Payoffs, Intensionality and Abstraction in Games

Chris Hankin and Pasquale Malacaria
(with thanks to Dusko Pavlovic and Fabrizio Smeraldi)

Imperial College London and Queen Mary University of London

June 3, 2013

Timeline

- 1981ish: London University Inter-collegiate PhD course on Theoretical Computer Science
- 1985: *The theory of strictness analysis for higher-order functions*, Burn, Hankin and Abramsky
  polymorphic invariance, logical relations, strictness logics
- 1987: *Abstract Interpretation of Declarative Languages*, Abramsky and Hankin (eds)
- 1994: *Full Abstraction of PCF*, Abramsky, Jagadeesan and Malacaria
- 1999: *Non-deterministic Games and Program Analysis: An application to Security* Malacaria and Hankin
- 2013: Happy birthday Samson!

Outline

- Context
- Payoffs
- Abstraction
- Conclusions

Games and Abstraction

- about 10 years of General Sum Stochastic Games in cyber security (2 players)
- even simple models soon become intractable
- Objective: ad hoc simplifications to rigorous abstractions
- use of (probabilistic) abstract interpretation techniques
- longer term plan to address games with
  - imperfect information (players do not know each other's states), and
  - incomplete games (players do not know each other's strategies).

Some motivations:

- Game Theory is sometimes criticised for giving unrealistic results
- Game Theory (in Security as in Economics and Social Sciences etc) needs good payoffs,
- The "wrong" payoffs may give misleading results
- Desiderata about games:
  1. "good" payoffs
  2. abstractable game solutions e.g. solutions resilient to payoffs perturbations.

Payoffs are tricky

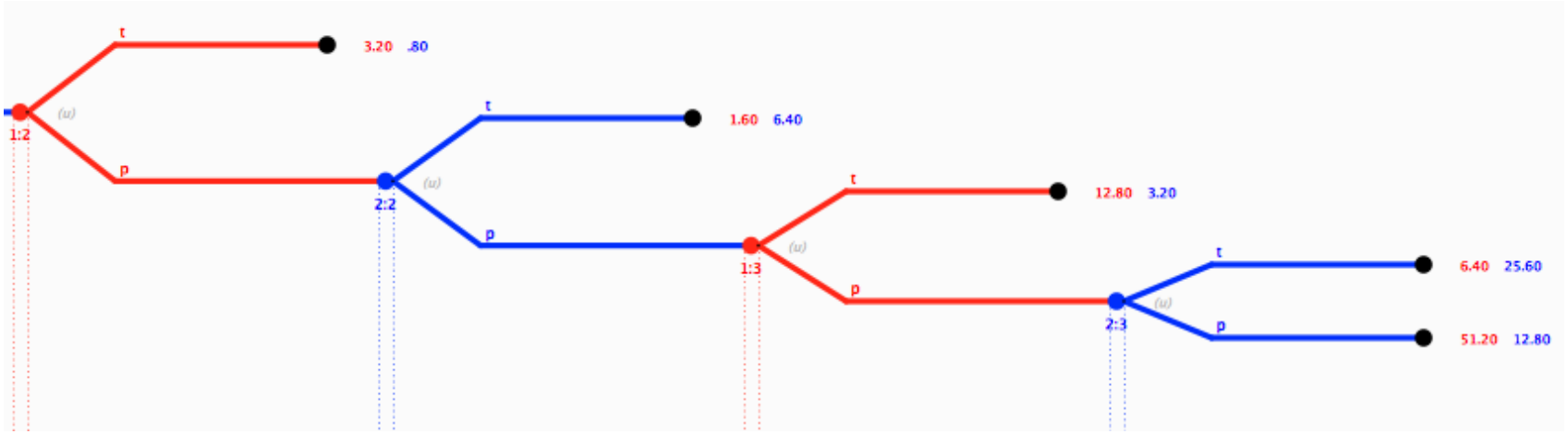

Figure: The Centipede game

- Game Theory tells players: stop at very beginning
- even if they could get arbitrary rich if they keep playing...
  1. experiments in the real world shows people keep playing. Are they smarter than Game Theory?
  2. but chess masters stops at the very beginning and stay poor....?!?!?

Payoffs are tricky

(A): so what's the problem?

real world people don't play the same game... they see different payoffs.... e.g. expectations on other player altruisms etc not expressed in the letter of the game; once payoffs fixed things work

A crypto-security related game:

- A player is given an odd number x and should decide whether x is prime or composite: correct guess gets \$2, incorrect guess gets \$-1000.The player can also choose not to play, and then gets \$1.
- Game solution=play, you always get \$2
- real life: only play if you can compute primality of x... (BITCOIN miners play this game every day)... GOTO (A)

Payoff time

How do we get *the* payoffs? e.g. what are the payoffs for the real world game of chess? for real world security?

- phishing or crypto-attacks?
  Game solutions: (naive payoffs) = crypto, (good payoffs) = phishing because crypto too costly
- in general how to model (as payoffs) complex (and ever changing) interaction between cyber-attacker and cyber-defender?
- computational complexity helps for crypto-attacks, in general intensional models of computation may help
- we revisited an intensional model (Game Semantics) in Game Theory terms: a taster in next slide

a taster: the system administrator dilemma

the prisoner dilemma revisited

a system decision (e.g. to give cash from a machine) depends on another system (bank authorization)

- in Game Semantics we are talking about the Game

$$((N \multimap B) \multimap B)$$

- resource manager controls the outer type, authentication controls subtype

$$(N \multimap B)$$

- cooperate=trust other system, defect=do not trust, e.g. refuse access. Game theoretical strategies can be combined with software modules...e.g. grim trigger trust policy

moving on: abstracting normal form games

	1		2		3	
1	2	2	-5	20	-5	22
2	-2	-1	0	-5	1	-7
3	-5	-5	10	-22	10	-20

Figure: A simple malware game with two similar attackers

- rows player (red) is the defender; actions= do nothing, alert, stop service
- columns player (blue) = honest user, malware , similar malware
- we see that the two pieces of malware are very similar in what they can do in terms of payoffs
- can we abstract them into one malware?

Abstractions

	1		2	
1	2	2	-5	21
2	-2	-1	0.5	-6
3	-5	-5	10	-21

Figure: The simple malware game abstracted

- rows player (red) is the defender; actions= do nothing, alert, stop service
- columns player (blue) = honest user, malware
- malware payoff=average of the malware(s) payoffs
- this transformation is a Moore-Penrose pseudo-inverse, providing the best-fit (least square error) for the original game

Abstractions

	1		2	
1	2	2	-5	21
2	-2	-1	0.5	-6
3	-5	-5	10	-21

Figure: The simple malware game abstracted

- the abstract and concrete game have similar equilibria
- so we can reason in the simpler scenario and get a good solution for the more complicated scenario
- But can we generalise?

Conclusions and future work

- we have seen some basic yet important questions, significant for "getting the right" model,
- We need to consider whether stochastic game models are the correct formalism – Stackelberg games may capture the asymmetry of cyber security better,
- We may need to model resources explicitly
- We look forward to apply soon some of these ideas on real data