Logic and Action

Jan van Eijck[1],[2]

[1]Centrum Wiskunde & Informatica (CWI)
Amsterdam

[2]Institute for Logic, Language and Computation (ILLC)
Amsterdam

AUC, May 19, 2016

Abstract

An action is something that takes place in the world, and that makes a difference to what the world looks like. Thus, actions are maps from states of the world to new states of the world. Actions can be of various kinds. The action of spilling coffee changes the state of your trousers. The action of telling a lie to your friend changes your friend's state of mind (and maybe the state of your soul). The action of multiplying two numbers changes the state of certain registers in your computer. Despite the differences between these various kinds of actions, we will see that they can all be covered under the same logical umbrella.

Outline

Modelling Actions and Reasoning About Actions

Hoare Logic and Propositional Dynamic Logic

Strategic Games: the Prisoner’s Dilemma

Key Notions: Best Response, Nash Equilibrium

Voting as a Multi-Agent Game

Group Strategies in Games

Extending PDL to Analyze Strategies in Games

Action in ‘Logic in Action’

`http://www.logicinaction.org/docs/ch6.pdf`

[BvDvEJ13]

The Pebble Puzzle

An urn cointains 70 pebbles; 35 of them are white and 35 are black. There is a pile of black pebbles available outside the urn.

Pebble Algorithm

- While there are still enough pebbles in the urn:
    - pick two pebbles;
      if they have the same colour, put back a black pebble
      otherwise, put back the white pebble.

In every step of the algorithm one pebble gets removed. After 69 steps, there is one pebble left. What is its colour?

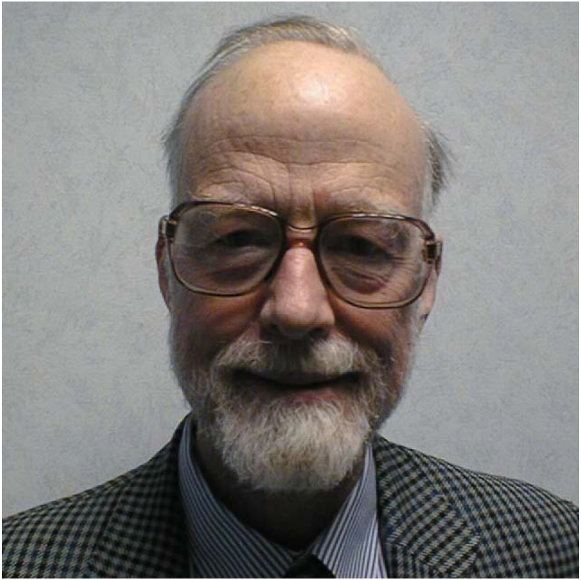
Pebble Algorithm as a Program

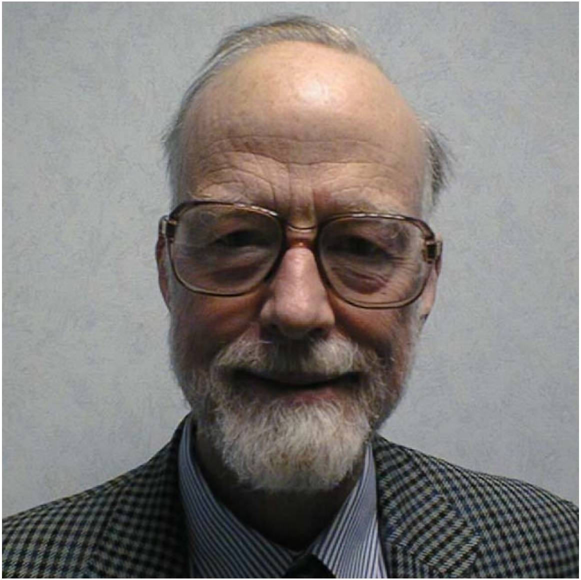
```
module Pebbles where
data Color = W | B deriving (Eq,Show)

drawPebble :: [Color] -> [Color]
drawPebble [] = []
drawPebble [x] = [x]
drawPebble (W:W:xs) = drawPebble (B:xs)
drawPebble (B:B:xs) = drawPebble (B:xs)
drawPebble (W:B:xs) = drawPebble (W:xs)
drawPebble (B:W:xs) = drawPebble (W:xs)

numberW, parityW :: [Color] -> Int
numberW = length . (filter (\x -> x == W))
parityW xs =  mod (numberW xs) 2

prop_invariant = \xs ->
  parityW xs == parityW (drawPebble xs)
```





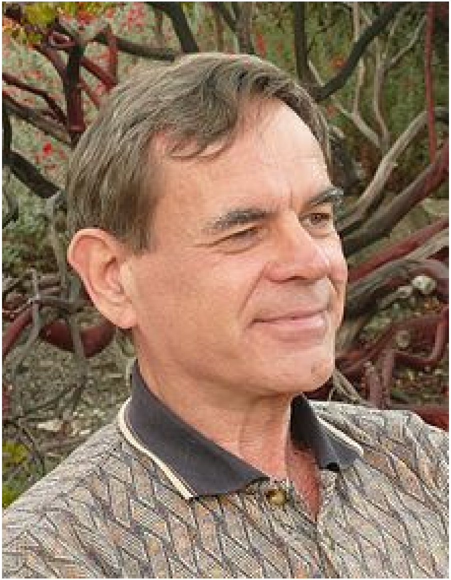
Sir Tony Hoare

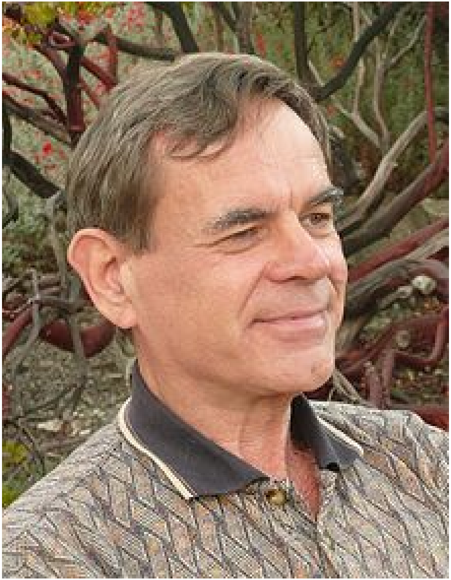
Formal Specification With Hoare Triples

- $\{P\}\ S\ \{Q\}$
- $\{$ **initial state** $\}$ **statement** $\{$ **final state** $\}$
- If execution of S in a state that satisfies P terminates, then the termination state is guaranteed to satisfy Q.
- Triples $\{P\}\,S\,\{Q\}$ are called *Hoare triples* after Tony Hoare [Hoa69].
- The predicate for the initial state is called the *precondition*, and the predicate for the final state is called the *postcondition*.

Hoare Calculus

assignment	$\dfrac{}{\{\phi^v_a\}\ v := a\ \{\phi\}}$
skip	$\dfrac{}{\{\phi\}\ \text{SKIP}\ \{\phi\}}$
sequence	$\dfrac{\{\phi\}\ C_1\ \{\psi\} \qquad \{\psi\}\ C_2\ \{\chi\}}{\{\phi\}\ C_1 ; C_2\ \{\chi\}}$
conditional choice	$\dfrac{\{\phi \wedge B\}\ C_1\ \{\psi\} \qquad \{\phi \wedge \neg B\}\ C_2\ \{\psi\}}{\{\phi\}\ \text{ if } B \text{ then } C_1 \text{ else } C_2\ \{\psi\}}$
guarded iteration	$\dfrac{\{\phi \wedge B\}\ C\ \{\phi\}}{\{\phi\}\ \text{ while } B \text{ do } C\ \{\phi \wedge \neg B\}}$
precondition strengthening	$\dfrac{\mathbb{N} \models \phi' \to \phi \qquad \{\phi\}\ C\ \{\psi\}}{\{\phi'\}\ C\ \{\psi\}}$
postcondition weakening	$\dfrac{\{\phi\}\ C\ \{\psi\} \qquad \mathbb{N} \models \psi \to \psi'}{\{\phi\}\ C\ \{\psi'\}}$





Vaughan Pratt

Hoare Logic as a Fragment of Dynamic Logic

- Hoare logic is a fragment of a more general system of (propositional) dynamic logic.
- The language of propositional dynamic logic was defined by Pratt in [Pra76, Pra80] as a generic language for reasoning about computation.
- Axiomatisations were given independently by Segerberg [Seg82], Fisher/Ladner [FL79], and Parikh [Par78].
- These axiomatisations make the connection between propositional dynamic logic and modal logic very clear.

PDL Language

- Let p range over the set of basic propositions P, and let a range over a set of basic actions A.
- The formulae ϕ of propositional dynamic logic are given by:

$$\phi \quad ::= \quad \top \mid p \mid \neg\phi \mid \phi_1 \vee \phi_2 \mid \langle\alpha\rangle\phi$$

- The programs α of propositional dynamic logic are given by:

$$\alpha \quad ::= \quad a \mid ?\phi \mid \alpha_1 ; \alpha_2 \mid \alpha_1 \cup \alpha_2 \mid \alpha^*$$

- Abbreviation:

$[\alpha]\phi$ abbreviates $\neg\langle\alpha\rangle\neg\phi$.

Expressing Hoare Triples in PDL

- Floyd-Hoare correctness assertions are expressible in PDL.
- If ϕ, ψ are PDL formulae and α is a PDL program, then

  $$\{\phi\}\ \alpha\ \{\psi\}$$

  translates into

  $$\phi \to [\alpha]\psi.$$
- $\{\phi\}\ \alpha\ \{\psi\}$ holds in a state in a model iff $\phi \to [\alpha]\psi$ is true in that state in that model.

PDL Axiomatisation

Aioms are all propositional tautologies, plus the following axioms (we give box ($[\alpha]$)versions here, but every axiom has an equivalent diamond ($\langle\alpha\rangle$) version):

$$\begin{array}{rll}
\text{(K)} & \vdash & [\alpha](\phi \to \psi) \to ([\alpha]\phi \to [\alpha]\psi) \\
\text{(test)} & \vdash & [?\phi_1]\phi_2 \leftrightarrow (\phi_1 \to \phi_2) \\
\text{(sequence)} & \vdash & [\alpha_1 ; \alpha_2]\phi \leftrightarrow [\alpha_1][\alpha_2]\phi \\
\text{(choice)} & \vdash & [\alpha_1 \cup \alpha_2]\phi \leftrightarrow [\alpha_1]\phi \wedge [\alpha_2]\phi \\
\text{(mix)} & \vdash & [\alpha^*]\phi \leftrightarrow \phi \wedge [\alpha][\alpha^*]\phi \\
\text{(induction)} & \vdash & (\phi \wedge [\alpha^*](\phi \to [\alpha]\phi)) \to [\alpha^*]\phi
\end{array}$$

and the following rules of inference:

(modus ponens) From $\vdash \phi_1$ and $\vdash \phi_1 \to \phi_2$, infer $\vdash \phi_2$.

(modal generalisation) From $\vdash \phi$, infer $\vdash [\alpha]\phi$.

Deriving Hoare Rules in PDL

The Floyd-Hoare inference rules can now be derived in PDL. As an example we derive the rule for guarded iteration:

$$\frac{\{\phi \wedge \psi\}\ \alpha\ \{\psi\}}{\{\psi\}\ \text{WHILE}\ \phi\ \text{DO}\ \alpha\ \{\neg\phi \wedge \psi\}}$$

Let the premise $\{\phi \wedge \psi\}\ \alpha\ \{\psi\}$ be given, i.e. assume (1).

$$\vdash (\phi \wedge \psi) \rightarrow [\alpha]\psi. \tag{1}$$

We wish to derive the conclusion

$$\vdash \{\psi\}\ \text{WHILE}\ \phi\ \text{DO}\ \alpha\ \{\neg\phi \wedge \psi\},$$

i.e. we wish to derive (2).

$$\vdash \psi \rightarrow [(?\phi; \alpha)^*; ?\neg\phi](\neg\phi \wedge \psi). \tag{2}$$

From (1) by means of propositional reasoning:

$$\vdash \psi \rightarrow (\phi \rightarrow [\alpha]\psi).$$

Deriving Hoare Rules in PDL, ctd

$$\vdash \psi \to (\phi \to [\alpha]\psi).$$

From this, by means of the test and sequence axioms:

$$\vdash \psi \to [?\phi; \alpha]\psi.$$

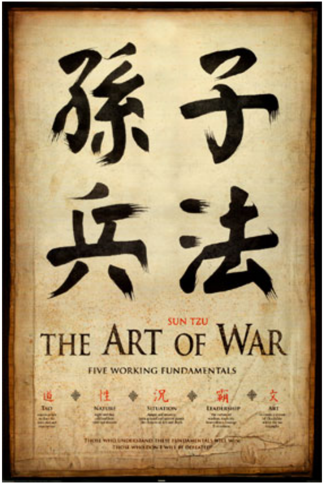
Applying the loop invariance rule gives:

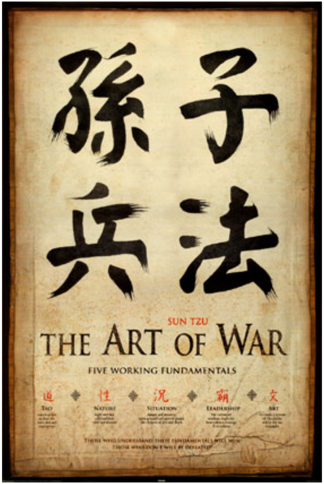
$$\vdash \psi \to [(?\phi; \alpha)^*]\psi.$$

Since ψ is propositionally equivalent with $\neg\phi \to (\neg\phi \wedge \psi)$, we get from this by propositional reasoning:

$$\vdash \psi \to [(?\phi; \alpha)^*](\neg\phi \to (\neg\phi \wedge \psi)).$$

The test axiom and the sequencing axiom yield the desired result (2).


孫子
兵法
SUN TZU
THE ART OF WAR
FIVE WORKING FUNDAMENTALS
道
TAO
性
NATURE
況
SITUATION
霸
LEADERSHIP
文
ART
THOSE WHO UNDERSTAND THESE FUNDAMENTALS WILL WIN
THOSE WHO DON'T WILL BE DEFEATED


孫子兵法
SUN TZU
THE ART OF WAR
FIVE WORKING FUNDAMENTALS
道
TAO
性
NATURE
況
SITUATION
霸
LEADERSHIP
文
ART
THOSE WHO UNDERSTAND THESE FUNDAMENTALS WILL WIN
THOSE WHO DON'T WILL BE DEFEATED

Sun Tzu

Carl von Clausewitz

Strategic Games: The Prisoner's Dilemma

	cooperate	defect
cooperate	c, c	c, d
defect	d, c	d, d

Strategic Games: The Prisoner's Dilemma

	cooperate	defect
cooperate	c, c	c, d
defect	d, c	d, d

With output function $o : \{c, d\}^2 \to \{x, y, z, u\}^2$:

	cooperate	defect
cooperate	x, x	y, z
defect	z, y	u, u

Strategic Games: The Prisoner’s Dilemma

	cooperate	defect
cooperate	c, c	c, d
defect	d, c	d, d

With output function $o : \{c, d\}^2 \rightarrow \{x, y, z, u\}^2$:

	cooperate	defect
cooperate	x, x	y, z
defect	z, y	u, u

Fixing the preferences of the players: $z > x > u > y$.

Strategic Games: The Prisoner's Dilemma

	cooperate	defect
cooperate	c, c	c, d
defect	d, c	d, d

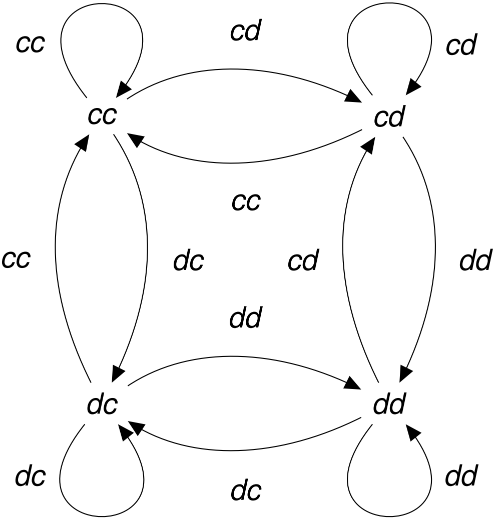
With output function $o : \{c, d\}^2 \to \{x, y, z, u\}^2$:

	cooperate	defect
cooperate	x, x	y, z
defect	z, y	u, u

Fixing the preferences of the players: $z > x > u > y$.
With numerical utilities:

	cooperate	defect
cooperate	$2, 2$	$0, 3$
defect	$3, 0$	$1, 1$

Group Strategies in PD Game are the Strategy Profiles

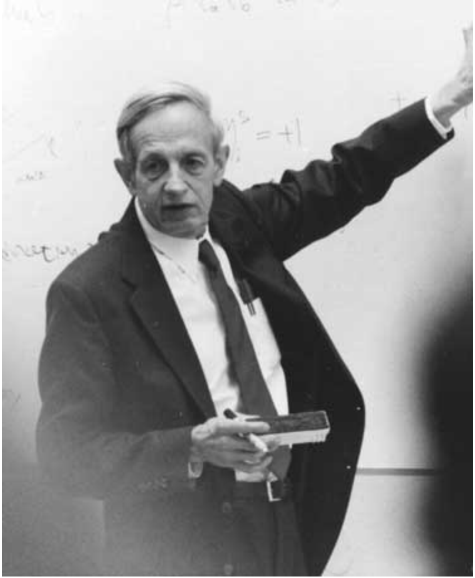


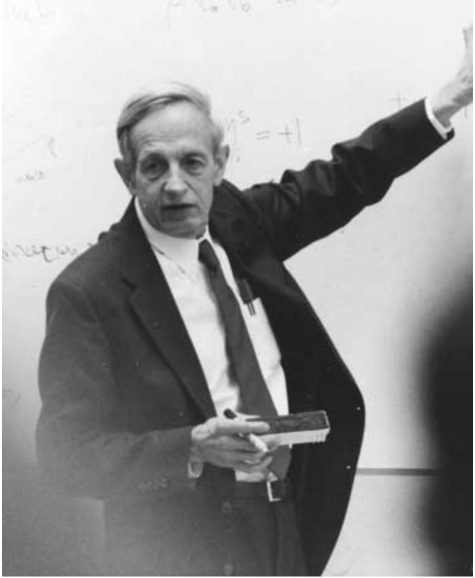
Key Notions: Best Response

- Let (s_i', s_{-i}) be the strategy profile that is like s for all players except i, but has s_i replaced by s_i'.
- A strategy s_i is a *best response* in s if

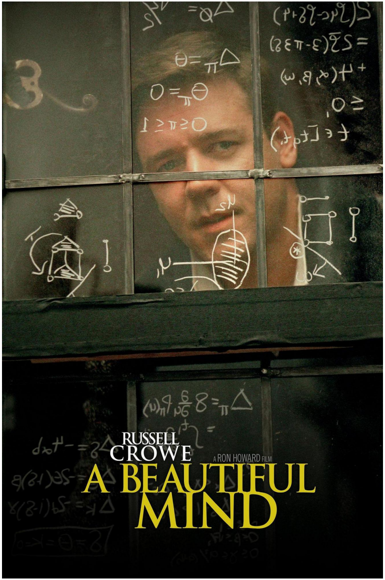
$$\forall s_i' \in S_i \; u_i(s) \geq u_i(s_i', s_{-i}).$$

- Example in PD game. Let $s = (d, c)$. The first player defects, the second player cooperates.
- Is d a best response for player 1 in (d, c)?
- Yes, because (d, c) gives payoff 3 for player 1, while the alternative (c, c) only gives payoff 2. So player 1 cannot do better than play d.


= +1



John Nash


RUSSELL
CROWE
A RON HOWARD FILM
A BEAUTIFUL
MIND

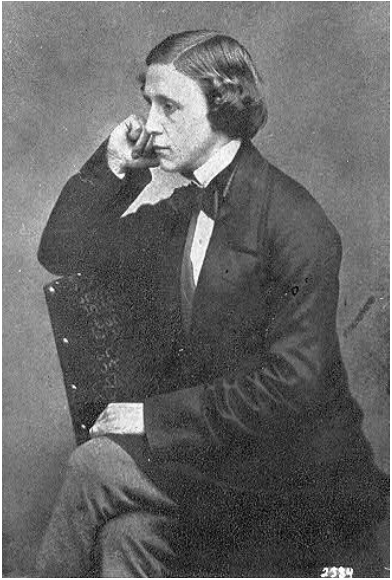
JOHN NASH
JUNE 13, 1928–MAY 23, 2015
GOLDEN GLOBE AWARDS
GOLDEN GLOBE AWARDS
CNN
11:06 AM PT
WSROOM

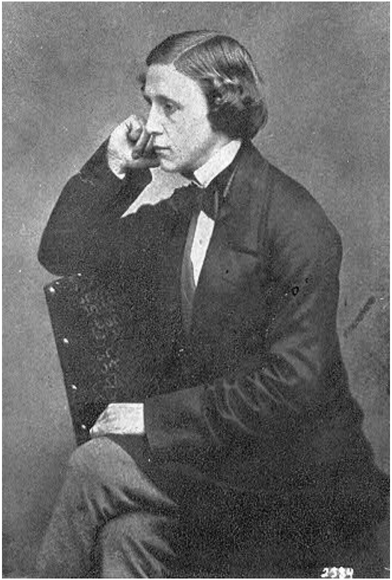
Key Notions: Pure Nash Equilibrium

- A strategy profile s is a (pure) Nash equilibrium if each s_i is a best response in s:

$$\forall i \in N \, \forall s_i' \in S_i \; u_i(s) \geq u_i(s_i', s_{-i}).$$

- A game G is *Nash* if G has a (pure) Nash equilibrium.
- (d, d) is a Nash equilibrium for the PD game, so the PD game is Nash.


2584



Charles Dodgson, also known as Lewis Carroll

Voting as a Multi-Agent Game

- Voting can be seen as a form of multi-agent decision making, with the voters as agents [Far69].
- Voting is the process of selecting an item or a set of items from a finite set A of alternatives, on the basis of the stated preferences of a set of voters.
- We assume that the preferences of a voter are represented by a ballot: a linear ordering of A. Let $\mathbf{ord}(A)$ be the set of all ballots on A.
- If there are three alternatives a, b, c, and a voter prefers a over b and b over c, then her ballot is abc.

Example

- Assume there are three voters $\{1, 2, 3\}$.
- Assume there are three alternatives $\{a, b, c\}$.
- Then profiles are vectors of ballots.
- Example profile where the first voter has ballot *abc*, the second voter has ballot *abc*, the third voter has ballot *bca*, and so on:
$$(abc, abc, bca).$$
- Absolute majority selects *a* as winner, for *a* has two votes, *b* has one.

Group Actions in Voting Games

Voters: $1, 2, 3$. Alternatives: a, b, c. Voting rule: absolute majority.

Group Actions in Voting Games

Voters: $1, 2, 3$. Alternatives: a, b, c. Voting rule: absolute majority.

1 is the row player, 2 the column player, and 3 the table player

a:

	a	b	c
a	a	a	a
b	a	b	a, b, c
c	a	a, b, c	c

b:

	a	b	c
a	a	b	a, b, c
b	b	b	b
c	a, b, c	b	c

c:

	a	b	c
a	a	a, b, c	c
b	a, b, c	b	c
c	c	c	c

Payoffs

- Assume ballot *abc*.
- Then $a > b > c$, and plausibly $a > \{a, b, c\} > c$.
- Assume $\{a, b, c\}$ and b give the same payoff.
- Fix the payoff function for voters i of type *abc* as

$$u_i(a) = 2, u_i(b) = u_i(\{a, b, c\}) = 1, u_i(c) = 0.$$

- If we do similarly for the other voter types, then this fixes the strategic game for voting according to the majority rule over the set of alternatives $\{a, b, c\}$.

Suppose 1 has ballot *abc*, 2 has *bca*, and has 3 *cab*

This fixes the strategic game:

a:

	a	b	c
a	(2, 0, 1)	(2, 0, 1)	(2, 0, 1)
b	(2, 0, 1)	(1, 2, 0)	(1, 1, 1)
c	(2, 0, 1)	(1, 1, 1)	(0, 1, 2)

b:

	a	b	c
a	(2, 0, 1)	(1, 2, 0)	(1, 1, 1)
b	(1, 2, 0)	(1, 2, 0)	(1, 2, 0)
c	(1, 1, 1)	(1, 2, 0)	(0, 1, 2)

c:

	a	b	c
a	(2, 0, 1)	(1, 1, 1)	(0, 1, 2)
b	(1, 1, 1)	(1, 2, 0)	(0, 1, 2)
c	(0, 1, 2)	(0, 1, 2)	(0, 1, 2)

Nash Equilibrium

- If they all cast their vote according to their true ballot, then 1 votes a, 2 votes b and 3 votes c.
- Then the outcome is a tie, $\{a, b, c\}$, with payoff $(1, 1, 1)$.
- This is a Nash equilibrium: the vote cast by each player is a best response in the strategy profile.

Changing the Game

- Now let's change the voting rule slightly, by switching to majority voting with tie breaking, where *abc* as the tie breaking order.
- This changes the majority rule into a resolute voting rule.
- It also changes the strategic game . . .

New Strategic Game

a:

	a	b	c
a	(2, 0, 1)	(2, 0, 1)	(2, 0, 1)
b	(2, 0, 1)	(1, 2, 0)	(2, 0, 1)
c	(2, 0, 1)	(2, 0, 1)	(0, 1, 2)

b:

	a	b	c
a	(2, 0, 1)	(1, 2, 0)	(2, 0, 1)
b	(1, 2, 0)	(1, 2, 0)	(1, 2, 0)
c	(2, 0, 1)	(1, 2, 0)	(0, 1, 2)

c:

	a	b	c
a	(2, 0, 1)	(2, 0, 1)	(0, 1, 2)
b	(2, 0, 1)	(1, 2, 0)	(0, 1, 2)
c	(0, 1, 2)	(0, 1, 2)	(0, 1, 2)

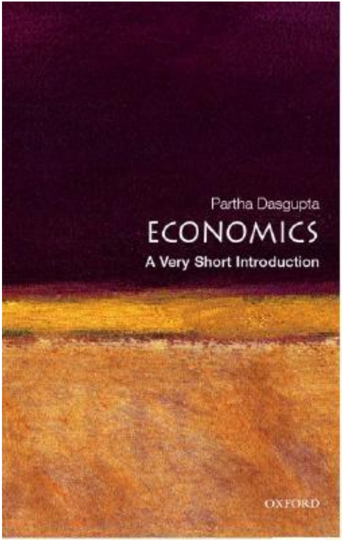
Coalition Formation

- If the players all vote according to their true preference, the outcome is *a* because of the tie breaking, with payoff given by $(2, 0, 1)$.
- But this is no longer a Nash equilibrium, for player 2 can improve his payoff from 0 to 1 by casting vote *c*, which causes the outcome to change into *c*, with payoff $(0, 1, 2)$.
- The strategy triple (a, c, c) is a Nash equilibrium.
- The voting rule seems to favour voter 1 with ballot *abc*, because the tie breaking rule uses this order for tie breaking.
- Still, the voter with this ballot ends up losing the game, because the other two players have an incentive to form a coalition against player 1.

Extending PDL

- PDL can be extended to a language for multi-agent strategic reasoning [Eij13].
- PDL can also be extended to a language that combines action and knowledge.
- Analysis of knowledge-based obligation [PPC06].
- Analysis of claims, duties, liberties in legal reasoning [EJ16].
- PDL and Deontic Logic: distinguish between *good* and *bad* states. Actions are *good* if they do not change the state from *good* to *bad* . . .

Analysis of Rational Action: Friends Elsewhere

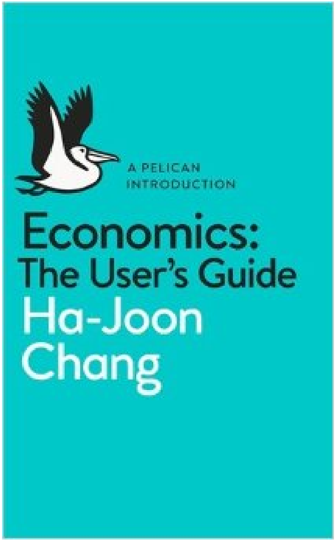


Partha Dasgupta

ECONOMICS

A Very Short Introduction

OXFORD



A PELICAN INTRODUCTION

Economics: The User's Guide

Ha-Joon Chang

References

- Johan van Benthem, Hans van Ditmarsch, Jan van Eijck, and Jan Jaspars.
  Logic in Action.
  Internet, 2013.
  Electronic book, available from `www.logicinaction.org`.

- Jan van Eijck.
  PDL as a multi-agent strategy logic.
  In Burkhard C. Schipper, editor, *TARK 2013 – Theoretical Aspects of Reasoning About Knowledge, Proceedings of the 14th Conference – Chennai, India*, pages 206–215, 2013.

- Jan van Eijck and Fengkui Ju.
  Modelling legal relations.
  In *Proceedings of LOFT 2016*, 2016.
  to appear.

- Robin Farquharson.

Theory of Voting.
Blackwell, 1969.

M.J. Fischer and R.E. Ladner.
Propositional dynamic logic of regular programs.
Journal of Computer and System Sciences, 18(2):194–211, 1979.

C.A.R. Hoare.
An axiomatic basis for computer programming.
Communications of the ACM, 12(10):567–580, 583, 1969.

Rohit Parikh.
The completeness of propositional dynamic logic.
In *Mathematical Foundations of Computer Science 1978*, pages 403–415. Springer, 1978.

Eric Pacuit, Rohit Parikh, and Eva Cogan.
The logic of knowledge based obligation.
Synthese, 31:311–341, 2006.

V. Pratt.

Semantical considerations on Floyd–Hoare logic.
Proceedings 17th IEEE Symposium on Foundations of Computer Science, pages 109–121, 1976.

V. Pratt.
Application of modal logic to programming.
Studia Logica, 39:257–274, 1980.

K. Segerberg.
A completeness theorem in the modal logic of programs.
In T. Traczyck, editor, *Universal Algebra and Applications*, pages 36–46. Polish Science Publications, 1982.