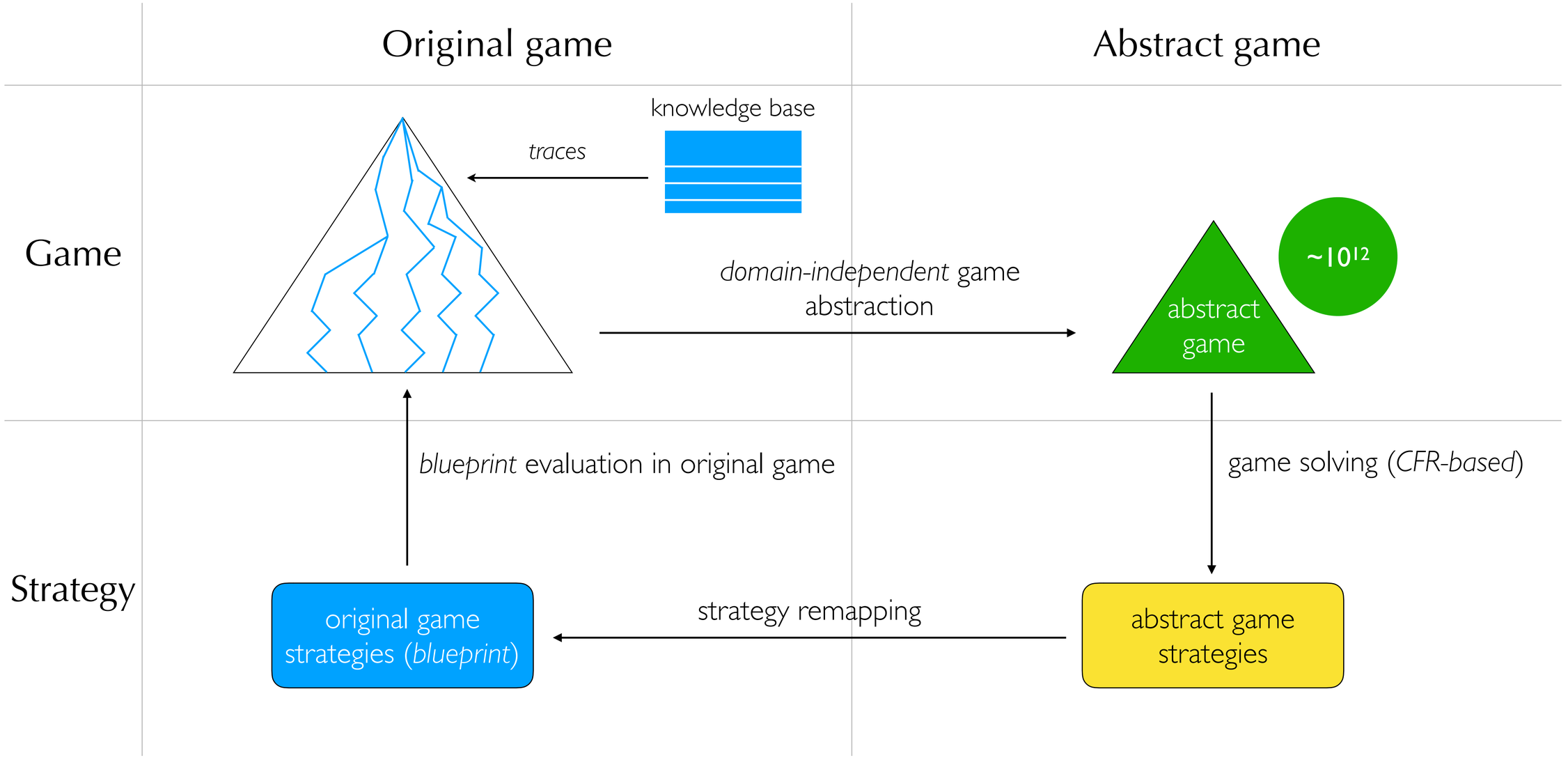
Idea


Original game
Abstract game
Game
Strategy
knowledge base
traces
domain-independent game abstraction
~10^12
abstract game
game solving (CFR-based)
abstract game strategies

Idea

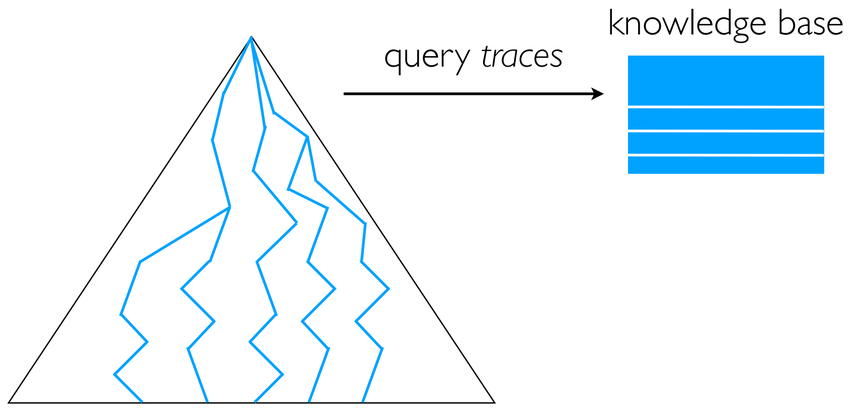

Original game
Abstract game
Game
Strategy
knowledge base
traces
domain-independent game abstraction
~10^12
abstract game
game solving (CFR-based)
original game strategies (blueprint)
strategy remapping
abstract game strategies

Idea


Original game
Abstract game
Game
Strategy
knowledge base
traces
domain-independent game abstraction
~10^12
abstract game
blueprint evaluation in original game
game solving (CFR-based)
original game strategies (blueprint)
strategy remapping
abstract game strategies

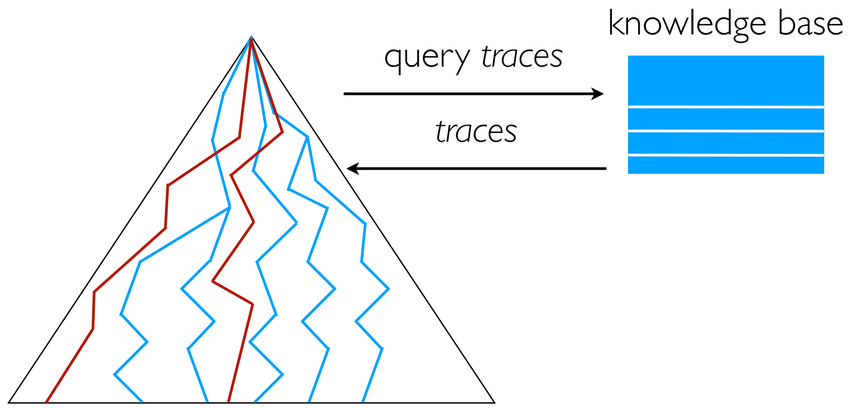
Idea

Blueprint exploitability evaluation
Traces gathering to refine abstraction

	Original game	Abstract game
Game		
Strategy	*blueprint* evaluation in original game original game strategies (*blueprint*)	

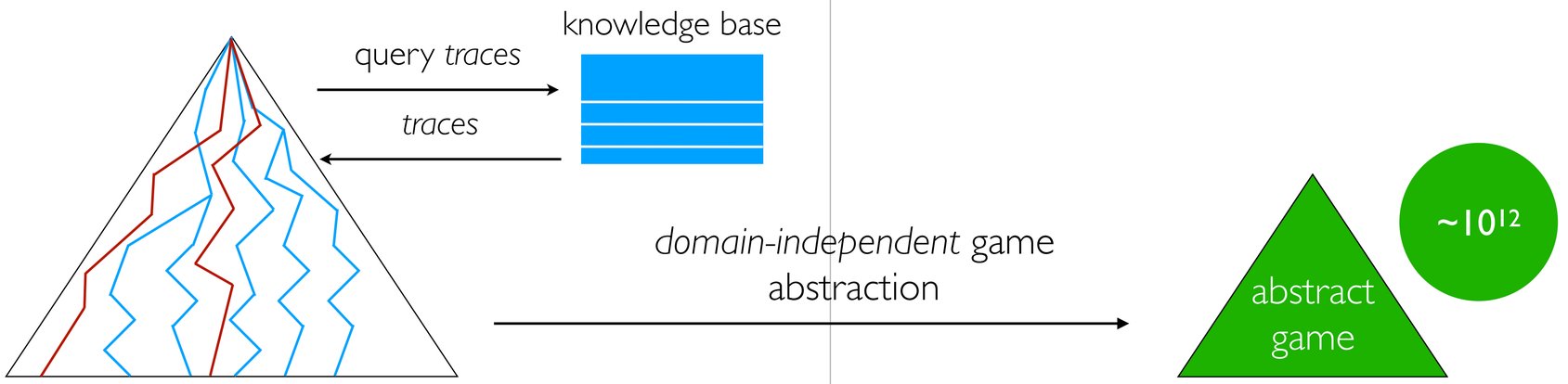
Idea

Blueprint exploitability evaluation
Traces gathering to refine abstraction

	Original game	Abstract game
Game		
Strategy	*blueprint* evaluation in original game original game strategies (*blueprint*)	

Idea

Blueprint exploitability evaluation
Traces gathering to refine abstraction

	Original game	Abstract game
Game		
Strategy		

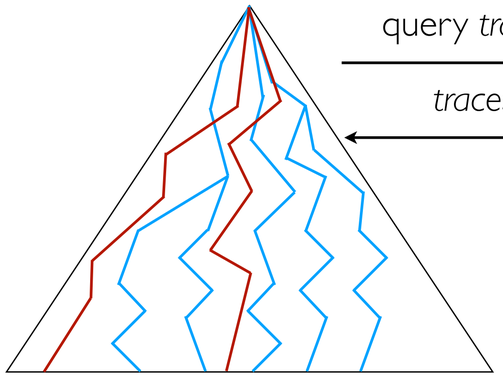
Idea

Blueprint exploitability evaluation
Traces gathering to refine abstraction

Original game

Abstract game

Game

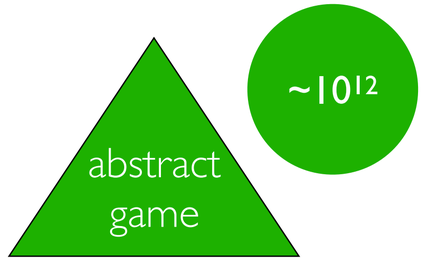


query *traces*

traces

knowledge base

domain-independent game abstraction



~10^{12}

Strategy

game solving (*CFR-based*)

abstract game strategies

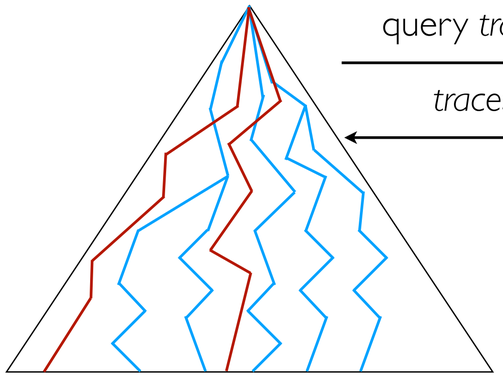
Idea

Blueprint exploitability evaluation
Traces gathering to refine abstraction

Original game

Abstract game

Game



query *traces*

traces

knowledge base

domain-independent game abstraction

abstract game

~10^{12}

game solving (*CFR-based*)

Strategy

original game strategies (*blueprint*)

strategy remapping

abstract game strategies

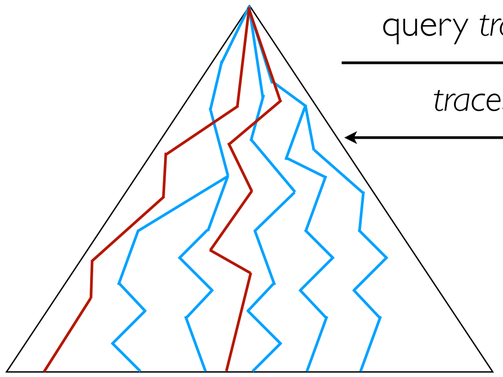
Idea

Blueprint exploitability evaluation
Traces gathering to refine abstraction

Original game

Abstract game

Game

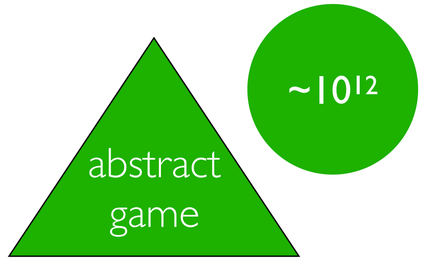


query *traces*

traces

knowledge base

domain-independent game abstraction



$\sim 10^{12}$

blueprint evaluation in original game

game solving (*CFR-based*)

Strategy

original game strategies (*blueprint*)

strategy remapping

abstract game strategies

In a nutshell...

- Real-world strategic conditions are *too large* to be represented and analyzed
- Need for a *domain-independent* way to solve large games
- Exploit *data availability* (*traces*) and *artificial learning techniques*

In a nutshell...

- Real-world strategic conditions are *too large* to be represented and analyzed
- Need for a *domain-independent* way to solve large games
- Exploit *data availability* (*traces*) and *artificial learning techniques*

- How to abstract the game starting from *traces*?
- How to choose future *traces*?

Applications

Contract Bridge

Applications

Contract Bridge

Car Racing

Applications

Contract Bridge

Car Racing

Cybersecurity