Games for Controls

Krishnendu Chatterjee
EECS, UC Berkeley.

Radha Jagadeesan*
School of CTI, DePaul University.

Corin Pitcher
School of CTI, DePaul University.

Abstract

We argue that games are expressive enough to encompass (history-based) access control, (resource) usage control (e.g., dynamic adaptive access control of reputation systems), accountability based controls (e.g., insurance), controls derived from rationality assumptions on participants (e.g., network mechanisms), and their composition. *Building on the extensive research into games, we demonstrate that this expressive power coexists with a formal analysis framework comparable to that available for access control.*

1. Introduction

We advocate two player turn-based games (perhaps with quantitative and probabilistic information) as a framework to describe policies on shared resources. The two players in question are the System (the owner of the resource) and the Player (the entity requesting access). Games permit the incorporation of a broad spectrum of assumptions about Player models (ranging from an adversarial model to more cooperative viewpoints modelling behavior in rational self-interest) and System objectives (ranging from exact controls for absolute correctness to risk management approaches that bound the utility lost to undesirable usage). Games also enable a uniform description of controls that differ in when checks are done (such as prior to granting access, or ongoing regulations while the resource is being used or obligations discharged after the fact).

We substantiate this argument by examining existing mechanisms for controls from a game viewpoint.

In (stack [47, 46] or history-based [5]) access control, most interesting moves, such as the initial request and the moves recorded in the examined history, are made by the Player. The only move made by the System is a grant/deny move at the end of the interaction. In the parlance of game theory, this access control model is a "1-player game", aka transition system. Thus, the operational model for (history-based) access control is (finite state) automata (e.g., [41]), algebraic declarative models for access control are based on regular expressions (e.g., [4]) and logic-based declarative approaches are usually in some fragment of many-sorted first-order predicate logic with sorts for roles and time [24]). The compositional approaches to access control policy languages (e.g., [12, 48, 11, 16] to name but a few), reflect this viewpoint.

The above analysis ceases to hold when we move to more flexible and dynamic derivatives of access control, such as the (resource) usage control model [37, 36]. In this view, a complete usage process consists of three phases: before-usage, ongoing-usage, and after-usage, with System and Player actions permitted in all three phases. Thus the (resource) usage control model is fundamentally of ongoing interaction between the Player and the System. Consequently, the design and analysis of such mechanisms includes situations in which Player (resp. System) want to respond to strategic behavior by System (resp. Player). Consider adaptive access control based on reputation (e.g., [28]).

EXAMPLE 1. The following policies are reproduced verbatim from eBay.

- Acceptable payment methods include Credit Cards or Bank Transfers via PayPal or US Postal Money Order (Bank Money Orders will delay shipment). If your feedback rating is 50 or higher with no negative comments, we will consider accepting a personal or business check and holding shipment until the check clears.

- Please contact me before bidding if you have more than one negative feedback within the last six months or if your feedback or identity is hidden. I reserve the option to not accept your bids otherwise.

■

[*] Supported in part by NSF Cybertrust 0430175

The game, as described informally in the above example, specifies the policy. System moves include changes to future player reputation, and providing a suite of payment alternatives that depend on the current Player reputation. Since the System cannot control Player moves, and vice versa, the key issue in this context is “controllability”. For example, does the Player have a strategy to ensure that she can always ensure that her bid is always accepted? In such reasoning, the Player has to reason against *all* possible moves permissible for the System, using *existing* Player moves to achieve her objective.

Such mixed quantifier reasoning — universal on the moves of one participant and existential on the moves of the other participant —- is not supported directly by existing (linear-time) temporal logic based approaches [49] to usage control.

Games also open up the possibility of applying quantitative methods to controls. This permits specification and analysis that is difficult to formulate in the purely (boolean) 0/1 world of traditional access/usage control, such as designs incorporating rationality assumptions on protagonist behavior. While not universally applicable, such assumptions, when they hold, can simplify the design of access control. The following policy captures some of the ingredients of the class of protocols exemplified by bankable postage for network services [3].

EXAMPLE 2. The following policy enforced at a local grocery store aims to encourage customers to return the carts to a predetermined location in the store.

Patrons rent a cart at a nominal charge, e.g., 25¢, from a location inside the store. Every cart is equipped with a device that has two connected slots. This device is long enough to accommodate one quarter. A chain with a coin-sized flat piece of metal is also attached to the cart. Free carts are lined up, where the chain from one cart goes into the device slot of the next cart.

When a customer arrives, she deposits a coin into the slot. The coin nudges out the chain coming in from the previous cart, thus detaching the first cart from the chain of carts. When the customer returns the cart to this location, the process is reversed: the chain from the previous cart is inserted into the slot, pushing out the coin. The customer gets back her coin and the returned cart becomes part of the chain of carts.

This system works: almost always, there are no carts left in the parking lot. ∎

This example motivates the use of quantitative cost information on transitions in games: the initial player move costs the Player 25¢ and the return move pays back 25¢ to the Player. How to analyze such a system? Worst case adversarial assumptions are not useful to justify the success of this access mechanism, since the customer is not forced to return the cart in the right spot. Rather, what is being employed is a gentle appeal to the rationality of the customer. 25¢ seems to suffice to convince the customers to indulge in behavior that results in maximum utility for both the store and the customer.

More generally, quantitative approaches apply to scenarios where active and complete policy enforcement is infeasible, inconvenient or too costly in practice. In such examples, access is granted when resources are requested, with auditing used later to establish accountability and check whether the requestor had the required privileges [17]—[14, 39] delineate criteria on the systems that can use such policies. Insurance is a classical example where auditing is used to enforce accountability.

EXAMPLE 3. [http://www.bls.gov/oco/ocos125.htm] Individuals purchase insurance policies to protect against monetary losses from incidents such as fire. As part of the insurance application, information about number and placement of fire detection systems is requested. This information is usually not checked but recorded. In the event of a loss, policyholders submit claims seeking compensation for their loss. Claims examiners and insurance investigators verify the policy-holder’s statements about the fire detection equipment before settling the claims upon the policy. ∎

In such games, in addition to normal System access-control moves that grant or deny policies upfront, there are also System auditing moves to demand evidence, and grant or deny claims. The auditing moves are perforce best modelled probabilistically to quantify the success of the auditing process with costs on appropriate edges to indicate the payout/penalty to Player and System. As in the previous example, the Player can violate policy at risk of being traced, so the iron-clad guarantees of access/usage control are usually not achieved here. Instead, what the System is attempting to do is risk management, i.e. analyze and control the expected worst case loss.

To go along with expressiveness, we now argue that games support effective reasoning and analysis principles analogous to those of access control alluded to earlier.

- Games provide a direct operational specification of policies, including those with history and quantitative information. We demonstrate that our definitions support the existence of equilibria required to reason about rational behavior. Our proofs are constructive and yield strategies for the participants to enforce equilibria behavior on the opponents.

- We show that policy specifications (aka games) can be compositionally constructed using an algebra of operators. These include game interpretations of propositional operators, such as conjunction and disjunction; of temporal operators, such as sequencing and iteration; and spatial conjunction operators to combine policies on multiple resources.

- We provide a logic to reason about properties of individual games.

  The quantitative information in these games, such as costs and probabilities, are usually inspired by empirical data. Thus, these numbers are to be viewed as coming with an error estimate. This motivates the idea that the analysis should vary smoothly with perturbations in the numerical values.

  We delineate an approximate reasoning approach based on (metric) distances between states. For games without quantitative information, the metric approach specializes down to usual exact equational reasoning, e.g., 0-distance coincides with bisimilarity.

  We show that closeby states have closeby quantitative properties, for *both* worst case and rational assumptions on the protagonists. We also show that most of the game combinators preserve closeness, e.g., if games P_1, Q_1 (resp. P_2, Q_2) are closeby, then the product games $P_1 \times P_2$ and $Q_1 \times Q_2$ are closeby.

When restricted to transition systems, these formal techniques yield the usual ones for access control: e.g., metrics for games reduce to bisimulation for transition systems.

1.1. Related work

Games provide a basic model for interaction, perhaps with quantitative elements — so, it is unsurprising that they have been used in a wide variety of contexts, from economics [45] to optimal control [22].

In the context of security, there is a growing interest in the economic aspects of information security, e.g., see [10] for a broad overview of research in this general area. Examples 2 and 3 of the introduction, echo the themes underlying such research. However, our focus is intentionally narrow and limited to the general area of controls, and reflects our attempt to formalize the first half of [29].

Mean-payoff games. Perfect-information (turn-based) stochastic games have been widely studied in stochastic game theory [40]. [30] studied zero-sum mean-payoff games and established existence of values in such games and characterized existence of simple optimal strategies. [42, 43] prove existence of equilibrium in nonzero-sum mean-payoff games by an application of *threat* strategies.

Cost-based frameworks for analysis of denial of service [33] can be viewed as games with quantitative information (albeit without stochastic information) in the form of more general cost functions than we permit in this paper.

In this work we extend the construction of threat strategies to obtain a useful class of Nash equilibria.

Logics. The logics to enable specification of open or multiagent systems where the different agents may represent different components of the system and the environment have been developed in two different traditions. On the one hand, there is Alternating Temporal Logic (ATL) [8] developed as a game-based extension of temporal logic in research into computer aided verification and control. There are also the coalition/game logics — see [38] that includes a historical survey — developed in investigations into multiagent systems based on epistemic logic. Alternating temporal logics have already been used in the specification and verification of a variety of protocols, e.g., [32, 27].

In this paper, we enhance alternating temporal logics by combining them with the logics developed for probabilistic systems [13, 20].

Game algebras. Game algebras have been investigated in [34, 35], albeit under the nomenclature of “game logics” [1] — see [44] for a detailed presentation in course notes. Game constructors have been explored in the context of semantics of Linear logic [15, 6] and programming languages [25, 7]. None of this prior work was meant to address costs and probabilities[2].

In this paper, we select operators, guided by relevance to the application of interest, accounting for the extra features of costs and probabilities in our formal development. For each operator, we examine compositional reasoning to construct strategies in the composite game from those for the component games.

Approximate reasoning. The arguments for approximate reasoning and an “approximate” notion of equality of processes are by now well-known [26, 31]. These remarks were made for probabilistic systems, but the same remarks apply, mutis mutandis, to costs as well. In the probabilistic context, both these papers propose that the correct formulation of the “nearness” notion for approximate reasoning is via a metric. Prior research in this area includes our investigations into metrics for Markov processes [21], the subsequent study of metrics for probabilistic games [19] and the study for generalized semi-Markov processes [23].

In this paper, we adapt [19] to address games with costs. In addition we demonstrate that closeby games have closeby strategies, both for adversarial and rational situations. We also show that the operators are robust for perturbations of numerical values.

[1] In this paper, we reserve the term “ game logic” for logics that talk about the properties of individual games, and use “game algebra” for methods to construct composite games from simpler ones.

[2] Game studies of probabilities in programming languages [18] incorporate probabilities in strategies, not in the game perse.

1.2. Rest of the paper

In section 2, we review the basic definitions for the games that we use in this paper. In section 3 we sketch the foundations for the reasoning methods. It is possible to skim this section initially and read in detail in a demand-driven fashion as the following section is read. In section 4, we describe the game algebra.

The rest of the paper is punctuated with examples to make the ideas concrete and reinforce the idea that games permit mixtures of a varieties of controls.

In the interests of space and exposition, the main sections of the paper focus on the novel features. We elide most proofs in this extended abstract.

2. Background

Notation. For $i \in \{1,2\}$, we use $\bar{i}$ for $i \mathtt{~mod~} 2+1$. We write $D(U)$ for the set of probability distributions over the state space U. We use $\uplus$ for disjoint sum of sets/relations/probability distributions. In the formal development, we treat the Player and System symmetrically for conciseness — so, we just talk of player- 1 and player- 2 symmetrically.

2.1. Turn-based probabilistic games

DEFINITION 4. Let L be a set of labels. A *turn-based probabilistic game graph* ($2\,{}^1\!/\!_2$*-player game graph*) $G = ((S,E),(S_1,S_2,S_{1\bigcirc},S_{2\bigcirc}),\delta)$ consists of a directed graph (S,E), a partition $(S_1,S_2,S_{1\bigcirc},S_{2\bigcirc})$ of the finite set S of states, such that $E \subseteq \cup_i S_i \times L \times S_{i\bigcirc}$ and δ: $S_{i\bigcirc} \to D(S_{\bar{i}})$ yields a probability distribution at each state in $\cup_i S_{i\bigcirc}$.

A *rooted game graph* is a turn-based probabilistic game graph with a specified start state $s \in S_1 \cup S_2$. ∎

An evolution in such a game graph proceeds as follows. The states in S_1 are the *player-1* states, where player 1 decides the successor state, which is a state in $S_{1\bigcirc}$ by definition. At an $S_{1\bigcirc}$ state, the successor is chosen as per the prescribed probability distribution — this successor state is a state from S_2 by definition. Evolution at such a probabilistic state is totally autonomous without input from either player. Symmetrically for states in S_2 that are the *player-2* states, where player 2 decides the successor state.

In contrast to standard presentations, we do not require every state to have a successor state: this extra bit of generality is useful to indicate termination of a policy. We sometimes call a state without successors a *blocking* state. We include labels on edges to facilitate compositions of games.

The above definition incorporates rigid alternation requirements to lighten the notation in the technical development. The essential technical restriction is that every state has an uniquely identified player determining successor states. The other constraints — e.g., the strict alternation between $S_1, S_{1\bigcirc}$ (resp. $S_2, S_{2\bigcirc}$) — are inessential. Indeed, when describing actual examples, we will tend to be loose wrt the strict alternation.

EXAMPLE 5. An approach to a simplified version of example 1 is the game specification of the seller policy in figure 1. In this specification, the bidder's reputation is good iff they paid on their last transaction or the seller chose to accept a new bidder as good for their first transaction. In this figure, the trailing `GOOD` (resp. `BAD`) at the right hand end of the figure are intended as loops back to `P GOOD` (resp. `BAD P`) at the left of the figure.

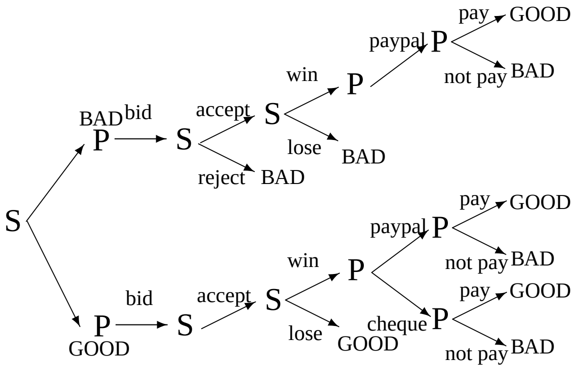

Figure 1. Auction

In order to minimize the clutter in the diagram, we have elided intermediate states that do not play a significant role. In a specification that is in complete conformance with definition 4, we would have to introduce intermediate states to enforce strict alternation. For example, the `bid` transition from `P GOOD` is missing an intermediate probabilistic state with a probability 1 transition to the System state. ∎

Let $s \in S_{i\bigcirc}, U \subseteq S_{\bar{i}}$ we write $\delta(s)(U)$ for the (probability) measure of U, i.e. $\delta(s)(U) = \sum \delta(s)(t) \mid t \in U$. Let $S_{\bigcirc} = S_{1\bigcirc} \cup S_{2\bigcirc}$. For $s \in S_{\bigcirc}$ and $t \in S$, we write $E(s)$ to denote the set of possible successors of s, i.e. the set of all t to which s has an edge with non-zero probability.

The *turn-based deterministic game graphs* (*2-player game graphs*) are the special case of the $2\,{}^1\!/\!_2$-player game graphs such that for all $s \in S_{\bigcirc}$ with have $|E(s)| = 1$. The *Markov decision processes* ($1\,{}^1\!/\!_2$*-player game graphs*) are the special case of the $2\,{}^1\!/\!_2$-player game graphs such that for all states $s \in S_2$ we have $|\{t \mid (s,l,t) \in E\}| = 1$.

Plays. A path in the game graph is a finite or infinite sequence $\omega = \langle s_0, l_0, s_1, l_1, s_2, \ldots \rangle$ of states and labels such that $(s_k, l_k, s_{k+1}) \in E$. A *play*, of the game graph G is either an infinite path of states or a finite terminated path, i.e.

a sequence $\langle s_0, l_0, s_1, l_n \ldots, s_n \rangle$ such that s_n has no outgoing edges. We write Ω for the set of all plays, and for a state $s \in S$, we write $\Omega_s \subseteq \Omega$ for the set of plays that start from the state s.

Strategies. A strategy for a player is a recipe to extend a play, i.e., given a finite sequence of states, representing the history of the play, a strategy for a player chooses the successor state to extend the play. In this paper, we will be concerned only with *pure* strategies, i.e. the action taken by a player yields a unique successor state. In the rest of this paper, we will just use "strategy" for "pure strategy"[3].

Let $\mathtt{M}$ be a set called *memory* that encodes the information about the history of the play. A player-1 strategy can be described as a pair of functions: a *memory-update* function σ_u: $S \times \mathtt{M} \rightarrow \mathtt{M}$ and a *next-move* function σ_m: $S_1 \times \mathtt{M} \rightarrow S_{1\bigcirc} \times L$. A strategy must prescribe only available moves, i.e., for all $s \in S_1$, for all $m \in \mathtt{M}$, and for all $t \in S_{1\bigcirc}$, if $\sigma(s,m) = (t,l)$, then $(s,l,t) \in E$. We denote by Σ the set of strategies for player 1. Analogously we define the corresponding strategy family Π for player 2.

The strategy (σ_u, σ_m) is *finite-memory* if the memory $\mathtt{M}$ is finite. We denote by Σ^F the set of (pure) finite-memory strategies for player 1. The strategy (σ_u, σ_m) is *memoryless* if $|\mathtt{M}| = 1$; that is, the next move does not depend on the history of the play but only on the current state. A memoryless strategy for player 1 can be represented as a function σ: $S_1 \rightarrow S_{1\bigcirc} \times L$.

Interaction of strategies. Player 1 follows the strategy σ if in each player-1 move, she chooses the next state according to σ. Once a starting state $s \in S$ and strategies $\sigma \in \Sigma$ and $\pi \in \Pi$ for the two players are fixed, the outcome of the game is a random walk $\omega_s^{\sigma,\pi}$ for which the probabilities of events are uniquely defined, where an *event* $A \subseteq \Omega$ is a measurable set of paths.

EXAMPLE 6. In the game of figure 1, the buyer has a strategy to ensure that a bid is accepted, e.g., by always choosing the `pay` move. Similarly, the System has a strategy to ensure that it has the option of rejecting the next bid if the bidder did not pay the prior successful one, e.g., by choosing resp. `BAD P` in case the Player has not paid. Since the game records the Player reputation in the state, the System has a memoryless strategy to achieve this objective.

The interaction of these two strategies leads to a path in the bottom half of the game tree that always ends in a `P GOOD`. ∎

2.2. Mean-payoff Games

DEFINITION 7 (MEAN-PAYOFF GAME.). A *mean-payoff game* $G = (G, r_1, r_2)$ consists of a $2\frac{1}{2}$-player game graph G, and two reward functions $r_1 : E \rightarrow \mathbb{R}$ and $r_2 : E \rightarrow \mathbb{R}$, where E is the set of edges in G. ∎

Both players win the "long-run average" of the corresponding rewards of a play. Formally, given a finite play of even length $\omega = \langle s_0, l_0, s_1, \ldots, s_{2n} \rangle$ the value for player 1:

$$valpath_1(\omega) = \frac{1}{2n} \sum_{i=0}^{2n-1} r_1((s_i, l_i, s_{i+1}))$$

where (s_i, l_i, s_{i+1}) denotes the edge labeled l_i from s_i to s_{i+1}. Similarly for player 2 using r_2 instead of r_1. For an infinite play $\omega = \langle s_0, l_0, s_1, l_1, s_2, \ldots \rangle$, both players win the "long-run average" of the corresponding rewards of a play, i.e.

$$valpath_1(\omega) = \liminf_{n \rightarrow \infty} \frac{1}{2n} \sum_{i=0}^{2n-1} r_1((s_i, l_i, s_{i+1}))$$

and similarly for $valpath_2(\omega)$.

A zero-sum game is the special case when the gain $r_1(e)$ for player 1 is the loss of player 2.

DEFINITION 8. A *zero-sum mean payoff* game is a mean-payoff game in which $r_1(e) = -r_2(e)$ for all edges $e \in E$. ∎

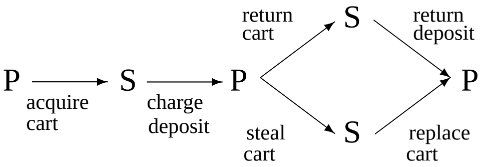

Figure 2. Grocery Cart

EXAMPLE 9. The game for example 2 of the introduction is described in figure 2. In this game, the `charge-deposit` has cost 25c (resp. gain 25c) for the Player (resp. System) and the edge `return-deposit` reverses the payments. The `replacecart` transition has a large cost (order of dollars) for the System.

There is only one System strategy since there are no choices in System moves. ∎

EXAMPLE 10. Illinois uses a system known as I-PASS to collect payments on toll roads. Drivers with an I-PASS account and a transponder in their car may drive through

[3] A strategy that is not necessarily pure, i.e. uses randomization, is called *randomized*. We do not consider randomized strategies in this paper.

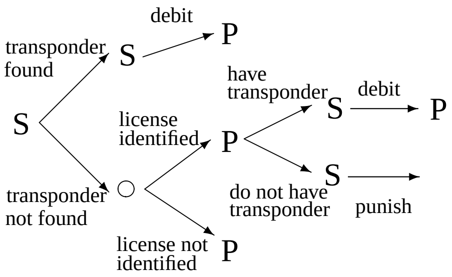

Figure 3. I-PASS

specially-marked fastlanes where traffic need not, and must not, stop. In this case, the toll payment is debited from their I-PASS account. Failure to detect a transponder in an I-PASS lane causes an attempt to punish the owner of the car by tracing the license plate.

Figure 3 presents a game describing the policy for such a fastlane. Initially, the system tries to detect a transponder. The cost (resp. gain) on the `debit` edge reflects the payment from (resp. to) the Player (resp. System). The probabilistic state (represented by a circle in figure 3) is intended to model the image recognition techniques that are claimed to be used if a transponder is not detected. ∎

Values and optimal strategies. The values for the players are the maximal payoff that each player can guarantee.

Recall that the the result of the interaction of strategies σ, π is a random walk $\omega_s^{\sigma,\pi}$ for which the probabilities of all measurable sets of paths are uniquely defined. Since each path has a player reward as defined above, the objectives achieved by each players at a state s for such an interaction is the expected value induced by this random walk over the set of paths: we denote these by $\mathrm{E}_s^{\sigma,\pi}[valpath_1]$ and $\mathrm{E}_s^{\sigma,\pi}[valpath_2]$.

Finally, for a state s, the values for the player i at state s is defined by considering all player i strategies against all strategies of the opposing player. Formally, these are given by functions $val_1 : S \longrightarrow \mathbb{R}$ and $val_2 : S \longrightarrow \mathbb{R}$ defined as follows: $val_1(s) = \sup_{\sigma \in \Sigma} \inf_{\pi \in \Pi} \mathrm{E}_s^{\sigma,\pi}[valpath_1]$ and $val_2(s)$ is defined similarly.

A strategy σ for player 1 is optimal for a state s, if we have $val_1(s) = \inf_{\pi \in \Pi} \mathrm{E}_s^{\sigma,\pi}[valpath_1]$, i.e., the strategy ensures player 1 the payoff of at least $val_1(s)$ against all player 2 strategies. The definition for optimal strategies for player 2 is analogous.

EXAMPLE 11. The optimal strategy for Player in the grocery game of figure 2 is the strategy that returns the cart — it has value 0.

The sole strategy for System has a negative value corresponding to the cost of replacing the cart: this value is realized while interacting with Player strategy that does not return the cart. ∎

EXAMPLE 12. The optimal strategy for Player in the IPASS game of figure 2 depends on the relationship between two numbers: (a) the expected cost of being caught cheating (i.e. probability of the license identified times the cost on `punish` in the figure) and (b) the cost of the charge debited in the `debit` transitions. If (a) $>$ (b), then the Player is best served by having a transponder.

In this example, a good analysis of System strategies requires a (probabilistic?) modelling of the soundness of transponder detection that is not done in our model. Note however, that the optimal strategy for Player protects against the System deciding to go with a very faulty transponder detection system. ∎

2.3. Nash equilibrium.

A notion of rational behavior in nonzero-sum game is captured by the notion of Nash equilibrium.

In a nonzero-sum mean-payoff game, a strategy profile (σ^*, π^*) is a Nash equilibrium if none of the players gain by unilateral deviation. Formally, (σ^*, π^*) is a Nash equilibrium if and only if the following conditions hold:

$$\forall \sigma \in \Sigma.\ \mathrm{E}_s^{\sigma^*,\pi^*}[valpath_1] \geq \mathrm{E}_s^{\sigma,\pi^*}[valpath_1]$$

$$\forall \pi \in \Pi.\ \mathrm{E}_s^{\sigma^*,\pi^*}[valpath_2] \geq \mathrm{E}_s^{\sigma^*,\pi}[valpath_2].$$

In this case, we call $(\mathrm{E}_s^{\sigma^*,\pi^*}[valpath_1], \mathrm{E}_s^{\sigma^*,\pi^*}[valpath_2])$ an equilibrium value profile.

EXAMPLE 13. In figure 2 the Player strategy that returns the grocery cart and the uniquely determined System strategy are in a Nash equilibrium yielding value 0 for both. Indeed, this is a Pareto equilibrium that maximizes value for both participants.

Contrast against example 11 to see the benefits of rationality assumption for the System. ∎

3. Reasoning

We discuss three issues in this section: concrete construction of Nash equilibria, a logic for specifying properties and an approximate metric bisimulation reasoning principle.

3.1. Constructing Nash equilibria

Our approach is based on the explicit construction of *threat* strategies in repeated games. The concrete operational content of the proof is relevant to the implementation of system-level strategies.

Recall a basic result for zero-sum mean-payoff games.

THEOREM 14 ([30]). *For all zero-sum mean-payoff games, for all states s we have $val_1(s) + val_2(s) = 0$. Pure memoryless optimal strategies exist for both players in mean-payoff games.*

Given a nonzero-sum mean-payoff game $G = (G, r_1, r_2)$ consider two zero-sum mean-payoff games:

- $G_1 = (G, r_1, \bar{r}_1)$ where $\bar{r}_1(e) = -r_1(e)$ for all edges e.
- $G_2 = (G, \bar{r}_2, r_2)$ where $\bar{r}_2(e) = -r_2(e)$ for all edges e.

Let σ_1, $\bar{\pi}_1$ be any pure optimal strategies for player 1 and player 2, respectively, in G_1 (such optimal strategies exist by Theorem 14). Let $\bar{\sigma}_2$, π_2 be any pure optimal strategies for player 1 and player 2, respectively, in G_2. The strategies $\bar{\pi}_1$ and $\bar{\sigma}_2$ are threat strategies for player 2 and player 1, respectively. Consider strategy $(\sigma^*, \pi^*) = (\sigma_1 + \bar{\sigma}_2, \pi_2 + \bar{\pi}_1)$ described as follows: (a) player 1 follows strategy σ_1 as long as player 2 follows the strategy π_2, and as soon as player 2 deviates from π_2 player 1 switches to the threat strategy $\bar{\sigma}_2$; (a) player 2 follows strategy π_2 as long as player 1 follows the strategy σ_1, and as soon as player 1 deviates from σ_1 player 2 switches to the threat strategy $\bar{\pi}_1$. We argue the strategy (σ^*, π^*) is a Nash equilibrium. Observe that since the strategies σ_1 and π_2 are pure, any deviation of the strategies can be immediately observed by the other player. Consider any strategy for player 1: if player 1 follows σ_1, then she is guaranteed at least the value of the zero-sum game G_1 for all positions of the play (by optimality of σ_1), and if player 1 deviates then the threat strategy of player 2 ensures that player 1 gets no more than the value of zero-sum game G_1 from the point of deviation in the play. Hence player 1 has no incentive for unilateral deviation. Similar argument holds for player 2. This establishes that (σ^*, π^*) is a Nash equilibrium. Also (σ^*, π^*) have low-memory requirements: $O(n)$ for games with n states. Formally we have the following theorem.

THEOREM 15. *Given a nonzero-sum mean-payoff game $G = (G, r_1, r_2)$ consider two zero-sum games as follows:*

- *$G_1 = (G, r_1, \bar{r}_1)$ where $\bar{r}_1(e) = -r_1(e)$ for all edges e.*
- *$G_2 = (G, \bar{r}_2, r_2)$ where $\bar{r}_2(e) = -r_2(e)$ for all edges e.*

Let σ_1, $\bar{\pi}_1$ be any pure optimal strategies for player 1 and player 2, respectively, in G_1. Let $\bar{\sigma}_2$, π_2 be any pure optimal strategies for player 1 and player 2, respectively, in G_2. The strategy $(\sigma^*, \pi^*) = (\sigma_1 + \bar{\sigma}_2, \pi_2 + \bar{\pi}_1)$ is a Nash equilibrium for G.

3.2. Logic

We adapt a combination of the logic ATL* [8] and the probabilistic logics [13, 20] to address edge labels, edge costs and values. This logic is tuned to capturing worst-case properties: equilibrium properties can only be captured approximately.

In the presentation of this subsection, we rely heavily on prior research [8, 13] for background motivation and detailed examples. In this paper, limited by space constraints, we focus on the novelties.

Let $\bowtie \in \{=, <, \leq, >, \geq\}$, $r \in \mathbb{R}$, $q \in [0,1]$, $A \subseteq \{1,2\}$. The state (ϕ), path (ψ) and cost/probability formulas (η) are given by the following grammar:

$$\begin{array}{rcl}
\eta & ::= & \mathtt{true} \mid \neg\eta \mid \eta \vee \psi \mid v^1 \bowtie r \mid v^2 \bowtie r \mid \mathtt{prob} \bowtie q \\
\phi & ::= & \mathtt{true} \mid \neg\phi \mid \phi \vee \phi \mid \langle\langle A \rangle\rangle_\eta \psi \\
\psi & ::= & \mathtt{true} \mid l \mid \phi \mid \neg\psi \mid \psi \vee \psi \mid \bigcirc\psi \mid \psi\, U\, \psi
\end{array}$$

A path satisfies a path formula l if the first edge in the path has label l.

We illustrate strategy quantifiers and cost formulas by considering a concrete example. Let $\eta = v^1 \geq r_1 \wedge\ v^2 \geq r_2 \wedge \mathtt{prob} \geq q$. We use a path formula ψ to also stand for the set of paths that satisfy it (i.e. eliding semantic brackets). The strategy quantifier $\langle\langle 1 \rangle\rangle_\eta \psi$ is true at a state s if player 1 has a strategy σ such that for any strategy π for player 2:

- The probability of the paths resulting from (σ, π) satisfying ψ is at least q.
- $\mathrm{E}_s^{\sigma,\pi}[\psi[valpath_1]\ ] \geq r_1$: the expectation of player 1 value over the ψ-paths resulting from (σ, π) is $\geq\ r_1$.
- $\mathrm{E}_s^{\sigma,\pi}[\psi[valpath_2]\ ] \geq r_2$: the expectation of player 2 value over the ψ-paths resulting from (σ, π) is $\geq\ r_2$.

In the following examples, we freely use derived operators $\wedge$ (conjunction) and the LTL path connectives $\Box$ (always in the future) and $\Diamond$(eventually in the future) with traditional meanings.

EXAMPLE 16.

- In example 5, the buyer has a strategy to ensure that a bid is accepted: $\langle\langle Player \rangle\rangle_{\mathtt{prob}=1} \Diamond\, \mathtt{accept}$.
- In example 5, the game satisfies $\langle\langle System \rangle\rangle_{\mathtt{prob}=1}\ \Box[\mathtt{notpay} \Rightarrow\ (\Diamond\, \mathtt{reject}\, U\, \mathtt{pay})]$: the system has a strategy to ensure that it has the option of rejecting the next bid if the bidder did not pay the prior successful one.
- In example 9, the Player has a strategy to ensure that they never lose the coins deposited for getting a grocery cart: $\langle\langle Player \rangle\rangle_{v_{\mathtt{Player}} \geq 0}\, \mathtt{true}$.
- In example 9, the game only satisfies $\langle\langle System \rangle\rangle_{v_{\mathtt{System}} \leq r_{\mathtt{System}}(\mathtt{replacecart})} \mathtt{true}$: the System can only guarantee that they never lose more than the cost to replace the cart.

- In example 12, the game satisfies $\langle\langle System \rangle\rangle_{v_{\texttt{System}} \geq x}$ `true`, where x is the minimum of (a) the expected gain from Player caught cheating and (b) the System gain from the `debit` transitions.

∎

3.3. Alternating metric-bisimulation

In this subsection, we describe a coinductive reasoning principle to calculate how close two game-states are — to show that the distance between two states is less than ε, it suffices to produce a (metric) bisimulation that sets the distance between the states to be less than ε. We show that perturbing numerical values of costs yields a closeby game and show that closeby games have closeby optimal values and equilibria values.

As a technical warmup, we begin by defining a game version of bisimulation. The definition combines and adapts the definitions for games [9] and labeled Markov processes [20]. Given an equivalence relation R on the state set, and two probability distributions P_1, P_2, we say $P_1 \; R \; P_2$ if for all U such that $\{s \mid (t,s) \in R, t \in U\} \subseteq U$, it is the case that $P_1(U) = P_2(U)$.

DEFINITION 17. Let $G = ((S,E),(S_1,S_2,S_{1\bigcirc},S_{2\bigcirc}),\delta)$. An equivalence relation $R \subseteq [\cup_i S_i] \times [\cup_i S_i]$ is a *bisimulation* if for all $i \in \{1,2\}$:

- $s,t \in S_i, s \; R \; t \Rightarrow \; (\forall (s,l,s') \in E) \; (\exists (t,l,t') \in E) \; \delta(s') \; R \; \delta(t')$
- $s \in S_i, t \in S_{\bar{i}}, s \; R \; t \Rightarrow$
    - $(\exists (s,l,s') \in E) \Rightarrow \; (\exists (t,l,t') \in E)$
    - $(\forall (s,l,s') \in E) \; (\forall (t,l,t') \in E) \; \delta(s') \; R \; \delta(t')$

∎

The second case of the definition permits us to potentially equate player 1 and player 2 states. There is a maximum bisimulation, that we denote $\approx$.

EXAMPLE 18. Bisimulation is sound for logic. If $s \approx t$, then for all ϕ, s satisfies ϕ iff t satisfies ϕ. ∎

Definition 17 is very sensitive to perturbations of numerical values: a small change in numbers yields an inequivalent process. So, we describe an approximate approach based on distances between states. We model distances standardly as pseudometrics [4].

Given a pseudometric d on the state set and given two probability distributions P_1, P_2, the Wasserstein distance, written $d(P_1,P_2)$, lifts the metric to the space of probability distributions on the statespace[5].

If $(s,l,s'),(t,l,t') \in E$, we write $d((s,l,s'),(t,l,t')) = \max(d(\delta(s'),\delta(t')), |r_1(s,l,s') - r_1(t,l,t')|, |r_2(s,l,s') - r_2(t,l,t')|)$, thus accounting for both the distance from the probabilities and the distance caused by difference of costs.

The following definition parallels definition 17, roughly replacing every "equality" of that definition by "atmost ε".

DEFINITION 19. Let $G = ((S,E),(S_1,S_2,S_{1\bigcirc},S_{2\bigcirc}),\delta)$. A metric d on $\cup_i S_i$ is a *metric-bisimulation* if if $d(s,t) < \varepsilon$ implies for all $i \in \{1,2\}$:

- $s,t \in S_i \Rightarrow \; (\forall (s,l,s') \in E) \; (\exists (t,l,t') \in E) \; d((s,l,s'),d(t,l,t')) < \varepsilon$
- $s \in S_i, t \in S_{\bar{i}} \Rightarrow \; (\forall (s,s') \in E)$
    - $(\exists (s,l,s') \in E) \Rightarrow \; (\exists (t,l,t') \in E)$
    - $(\forall (t,l,t') \in E) \; d((s,l,s'),d(t,l,t')) < \varepsilon$

∎

There is a minimum metric[6], that we denote M.

The close correspondence between definitions 19 and 17 permits us to use analogues of traditional methods to reason about metric distances. For example, to deduce that two states are equivalent, it suffices to produce a bisimulation that relates the states. Similarly, to show that the distance between two states is less than ε, it suffices to produce a (metric) bisimulation that sets the distance between them to be less than ε.

EXAMPLE 20. Small perturbations in costs yield small distances. Let $G = (G,r_1,r_2)$. Let $G' = (G,r'_1,r'_2)$, where for all $e \in E$, for all $i \in 1,2$, $|r_i(e) - r'(e)| < \varepsilon$. Writing s_G (resp. $s_{G'}$) for the copies of state s in G (resp. G'), we deduce for all states s: $M(s_G, s_{G'}) \leq \varepsilon$ via the following metric-bisimulation d:

- $d(s_G, s_{G'}) = d(s_{G'}, s_G) = \varepsilon$
- $d(t,t') = d(t',t) = \infty$, if $\neg(\exists s)\{t,t'\} = \{s_G\}$

∎

Similarly, small perturbations in probabilities of transitions not involved in loops yield small distances (see [23]).

EXAMPLE 21. Bisimulation = 0 distance. For any metric-bisimulation d, it is the case that $d^{-1}(0)$ is a bisimulation. More generally: $\approx = \{(s,t) \mid M(s,t) = 0\}$. ∎

[4] A pseudometric d on a state space S is a function $S \times S \rightarrow \texttt{Reals}$ such that: $d(x,x) = 0$, $d(x,y) = m(y,x)$, $d(x,z) \leq m(x,y) + m(x,z)$.

[5] Due to space constraints, we refer the reader to [23] for a review of relevant probability theory in the context of coinductive definitions.

[6] For pseudometrics d_1, d_2, $d_1 \leq d_2$ if for all states s, $d_1(s) \leq d_2(s)$.

EXAMPLE 22. Closeby states have closeby logical properties. The key case follows — a more complete treatment including probability variations is elided in the interests of space.

Let η be a cost formula in negation normal form, with only value constraints[7]. For $\varepsilon > 0$, let η^{ε} be the ε enlargement of η: e.g., replace $v^1 \geq r$ (resp. $v_1 \leq r$) by $v_1 \geq r - \varepsilon$ (resp. $v_1 \leq r + \varepsilon$), and extend this homomorphically over disjunctions and conjunctions of such η formulas. Let ψ be a path formula whose only η sub-formula is `true`.

Then, if s satisfies $\langle\langle A \rangle\rangle_{\eta}\psi$ and $M(s,t) < \varepsilon$, then, t satisfies $\langle\langle A \rangle\rangle_{\eta^{\varepsilon}}\psi$. ∎

EXAMPLE 23. Closeby states have closeby quantitative properties. Let $G = (G, r_1, r_2)$. Let $M(s,t) < \varepsilon$. Then:

- (Worst case reasoning:) $|val_1(s) - val_1(t)| < \varepsilon, |val_2(s) - val_2(t)| < \varepsilon$
- (Rational case reasoning:) For every equilibrium value profile (v_1, v_2) at s, there is an equilibrium value profile (v'_1, v'_2) at t such that for $i = 1,2$, $|v_i - v'_i| < \varepsilon$.

∎

Proofs of all the above examples rely on showing local coinductive matching between the strategies from the states using metric bisimulation.

4. Game Algebra

In this section, we describe operators (synchronous product, restriction, sequencing, iteration, player choice, probabilistic choice, and tensor) to build up composite games from simpler games. In the special case of transition systems, all but the tensor operator specialize to the (familiar) one of the same name. Tensor is spatial conjunction that corresponds to interleaving of transition systems. In some cases, e.g., choice and tensor, the games definitions have subtle flavors without direct analogues in transition systems.

In some of the following constructions, we make assumptions about which player is to make the starting move, and/or which player is to move at blocked states. If a game doesn't already satisfy these assumptions, they can be met by adding (conceptually redundant) moves.

4.1. Choice

The choice operator $\oplus_i$ enables player i to choose between games with player i start states. The definition adapts process algebraic choice with the cost structure inherited from the cost structure for the underlying component games.

DEFINITION 24 (CHOICE OF ROOTED GAME GRAPHS). Let $G^A = ((S^A, E^A), (S^A_1, S^A_2, S^A_{1\bigcirc}, S^A_{2\bigcirc}), \delta^A)$ and $G^B = ((S^B, E^B), (S^B_1, S^B_2, S^B_{1\bigcirc}, S^B_{2\bigcirc}), \delta^B)$ be rooted game graphs with start states $s^A \in S^A_i, s^B \in S^B_i$. Then, $G^A \oplus_i G^B = ((S,E), (S_1, S_2, S_{1\bigcirc}, S_{2\bigcirc}), \delta)$ with start state $\langle s^a, s^B \rangle \in S_i$ is defined as:

- $S_i = S^A_i \uplus \; S^B_i \uplus \{\langle s^A, s^B \rangle\}$. $S_{\bar{i}} = S^A_{\bar{i}} \uplus \; S^B_{\bar{i}}$. For $i \in \{1,2\}$ $S_{i\bigcirc} = S^A_{i\bigcirc} \uplus \; S^B_{i\bigcirc}$
- $E = E^A \uplus E^B \uplus \{(\langle s^A, s^B \rangle, l, t) \mid (s^A, l, t) \in E^A \vee \; (s^B, l, t) \in E^B\}$
- $\delta = \delta^A \uplus \delta^B$

∎

DEFINITION 25. Let $G^A = (G^A, r^A_1, r^A_2)$ and $G^B = (G^B, r^B_1, r^B_2)$ be rooted mean-payoff game graphs with start states $s^A \in S^A_i, s^B \in S^B_i$. $G^A \oplus_i G^B$ is defined as follows:

- $G = G^A \oplus_i G^B$
- For $j = 1,2$:

$$r_j(e) = \begin{cases} r^A_j(e), \text{ if } e \in E^A \\ r^A_j((s^A, l, t)), \text{ if } e = (\langle s^A, s^B \rangle, l, t) \wedge \; (s^A, l, t) \in E^A \\ r^B_j(e), \text{ if } e \in E^B \vee \; e = (\langle s^A, s^B \rangle, t) \wedge \; (s^B, t) \in E^B \\ r^B_j((s^B, l, t)), \text{ if } e = (\langle s^A, s^B \rangle, l, t) \wedge \; (s^B, l, t) \in E^B \end{cases}$$

∎

EXAMPLE 26. In the highway tolling example 10, we have described a game (say $G^{\texttt{fastlane}}$), with costs and probabilities, for fastlanes. Let $G^{\texttt{regular}}$ be a game specification of a standard access control policy for a regular lane: wait for suitable coins from the Player before granting access (we elide the details in the interests of space).

A game specification of a composite policy for a tollbooth, with Player choosing the lane type, is given by $G^{\texttt{tollbooth}} = G^{\texttt{fastlane}} \oplus_{\texttt{Player}} G^{\texttt{regular}}$. ∎

Choice of closeby games yields closeby games, i.e. the choice combinator does not expand or increase distances, as shown by a standard metric-bisimulation proof.

THEOREM 27. *If* $M(s^A_1, s^A_2) \leq \varepsilon_1$ *and* $M(s^B_1, s^B_2) \leq \varepsilon_2$, *then* $M(\langle s^A_1, t^A_1 \rangle \langle s^B_2, t^B_2 \rangle) \leq \varepsilon_1 + \varepsilon_2$.

In the case when $\varepsilon_1 = \varepsilon_2 = 0$, this yields that bisimulation is a congruence for choice.

Choice is asymmetric between the players.

- $\langle\langle i \rangle\rangle_{\eta}\psi$ is true for the resulting game if it is true in either game. On the other hand, $\langle\langle \bar{i} \rangle\rangle_{\eta}\psi$ is true for the resulting game only if it is true in both games.
- The value for player i is the maximum of the values for the individual games. The value for player $\bar{i}$ is the minimum of the values for the individual games.

[7] i.e. given in the following restricted grammar: $\eta ::= \eta \wedge \; \eta \mid \eta \vee \eta \mid v^1 \bowtie \; r \mid v^2 \bowtie r$.

- An equilibrium profile $(v_i, v_{\bar{i}})$ of a component game is an equilibrium profile of the choice game iff there is no other equilibrium profile in the other component with a higher value for player i.

When applied to example 26, the first two items above yield that the Player has strategies to choose the access lane that minimizes her cost; System on the other hand is only guaranteed the minimum of the payments yielded by the two options.

4.2. Synchronous parallel composition

The synchronous product is used to build conjunctions of policies that share the same history of interaction.

The definition assumes that both games are started by player i. The construction of the product probability distribution in the definition corresponds to treating the two underlying random variables as independent, so the probabilities are multiplied. An edge in the synchronous product is a pair of edges, one each from the two component games. The labels used in the synchronous product game are products of labels from the individual games. The cost function on edges is the sum of the cost function on the two edges constituting the pair.

DEFINITION 28. [Synchronous Product]
Let $G^A = ((S^A, E^A), (S^A_1, S^A_2, S^A_{1\bigcirc}, S^A_{2\bigcirc}), \delta^A)$ and $G^B = ((S^B, E^B), (S^B_1, S^B_2, S^B_{1\bigcirc}, S^B_{2\bigcirc}), \delta^B)$ be two rooted game graphs with start states $s^A \in S^A_i, s^B \in S^B_i$. Then, the product game graph $G^A \times G^B = ((S, E), (S_1, S_2, S_{1\bigcirc}, S_{2\bigcirc}), \delta)$ is defined as follows:

- For $i \in \{1,2\}$, $S_i = S^A_i \times S^B_i$, $S_{i\bigcirc} = S^A_{i\bigcirc} \times S^B_{i\bigcirc}$
- For all $\langle s^A, s^B\rangle \in S$: $(\langle s^A, s^B\rangle, \langle l^A, l^B\rangle, \langle t^A, t^B\rangle) \in E \Leftrightarrow (s^A, l^A, t^A) \in E^A \wedge (s^B, l^B, t^B) \in E^B$
- For all $\langle s^A, s^B\rangle \in S_{1\bigcirc} \cup S_{2\bigcirc}$: $\delta(\langle s^A, s^B\rangle)(\langle t^A, t^B\rangle) = \delta(s^A, t^A) \star \delta(s^B, t^B)$

▮

DEFINITION 29. Let $G^A = (G^A, r^A_1, r^A_2)$ and $G^B = (G^B, r^B_1, r^B_2)$ be two rooted mean-payoff games with start states $s^A \in S^A_i, s^B \in S^B_i$. The product mean-payoff game $G = (G, r_1, r_2)$ is defined as follows:

- $G = G^A \times G^B$
- For $j \in \{1,2\}$, for $e = \langle s^A, s^B\rangle, \langle l^A, l^B\rangle, \langle t^A, t^B\rangle \in E$, $r_j(e) = r^A_j(s^A, l^A, t^A) + r^B_j(s^B, l^B, t^B)$.

▮

As in choice, a simple metric-bisimulation proof shows the analogue of Theorem 27 for synchronous product.

The only way for a strategy in one component in the synchronous product to affect the strategy in the other component is by being a blocking state and not having any transitions. We formalize this below. Given a strategy pair (σ^A, π^A) for a rooted game graph G^A with start state s (similarly (σ^B, π^B) for rooted game graph G^B with start state t), we say they are *compatible* if all maximal paths in both $\omega_s^{\sigma^A,\pi^A}$ and $\omega_t^{\sigma^B,\pi^B}$ are of the same length, i.e. they consist only of infinite paths or all maximal paths are finite and of the same length[8]. In this case,

$$\mathrm{E}^{\langle\sigma^A,\sigma^B\rangle,\langle\pi^A,\pi^B\rangle}_{\langle s,t\rangle}[valpath_1] = \mathrm{E}^{\sigma^A,\pi^A}_s[valpath_1] + \mathrm{E}^{\sigma^B,\pi^B}_s[valpath_1]$$

Similarly for $valpath_2$. This enables us to deduce that values (resp. equilibrium value profiles) in the composite game are sums of values (resp. pointwise sum of equilibrium value profiles) in the component games when the compatibility assumptions are met.

Other process algebraic combinators. Given choice and synchronous product, the definitions for probabilistic choice, sequential composition, restriction and iteration are routine. Probabilistic choice, sequential composition and restriction satisfy analogues of Theorem 27. However, iteration does not — intuitively, since iteration involves potential repeated use, small differences can add up.

In the main text, we content ourselves with an example that uses several of these combinators to construct a game that combines different mechanisms of control.

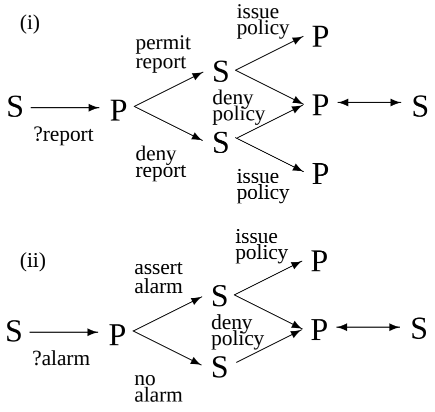

Figure 4. Insurance: Policy Issue

EXAMPLE 30. Arranging a home insurance policy requires up-front checks, in the access control tradition (e.g., CLUE reports for US home insurers) before a policy is issued. Some checks require permission from the applicant. G^{CLUE}, the game for this procedure appears in Figure 4(i). If the insurer denies a policy to the applicant, the parties no longer

[8] This is automatically satisfied by the traditional games that have no blocking states.

have an useful interaction, modelled as a loop between the Player and System.

Insurers also rely upon unverified information, e.g., an insurer may not verify the existence of a fire alarm asserted by the applicant. $G^{assert-alarm}$, the game in Figure 4(ii), describes such a procedure in which the insurer refuses to issue a policy unless the Player claims to have an alarm. The entire preissue game, $G^{\texttt{preissue}}$ is:

$$(\nu\langle\texttt{deny},\texttt{accept}\rangle)\;(\nu\langle\texttt{accept},\texttt{deny}\rangle)[G^{\texttt{CLUE}} \times G^{assert-alarm}]$$

In $G^{\texttt{preissue}}$, the combination of synchronous product and restriction achieves the effect of synchronizing on accept/deny. The product in this game satisfies the hypothesis of the compositional techniques alluded to earlier.

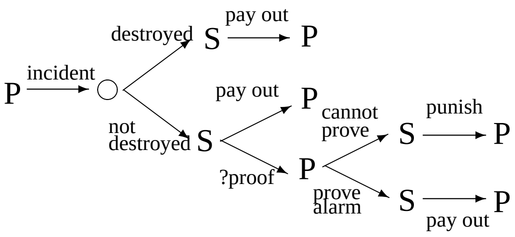

Figure 5. Insurance: Claim reimbursement

$G^{\texttt{incident}}$, of Figure 5, addresses the interaction after an incident. The edges `punish` (and `payout`) has associated Player cost (resp. benefit) and System benefit (resp. cost). The probabilistic state (shown as a circle) models whether, after the incident, it is possible to check that a fire alarm was present. If so, the insurer punishes the policy holder if no evidence for a fire-alarm is found. The System strategy that chooses the `?proof` transition in $G^{\texttt{incident}}$ reduces the (expected) risk of payout to dishonest Players to the product of the `destroyed` probability and the payout costs.

The full game combining access-control based issue and audit-based reimbursement is: $G^{\texttt{preissue}};G^{\texttt{incident}}$. In this game, at each state after a policy has been issued, the applicant has a strategy that leads to a pay out and avoids punishment, i.e. $\langle\langle Player\rangle\rangle_{\texttt{prob=1}}(\Diamond\texttt{issuepolicy} \Rightarrow \Box[\neg\texttt{punish} \wedge \Diamond\ \texttt{payout}])$. ∎

4.3. Tensor conjunction

The aim of tensor is to combining policies on multiple resources. The tensor achieves this by partitioning the System/Player interaction into the individual components. None of the other combinators yield the flexible temporal overlap of the component games permitted by tensor — the product shares the same Player/System interaction between the two component games; choice forces the selection of one of the two component games, and sequential composition orders the two component games.

The tensor game graph yields interleavings of the evolutions of the game graphs of the two component games. The restriction placed on the interleaving is that in $G^A \otimes_{\bar{i}} G^B$, only player i gets to shift between the two games.

DEFINITION 31. [Tensor of game graphs]
Let $G^A = ((S^A,E^A),(S_1^A,S_2^A,S_{1\bigcirc}^A,S_{2\bigcirc}^A),\delta^A)$ and $G^B = ((S^B,E^B),(S_1^B,S_2^B,S_{1\bigcirc}^B,S_{2\bigcirc}^B),\delta^B)$ be two rooted game graphs with start states $s^A \in S_i^A, s^B \in S_i^B$. Then, the rooted tensor game graph $G^A \otimes_{\bar{i}} G^B = ((S,E),(S_1,S_2,S_{1\bigcirc},S_{2\bigcirc}),\delta)$ is defined as follows:

States: Start state is $\langle s^A,s^B\rangle$

- $S_i = \{\langle t^A,t^B\rangle \mid t^A \in S_i^A, t^B \in S_i^B\}$
- $S_{\bar{i}} = \{\langle t^A,t^B\rangle \mid t^A \in S_i^A, t^B \in S_{\bar{i}}^B\} \cup \{\langle t^A,t^B\rangle \mid t^B \in S_i^B, t^A \in S_{\bar{i}}^A\}$
- $S_{i\bigcirc} = \{\langle t^A,t^B\rangle \mid t^A \in S_{i\bigcirc}^A, t^B \in S_i^B\} \cup \{\langle t^A,t^B\rangle \mid t^A \in S_i^A, t^B \in S_{i\bigcirc}^A\}$
- $S_{\bar{i}\bigcirc} = \{\langle t^A,t^B\rangle \mid t^A \in S_{\bar{i}\bigcirc}^A, t^B \in S_i^A\} \cup \{\langle t^A,t^B\rangle \mid t^A \in S_i^A, t^B \in S_{\bar{i}\bigcirc}^A\}$

Edges: $E = \{(\langle t^A,t_1^B\rangle, l^B, \langle t^A,t_2^B\rangle) | (t_1^B,l^B,t_2^B) \in E^B\} \cup \{(\langle t_1^A,t^B\rangle, l^A, \langle t_2^A,t^B\rangle) | (t_1^A,l^A,t_2^A) \in E^A\}$

Distributions:

- $\delta(\langle t^A,t^B\rangle) = \delta^A(t^A)$ if $t^A \in S_{1\bigcirc}^A \cup S_{2\bigcirc}^A$.
- $\delta(\langle t^A,t^B\rangle) = \delta^B(t^A)$ if $t^B \in S_{1\bigcirc}^B \cup S_{2\bigcirc}^B$.

∎

DEFINITION 32. Let $G^A = (G^A,r_1^A,r_2^A)$ and $G^B = (G^B,r_1^B,r_2^B)$ be two rooted mean-payoff game graphs with start states $s^A \in S_i^A, s^B \in S_i^B$. The mean-payoff game $G = (G,r_1,r_2) = (G^A,r_1^A,r_2^A) \otimes_{\bar{i}} (G^B,r_1^B,r_2^B)$ is defined as follows:

- $G = G^A \otimes_{\bar{i}} G^B$
- For $j \in \{1,2\}$:
  - $r_j(\langle t^A,t_1^B\rangle, l^B, \langle t^A,t_2^B\rangle) = r_j^B(t_1^B,l^B,t_2^B)$
  - $r_j(\langle t_1^A,t^B\rangle, l^A, \langle t_2^A,t^B\rangle) = r_j^A(t_1^A,l^A,t_2^A)$

∎

Each edge in the tensor game graph arises from an edge in one of the two component game graphs. So, the cost function is directly inherited from the cost functions of the component games. The sum of costs on a path for either player is the sum of the projections of the path on the component games — this statement is of course not true for the limit average definition of values of paths.

EXAMPLE 33. Example 9 describes a game specification, say $G^{\texttt{cart}}$, for a grocery cart. Let $G^{\texttt{checkout}}$ be a game that specifies a policy for checkout at the grocery store (this could be a standard access control policy whose details we elide in the interest of space).

The policy that appropriately combines the two games is given by $G^{\texttt{cart}} \otimes_{\text{System}} G^{\texttt{checkout}}$. The interleaving and Player switching supported by $\otimes_{\text{System}}$ permits the Player to construct the desired interleaving of the two games, namely pick a cart (from $G^{\texttt{cart}}$), checkout (from $G^{\texttt{checkout}}$) and return cart (from $G^{\texttt{cart}}$). ∎

As in choice, a simple metric-bisimulation proof shows the analogue of Theorem 27 for tensor.

A pair of strategies (say π_1, π_2) from the component games yields a strategy, say $\langle \pi_1, \pi_2 \rangle$, for player $\bar{i}$ in the tensor game. For player i, a strategy in the tensor game can be built up from strategies (say σ_1, σ_2) for the component games, and an interleaving strategy between the components.

With a fair interleaving strategy by player i, the values for players resulting from playing $\langle \sigma_1, \sigma_2 \rangle$ against $\langle \pi_1, \pi_2 \rangle$ can be calculated when values of paths are calculated in terms of total costs, rather than limit averages, as follows. For a path $\omega = s_0, l_0, s_1, l_1, s_{i+1}, \ldots$, let $valpath_1(\omega) = \sum r_1((s_i, l_i, s_{i+1}))$; $valpath_2(\omega) = \sum r_2((s_i, l_i, s_{i+1}))$. In this case, the interaction yields value $\mathrm{E}_{s^A}^{\sigma_1,\pi_1}[valpath_1] + \mathrm{E}_{s^B}^{\sigma_2,\pi_2}[valpath_1]$ for player 1 and value $\mathrm{E}_{s^A}^{\sigma_1,\pi_1}[valpath_2] + \mathrm{E}_{s^B}^{\sigma_2,\pi_2}[valpath_2]$ for player 2.

In example 33, this yields that the way for Player to maximize value (for both worst case and rational assumptions) is to use the optimal strategies for the $G^{\texttt{checkout}}$ and $G^{\texttt{cart}}$ games. Unfortunately, we are not aware of a similar simple characterization in the standard calculation of values based on limit averages.

Unfair interleaving strategies by player i can lead to asymmetry: disjunctive for player i and conjunctive for the other, similar to choice. For example, let ψ be a path formula that is not reliant on reaching blocked states[9]. For such a formula, player i has a strategy to achieve ψ in the tensor game (i.e. $\langle\langle i \rangle\rangle_{\texttt{prob}\geq 1}\psi$ is true at the start state) if she has a strategy to achieve ψ in either of the component games. On the other hand, $\langle\langle \bar{i} \rangle\rangle_{\texttt{prob}\geq 1}\psi$ is true for the tensor game only if player $\bar{i}$ has a strategy to achieve ψ in both component games.

5. Conclusion

The need for controls arises when the owner of a resource has to share it with other parties. The requirements on these controls depend on the underlying architectural assumptions: the model of the requestor (worst case vs. rational), owner objectives (absolute correctness vs risk management) and when are controls exercised (before, during or after the access to the resource). The motivation for this paper is that several useful applications require a mixture of different kinds of controls.

We have argued that games provide a unified framework to address this issue. Our formalization shows that games provide good composition mechanisms. Several small examples illustrate the benefits derived for applications of interest.

We have developed formal results to analyze games. A key feature of our analysis methods is the explicit concession to approximate reasoning with quantitative information: our methods are robust with respect to perturbations of numerical values.

Extant research provides automated analysis methods for important subcases: e.g., probabilistic systems, qualitative games. For example, MOCHA [1] provides automated analysis of deterministic games with ATL specifications and TICC [2] provides support for interface compatibility and composition using symbolic game algorithms. However, a suite of algorithms to automate analysis for all the games of interest is as yet unavailable. This will be a topic of future work.

References

[1] Mocha. http://embedded.eecs.berkeley.edu/research/mocha/.

[2] Ticc. http://dvlab.cse.ucsc.edu/dvlab/Ticc.

[3] M. Abadi, A. Birrell, M. Burrows, F. Dabek, and T. Wobber. Bankable postage for network services. In V. A. Saraswat, editor, *ASIAN*, volume 2896 of *Lecture Notes in Computer Science*, pages 72–90. Springer, 2003.

[4] M. Abadi, A. Birrell, and T. Wobber. Access control in a world of software diversity. In *Proc. of the Tenth workshop on Hot Topics in Operating Systems*, 2005.

[5] M. Abadi and C. Fournet. Access control based on execution history. In *Proc. Network and Distributed System Security Symp.*, 2003.

[6] S. Abramsky and R. Jagadeesan. Games and full completeness for multiplicative linear logic. *J. Symb. Log.*, 59(2):543–574, 1994.

[7] S. Abramsky, R. Jagadeesan, and P. Malacaria. Full abstraction for PCF. *Inf. Comput.*, 163(2):409–470, 2000.

[8] R. Alur, T. Henzinger, and O. Kupferman. Alternating-time temporal logic. *Journal of ACM*, 49:672–713, 2002.

[9] R. Alur, T. A. Henzinger, O. Kupferman, and M. Y. Vardi. Alternating refinement relations. In D. Sangiorgi and R. de Simone, editors, *CONCUR*, volume 1466 of *Lecture Notes in Computer Science*, pages 163–178. Springer, 1998.

[10] R. Anderson. Why information security is hard - an economic perspective. In *Annual Computer Security Applications Conference (ACSAC)*, December 2001.

[11] S. Barker and P. J. Stuckey. Flexible access control policy specification with constraint logic programming. *ACM Trans. Inf. Syst. Secur.*, 6(4):501–546, 2003.

[9] This can be formalized by saying that ψ has at least one infinite path in its set of models.

[12] E. Bertino, P. A. Bonatti, and E. Ferrari. TRBAC: A temporal role-based access control model. *ACM Trans. Inf. Syst. Secur.*, 4(3):191–233, 2001.

[13] A. Bianco and L. de Alfaro. Model checking of probabilistic and nondeterministic systems. In P. S. Thiagarajan, editor, *FSTTCS*, volume 1026 of *Lecture Notes in Computer Science*, pages 499–513. Springer, 1995.

[14] B. Blakley. The emperor's old armor. In *NSPW '96: Proceedings of the 1996 workshop on New Security Paradigms*, pages 2–16, New York, NY, USA, 1996. ACM Press.

[15] A. Blass. A game semantics for linear logic. *Ann. Pure Appl. Logic*, 56(1-3):183–220, 1992.

[16] P. Bonatti, S. D. C. di Vimercati, and P. Samarati. An algebra for composing access control policies. *ACM Trans. Inf. Syst. Secur.*, 5(1):1–35, 2002.

[17] J. G. Cederquist, R. Corin, M. A. C. Dekker, S. Etalle, and J. I. den Hartog. An audit logic for accountability. In *POLICY*, pages 34–43. IEEE Computer Society, 2005.

[18] V. Danos and R. Harmer. Probabilistic game semantics. *ACM Trans. Comput. Log.*, 3(3):359–382, 2002.

[19] L. de Alfaro. Quantitative verification and control via the mu-calculus. In R. M. Amadio and D. Lugiez, editors, *CONCUR*, volume 2761 of *Lecture Notes in Computer Science*, pages 102–126. Springer, 2003.

[20] J. Desharnais, A. Edalat, and P. Panangaden. Bisimulation for labelled Markov processes. *Inf. Comput.*, 179(2):163–193, 2002.

[21] J. Desharnais, V. Gupta, R. Jagadeesan, and P. Panangaden. Metrics for labelled Markov processes. *Theor. Comput. Sci.*, 318(3):323–354, 2004. Preliminary paper appeared in the proceedings of CONCUR 1999.

[22] J. Filar and K. Vrieze. *Competitive Markov Decision Processes*. Springer, 1997.

[23] V. Gupta, R. Jagadeesan, and P. Panangaden. Approximate reasoning for real-time probabilistic processes. *Logical Methods in Computer Science*, 2(1), 2006.

[24] J. Y. Halpern and V. Weissman. Using first-order logic to reason about policies. In *CSFW '03: Proceedings of the 17th IEEE Computer Security Foundations Workshop (CSFW'03)*, pages 118–130. IEEE Computer Society, 2003.

[25] J. M. E. Hyland and C.-H. L. Ong. On full abstraction for PCF: I, II, and III. *Inf. Comput.*, 163(2):285–408, 2000.

[26] C.-C. Jou and S. A. Smolka. Equivalences, congruences, and complete axiomatizations for probabilistic processes. In J. C. M. Baeten and J. W. Klop, editors, *CONCUR*, volume 458 of *Lecture Notes in Computer Science*, pages 367–383. Springer, 1990.

[27] S. Kremer and J.-F. Raskin. Game analysis of abuse-free contract signing. In *Proc. of the 15th IEEE Computer Security Foundations Workshop*, pages 206–220, 2002.

[28] K. Krukow, M. Nielsen, and V. Sassone. A framework for concrete reputation-systems with applications to history-based access control. In *CCS 2005: Proceedings of the 12th ACM Conference on Computer and Communications Security*, pages 260–269, 2005.

[29] B. Lampson. Computer security in the real world. Invited talk at CyberTurst 2005, http://www.ics.uci.edu/~cybrtrst.

[30] T. A. Liggett and S. A. Lippman. Stochastic games with perfect information and time average payoff. *SIAM Review*, 11:604–607, 1969.

[31] P. Lincoln, J. C. Mitchell, M. Mitchell, and A. Scedrov. A probabilistic poly-time framework for protocol analysis. In *ACM Conference on Computer and Communications Security*, pages 112–121, 1998.

[32] A. Mahimkar and V. Shmatikov. Game-based analysis of denial-of-service prevention protocols. In *CSFW*, pages 287–301. IEEE Computer Society, 2005.

[33] C. Meadows. A cost-based framework for analysis of denial of service networks. *Journal of Computer Security*, 9(1/2):143–164, 2001.

[34] R. Parikh. The logic of games and its applications. In *Selected papers of the international conference on "foundations of computation theory" on Topics in the theory of computation*, pages 111–139, New York, NY, USA, 1985. Elsevier North-Holland, Inc.

[35] R. Parikh. Propositional game logic. In *Foundations of Computer Science*, pages 195–200, 1993.

[36] J. Park. *Usage control: a unified framework for next generation access control*. PhD thesis, School of Information Technology and Engineering, George Mason University, 2003.

[37] J. Park and R. S. Sandhu. The UCON_{ABC} usage control model. *ACM Trans. Information System Security*, 7(1):128–174, February 2004.

[38] M. Pauly. *Logic for Social Software*. PhD thesis, Institute for Logic, Language and Computation, Universiteit van Amsterdam, 2001. ILLC Dissertation Series 2001-10.

[39] D. Povey. Optimistic security: a new access control paradigm. In *NSPW '99: Proceedings of the 1999 workshop on New Security Paradigms*, pages 40–45, New York, NY, USA, 2000. ACM Press.

[40] T. Raghavan and J. Filar. Algorithms for stochastic games — a survey. *ZOR — Methods and Models of Operations Research*, 35:437–472, 1991.

[41] F. B. Schneider. Enforceable security policies. *ACM Trans. Inf. Syst. Secur.*, 3(1):30–50, 2000.

[42] F. Thuijsman. *Optimality and Equilibria in Stochastic Games*. CWI-Tract 82, CWI, Amsterdam, 1992.

[43] F. Thuijsman and T. Raghavan. Perfect information stochastic games and related classes. *International Journal of Game Theory*, 26:403–408, 1997.

[44] J. van Benthem and M. Pauly. ESSLLI course on logic and games. www.stanford.edu/~pianoman/GameLogic/esslli-course.html.

[45] J. von Neumann and O. Morgenstern. *Theory of Games and Economic Behavior*. New York: John Wiley and Sons, 1944.

[46] D. Wallach. *A New Approach to Mobile Code Security*. PhD thesis, Princeton University, 1999.

[47] D. Wallach, D. Balfanz, D. Dean, and E. Felten. Extensible security architectures for Java. In *Proceedings of the Sixteenth Symposium on Operating System Principles*, Saint-Malo, France, Oct 1997.

[48] D. Wijesekera and S. Jajodia. Policy algebras for access control — the predicate case. In *CCS '02: Proceedings of the 9th ACM conference on Computer and Communications Security*, pages 171–180. ACM Press, 2002.

[49] X. Zhang, J. Park, F. Parisi-Presicce, and R. Sandhu. A logical specification for usage control. In *SACMAT '04: Proceedings of the ninth ACM Symposium on Access Control Models and Technologies*, pages 1–10. ACM Press, 2004.