Organizing Software Development Using Competitions

Karl Lieberherr, Northeastern University,
College of Computer and Information Science, PRL
joint work with
Ahmed Abdelmeged (PhD Dissertation)
Ruiyang Xu

Quick Introduction

- Motivation
- 3 Examples

Solving Problems using the Crowd

They look at the administrator who only acts as **referee**. We use the crowd for **fair peer evaluation**!

They want to be heard.

Some are better than others: **meritocracy**.

Some want to game the system by colluding/lying.

We want to extract a **provably** reliable signal ("best", **collusion-resistant** solution) from the complex behavior of the crowd.

- In 2011, researchers from the Harvard Catalyst Project were investigating the potential of crowdsourcing genome-sequencing algorithms.

- So, they collected a <u>few million sequencing problems</u> and developed an electronic judge that evaluates sequencing algorithms by how well they solve these problems.

- And, they set up a two-week open online competition on TopCoder with a total prize pocket of $6000.

- The results were astounding!

-- Nature Biotechnology, 31(2):pp. 108–111, 2013.

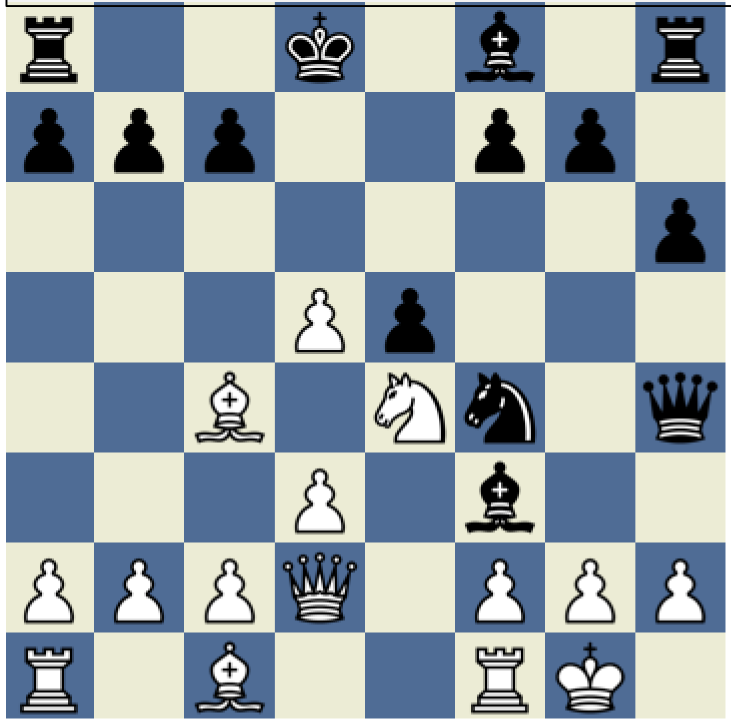
- “... A two-week online contest ... produced over 600 submissions ... . Thirty submissions exceeded the benchmark performance of the US National Institutes of Health’s MegaBLAST. The best achieved both greater accuracy and speed (1,000 times greater).”

- We want to lower the barrier to entry for organizing such competitions by having “meaningful” competitions where participants assist the administrator in evaluating their peers.
    - Administrator: WHAT
    - Players = Participants: HOW

Organization is based on side-choosing games. What is a side-choosing game (SCG)?

It is about a claim C.
Structure: ChessBoard.
Logical Sentence:
For given ChessBoard-instance:
Exists move for Black
ForAll moves of White
Exists move for Black:
  White King is mate

Black moves first and mates in 2 moves



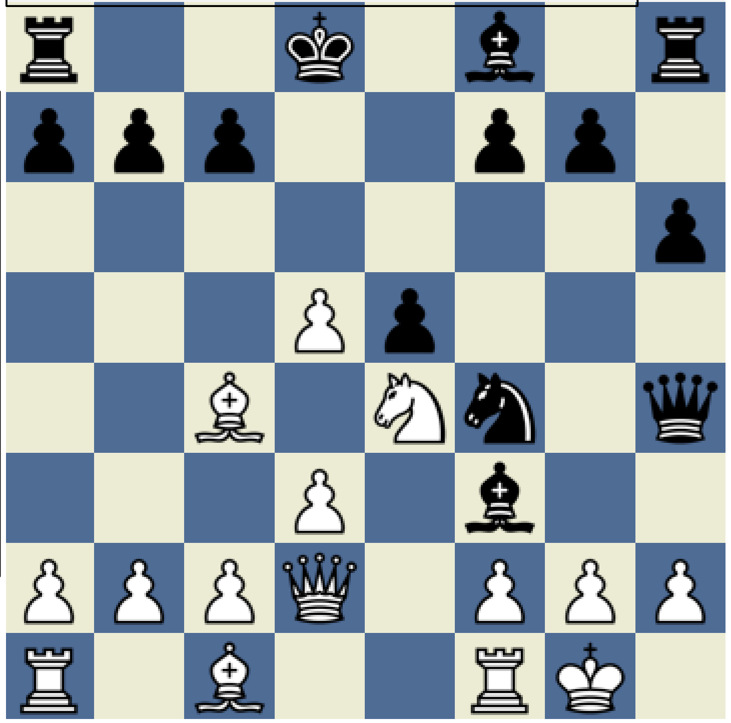
We ask 2 players x and !x:
x is a Proponent
!x is an Opponent

x and !x must defend their side-choice by winning the game.

What is a side-choosing game?

There is always a winner and a loser. No ties. Protocol for discourse is determined by logical sentence.

Claim: Black mates in 2 moves



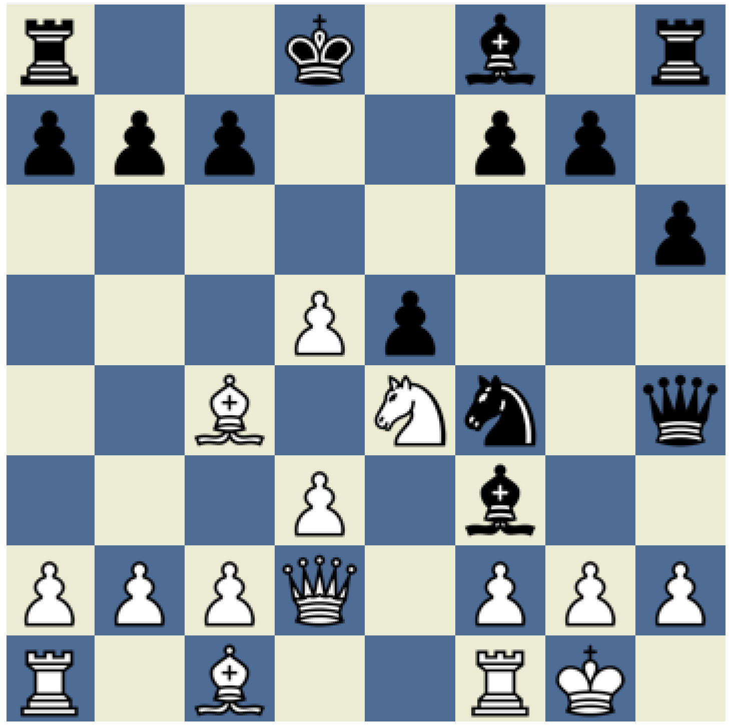
It is about a claim.
Structure: ChessBoard.
Logical Sentence:
For given ChessBoard-instance cb:
Exists move b1 for Black
ForAll moves w1(b1) of White
Exists move b2(b1,w1) for Black:
  WhiteKingIsMate(b1,w1,b2)

We ask 2 players x and !x:
x is a **Proponent**
!x is a **Proponent**

We have one of them play as devil's advocate. Say x is devil's advocate. If x wins, !x was not a serious Proponent. Devil's Advocate = **Forced**.

Side-Choosing Game (SCG) Cases

Sx (Side(x))	S!x (Side(!x))	P (Proponent)	W (Winner)
P	O	x	x
P	O	x	!x
P	P	x	x
P	P	x	!x
P	P	!x	x
P	P	!x	!x
…			



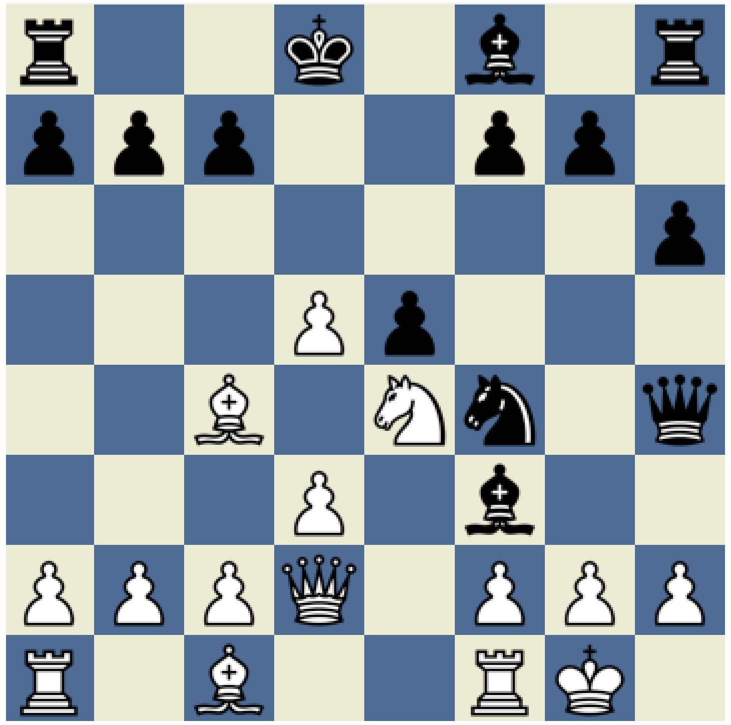
How should we rank players? Player with the most wins? But there are cheap wins: when the other is forced. Should we rank based on wins where other is not forced???

Another Issue: Distributing the Evaluation Work.

- Administrator: Defines claim; checks that rules are followed; determines who wins and loses and keeps track of results.
- Does Administrator have to solve the problems = develop winning strategies for claims?
- No! We want the administrator only be a **referee** who is interested in problem solutions but who wants to get them from the players.
- How can we make sure that the **peer evaluation** is **fair**?

Voting with Justification

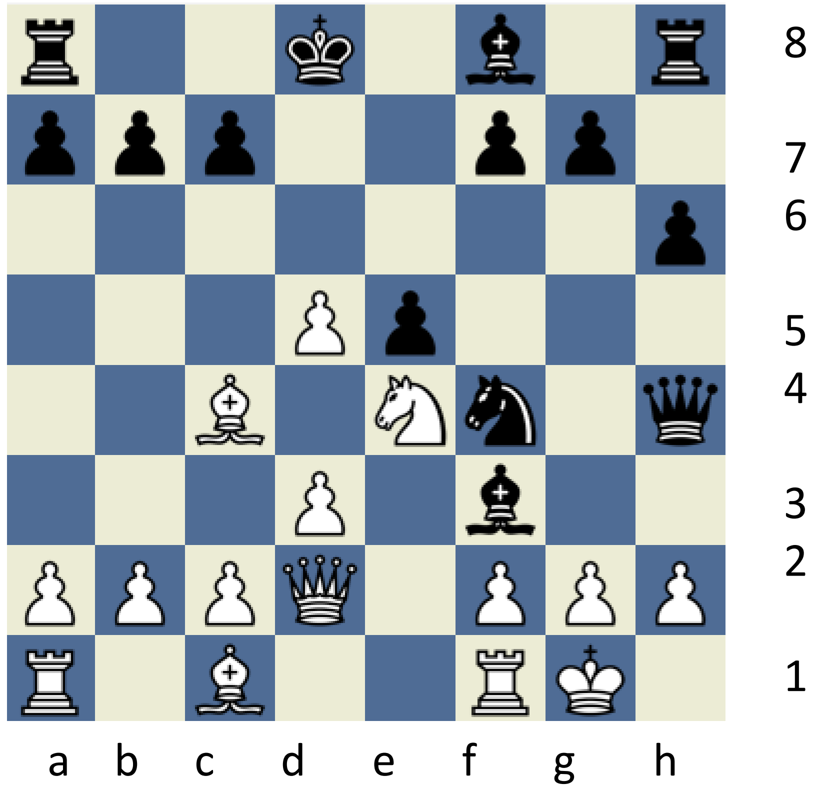
- You cannot just say: I am a Proponent.
- You must justify your choice by game play.
- As Proponent: you must win.
- As Opponent: you must prevent the other from winning, i.e., you must win.



To play this SCG

W	L	F
?	?	?

Claim: Black mates in 2 moves



Proponents?
Opponents?

W: Winner
L: Loser
F: Forced

Software Development: specifying a function

- Pre and Post conditions for requirements
- Exists gcd in Function(Nat,Nat -> Nat) ForAll x,y in Nat Exists d in Nat:
  - d=gcd(x,y) $\wedge$
  - divides(d,x) $\wedge$ divides(d,y) $\wedge$
  - ! Exists s in Nat ((s>d) $\wedge$ divides(s,x) and divides(s,y)) $\wedge$
  - if x,y < C then Runtime(gcd,(x,y)) < RC
- E.g., C = 10^{10}, RC = 10 milliseconds.

Gamification of Software Development for Computational Problems

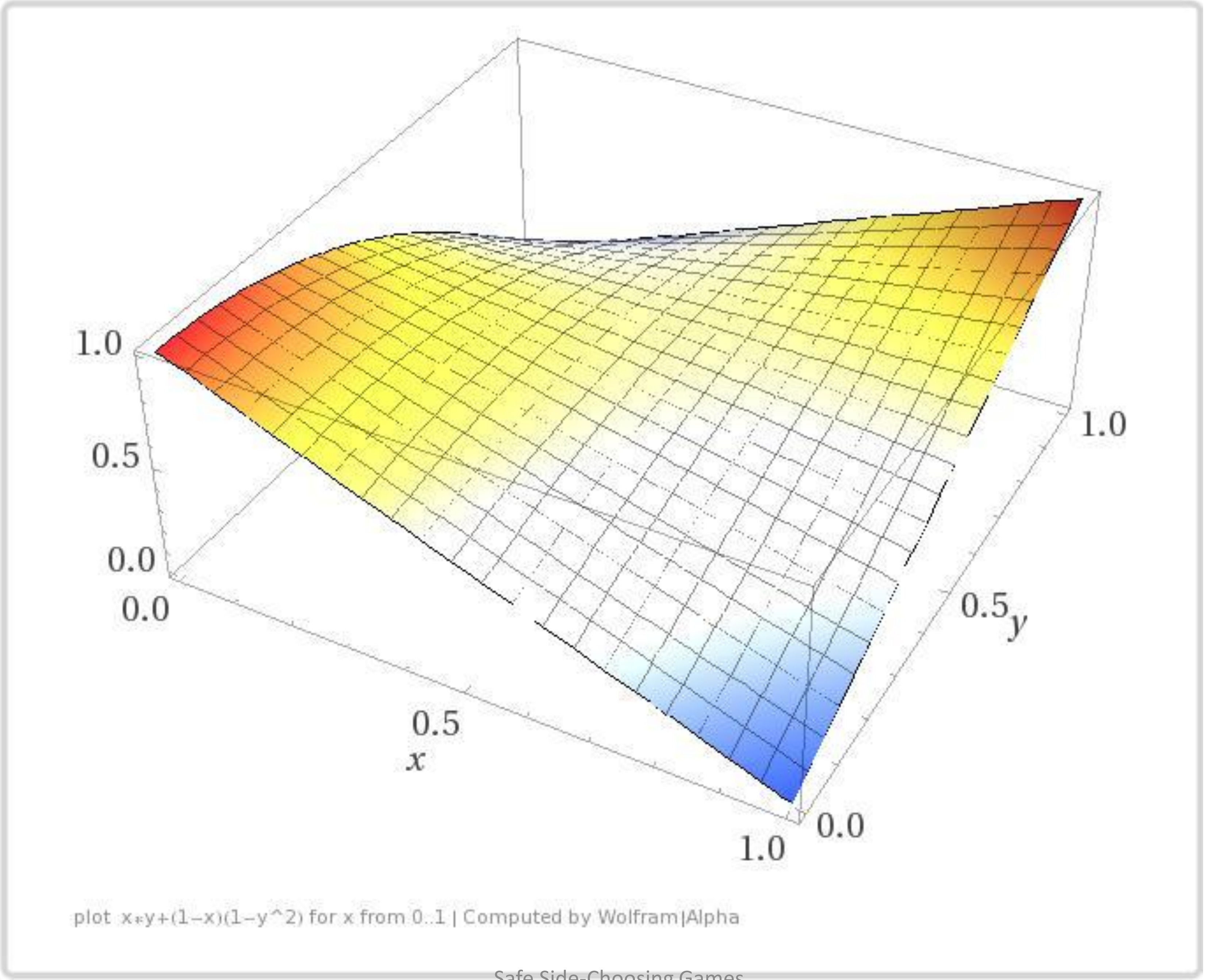
- Want reliable software to solve a computational problem? Design an SCG lab where the winning team will create the software you want.

Formal Science Claims: Saddle Point / Silver Ratio

claim

G(c) = ForAll x in [0,1] Exists y in [0,1]: x*y + (1-x)*(1-y^2) >= c

- Strategy chosen depends on c.
    - G(0.5)
    - G(0.615)
    - G(0.616)



plot x*y+(1−x)(1−y^2) for x from 0..1 | Computed by Wolfram|Alpha

Claim⟨ϕ, ***A***⟩

- What is common to
  - Chess Puzzle claim
  - Software Development claim
  - Formal Science claim
    - Claim⟨ϕ, ***A***⟩
      - ϕ is a well-formed formula.
      - ***A*** is a structure, often consisting of several substructures. Think of ***A*** as a collection of data types that are needed to define the claim.
        - ϕ refers to the functions defined in those data types.

Outline for the Talk

- Theory: Develop ranking theory for side-choosing games to find the most meritorious player.
  - Novelties:
    - Side-Choosing games.
    - Meritocracy Management for Side-Choosing Games: Ranking functions for side-choosing games. Map game results to a ranking of players.
  - Axiomatic approach:
    - Formulate desirable axioms for ranking functions.
    - Find representation theorem for ranking functions satisfying axioms.
- Applications/Results:
  - Lower barrier of entry for competition designers and participants
    - Simplify work of administrator
    - Make participation fun (collaborative, learning component)
  - Organize communities for experience-based learning)
  - Software Development

Outline Theory

- Why Side-Choosing Games: Benefits
- Side-ChoosingGame = Side-Choice x GeneralSemanticGame
  - GeneralSemanticGame: details of protocol not important
    - Game outcome must satisfy certain rules
  - Winning strategies
- Examples of families of semantic games
  - Logics
    - concrete example: Integer inequality.
  - Positions in explicit-form games
    - concrete example: Chess: mate in two.
- Meritocracy management
  - Tables with base and derived fields
  - Important table: SCG-Table: (W,L,F)
  - How to get to the SCG-Table?
  - Axioms for Ranking and Representation Theorem: Surprise

benefits: Objective, Low Overhead, Correct, Targeted Feedback, Well-Defined Interfaces.

Benefits of Side-Choosing Games

- Objective: The result depends on how well the participants solve the computational problems coming from the claim and protocol.
- Low Overhead on Administrator: Prepare claim and protocol and check that the protocol is followed during debates.
- Correct: The winners demonstrate their opponent's lack of skills for current claim.
- Targeted Feedback: protocol gives losers specific feedback.
- Participants interact through well-defined interfaces. Choosing side and following protocol.

What is a Side-Choosing Game?

- Claim C: precisely formulated
  - Truth value not known
  - Side-Choice (P (Proponent) or O (Opponent))

Sx (Side(x))	S!x (Side(!x))
P	O
O	P
P	P
O	O

- Semantic Game
  - Our notion of Semantic Game is more general than the traditional notion.

P (Proponent)	Winner
x	x
x	!x
!x	x
!x	!x

One P against one O!

General Semantic Game Definition

- From: “Semantic Games in Logic and Epistemology” by A-V Pietarinen (section 3):
  - You and I confront one another, observing a set of rules telling us which moves are legal.
  - We both try to win the game by winning any play of it, and if one of us finds a systematic way of doing so, he or she has a winning strategy.
  - The set of game rules is fixed by the logically active components in language. In the case of first-order languages the logically active components comprise the existential and universal quantifiers.

Important Property of Semantic Games

- Claim is
  - true
    - Proponent has a winning strategy
  - false
    - Opponent has a winning strategy
  - Truth value is UNKNOWN!
- If Proponent or Opponent loses: did not have the skill to find the winning strategy.
  - A correct statement independent of whether claim is true or false.
  - Loser demonstrates lack of skill: got into a contradiction.

Creating Semantic Games

- Sentences in various logics
    - Propositional
    - First-order
    - Independence-Friendly
- Positions in 2-person extensive-form games with perfect information. Choose a node (position) and ask: is it winning? Example: Mate in 2.

Examples of Semantic Games

- First from logic.
- Second from game positions.

Semantic Games (SGs) for interpreted formulas

- A semantic game for a given claim⟨ϕ, A⟩ is a game played by a Proponent and an Opponent, denoted SG(⟨ϕ, A⟩, Proponent, Opponent), such that:
  - A |= ϕ <=> the verifier has a winning strategy for ϕ, given structure A.

Toy Example: SCG Trace

✓ Proponent
✗ Opponent

SG($\forall x \in [0,1]: \exists y \in [0,1]: x + y > 1.5$, ✓, ✗)

✗ Provides 1 for x | Weakening (too much!)

SG($\exists y \in [0,1]: 1 + y > 1.5$, ✓, ✗)

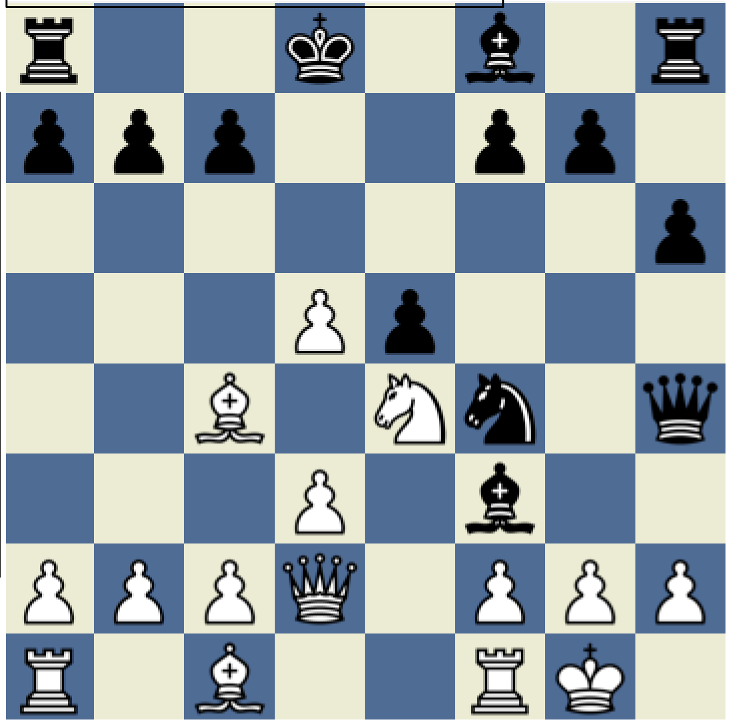
✓ Provides 1 for y | Strengthening

SG($1 + 1 > 1.5$, ✓, ✗) ✓ Wins

SCG = Side-Choosing Game
SCG = Scientific Community Game
SCG = Specker Challenge Game
SCG = SemantiC Game
all 4 make are meaningful

Semantic Games from Game Positions

Black mates in 2 moves



It is about a claim C.
Structure: ChessBoard.
Logical Sentence:
For given ChessBoard-instance cb:
Exists move b1 for Black
ForAll moves w1(b1) of White
Exists move b2(b1,w1) for Black:
WhiteKingIsMate(b1,w1,b2)

We ask 2 players x and !x:
x is a **Proponent**
!x is an **Opponent**

Approach: Table Overloading

- SCG Cases
  - Possible rows that are the result of binary SCGs
- SCG Tournament Results
  - Tables of tournament results: each row describes one binary SCG
  - concrete SCG-Table -> abstract SCG-Table
- Use same representation for cases and games
  - All 12 possible SCG cases can be viewed as the result of a tournament between two players involving 12 games.
  - Therefore we use the same representation for cases and tournament results: SCG-Tables

What is a Side-Choosing Game?

- A game that produces an SCG-Row in an SCG-table. The game is about a claim C involving a set of players Players. First move is simultaneous: choose a side: Proponent (P) or Opponent (O).
- What is a row in an SCG-table?
  - Presents one game result between two distinct participants p and q in Players.
  - Columns are: **W,L,F** (for **W**inner, **L**oser, **F**orced) (SCG-Table Rule 0).
  - **W**,**L** contain either p or q. **W≠L**. There are no ties (SCG-Table Rule 1).
  - **F** contains “none” or **W** or **L** (SCG-Table Rule 2).
    - A participant is Forced if it has to take the opposite side than it has chosen. Synonym: Devil’s Advocate.

What is a Side-Choosing Game? (continued)

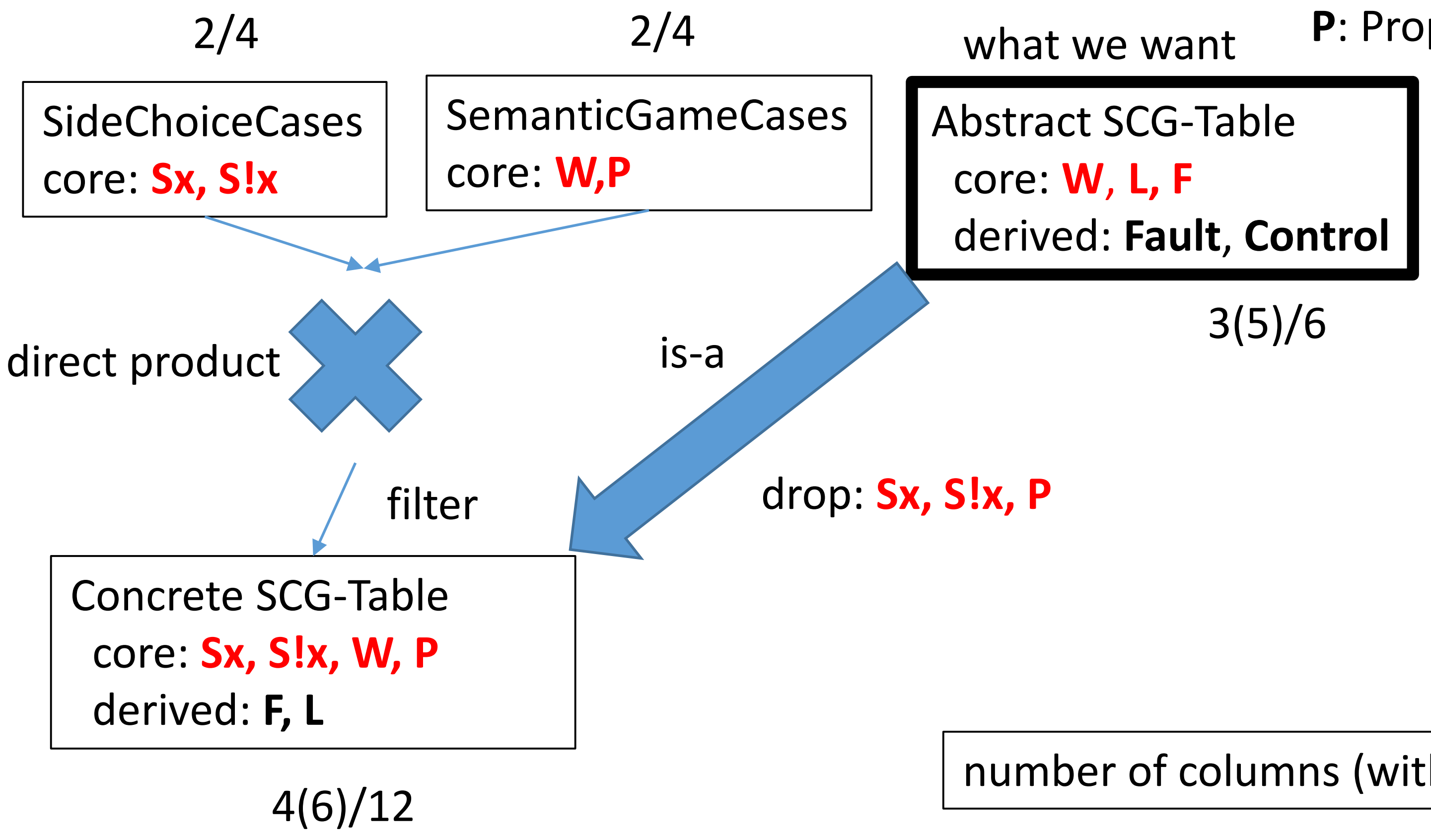
- What is an SCG-Table?
  - A table of SCG rows satisfying rules 0-2.
  - Multiple rows may involve the same two participants (SCG-Table Rule 3).
- To determine who wins the side war requires the execution of some protocol between the two participants. The SCG-Table definition does not specify the protocol: separation of concerns. The protocol language must guarantee: If a claim is true, it is possible to define a winning strategy in the protocol language. The protocol language of “standard” semantic games guarantees this.

How to get to the SCG-Table?

- From raw game results to an abstract representation important for ranking.
- Give sketch.

SCG-Tables

columns Concrete SCG-Table:
x: player
!x: other player
Sx: side of player x
S!x: side of player !x
P: Proponent



core columns abstract SCG-Table:
W: Winner
L: Loser
F: Forced

derived columns SCG-Table:
Fault: Loser is not forced
Control: Loser in control

number of columns (with derived)/number of rows

Problem we solve next: Meritocracy Finding

- Given a tournament of side-choosing games among a set of Players, how can we find the most meritorious players?
- Note: There is a lot of noise produced by the tournament:
    - We don’t know whether claim is true.
    - true claim might be refuted.
    - false claim might be defended.
    - players switch their sides between different games.
    - players may lie about their strength and lose intentionally to help a friend become more meritorious (collusion among players).
    - The more weak players or the more collusion, the more noise.
- HOW CAN WE FIND ORDER IN THIS COMPLEXITY?

Informal Reasoning

Case Abbreviation	Winner/Loser	Forced/Unforced
WF	Win	Forced
WU	Win	Unforced
LF	Lose	Forced
LU	Lose	Unforced

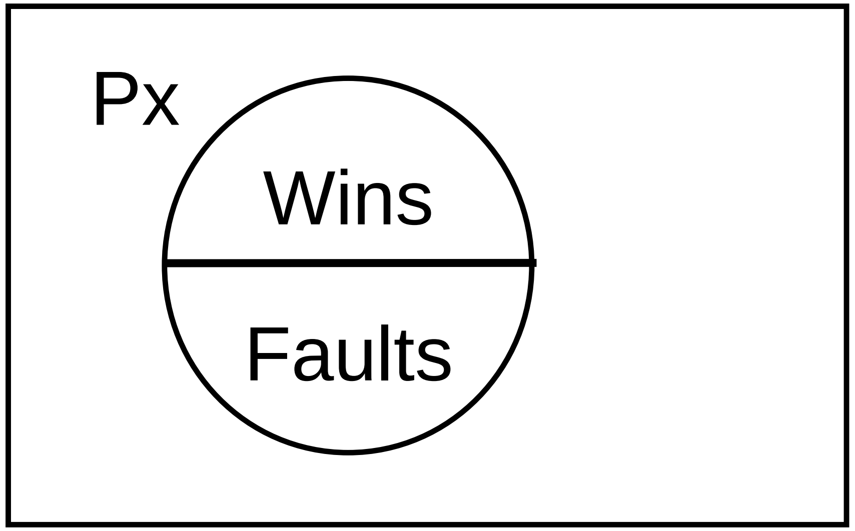
Which of the four statistics is a reliable indicator of strength or weakness?
WF: strength: no (because of collusion)
WU: strength: no (because of collusion)
LF: weakness: no
LU (Fault): weakness: YES. Player is contradictory!
  Loser is Proponent: should have won
  Loser is Opponent: should have prevented the other from winning.

Informal argument why counting Faults is interesting.

Fair Peer-Based Evaluation for n participants

- Builds on Two-Participant Evaluation
- Ranking systems for side-choosing games
- Axiomatic treatment: collusion-resistant
  - Introduce 3 axioms

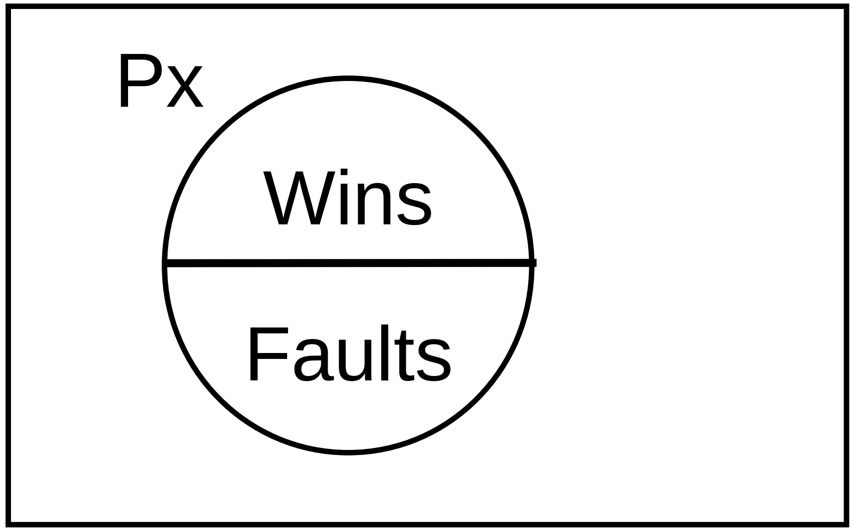
Non-Negative Effect For Wins (Axiom 1: NNEW)



Additional wins cannot worsen Px's rank w.r.t. other participants.

undisputed: Wins don't lower rank.

Non-Positive Effect For Losses (Axiom 2: NPEL)



Implies:

Additional losses cannot improve Px's rank w.r.t. other participants.

undisputed: Losses don't increase rank.

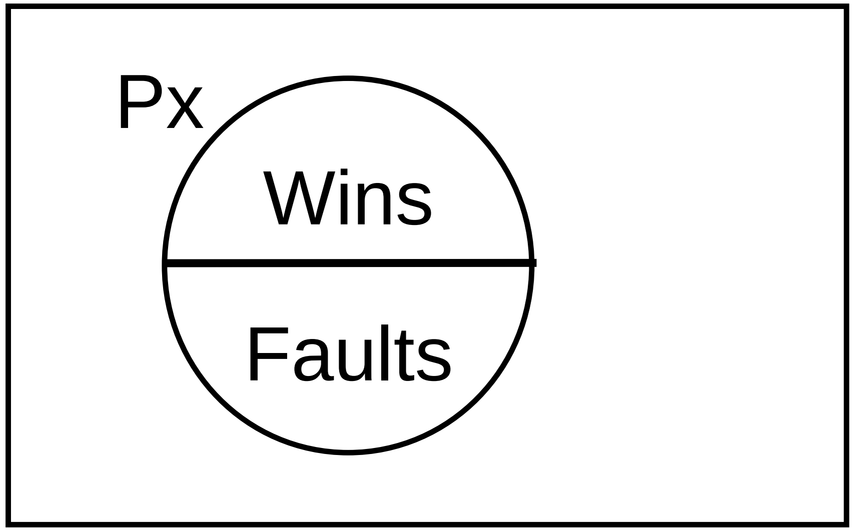
Ranking Functions (Anonymity)

- Output ranking is independent of participants’ identities.
- Ranking function ignores participants’ identities.
- Participants also ignore their opponents’ identities.

Collusion-Resistance

- Slightly weaker notion than anonymity.
- What you want in practice.
- A participant Py can choose to lose on purpose against another participant Px, but that won’t make Px get ahead of any other participant Pz.

Collusion-Resistance (Axiom 3: CR)



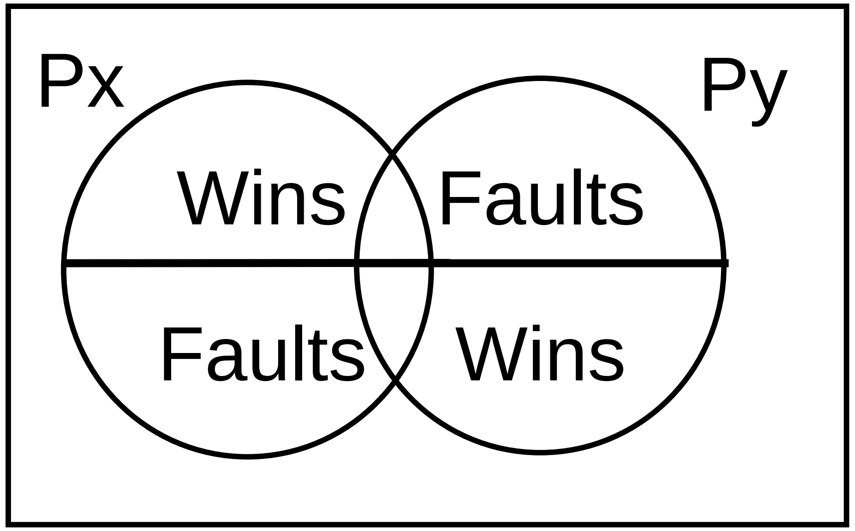
Games outside Px's control cannot worsen Px's rank w.r.t. other participants.

Px is in control if (Px in W(inner)) or (not(x in F(orced)))

Collusion-Resistant Axiom: Explanation

- When additional rows are added to T
  - where you did not participate, your rank cannot come down
  - where you won, your rank cannot come down.
  - where you were forced, your rank cannot come down.

Preparing for the Discovery/Surprise: Locally Fault Based (LFB)



Relative rank of Px and Py depends only on faults made by either Px or Py.

“Locally” is used to exclude faults made by some third player z when ranking x versus y.

Discovery: Property of SCG-Tables

- A useful design principle for ranking functions.
- Under NNEW, NPEL : CR = LFB
- LFB is quite unusual.
- LFB lends itself to implementation.
- Not only are faults important; they are fundamental.

$$\boxed{\begin{array}{l} \text{NNEW} \land \text{CR} \Rightarrow \text{LFB} \\ \text{NPEL} \land \text{LFB} \Rightarrow \text{CR} \\ \text{NNEW} \land \text{NPEL} \Rightarrow (\text{CR} \Leftrightarrow \text{LFB}) \end{array}}$$

≥ monotonically non-decreasing
≤ monotonically non-increasing

WF(x), WF(y),LF(x),LF(y) shown in diagram
1∪3-WF(x)=WU(x); 4∪6-WF(y)=WU(y); 2∪4:LU(x); 3∪5:LU(y)

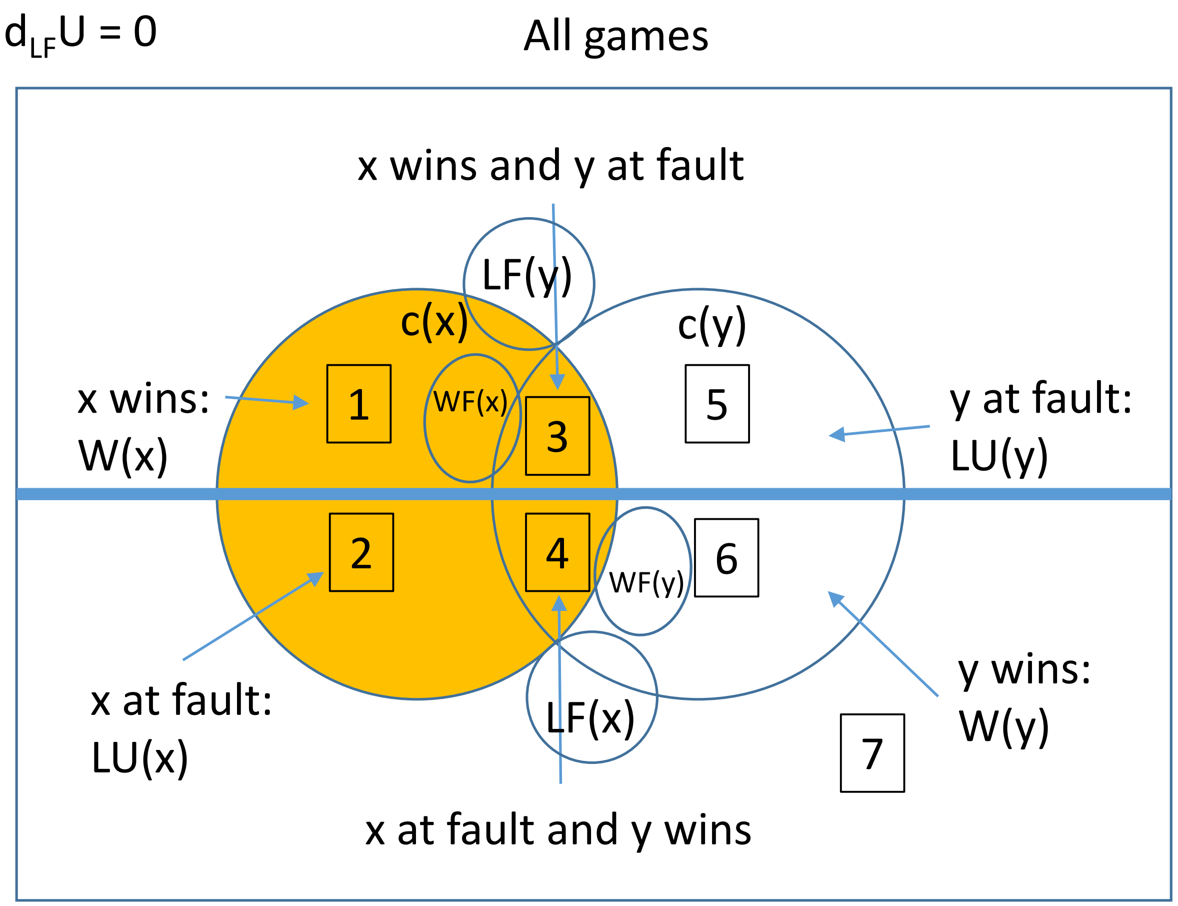
Proof
Venn Diagram for Game Kinds involving x,y

LFB: $d_W U = 0$, $d_{LF} U = 0$

games that cannot improve rank of x wrt. y:
6 by NNEW
1,7 by CR

games that cannot worsen rank of x wrt. y:
6,7 by CR
1 by NNEW

Proves:
NNEW and CR => LFB



2, 3, 4, 5: LFB

1 [!fl y] [w x] = [!c y] [w x]
2 [!w y] [fl x] = [!c y] [fl x]
3 [fl y] [w x] = [c y] [w x]
4 [w y] [fl x] = [c y] [fl x]
5 [!w x] [fl y] = [!c x] [fl y]
6 [!fl x] [w y] = [!c x] [w y]
7 [!c x] [!c y]

c(z) = z in control (full circle)
= (z wins) or (z not forced)

NNEW: $d_W U \leq 0$, NPEL: $d_L U \geq 0$, CR: $d_W U \geq 0$, $d_{LF} U = 0$, $d_{LU} U \geq 0$

Ranking Axioms Imply

$\geq$ monotonically non-decreasing
$\leq$ monotonically non-increasing

Monotonicity Constraints

- NNEW: $d_{WF}U \leq 0 \wedge d_{WU}U \leq 0$
- NPEL: $d_{LF}U \geq 0 \wedge d_{LU}U \geq 0$
- CR: $d_{WF}U \geq 0 \wedge d_{WU}U \geq 0 \wedge d_{LF}U = 0 \wedge d_{LU}U \geq 0$
- LFB: $d_{WF}U = 0 \wedge d_{WU}U = 0 \wedge d_{LF}U = 0$

Above implies:
NNEW $\wedge$ CR => LFB
NPEL $\wedge$ LFB => CR
NNEW $\wedge$ NPEL => (CR $\Leftrightarrow$ LFB)

Fault-Counting Scoring Function

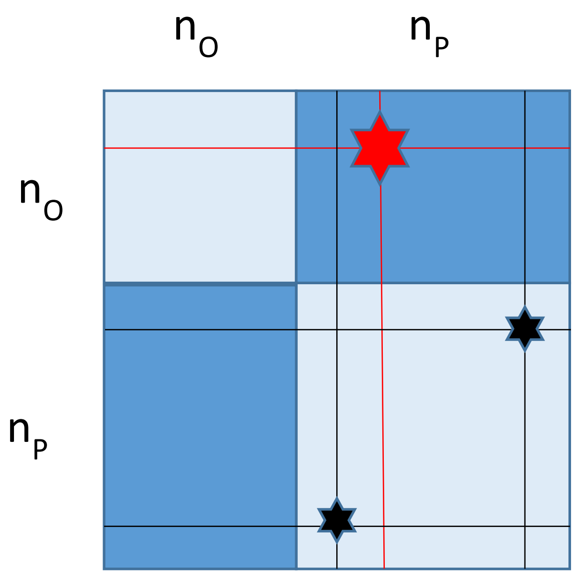
- Provide a concrete example of ranking function which is NNEW, NPEL and LFB and therefore CR.
  - Players are ranked according to their score: the number of faults they make. The fewer the number of faults the higher the rank.
- Satisfies the NNEW and NPEL
  - Clearly, Fault Counting is LFB and therefore CR (by previous theorem).
- Note: There is an infinite family of ranking functions that are LFB.

Semantic Game Tournament Design

- Full Round-Robin with adaptation: For every pair of players:
  - If choosing different sides, play a single SG.
  - If choosing same sides, play two SGs where they switch sides.

Neutrality

- Each player plays $n_P + n_O - 1$ SGs in their chosen side, those are the only games in which it may make faults.
- P = Proponent, O = Opponent



▨ 2 games; one player in chosen side

🟦 1 game; both players in chosen side

Summary Theory

- Side-choosing games require a side-choice at the beginning and then resort to forcing if both choose same side. A side-choosing game finishes with a generalized semantic game.
- Forcing is needed because generalized semantic games require a Proponent and an Opponent in each binary game.
- The axiomatic ranking theory producing the representation theorem: NNEW $\wedge$ NPEL => (CR $\Leftrightarrow$ LFB), only depends of SCG-Tables satisfying rules 0-3 and not on the details of semantic games. The 4 rules provide an abstraction barrier from the details of protocols.
- Simple Result: Full-Round Robin with Fault-Counting.

Solving Problems using the Crowd
USE SAFE SIDE-CHOOSING GAMES!

They look at the administrator who only acts as **referee**. We use the crowd for **fair peer evaluation**!

They want to be heard.

Some are better than others: **meritocracy**.

Some want to game the system by colluding/lying.

We want to extract a **provably** reliable signal ("best", **collusion-resistant** solution) from the complex behavior of the crowd.

Lower barrier of entry for competition designers

- What can be done with SCGs?
  - LFB ranking mechanisms, e.g., fault-counting.
  - Shifting evaluation from administrator to players!
  - Tournaments that fairly evaluate players!
    - collusion-resistance is guaranteed.

Methods

- Use piazza.com
  - Use JSON for scientific discourse (objects sent back and forth during semantic game).
  - Divide class into teams of 3.
- For strategies in software
  - We developed
    - a generator for baby strategies
    - administrator automated
- Also used simulation (synthetic strategies) to help invent the theory.

Applications of SCG (1)

- Teaching Software Development
  - Students get Baby-Strategy to start (automatically generated from claim).
  - Students add intelligence to their strategy to outperform the strategies of their peers.
  - A strategy consists of a function for side choice and a function for each quantifier (we use simplified semantic games).
  - Strategy with fewest losses wins. Strategies use web for fight!
  - Weekly tournaments with slightly modified claim: encourage well-modularized software that is easy to change.

Applications of SCG (2)

- Bring order to literature on solving computational problems.
  - Authors are required to provide a strategy to defend the claim.
  - Knowledge becomes active on the web.
  - A newcomer might beat all current strategies. Easily verified by a tournament.
  - Develop a Wikipedia for computational problems.

Applications of SCG (3)

- Teaching Formal Sciences (e.g., Algorithms).
  - Use a platform like Piazza to execute the protocol.
  - Avoid claims where the semantic game gives away the solution.
  - Divide class into groups of size 3. Balance skills in each group. Play tournaments within groups to prevent information overload for class.
  - Students came up with solutions that were about 10 years behind the state-of-the-art.

Results

- SCG usage for teaching using forum
  - Innovation Success with Undergraduates using SCG on piazza.com: Qualitative Data Sources & Analysis
  - Perfect for creating interaction between students (peer teaching)
- Strategy competitions are useful for teaching (and good for competitive innovation).
  - Some students stayed up to see their strategy succeed in the tournament.

Related Work

- Erlanger Konstruktivismus
  - Paul Lorenzen, Dialogische Logik
  - Konstruktive Wissenschaftstheorie, Suhrkamp, 1974.
- Rating and Ranking Functions
- Tournament Scheduling
- Match-Level Neutrality

Rating and Ranking Functions (I)

- Dominated by heuristic approaches
  - Elo ratings.
  - Who’s #1?
- There are axiomatizations of rating functions in the field of Paired Comparison Analysis.
  - CR not on radar

Rating and Ranking Functions (II)

- Rubinstein[1980]:
  - points system (winner gets a point) characterized as:
    - Anonymity : ranks are independent of the names of participants.
    - Positive responsiveness to the winning relation which means that changing the results of a participant p from a loss to a win, guarantees that p’s rank would improve.
    - IIM: relative ranking of two participants is independent of matches in which neither is involved.
  - “beating functions” are restricted to complete, asymmetric relations.

Tournament Scheduling

- Neutrality is off radar.
  - Maximizing winning chances for certain players.
  - Delayed confrontation.

Match-Level Neutrality

- Dominated by heuristic approaches
  - Compensation points.
  - Pie rule.

Conclusions

- Semantic games (and their generalization: side-choosing games) of interpreted logic sentences provide a useful foundation to organize computational problem solving and formal science communities for research and experience-based learning.
- We found a solution to the problem of lowering the barrier of entry for competition organizers by shifting evaluation tasks from the administrator to the players.
- We show the fundamental nature of locally fault-based evaluation in the presence of collusion-resistance.
- Simple result: 3 axioms, representation theorem, adapted full round-robin tournament with fault-counting gives fair, collusion-resistant evaluation of SCGs.

CCIS

- 45 faculty + 9 researchers
  - 7 new faculty in 2012
  - 6 new faculty in 2013
- 634 undergrads, 606 MS, 97 PhDs
- $7.6 million in grant funding in 2012
- PhD Programs in:
  - Computer Science
  - Information Assurance
  - Health Informatics
  - Network Science

Research at CCIS

- Security
  - 19+ papers at *NDSS*, *CSS*, *S&P*, and *Usenix Security* since '08
  - Funding from DARPA, Symantec, and Verisign
- Programming Languages
  - 19 papers in *POPL*, *OOPSLA*, and *ICFP* since '08
  - Active in the ECMA Javascript standards body
- DB, IR, and ML
  - Papers in *SIGIR*, *CIKM*, *KDD*, *SIGMOD*, *VLDB*, *ICDM*
  - New research center for digital humanities
- Formal Methods
  - 10 papers in *FMSD*, FMCAD, *TOPLAS*, and *CAV* since '08
  - NSF CAREER for building dependable concurrent software
- Robotics and Computer Vision
  - NSF CAREER for building robots that handle uncertainty
  - New hire for 2013: Rob Platt
- Network Science
  - Founders of the field: Laszlo Barabasi and Alex Vespignani
  - 20+ papers in *Nature*, *Science*, and *PNAS* since '08

Questions?

Karl Lieberherr
lieber@ccs.neu.edu

Thank You!