Theory of Side-Choosing Games to Create and Disseminate Knowledge in Formal Sciences

AHMED ABDELMEGED, Northeastern University
RUIYANG XU, Northeastern University
RAGHAVENDRA GALI, Northeastern University
KARL LIEBERHERR, Northeastern University

We solve the problem of collusion in massive online tournaments (of formal scientific communities) which, surprisingly, has a simple yet rather counterintuitive solution. The solution requires that we collect the participants' side-choices of whether they want to be proponent or opponent of a given claim. In the presence of side-choice, we have succeeded in comprehensively analyzing the design space for a broad class of incentive-compatible mechanisms for side-choosing games and their tournaments. We model the ranking of participants using three axioms NNEW (Non-Negative Effect for Winning), NPEL (Non-Positive Effect for Losing) and the crucial axiom called collusion-resistance (CR). We prove that any ranking function satisfying the three axioms must be based on a special kind of loss, called a fault. We are used to building meritocracy based on winning, but in the world of side-choosing games, surprisingly, the standard approach does not work in general for any tournament but there is a simple alternative and we have two theorems to prove it: The Representation Theorem and the Meritocracy Theorem. Version 2/5/2015.

ACM Reference Format:
Ahmed Abdelmeged, Ruiyang Xu, Raghavendra Gali, Karl Lieberherr, 2015. Theory of Side-Choosing Games to Create and Disseminate Knowledge in Formal Sciences. *ACM* X, X, Article X (February 2015), 20 pages.
DOI: http://dx.doi.org/10.1145/0000000.0000000

1. INTRODUCTION

How do we create knowledge? We use the language of side-choosing games to instruct humans about which knowledge needs to be created, i.e., which problem needs solving. The language of side-choosing games becomes the domain-specific language for the human computation system for formal sciences.

What is unique about our approach to create knowledge? To support a claim, a game is played. Playing a game only involves cleverly assigning values to variables and not knowledge about a formal proof system to support the claim. The side-choosing games support a low key approach to defending a claim and are suitable for many skill levels.

What is our theory: We study how to map side-choosing game results into ranking relations between players with guaranteed and desirable properties. We borrow ideas from social choice theory instead of using heuristics. Our theory informs the design of socio-technical systems for problem solving and teaching.

Our main contributions are (1) solving the collusion problem (2) the concept of a side-choosing game with an illustration of its usefulness, (3) a representation theorem for an axiomatic treatment of ranking relations for side-choosing games, and (4) a

---

Author's addresses: A. Abdelmeged, R. Xu, R. Gali and K. Lieberherr, College of Computer and Information Science, Northeastern University; email: lieber@ccs.neu.edu

DOI: http://dx.doi.org/10.1145/0000000.0000000

meritocracy theorem showing that our mechanism is incentive-compatible, i.e., quasi-perfect players will be top-ranked iff the critical property of collusion-resistance holds.

In this theory paper we discuss and analyze the design space for a socio-technical system called SCG (for Side-Choosing Game or Scientific Community Game) which manages the creation and dissemination of formal knowledge. Our motivations for designing such a system are both economical benefit (social welfare created in the form of practical solutions by goal-oriented players who want to win) and a positive experience for the participating players.

Side-choosing games (SCGs) have the following advantages: (1) distributed evaluation of players without a central authority. The players receive targeted, objective feedback from their peers. SCGs provide an educational system for checking the mastery of skills. There is low overhead for the "administrator" who organizes a competition for a given claim. (2) collusion resistance: It is impossible for players to collude and make a skilled or even perfect player lose without justification. (3) The overhead of learning SCGs is amortized over a large number of applications. (4) Low entry barrier: Playing the dialog game involves cleverly assigning values to variables. It does not require knowledge about a proof system and the skill of providing formal proofs. Clever ideas are demonstrated by practice: systematically defending the chosen side. (5) They are objective: the result depends on how well the participants solve the computational problems underlying the claim.

There are already SCG-like systems in place, e.g., TopCoder, Kaggle, SAT competitions or conferences and journals about formal sciences.

A side-choosing game is about a claim C which has an associated two-player, win-lose game $Game(C)$ between a proponent and an opponent of C such that a win of the proponent is an indication that C is true and a win of the opponent is an indication that C is false. $Game(C)$ does not allow for draws. If a proponent of C consistently wins against opponents, the proponent is said to have a winning strategy for C.

The first move in a side-choosing game is to choose a side for C: proponent or opponent. Player $x(y)$ chooses side $d(x)(d(y))$ (d = design-time). The side-choosing move may be simultaneous or sequential. For simplicity, we assume it is simultaneous. The side-choosing game includes an agreement algorithm that maps $x, y, d(x), d(y)$ into a set of games to be played where the design-time choices have been mapped into run-time choices $r(x), r(y)$ so that $r(x) \neq r(y)$. This requires that at most one of the players will be forced because it might be that $d(x) = d(y)$. A forced player is also called a devil's advocate especially when we think of a side-choosing game as a model for debates.

What is important to our theory is that for a claim C we get a table of game results $(winner = x, loser = y, d(x), d(y), r(x), r(y))$ for the games played between x and y. We simplify such a row to $(winner = x, loser = y, forced = z)$, where z is either x, y or 0. We assume that the Players are $1, 2, 3, ...$ If $z = 0$ none of the players is forced. The reason why this simplification works is because the only parameters of the game results that the ranking depends on is who the winner and loser is in a game and if the players' design and run time side choices differ. It does not depend on the exact choice of sides.

1.1. Motivating Side-Choosing Games (SCGs)

SCG is a new concept that we introduce in this paper. Why is the concept important and worthy of study? Many real-life situations can be modeled with side-choosing games. Studying how to fairly evaluate tournaments of side-choosing games is very important because collusion between the players could distort the outcome. Rather than looking for collusion detection algorithms we design our system such that a dangerous kind of collusion is impossible. We use an axiomatic approach to ranking the players and postulate collusion-resistance as an axiom.

We motivate the importance of SCGs by describing applications, users and owners.

Table I. A Table T of game results

GRID	Winner	Loser	Forced
gr1	1	2	1
gr2	1	2	1

- **SCG Applications.**
  - **Formal Sciences** Formal sciences are disciplines concerned with formal systems, such as logic, mathematics, statistics, theoretical computer science, information theory, game theory, systems theory, and decision theory. A claim is defined using an *interpreted* predicate logic sentence. This is *not* an exercise in logic as the quantifiers are only used to define the tasks that the users must perform. The sentence is interpreted in a structure which might be defined by a complex program encoding the functionality best executed by computers.
  - **Formal Claims based on Simulation Environments** Robotics and biological sciences, etc. fall into this category. The structure in which the claim is interpreted is the simulation environment.
- **SCG Users.** Users are problem solvers or learners and they operate directly or indirectly. In **direct mode**, the users perform the moves themselves, maybe using software. In **indirect mode**, the users produce software that plays the SCG on their behalf. There is a simple SCG-interface that the software has to follow. Of course, indirect users must have software development skills.
  The indirect mode is of central interest to us because it is a novel approach to develop software for computational problems using a group of people. The quality control of the software is automated by running an online or offline tournament to determine the top-ranked software. The claim under consideration determines what quality means. Note that the SCG-interface implies that testing is an integral part of the solution.
  Users of SCGs include:
  - **Students in high schools and universities.** They must understand the concept of a claim. Focus is on dissemination of knowledge through peer teaching and peer evaluation.
  - **Researchers.** Focus is on creation of new knowledge and its peer evaluation. Researchers propose claim variations.
  - **Citizen Scientists.** They might find innovative constructions that are imperfect. Experts might benefit from those ideas and correct them.
- **SCG Owners.**
  Owners define claims. Some users also play the role of owners. Owners don't need expertise how to solve the problems.
  Owners include: (1) Teachers and Professors. (2) Research Directors, Heads of Research Programs, Organizations like NSF, DARPA, ONR etc. (3) Program Chairs of conferences and Journal Editors. (4) Companies who need a specific computational problem solved for which no off-the-shelve solution is available. (5) Companies who are looking for employees with skills in a specific domain. E.g., Facebook organized a competition on kaggle.com and the winner got a Facebook job.

1.2. Players and Game Results

Let P be the set of all the players involved in the competition. Each game result has three columns corresponding to the winner, loser and the player forced to choose a side, if any. To represent a table T of game results we use a unique identifier for each row, called $GRID$ (Game Result ID). This guarantees that T will have no duplicates. Table I represents two game results where 1 played against 2 and where 1 won although it was forced. The game result tables can grow to any size as we allow the same players to play again. Of course, the game history for gr1 and gr2 might be very different. $GR(P)$ is the set of the all possible game results without the unique identifier $GRID$. $GR(P)$ contains $n \cdot (n-1) \cdot 3$ rows where n is the number of players in P. $G(P)$ is the set of all possible game results with unique identifiers.

1.3. Examples of Side-Choosing Games

The definition of a side-choosing game given above is abstract and only useful if we can give several interesting concrete versions.

— **Combinatorial Games** We choose a combinatorial game [Demaine 2001] and a position pos. The claim is: the position pos is a winning position.
— **Semantic Games** We choose a logic which supports semantic games. I.e., each sentence in the logic is mapped to a game between proponent and opponent so that the claim is true iff the proponent has a winning strategy. The sentences are interpreted in some structure. Most logics have semantic games. Some prominent examples are first and higher-order logics and independence-friendly logic [Tulenheimo 2013].
Semantic games are a huge application domain for side-choosing games and we arrived at side-choosing games through the study of semantic games, also called "outdoor" games by Hintikka because of their simplicity [Kulas and Hintikka 1983].
The connection between proofs and winning strategies is an active topic in logic [Boyer and Sandu 2012]. One of the attractions of side-choosing games is that you don't need a proof for a claim to perform well in the side-choosing game for the claim. Side-choosing games are more easily accessible than formal proofs. We provide examples of semantic-game-based claims online[1].

1.4. Applications of Side-Choosing Games to Existing Systems

Our study of side-choosing games is motivated by their potential to organize problem-solving competitions and by their successful use in education at Northeastern University. We believe SCGs are a foundation for platforms like TopCoder or Kaggle or scientific human-computation tools like Fold-It [Cooper et al. 2010].

— **Education in Formal Sciences** Our favorite way of summarizing learning objectives for a formal science domain is to say that learners must demonstrate the skill of judging claims in the domain, choosing their side on the claim and then defending their side choice through game play against other students. The resulting peer-teaching and peer-grading is very attractive. A claim is representing a lab in which students learn and is chosen in such a way that solving the problem requires skills that students should have.
Using piazza.com To post claims and to organize the playing of games related to those claims we used piazza.com. This worked very well, especially when we divided the Algorithms class into small groups of three students and kept the games in those small groups. The undergraduate students solved challenging problems like finding the worst-case input for the Gale-Shapley algorithm or optimally solving a product stress testing problem.
Using our own software In software development classes we had the students develop "avatars" to play the game and we did a full-round-robin tournament evaluation of the avatars. The problem to be solved was a maximum constraint satisfaction problem.
— **Improving Evaluation in Problem-Solving Competitions for Computational Problems** A significant advantage of our approach is that the evaluation of solutions is done by peers and not the competition organizer. This is relevant to systems

[1] http://www.ccs.neu.edu/home/lieber/Claims/

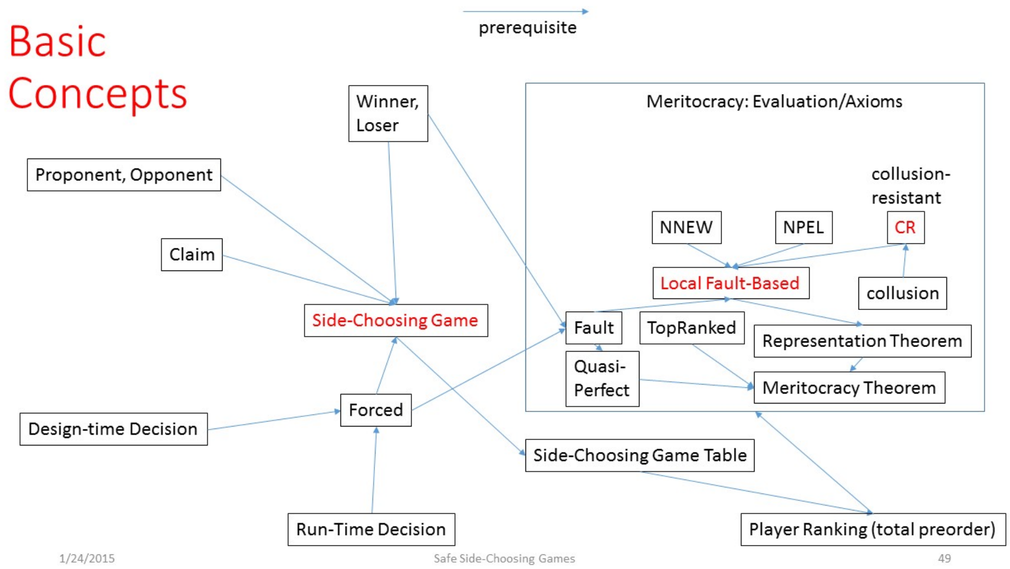

Fig. 1. Basic concepts

like topcoder.com and various competitions like SAT-solver competitions. The competition organizer only acts in a role as referee. Instead of static benchmarks, dynamic benchmarks are developed through game play.
The quality of the solutions produced depends on the skills of the participants who might not be motivated or not have the knowledge necessary to solve the problem. To attract strong participants either money or fame has to be given; a common theme in human computation.

1.5. Organization of the paper

In section 2 we start with a discussion of related work touching on philosophy of science, social choice theory, logic, and heuristic rating methods. In section 3 we describe our table model SCG-tables and related predicates and operations. Our table model is the foundation for our theory. The central concepts of ranking and collusion-resistance are introduced in section 4. Section 5 provides the link between ranking function properties and monotonicity constraints which are used to prove theorems. Our Representation Theorem is introduced and proved in section 6. Section 7 talks about ramifications of the representation theorem, including incentive compatibility and the shape of "good" scoring functions. We discuss how parameters of those scoring functions can be used to influence the behavior of the players. Section 8 rounds up the paper by introducing the concept of quasi-perfection which leads to quasi-equilibria[2]. Our Meritocracy Theorem connects quasi-perfection and being top-ranked.

The appendix contains the definition of side-choosing games and examples of claims. Figure 1 contains the list of important concepts and their prerequisite dependencies.

[2]Our notion of equilibrium is non-standard. In a quasi-equilibrium players have a strong incentive to change their behavior if they can push the others into faults!

2. RELATED WORK

Our concept of a side-choosing game is very broad but has not been formally studied before. We were influenced by semantic games which have a long history in logic. Falsifiability as promoted by Karl Popper and many others was another strong influence. A claim is falsifiable if there is an argument which proves the claim to be false. We use a weaker form of falsifiability which we call personalized testability. A claim C for which participant p_x is a proponent is personally testable by a participant p_y if there is an argument that brings p_x into a contradiction with respect to C. The argument is an interactive "debate" (p_y winning a game against p_x) but it does not prove that the claim is false, in general.

[Kulas and Hintikka 1983] defines an important class of binary side-choosing games called semantic games and relates it to the foundations of logic.

In [Rubinstein 1980], Rubinstein provides an axiomatic treatment of tournament ranking functions that bears some resemblance to ours. Rubinstein's treatment was developed in a primitive framework where "beating functions" are restricted to complete, asymmetric relations. Rubinstein showed that the points system, in which only the winner is rewarded with a single point is *completely* characterized by the following three *natural* axioms:

— anonymity which means that the 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 would have a better rank than all other participants that used to have the same rank as p, and
— Independence of Irrelevant Matches (IIM) which means that the relative ranking of two participants is independent of those matches in which neither is involved.

Our LFB axiom is, in some sense, at least as strong as Rubinstein's IIM because, according to LFB, the relative rank of some participant p_x w.r.t. another participant p_y cannot be worsened by games that p_x does not participate in nor can it be improved by games that p_y does not participate in.

[Boyer and Sandu 2012] discusses the connection between winning strategies for semantic games and proofs. A recursive winning strategy for a semantic game of a sentence is a constructive proof of that sentence. They introduce the notion of CGTS-truth (computable game-theoretical semantics truth): a sentence ϕ is CGTS-true on a recursive model M exactly when there is a computable winning strategy for verifier in the semantical game played with ϕ on M.

They focus on the special case of Peano Arithmetic (PA). They investigate the following questions

— From proofs to winning strategies Do proofs in PA yield CGTS-truth?
— From winning strategies to proofs Can the CGTS-truth of a sentence be interpreted as a proof?

Side-choosing games with backward moves are important in the study of those questions. The backward moves allow for many more winning strategies.

Rating methods can be used to rank tournament participants. There is a vast body of literature on the topic of *heuristic* [Beasley 2006] rating methods aiming to estimate the skill level of participants such as the Elo [Elo 1978] rating method. [Langville and Meyer 2012] gives a recent comprehensive overview of rating methods used in sports tournaments. Our work differs from this vast body of literature in two important aspects. First, our axioms and ranking method are the first to be developed for an extended framework that we developed specifically to capture some of the peculiarities of side-choosing game tournaments such as forcing. Second, our work is the *first* to be concerned with collusion resistance.

An early version of SCG, then called Scientific Community Game, was published in [Lieberherr et al. 2010].

[Simpson 2014] provides a comprehensive overview of techniques to "Combined Decision Making with Multiple Agents".Our work differs by working with multiple arguing or debating agents who have to defend their decisions. The concept collusion is not mentioned in [Simpson 2014] while it is central to our analysis.

This paper is based on Ahmed Abdelmeged's dissertation [Abdelmeged 2014]. The dissertation is based on semantic games and does not explicitly define side-choosing games. However, the proof of the representation theorem does not rely on a specific logic. Therefore, we introduced side-choosing games in this paper to have an appropriate context for formulating and proving the representation theorem. The proofs in this paper have been simplified through the systematic use of monotonicity constraints.

3. MODEL

During SCGs many confusing events can happen: true claims might be refuted and false claims might be defended; the same player might be proponent in one game and opponent in another game; a player might lie about her strength and intentionally lose a game. In all this noise we would like to find order and look for quasi-perfect players and quasi-equilibria. A quasi-perfect player might not be perfect and a quasi-equilibrium might not be an equilibrium but none of the players can demonstrate this fact through game play.

3.1. Basic Objects

Our theory is about mapping a set of game results T to ranking relations which reveal the strongest players. Game results are represented by tables with four columns which we call SCG-tables: $GRID, winner, loser, forced$. Each row is of the form $(winner = x, loser = y, forced = z)$, where z is either $x, y \in P$ or 0. We assume that the set of players P is $1, 2, 3, ...$ If none of the players is forced then $z = 0$. We assume that $x \neq y$. $GRID$ stands for game result identifier which is unique for each row.

3.2. Abstraction Barrier

The above rules that the game results must satisfy define an important abstraction barrier for our theory. The tables which satisfy the above rules are called SCG-tables. Our theory is about adding and deleting rows to the SCG-tables and how those rows influence the rankings of players. To keep our paper self-contained we focus on SCGs with perfect information but the same theory also applies to games with imperfect information.

The output of the mapping of an SCG-table is a total preorder of the players. A preorder is a binary relation that is reflexive and transitive. A preorder is total if no pair of players is incomparable.

3.3. Basic Predicates and Operations

We use the following terminology: There is a design-time decision for P(roponent)/O(pponent). This is the design time for the winning strategy to demonstrate that the start position is winning or not. The side choice P says that the start position is winning; O that it is not winning. There is a run-time decision for P(roponent)/O(pponent). This is the decision used when the game executes and might involve forcing at most one participant.

The side-choosing game $SCG(G, x, y, dx, dy, rx, ry)$ is a game between x and y, where x makes design-time choice dx and y makes design-time choice dy. The run-time choices are rx and ry. x is forced if $rx =!dx$ and $ry = dy$. y is forced if $rx = dx$ and $ry =!dy$. At most one player is forced. The claim is: in the start position of G the player in the P role ($d* = P$) has a winning strategy. dx, dy, rx, ry have values in $\{P, O\}$. z has values in $\{x, y\}$. $!z = x$ if $z = y$. $!z = y$ if $z = x$. $!P = O, !O = P$.

Next we define a few basic predicates.
$\forall p_x \in P, \forall r \in G(P)$

$$participant(p_x, r) = p_x \text{ is a participant in the game } r$$
$$win(p_x, r) = p_x \text{ won the game } r$$
$$loss(p_x, r) = p_x \text{ lost the game } r$$
$$forced(p_x, r) = p_x \text{ is forced to choose a side}$$
$$\neg control(p_x, r) \Leftrightarrow \neg participant(p_x, r) \lor (loss(p_x, r) \land forced(p_x, r))$$
$$fault(p_x, r) \Leftrightarrow loss(p_x, r) \land \neg forced(p_x, r)$$

$\forall p_x \in P, \forall T \subseteq G(P)$

$$wf^T(p_x) = \text{the win count of } p_x \text{ in } T \text{ in a forced position}$$
$$wu^T(p_x) = \text{the win count of } p_x \text{ in } T \text{ in an unforced position}$$
$$lf^T(p_x) = \text{the loss count of } p_x \text{ in } T \text{ in a forced position}$$
$$lu^T(p_x) = \text{the loss count of } p_x \text{ in } T \text{ in an unforced position}$$
$$np^T(p_x) = \text{the number of games in } T \text{ where } p_x \text{ was not a participant}$$

The concept of a forced player cuts across our theory and influences the definition of the concept of collusion-resistance.

4. RANKING

In this section, we discuss ranking the players based on an SCG-table T under the axiom of collusion-resistance. When collusion-resistance does not hold, there are SCG-tables T for which a meritorious player is not top-ranked. This will frustrate meritorious players and therefore we enforce the axiom of collusion-resistance which is based on the concept of control. A player is not in control in a game if she does not participate or she loses while forced. Note that if a player is forced we cannot blame her when she loses. Collusion-resistance is formalized by expressing that adding a row where player p_x is not in control, will keep the ranking of p_x with respect to other players p_y invariant. It turns out that collusion-resistance is linked to the concept of fault: a player makes a fault if she loses while not forced.

Our ranking approach prevents sybil attacks. In an online competition, several sybils might enter and help others to win thereby preventing the strong players to win. In the presence of collusion-resistance sybils have no effect on determining the top-ranked players.

4.1. Ranking Relation

We define a preorder $\preceq_U^T$ called the weakly better relation $\forall T \subseteq G$ based on the scoring function $U : \mathbb{N} \times \mathbb{N} \times \mathbb{N} \times \mathbb{N} \to \mathbb{R}$. For convenience, we drop the subscript and refer to it simply as $\preceq^T$.

We want to assign each player a score solely based on the players' demonstration of ability. We use 4 statistics, based on wins and losses and whether a player was forced, to calculate a players' score. We formally define the ranking relation as,

$$\forall p_x, p_y \in P, \forall T \subseteq G(P)[\, p_x \preceq^T p_y \Leftrightarrow U(wf^T(p_x), wu^T(p_x), wf^T(p_x), lu^T(p_x)) \leq U(wf^T(p_y), wu^T(p_y), wu^T(p_y), lu^T(p_y))\,] \quad (1)$$

We want the ranking relation to have the following properties defined in terms of table extensions:

— NNEW: Winning cannot lower your rank:

$$\forall p_x, p_y \in P, \forall T \subseteq G(P), \forall r \in \{r \,|\, r \in G(P) \setminus T \,\wedge\, win(p_x, r)\}[\, p_x \preceq^T p_y \Rightarrow p_x \preceq^{T \cup \{r\}} p_y\,]$$

— NPEL: Losing cannot increase your rank:

$$\forall p_x, p_y \in P, \forall T \subseteq G(P), \forall r \in \{r \,|\, r \in G(P) \setminus T \,\wedge\, loss(p_y, r)\}[\, p_x \preceq^T p_y \Rightarrow p_x \preceq^{T \cup \{r\}} p_y\,]$$

— CR: Games you don't control don't affect your rank:

$$\forall p_x, p_y \in P, \forall T \subseteq G(P), \forall r \in \{r \,|\, r \in G(P) \setminus T \,\wedge\, \neg control(p_x, r)\}[\, p_x \preceq^T p_y \Rightarrow p_x \preceq^{T \cup \{r\}} p_y\,]$$

4.2. Universal Domain

From equation 1, it is clear that for every logically possible game result table T, we have a valid preorder. This implies that our ranking relation satisfies the Universal Domain property.

4.3. Anonymity

From equation 1 it is clear that the scoring function ignores the identity of the player in calculating the score. Hence, the ranking relation $\preceq^T$ is unaffected by changing labels and therefore anonymous.

4.4. Monotonicity of U and Notation

As we score a player solely based on the player's wins and losses, NNEW and NPEL imply that the function U is monotonic. One interesting property of the parameters of U for a particular player is that when we add a new game to the existing game result table T, at most one parameter increments. This allows us to use a notation that mimics the partial differential operator which simplifies the proofs in the original dissertation [Abdelmeged 2014].

$$\partial_x U \geq 0 \,:\, U \text{ is monotonically non-decreasing on the parameter } x$$
$$\partial_x U > 0 \,:\, U \text{ is monotonically increasing on the parameter } x$$
$$\partial_x U \leq 0 \,:\, U \text{ is monotonically non-increasing on the parameter } x$$
$$\partial_x U < 0 \,:\, U \text{ is monotonically decreasing on the parameter } x$$
$$\partial_x U = 0 \,:\, U \text{ is indifferent on the parameter } x$$

5. PROPERTIES OF RANKING RELATIONS

In this section, we formulate properties in predicate logic and derive their equivalent monotonicity constraints.

5.1. Collusion Resistance (CR)

$$\text{CR: } \forall p_x, p_y \in P, \forall T \subseteq G(P), \forall r \in \{r \,|\, r \in G(P) \setminus T \,\wedge\, \neg control(p_x, r)\}[\, p_x \preceq^T p_y \Rightarrow p_x \preceq^{T \cup \{r\}} p_y\,]$$

Here we have 2 cases: game results where p_x did not participate and game results where p_x lost when forced. For the first case, p_y may have won or lost in a forced or unforced position against some third player p_z.
Let us consider the case where p_y wins over p_z in a forced position, we have,

$$U(wf^T(p_x), wu^T(p_x), lf^T(p_x), lu^T(p_x)) \leq U(wf^T(p_y)+1, wu^T(p_y), lf^T(p_y), lu^T(p_y)) \quad (2)$$

From equations 1 and 2, we get the monotonicity constraint,

$$\partial_{wf} U \geq 0 \quad (3)$$

Let us consider the case where p_y wins over p_z in an unforced position, we have,

$$U(wf^T(p_x), wu^T(p_x), lf^T(p_x), lu^T(p_x)) \leq U(wf^T(p_y), wu^T(p_y)+1, lf^T(p_y), lu^T(p_y)) \quad (4)$$

From equations 1 and 4, we get the monotonicity constraint,

$$\partial_{wu} U \geq 0 \quad (5)$$

Using a similar argument, for the case where p_y loses over p_z in a forced position, we have

$$\partial_{lf} U \geq 0 \quad (6)$$

For the case where p_y loses over p_z in an unforced position, we have

$$\partial_{lu} U \geq 0 \quad (7)$$

Now we consider game results where p_x was forced to lose against some third player p_z,

$$U(wf^T(p_x), wu^T(p_x), lf^T(p_x)+1, lu^T(p_x)) \leq U(wf^T(p_y), wu^T(p_y), lf^T(p_y), lu^T(p_y)) \quad (8)$$

From equations 1 and 8, we get the monotonicity constraint,

$$\partial_{lf} U \leq 0 \quad (9)$$

Now, CR can be summarized in terms of monotonicity constraints as,

$$\partial_{wf} U \geq 0 \wedge \partial_{wu} U \geq 0 \wedge \partial_{lf} U = 0 \wedge \partial_{lu} U \geq 0 \quad (10)$$

5.2. Non Negative Effect of Winning (NNEW)

NNEW: $\forall p_x, p_y \in P, \forall T \subseteq G(P), \forall r \in \{r \mid r \in G(P) \setminus T \wedge win(p_x, r)\}[\, p_x \preceq^T p_y \Rightarrow p_x \preceq^{T \cup \{r\}} p_y \,]$

Let us consider a game result r where p_x won against a third player p_z. p_x could have won either in a forced or unforced position.
Let us consider the case where p_x wins over p_z in a forced position, we have,

$$U(wf^T(p_x)+1, wu^T(p_x), lf^T(p_x), lu^T(p_x)) \leq U(wf^T(p_y), wu^T(p_y), lf^T(p_y), lu^T(p_y)) \quad (11)$$

From equations 1 and 11, we get the monotonicity constraint,

$$\partial_{wf} U \leq 0 \quad (12)$$

For the case where p_x wins over p_z in an unforced position, we have

$$\partial_{wu} U \leq 0 \quad (13)$$

Summarizing the monotonicity constraints, we have,

$$\partial_{wf} U \leq 0 \wedge \partial_{wu} U \leq 0 \quad (14)$$

5.3. Non Positive Effect of Losing (NPEL)

$$\text{NPEL: } \forall p_x, p_y \in P, \forall T \subseteq G(P), \forall r \in \{r \mid r \in G(P) \setminus T \wedge loss(p_y, r)\}[p_x \preceq^T p_y \Rightarrow p_x \preceq^{T\cup\{r\}} p_y]$$

Let us consider a game result r where p_y lost against a third player p_z. p_y could have lost either in a forced or unforced position.
Let us consider the case where p_y loses over p_z in a forced position, we have,

$$U(wf^T(p_x), wu^T(p_x), lf^T(p_x), lu^T(p_x)) \leq U(wf^T(p_y), wu^T(p_y), lf^T(p_y)+1, lu^T(p_y)) \quad (15)$$

From equations 1 and 15, we get the monotonicity constraint,

$$\partial_{lf} U \geq 0 \quad (16)$$

For the case where p_y loses over p_z in an unforced position, we have

$$\partial_{lu} U \geq 0 \quad (17)$$

Summarizing the monotonicity constraints, we have,

$$\partial_{lf} U \geq 0 \wedge \partial_{lu} U \geq 0 \quad (18)$$

5.4. Local Fault Based (LFB)

As we want the ranking relation to satisfy all the three properties NNEW, NPEL and CR, from equations 10, 14 and 18, we get the monotonicity constraints,

$$\partial_{wf} U = 0 \wedge \partial_{wu} U = 0 \wedge \partial_{lf} U = 0 \wedge \partial_{lu} U \geq 0 \quad (19)$$

This tells us that the scoring function should be monotonically non-decreasing on faults and indifferent on other parameters. We call the ranking relation that uses a scoring function that satisfies equation 19 as Local Fault Based (LFB). The monotonicity constraints in equation 19 can be easily reformulated in predicate logic.

$$\text{LFB: } \forall p_x, p_y \in P, \forall T \subseteq G(P), \forall r \in \{r \mid r \in G(P) \setminus T \wedge \neg fault(p_x, r) \wedge \neg fault(p_y, r)\}[p_x \preceq^T p_y \Leftrightarrow p_x \preceq^{T\cup\{r\}} p_y]$$

5.5. Independence of Irrelevant Games (IIG)

$$\text{IIG: } \forall p_x, p_y \in P, \forall T \subseteq G(P), \forall r \in \{r \mid r \in G(P) \setminus T \wedge \neg participant(p_x, x) \wedge \neg participant(p_y, x)\}[p_x \preceq^T p_y \Leftrightarrow p_x \preceq^{T\cup\{r\}} p_y]$$

This property just means that relative ranking does not depend on the games where neither p_x nor p_y has participated. From the definitions of CR, LFB and IIG, it is simple to deduce that,

$$\text{CR} \Rightarrow \text{IIG} \quad (20)$$

$$\text{LFB} \Rightarrow \text{IIG} \quad (21)$$

But, both converses are not true.

6. REPRESENTATION THEOREM

In mathematics, a representation theorem is a theorem that states that every structure with certain properties is isomorphic to another structure. We show that every ranking relation which is NNEW, NPEL and CR is LFB. Therefore there is a simple way to represent collusion-resistant ranking relations by doing a variant of fault counting. See [Abdelmeged 2014] for an alternate proof.

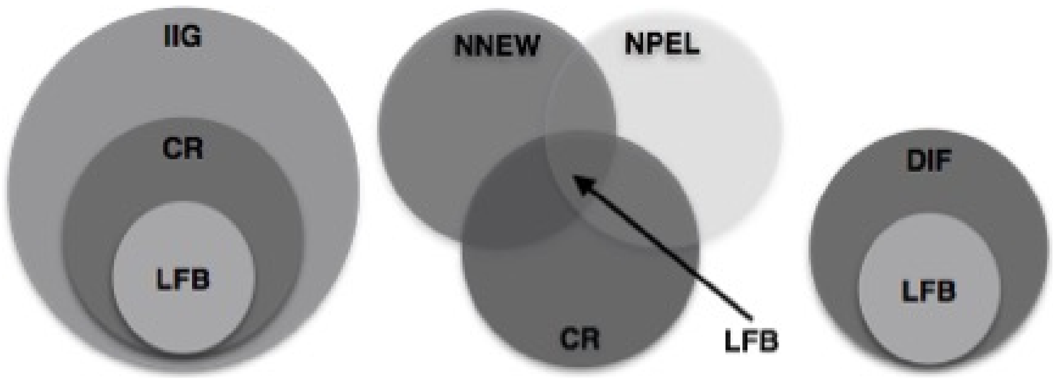

Fig. 2. Visual representation of relationships among the sets of ranking relations

6.1. Representation Theorem

$$\mathrm{NNEW} \wedge \mathrm{NPEL} \Rightarrow (\mathrm{CR} \Leftrightarrow \mathrm{LFB}) \tag{22}$$

Proof: It is easy to derive the following properties from the monotonicity constraints given above (constraints 10, 14, 18, and 19)

$$\mathrm{LFB} \Rightarrow \mathrm{NNEW}, \mathrm{LFB} \Rightarrow \mathrm{NPEL} \tag{23}$$

$$\mathrm{LFB} \Rightarrow \mathrm{CR}, \mathrm{NNEW} \wedge \mathrm{NPEL} \wedge \mathrm{CR} \Rightarrow \mathrm{LFB} \tag{24}$$

from which we deduce the Representation Theorem.

7. RAMIFICATIONS OF THE REPRESENTATION THEOREM

7.1. IIG and Representation theorem

We discuss if non participation as a parameter in the scoring function U would have any effect on the monotonicity constraints we have derived so far. For that, we have to derive the monotonicity constraint for IIG.
The number of games p_x has not participated in is defined as,

$$np^T(p_x) = |T| - wf^T(p_x) - wu^T(p_x) - lf^T(p_x) - lu^T(p_x) \tag{25}$$

From equations 1 and 25, it is clear that we can calculate $np^T(p_x)$ from $|T|$ and the existing 4 parameters of U. To account for non participation, we can redefine the ranking relation using $U : \mathbb{N} \times \mathbb{N} \times \mathbb{N} \times \mathbb{N} \times \mathbb{N} \to \mathbb{R}$,

$$\forall p_x, p_y \in P, T \subseteq G(P)[p_x \preceq^T p_y \Leftrightarrow U(wf^T(p_x), wu^T(p_x), wf^T(p_x), lu^T(p_x), |T|) \leq U(wf^T(p_y), wu^T(p_y), wu^T(p_y), lu^T(p_y), |T|)] \tag{26}$$

Now let us add a game result r where neither p_x nor p_y participate. From the definition of IIG we have,

$$U(wf^T(p_x), wu^T(p_x), lf^T(p_x), lu^T(p_x), |T|+1) \leq U(wf^T(p_y), wu^T(p_y), lf^T(p_y), lu^T(p_y), |T|+1) \tag{27}$$

For equations 26 and 27 to hold together, we need that equations 28 and 29 to hold. First, we focus on p_x and we need that,

$$U(wf^T(p_x), wu^T(p_x), lf^T(p_x), lu^T(p_x), |T|+1) \leq U(wf^T(p_x), wu^T(p_x), lf^T(p_x), lu^T(p_x), |T|) \tag{28}$$

Now, we focus on p_y and we need that,

$$U(wf^T(p_y), wu^T(p_y), lf^T(p_y), lu^T(p_y), |T| + 1) \geq \\ U(wf^T(p_y), wu^T(p_y), lf^T(p_y), lu^T(p_y), |T|) \quad (29)$$

This gives us the monotonicity constraints,

$$\partial_{|T|} U \leq 0 \wedge \partial_{|T|} U \geq 0 \quad (30)$$

This is the same as,

$$\partial_{|T|} U = 0 \quad (31)$$

From equation 31, it is clear that the IIG is the same as the scoring function U being indifferent to $|T|$.
From equations 20 and 21, it is clear that when CR holds, our initial formulation of not choosing $|T|$ as a parameter in U is reasonable. This means that we need not revise the monotonicity constraints.

7.2. Incentive Compatibility and Disincentivizing Faults

In economics, when a person must rely on others to solve a problem there is the important constraint of incentive compatibility: we want to make sure that people are motivated to behave in a manner consistent with the best solution. In our case, they may be motivated but might not have the skills to find the best solution. Incentive-compatibility in our case means that the best players are top-ranked. We will then use the solution of a top-ranked player as the best solution. The concept of "best" player is captured by quasi-perfection. These incentive-compatibility issues are covered in sections 8.2 and 8.4 where we prove the Meritocracy Theorem which implies that collusion-resistance leads to incentive compatibility.

Next we deal with a property, called DIF (DisIncentivizing Faults), which means that making a fault does not improve one's rank.

DIF: $\forall p_x, p_y \in P, \forall T \subseteq G(P), \forall r \in \{r \mid r \in G(P) \setminus T \wedge fault(p_y, r)\} [\, p_x \preceq^T p_y \Rightarrow p_x \preceq^{T \cup \{r\}} p_y \,]$

This is equivalent to incentivising not making faults. Let us first consider the game where p_y made a fault with a third player p_z. In that case,

$$U(wf^T(p_x), wu^T(p_x), lf^T(p_x), lu^T(p_x)) \leq \\ U(wf^T(p_y), wu^T(p_y), lf^T(p_y), lu^T(p_y) + 1) \quad (32)$$

From equations 1 and 32, we get the monotonicity constraint,

$$\partial_{lu} U \geq 0 \quad (33)$$

Now, let us consider the game where p_y made a fault in a game with p_x. In this case, there are 2 distinct possibilities. The first is p_x won the game and was forced, we have

$$U(wf^T(p_x) + 1, wu^T(p_x), lf^T(p_x), lu^T(p_x)) \leq \\ U(wf^T(p_y), wu^T(p_y), lf^T(p_y), lu^T(p_y) + 1) \quad (34)$$

From equations 32 and 34, we get the monotonicity constraint,

$$\partial_{wf} U \leq 0 \quad (35)$$

Table II. A modified Table T of game results for 3 players with an additional column for the opponent

GRID	Winner	Loser	Forced	Opponent
gr1	1	2	2	1
gr2	1	3	0	1

Let us consider the case where p_x won in an unforced position.

$$U(wf^T(p_x), wu^T(p_x)+1, lf^T(p_x), lu^T(p_x)) \leq U(wf^T(p_y), wu^T(p_y), lf^T(p_y), lu^T(p_y)+1) \tag{36}$$

From equations 32 and 36, we get the monotonicity constraint,

$$\partial_{wu} U \leq 0 \tag{37}$$

Summarizing the monotonicity constraints for DIF, we have,

$$\partial_{wf} U \leq 0 \wedge \partial_{wu} U \leq 0 \wedge \partial_{lu} U \geq 0 \tag{38}$$

From equations 19 and 38, we have that,

$$\text{LFB} \Rightarrow \text{DIF} \tag{39}$$

This confirms formally what we expect intuitively: LFB disincentivizes making faults. From equations 14 and 38, we have that,

$$\text{DIF} \Rightarrow \text{NNEW} \tag{40}$$

7.3. Example Scoring functions that are LFB

The representation theorem motivates us to explore the set of scoring functions that are LFB. Here, we list a set of 3 representative examples. It is useful to have families of LFB ranking functions to choose an appropriate member for a given competition.

7.3.1. Everyone get the same score. The simplest example of a LFB scoring function is where every player gets the same score.

$$U(wf^T(p_x), wu^T(p_x), lf^T(p_x), lu^T(p_x)) = 0 \tag{41}$$

It is easy to see that this scoring function satisfies the following monotonicity constraints,

$$\partial_{wf} U = 0 \wedge \partial_{wu} U = 0 \wedge \partial_{lf} U = 0 \wedge \partial_{lu} U = 0 \tag{42}$$

From equations 19 and 42, it is clear that this scoring function is LFB.

7.3.2. Regular fault counting. Another example would be to count faults.

$$U(wf^T(p_x), wu^T(p_x), lf^T(p_x), lu^T(p_x)) = lu^T(p_x) \tag{43}$$

This scoring function satisfies the following monotonicity constraints,

$$\partial_{wf} U = 0 \wedge \partial_{wu} U = 0 \wedge \partial_{lf} U = 0 \wedge \partial_{lu} U > 0 \tag{44}$$

From equations 19 and 44, it is clear that this scoring function is LFB.

7.3.3. Weighted fault counting. Now, we explore a more general version of fault counting. We consider positive weight assignment to fault classes where:

- The winner is a forced opponent, weakness in proponent (α fault): Weight $= \alpha$.
- The winner is a forced proponent, weakness in opponent (β fault): Weight $= \beta$.
- The winner is a non-forced opponent, weakness in proponent (γ fault): Weight $= \gamma$
- The winner is a non-forced proponent, weakness in opponent (δ fault): Weight $= \delta$

At this juncture, we need to define a few functions,

$$\forall p_x \in P, \forall T \subseteq G(P)[\, lu_\alpha^T(p_x) = \text{the number of } \alpha \text{ faults of } p_x \text{ in } T\,]$$
$$\forall p_x \in P, \forall T \subseteq G(P)[\, lu_\beta^T(p_x) = \text{the number of } \beta \text{ faults of } p_x \text{ in } T\,]$$
$$\forall p_x \in P, \forall T \subseteq G(P)[\, lu_\gamma^T(p_x) = \text{the number of } \gamma \text{ faults of } p_x \text{ in } T\,]$$
$$\forall p_x \in P, \forall T \subseteq G(P)[\, lu_\delta^T(p_x) = \text{the number of } \delta \text{ faults of } p_x \text{ in } T\,]$$

In the case of weighted fault counting, the table T is inadequate as we won't be able to decipher the exact type of fault. Another way to put it would be to say that the simplified game result is inherently lossy. To overcome this, we need to add another column in the game result table that lists the opponent O (see Table II). Now, for convenience, we redefine the scoring function as,

$$U(lu_\alpha^T(p_x), lu_\beta^T(p_x), lu_\gamma^T(p_x), lu_\delta^T(p_x)) = \alpha\cdot lu_\alpha^T(p_x) + \beta\cdot lu_\beta^T(p_x) + \gamma\cdot lu_\gamma^T(p_x) + \delta\cdot lu_\delta^T(p_x) \quad (45)$$

From equation 45, $\forall \alpha, \beta, \gamma, \delta \in \mathbb{R}^+$

$$\partial_{lu_\alpha} U > 0 \wedge \partial_{lu_\beta} U > 0 \wedge \partial_{lu_\gamma} U > 0 \wedge \partial_{lu_\delta} U > 0 \quad (46)$$

But,

$$lu^T(p_x) = lu_\alpha^T(p_x) + lu_\beta^T(p_x) + lu_\gamma^T(p_x) + lu_\delta^T(p_x) \quad (47)$$

From equations 46 and 47, it is clear that,

$$\partial_{lu_\alpha} U > 0 \wedge \partial_{lu_\beta} U > 0 \wedge \partial_{lu_\gamma} U > 0 \wedge \partial_{lu_\delta} U > 0 \Rightarrow \partial_{lu} U > 0 \quad (48)$$

From equation 48, it is clear that weighted fault counting is LFB. Now, we look at what tuning each of those weights imply.

- A high α encourages forced opponents to try to win. Tests agreement on proponent.
- A high β encourages forced proponents to try to win. Tests agreement on opponent.
- A high γ encourages non-forced opponents to try to win. Tests non-agreement or agreement on opponent.
- A high δ encourages non-forced proponents to try to win. Tests non-agreement or agreement on proponent.

All four might lead to a better quasi-equilibrium (see sections 8 and 8.3).

8. TOP RANKING AND MERITOCRACY

In this section we show a desirable implication of collusion-resistance. We introduce the concept of quasi-perfect player and show that quasi-perfect players are top-ranked iff the ranking relation is collusion-resistant. The quasi-perfect players are the players which have the minimum number of faults among all players. A quasi-perfect player may defend false claims and refute true claims and make "wrong" side choices. This differs from a perfect player who always makes the correct side-choices and successfully defends her choice.

To talk about top ranking, we need to first define the strictly better relation $\prec^T$. With $U : \mathbb{N} \times \mathbb{N} \times \mathbb{N} \times \mathbb{N} \to \mathbb{R}$,

$$\begin{aligned}\forall p_x, p_y \in P, \forall T \subseteq G(P)[\, p_x \prec^T p_y \Leftrightarrow \\ U(wf^T(p_x), wu^T(p_x), wf^T(p_x), lu^T(p_x)) < \\ U(wf^T(p_y), wu^T(p_y), wu^T(p_y), lu^T(p_y))\,]\end{aligned} \quad (49)$$

8.1. Top Ranking

A player p_x in a set of players P is said to be top-ranked $TR^T(p_x)$ in a set of game results T, if there is no player that is strictly better than p_x. More precisely,

$$TR^T(p_x) = \forall p_y \in P \mid \neg(p_y \prec^T p_x) \tag{50}$$

8.2. Quasi Perfection

A quasi-perfect player for a set of players is a player who makes the least number of faults within the set of players. A perfect player always chooses the correct side and always wins the defense and therefore is quasi-perfect. Such a player makes no faults.

8.3. Quasi-equilibrium

In economics, an equilibrium implies a position of rest characterized by absence of change. Assuming that the ranking relation is LFB, a table T is in quasi-equilibrium if there are no faults happening which means all wins must be against forced players. This implies that all players have chosen the same side and all forced players always lose and all non-forced players always win. A quasi-equilibrium is temporary if there exists a strategy which creates a fault for one of the players in the quasi-equilibrium. A quasi-equilibrium is stable if it is not temporary and is simply called an equilibrium. The games for a claim might go through several quasi-equilibria until the equilibrium is reached. When all players are perfect they produce an equilibrium. Quasi-equilibria are partially ordered. eq_1 is better than eq_2 if there is a strategy which creates a fault for a player in eq_2 but not in eq_1. Moving from one quasi-equilibrium to the next requires insight.

Simple claims have only one quasi-equilibrium while complex claims involving optimization might have several quasi-equilibria. Each equilibrium corresponds to a local optimum.

8.4. Meritocracy Theorem

— CR is necessary: If the ranking relation is not collusion-resistant (CR) but NNEW $\wedge$ NPEL, there exists a set of games where a quasi-perfect player is not top-ranked.
— CR is sufficient: NNEW $\wedge$ NPEL $\wedge$ CR imply that all quasi-perfect players are top-ranked.

Let us prove the first part of the theorem. First, we write down the monotonicity constraints for $\neg$CR and NNEW $\wedge$ NPEL. We have,

$$\neg\text{CR: } \partial_{wf}U < 0 \vee \partial_{wu}U < 0 \vee \partial_{lf}U > 0 \vee \partial_{lf}U < 0 \vee \partial_{lu}U < 0 \tag{51}$$

$$\text{NNEW} \wedge \text{NPEL: } \partial_{wf}U \leq 0 \wedge \partial_{wu}U \leq 0 \wedge \partial_{lu}U \geq 0 \wedge \partial_{lf}U \geq 0 \tag{52}$$

The equations 51 and 52 give us a clue into exploring a scoring function for our ranking relation that is $\neg$CR, but NNEW $\wedge$ NPEL. One proposal for such a scoring function is,

$$U(wf^T(p_x), wu^T(p_x), lf^T(p_x), lu^T(p_x)) = -(wf^T(p_x) + wu^T(p_x)) \tag{53}$$

To motivate the necessity of CR to hold for quasi perfect players to be top ranked, we consider an example. With the scoring function as specified by equation 53 (which is basically counting the total number of wins for a particular player), consider the table III of game results. Here, player 1 is top ranked with the 4 wins and 2 faults. But, player 2, the quasi perfect player with no faults is not top ranked.

8.4.1. Proof of the Meritocracy Theorem. We need that a quasi perfect player should be top ranked. Say, p_x made k faults and other players made at least k faults. We need to

Table III. A Table T of game results for 3 players

GRID	Winner	Loser	Forced
gr1	1	2	2
gr2	1	3	0
gr3	1	2	2
gr4	2	1	0
gr5	2	3	2
gr6	2	3	0
gr7	3	1	0
gr8	1	3	1

explore the monotonicity conditions for p_x to be top ranked. This requires that, if $lu^T(p_x) \leq lu^T(p_y), \forall p_y \in P \setminus \{p_x\}$, we need that,

$$U(wf^T(p_x), wu^T(p_x), lf^T(p_x), lu^T(p_x)) \leq U(wf^T(p_y), wu^T(p_y), lf^T(p_y), lu^T(p_y)) \quad (54)$$

As equation 54 should be true for any choice of the other 3 parameters of U, except for the constraint on faults, we can conclude that the only monotonicity constraint is that the scoring function should be non-decreasing on faults. So, the monotonicity constraint for quasi-perfect players to be top ranked is,

$$\partial_{wf}U = 0 \wedge \partial_{wu}U = 0 \wedge \partial_{lf}U = 0 \wedge \partial_{lu}U \geq 0 \quad (55)$$

From equations 19 and 55, it is clear that,

$$\text{Quasi Perfect player being Top Ranked} \Leftrightarrow \text{LFB} \quad (56)$$

From equations 22 (The Representation Theorem) and 56, when NNEW$\wedge$NPEL holds, we have,

$$\text{Quasi Perfect player being Top Ranked} \Leftrightarrow \text{LFB} \Leftrightarrow \text{CR} \quad (57)$$

From equation 57, we can see that $\neg$CR implies that Quasi Perfect player might not be top ranked. We notice that our proof is not constructive.

8.4.2. Observation. A small observation we need to make is that NNEW $\wedge$ NPEL impose monotonicity on the scoring function $U : \mathbb{N} \times \mathbb{N} \times \mathbb{N} \times \mathbb{N} \rightarrow \mathbb{R}$. This justifies our approach of expressing $\neg$CR in terms of monotonicity constraints. This is important as CR has a universal quantifier (see section 5.1), making $\neg$CR have an existential quantifier. So, a possible choice for a $\neg$CR scoring function is a non monotonic one, but such a function would not guarantee NNEW $\wedge$ NPEL.

9. CONCLUSIONS

In this paper we laid the foundations for organizing socio-technical systems for creating knowledge in Formal Sciences. The foundations are based on a new concept, called a side-choosing game (SCG), and a theory about mapping game results to player rankings in the presence of collusion-resistance (CR). CR is a crucial concept which says that games where you are not in control cannot affect your ranking. Our results show that in the presence of two non-controversial axioms, CR is equivalent to being local-fault-based (LFB). Therefore, if ”natural” scoring functions are used, like counting wins or counting wins against non-forced players, there is the danger of collusion which compromises truth. Collusion-resistance prevents gaming the game in a strong sense.

What comes next? Our plan is to put SCG-based applications on the web and reap the benefits of collective intelligence. So far we used SCG-based ideas and tools in designing courses at Northeastern University. We would like to build a tool that can be used in MOOCs and for algorithm competitions. The implementation of our domain-specific language for human computation for formal sciences requires several algorithms to be developed. Should those algorithms be challenging, we use human computation with side-choosing games to develop them!

An important area that needs further work is that participants can propose new claims. We want a modular approach to solving claims. For example, a complex claim C_1 might be reducible to a "simpler" claim C_2 so that a solution for C_2 implies a solution for C_1. We propose a formal study of claim relations which can themselves be captured as claims and approached with side-choosing games.

ACKNOWLEDGMENTS

We would like to thank Ravi Sundaram for his input to the paper. We would like to thank the audiences at the following universities for their valuable feedback. The last author has given talks in the summer of 2014 at (in chronological order) University of Erlangen-Nuernberg, Technical University of Darmstadt, ETH Zurich and Harvard University. We thank Reshef Meir for suggesting a flaw in our old definition of quasi-perfect.

We would like to thank Ian and Pamela Holland and several of Karl Lieberherr's students for their support of the Karl Lieberherr Fund at Northeastern University which has financed Ahmed Abdelmeged for the summer 2013 and Ruiyang Xu for the summer 2014. We thank Novartis, Switzerland for their support of the early years of this research.

REFERENCES

ABDELMEGED, A. 2014. Organizing computational problem solving communities via collusion-resistant semantic game tournaments. Ph.D. thesis, Boston, MA, USA. Advisor-Karl Lieberherr.

BEASLEY, J. 2006. *The Mathematics of Games*. Dover books on mathematics. Dover Publications, Incorporated.

BOYER, J. AND SANDU, G. 2012. Between proof and truth. *Synthese 187,* 3, 821–832.

COOPER, S., TREUILLE, A., BARBERO, J., LEAVER-FAY, A., TUITE, K., KHATIB, F., SNYDER, A. C., BEENEN, M., SALESIN, D., BAKER, D., AND POPOVIĆ, Z. 2010. The challenge of designing scientific discovery games. In *Proceedings of the Fifth International Conference on the Foundations of Digital Games*. FDG '10. ACM, New York, NY, USA, 40–47.

DEMAINE, E. D. 2001. Playing games with algorithms: Algorithmic combinatorial game theory. In *MFCS*, J. Sgall, A. Pultr, and P. Kolman, Eds. Lecture Notes in Computer Science Series, vol. 2136. Springer, 18–32.

ELO, A. 1978. *The rating of chessplayers, past and present*. Arco Pub.

KULAS, J. AND HINTIKKA, J. 1983. *The Game of Language: Studies in Game-Theoretical Semantics and Its Applications*. Synthese Language Library. Springer.

LANGVILLE, A. AND MEYER, C. 2012. *Who's #1?: The Science of Rating and Ranking*. Princeton University Press.

LIEBERHERR, K. J., ABDELMEGED, A., AND CHADWICK, B. 2010. The Specker Challenge Game for Education and Innovation in Constructive Domains. In *Keynote paper at Bionetics 2010, Cambridge, MA, and CCIS Technical Report NU-CCIS-2010-19*. http://www.ccs.neu.edu/home/lieber/evergreen/specker/paper/bionetics-2010.pdf.

RUBINSTEIN, A. 1980. Ranking the participants in a tournament. *SIAM Journal on Applied Mathematics 38,* 1, pp. 108–111.

SIMPSON, E. 2014. Combined decision making with multiple agents. Ph.D. thesis, University of Oxford.

TULENHEIMO, T. 2013. Independence friendly logic. In *The Stanford Encyclopedia of Philosophy* Fall 2013 Ed., E. N. Zalta, Ed.

APPENDIX

A. FORMAL DEFINITION OF SCG

A side-choosing game is a triple (G, SC, AA), where G is a two-person, draw-free, combinatorial game, SC is a side-choice configuration and AA is an agreement algorithm. G, SC and AA are defined separately. The important component is the combinatorial game G; SC and AA offer variation possibilities to define the side-choosing games. We will use simple instances of SC and AA for our side-choosing games.

A.1. Definition: Combinatorial Game

(1) There are two players. The game is sequential (turn-based) and Perfect-Information. There are no chance moves or hidden information. (But players may hide their winning strategies.)
(2) There is a finite set of possible positions of the game. There is a distinguished start position.
(3) The rules of the game specify, for both players and each position, which moves to other positions are legal moves.
(4) The game ends when a position is reached from which no moves are possible. A predicate on the final position determines who has won. There is an absolute winner: the first player to fulfill the winning condition. No ties or draws.
(5) The game ends in a finite number of moves.

A.2. Side-Choice Configuration Definition

The side choices are made by the players but the Side-choice Configuration (SC) defines the sequencing of the design-time decisions. We have two players, x and y, who make a choice dx and dy for the start position of the combinatorial game G. dx, dy are elements of $\{P, O\}$. If x chooses $dx = P$ then x claims to win the combinatorial game G from the game's start position. If x chooses $dx = O$ then x claims to prevent y from winning G from its start position. The players make the side choices but SC specifies the configuration. Examples for SC :

— simultaneous
  The two players make side choice independently. This is our preferred side-choice configuration.
— sequential
  With probability q for x to be the first player. Or we can choose other context-sensitive mechanisms to select the first player.

A.3. Agreement Algorithm Definition

The agreement algorithm maps two players x, y with their design-time side choices dx, dy, where $dx = dy$, into a set of plays between the two players with, for each play, run-time side choices rx, ry such that at most one player is forced.

— Example Agreement algorithm CAA (Competitive Agreement Algorithm):
  Randomly choose z, one of the players x, y and force z.
  — Play $SCG(G, x, y, dz, d!z, !dz, d!z)$; z is forced
  — Play $SCG(G, x, y, dz, d!z, dz, !d!z)$; $!z$ is forced
  Motivation for CAA: For some claims it is a disadvantage to have to move first because you give away a secret. Therefore, we choose the forced player randomly to balance the potential disadvantage. We play two games to give each player a chance to test the other. This is our preferred agreement algorithm.
— Alternative Example for Agreement Algorithm

CAA-1/2 only plays one game choosing the forced player randomly. CAA-1/2 (Competitive Agreement Algorithm 1/2): Randomly choose z, one of the players x, y and force z.
— Play $SCG(G, x, y, dz, d!z, !dz, d!z)$; z is forced

A.4. Example Claims

We provide three example claims to illustrate how side-choosing games work.We define the side-choosing games by considering the semantic game associated with an interpreted logical formula.

A.4.1. Saddle Point. We present a maximization problem as a claim here. Consider the function F described below,

$$F(\theta \in [0,1]) = \forall x \in [0,1]\, \exists y \in [0,1] \,|\, xy + (1-x)(1-y^2) \geq \theta$$

Now, we claim that,

$$\exists \theta \in [0,1] \,|\, F(\theta) \,\wedge\, (\forall \epsilon > 0 \,|\, \neg F(\theta + \epsilon))$$

This clearly is a claim that there exists a maximum θ beyond which $F(\theta)$ is false.

A.4.2. Distributing balls. Consider the problem of distributing $m \in \mathbb{Z}^+$ red and $n \in \mathbb{Z}^+$ green balls into two sacks. Let $S_1(r_1, g_1)$ and $S_2(r_2, g_2)$ such that $r_1 + r_2 = m$ and $g_1 + g_2 = n$ represent the distribution of the given set of balls in the sacks. Given, that picking one of the two sacks is equally likely, we have the following claim.
The only distribution of balls that maximizes the chance of drawing a red ball is when one of the sacks has only one ball and the color of that ball is red.
We have chosen to express the claim in plain English, which could be translated into a predicate logic formula that precisely defines the SCG.

A.4.3. Minimum Graph Basis

(1) *Size of minimum graph basis*: a basis of a directed graph G is defined as a set of nodes such that any node in the graph is reachable from some node in the basis. Formally, $MinBasisSize(G \in Digraphs, n \in \mathbb{N}) = BasisSize(G, n) \wedge \forall k\ s.t.\ k < n : \neg BasisSize(G, k)$ where $BasisSize(G \in Digraphs, n \in \mathbb{N}) = \exists s \in \mathcal{P}(nodes(G))\ s.t.\ |s| = n\ : \forall m \in nodes(G)\ \exists p \in paths(G) : last(p) = m \wedge first(p) \in s$.
(2) *Number of source nodes of a DAG*: a source node is a node with no incoming edges. Formally, $\#src(D \in DAGs, m \in \mathbb{N}) = \exists s \in \mathcal{P}(nodes(D))\ s.t.\ |s| = m\ : \forall v \in nodes(D) : inDegree(v) = 0 \Leftrightarrow v \in s$.

The relation between the two claim families $MinBasisSize(G \in Digraphs, n \in \mathbb{N})$ and $\#src(D \in DAGs, m \in \mathbb{N})$ can be described by $\forall G \in Digraphs, n \in \mathbb{N}\ : MinBasisSize(G, n) = \#src(SCCG(G), n)$ where $SCCG$ refers to Tarjan's the Strongly Connected Component Graph algorithm.
Note that the above claims don't express that we want the fastest algorithm.

Received February 2015; revised Month Year; accepted Month Year