Chapter 7

Logic, Games and Interaction

Overview When we bring the logical systems for information and for action from the preceding two chapters together, we get to a ubiquitous phenomenon that has a more "social character": processes of interaction, where different agents respond to each other. Some people think that logic is only about lonely thinkers on their own, but most rational activities involve many agents: think of a case in court, or a process of inquiry in a scientific research group.

This chapter will not present a new logical system, but will demonstrate the fruitfulness of looking at logic from the viewpoint of interaction. We look at argumentation as a game. We give an interactive account of evaluation of assertions in models. We will explain a fundamental result about finite zero-sum two-player games, and we will show you how to apply it. We introduce sabotage games and model comparison games, we explain backward induction and the notion of strategic equilibrium. This brings us into the realm of game theory proper, where we introduce and discuss the basic notions and point out connections with logic.

7.1 Logic meets Games

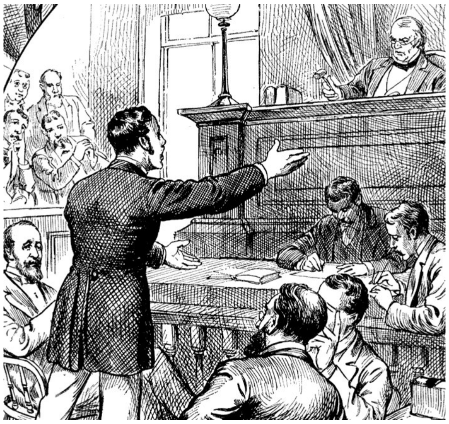
In many human core activities: conversation, argumentation, but also games that we play in general, social interaction is the heart of the matter, and people often follow rules for their responses over time, called *strategies*. These processes are subtle. In particular, in games, an activity we are all familiar with, strategies are chosen so as to best serve certain goals that the players have, depending on their preferences between different outcomes of the game. Cognitive scientists have argued that what makes us humans so special in the biological world is in fact this social intelligence.

Games fit naturally with the logics developed in this course. Contacts between logic and games go back to Antiquity, since logic arose in a context of argumentation, where valid inferences correspond to successful moves that participants can safely make. We will explain this connection a bit further in what follows, but just think of this. Argumentation is a sort of game, where you have to respond to others following certain rules, where timing of what you bring up matters, and where you can win or lose, depending on the quality of your strategies (and the sensibility of your claims). And here is an attractive idea that has recurred in the history of logic: players who stick to defending logically valid claims have a "winning strategy", a rule that guarantees success in winning debates.

Example 7.1 (Argumentation as a game) Here is an illustration making this a bit more precise. Consider this useful inference that you have encountered many times in Chapter 2:

from premises $\neg\varphi$, $\varphi \vee \psi$ to conclusion ψ.

Here is how we can see this function in an argumentation game.



A Proponent (player P) defends claim ψ against an Opponent (O) who has committed to the premises $\neg\varphi$, $\varphi \vee \psi$. The procedure is one where each player speaks in turn. We record some moves:

1 O starts by challenging P to produce a defense of ψ.

2 P now presses O on one of his commitments. $\varphi \vee \psi$, demanding a choice.

3 O must respond to this, having nothing else to say. There are two options here, which we list separately:

3' O commits to φ.

4' P now points at O's commitment to $\neg\varphi$, and wins because of O's self-contradiction.

3'' O commits to ψ.

4'' Now P uses this concession to make his own defense to 1. O has nothing further to say, and loses.

You see clearly how logical steps become moves in an argumentation scenario.

But argumentation is only one example of logic meeting games. Nowadays, there are many precise "logic games" for such tasks as evaluation of formulas in models, comparing models for similarity, finding proofs, and many other things of interest. We will discuss a few later, giving you an impression of what might be called *the game of logic*.

But there is more to the interface of logic and games. As we said already, interaction between many agents also involves their preferences, goals, and strategic behaviour where these are as important as the pure information that they have, or obtain. Such richer games have typically been studied, not in logic, but in the field of *game theory* which studies games of any sort: from recreational games to economic behaviour and warfare. Now, one striking recent development is the emergence of connections between logic and game theory, where logics are used to analyze the structure of games, and the reasoning performed by players as they try to do the best they can. The resulting *logics of games* are a natural continuation of the epistemic and dynamic logics that you have seen in the preceding chapters. We will also give you a brief glimpse of this modern link. Actually, this new interface developing today is not just an affair with two partners. It also involves computer science (the area of "agency" which studies complex computer systems plus their users) and philosophy, especially epistemology (the theory of knowledge) and philosophy of action. We will say something about these contacts in the Outlooks at the end.

This chapter will not gang up on you with one more core system that you must learn to work with, the way we have done in previous chapters. Its main aim is to give you an impression of how many earlier logical themes meet naturally in the arena of games, as a sort of combined finale. We start with a series of logical games, that should throw some new light on the logical systems that you have already learnt in this course. Following that, we discuss general games, and what logic has to say about them. All this is meant as a first introduction only. If you want to learn more about these interfaces, some very recent, you should go to a advanced course, or the specialized literature of today.

7.2 Evaluation of Assertions as a Logical Game

Our first example of a logical game is not argumentation, but something even simpler, the semantic notion of truth and falsity for formulas in models, as we have seen it in all our chapters, from propositional and predicate logic to epistemic and dynamic logic. Understanding complex logical expressions may itself be viewed as a game. A historical example is a famous explanation by Leibniz of the basic universal/existential quantifier combination that you have studied in Chapter 4. He did this in terms of two mathematicians discussing a logical formula of the form $\forall x \exists y \varphi(x, y)$. One of the mathematicians says to the other: if you challenge me with an x I will take on your challenge by giving you an y such that φ holds of x and y.

Here is a concrete example: the definition of continuity by Karl Weierstrass (1815–1897):

$$\forall x \forall \epsilon > 0 \exists \delta > 0 \forall y (|x - y| < \delta \rightarrow |f(x) - f(y)| < \epsilon).$$

This formula says something rather complicated about a function f, namely that f is continuous at every point x. The meaning can be unravelled by giving it the form of a dialogue.

Leibniz thought of a mathematician playing the universal quantifier as issuing a "challenge": any object for x. The other player, for the existential quantifier, then must come with an appropriate response, choosing some object for y that makes the assertion $\varphi(x, y)$ true. The following game is a generalization of this idea.

Remark on naming Logical games often have two players with opposing roles. There are many names for them: Players 1 and 2, Abelard and Eloise, Adam and Eve, $\forall$ and $\exists$, Spoiler and Duplicator, Opponent and Proponent, Falsifier and Verifier. In this chapter, we will use a selection from these, or we use the neutral I and II.

For a start, recall the basic semantic notion of predicate logic in Chapter 4, truth of a formula φ in a model $\mathcal{M}$ under an assignment s of objects to variables:

$$\mathcal{M}, s \models \varphi$$

Now, stepwise evaluation of first-order assertions can be understood dynamically as a game of evaluation for two players. Verifier **V** claims that φ is true in the setting $\mathcal{M}, s$, Falsifier **F** that it is false.

Definition 7.2 (Evaluation games) The natural moves of defense and attack in the first-order evaluation game will be indicated henceforth as

$$\text{game}(\varphi, \mathcal{M}, s)$$

The moves of evaluation games follow the inductive construction of formulas. They involve some typical actions that occur in games, such as *choice, switch*, and *continuation*, coming in dual pairs with both players **V** (Verifier) and **F** (Falsifier) allowed the initiative once:

Atoms Pd, Rde, …
V wins if the atom is true, **F** if it is false

Disjunction $\varphi_1 \vee \varphi_2$:
V chooses which disjunct to play

Conjunction $\varphi_1 \wedge \varphi_2$:
F chooses which conjunct to play

Negation $\neg\varphi$:
Role switch between the players, play continues with respect to φ.

Next, the quantifiers make players look inside $\mathcal{M}$'s domain of objects, and pick objects:

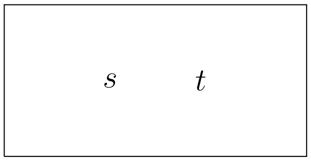
Existential quantifiers $\exists x\varphi(x)$:
V picks an object d, and then play continues with respect to $\varphi(d)$.

Universal quantifiers $\forall x\varphi(x)$:
The same, but now for **F**.

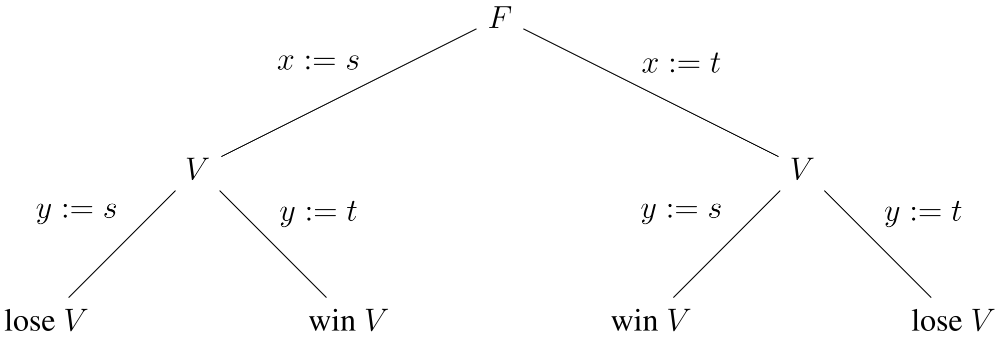
The game ends at atomic formulas: Verifier wins if it is true, Falsifier wins if it is false.

The schedule of the game is determined by the form of the statement φ. To see this in a very simple case, consider the following example.

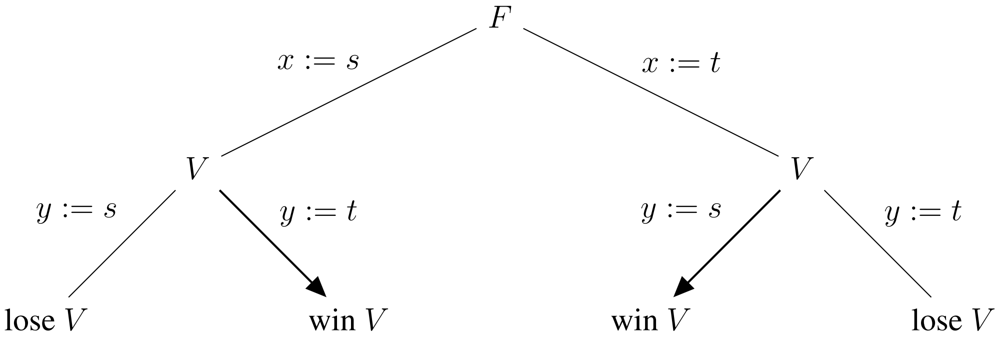
Example 7.3 (Evaluation Game With Two Objects) Let $\mathcal{M}$ be a model with two objects:



Here is the complete game for the first-order formula $\forall x\exists y x \neq y$ as a tree of moves, with scheduling from top to bottom (note that $x := s$ is shorthand for the action of picking object s for x):

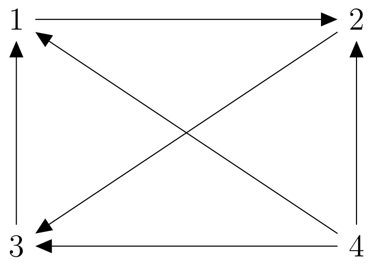


Falsifier starts, Verifier must respond. There are four possible plays, with two wins for each player. But Verifier is the player with a winning strategy, in an obvious sense: she has a rule for playing that will make her win no matter what Falsifier does: “choose the other object”. We can indicate this by high-lighting her recommended moves in bold-face:



Evaluation games for complex formulas in richer models can be more challenging.Here is an example going back to the graphs used in Chapters 3, 4 and 5.

Example 7.4 (Find non-communicators!) Consider the following communication network with arrows for directed links, and with all reflexive ’self-loops’ present but suppressed for convenience in the drawing:



In this setting, the predicate-logical formula

$$\forall x \forall y (Rxy \vee \exists z (Rxz \wedge Rzy))$$

claims that every two nodes in this network can communicate in at most two steps. Here is a possible run of the corresponding evaluation game:

F picks 2, game continues for $\forall y(R2y \vee \exists z(R2z \wedge Rzy))$

F picks 1, game continues for $(R21 \vee \exists z(R2z \wedge Rz1))$

V chooses $\exists z(R2z \wedge Rz1)$

V picks 4, game continues for $(R24 \wedge R41)$

F chooses $R41$.

test: V wins.

In this run, Falsifier started off with a threat by picking object 2, but then became generous towards Verifier, picking object 1. Verifier accepted the present by choosing the true right conjunct, but then tripped up by picking the wrong witness 4 instead of 3. But once again, Falsifier did not exploit this, by choosing the true right-hand conjunct. Obviously, however, Falsifier has a winning strategy in this game, exploiting the 'counter-example' of object 2, which cannot reach 1 in ≤ 2 steps. He even has more than one such strategy, as $x = 2, y = 4$ would also serve as a rule that always makes him win.

Exercise 7.5 Every finite network in which distinct points always have at least one directed link contains a 'Great Communicator': an object which can reach every other node in at most 2 steps. Prove this, and describe the general winning strategy for Verifier.

Truth and Verifier's winning strategies In our first example, participants were not evenly matched. Player **V** can always win: after all, she is defending the truth of the matter. More precisely, in the above terms, she has a *winning strategy*. As we said, such a strategy is a map from **V**'s turns to her moves following which guarantees, against any counterplay by **F**, that the game will end in outcomes that are won for V. By contrast, **F** has no winning strategy, as this would contradict **V**'s having one. (Playing two winning strategies against each other yields a contradiction.) Neither does have **F** the opposite power of a 'losing strategy': he cannot force **V** to win. Thus, players' powers of controlling outcomes may be quite different. Here is the fundamental connection between truth and games for evaluation games:

Lemma 7.6 (Success Lemma) The following are equivalent for all $\mathcal{M}, s$, and first-order φ:

(1) $\mathcal{M}, s \models \varphi$

(2) **V** has a winning strategy in game$(\varphi, \mathcal{M}, s)$.

A proof for this equivalence, while not hard at all, is beyond the horizon of this chapter.

Exercise 7.7 Prove the Success Lemma by induction on the construction of predicate-logical formulas. Hint: you will find it helpful to show two things simultaneously: (a) If a formula φ is true in $(\mathcal{M}, s)$, then Verifier has a winning strategy, (b) If a formula φ is false in $(\mathcal{M}, s)$, then Falsifier has a winning strategy.

Exercise 7.8 The above definition of evaluation games can be rephrased as follows. There are two kinds of *atomic games*: (a) testing atomic formulas for truth or falsity, but also an operation of (b) picking some object as a value for a variable. Complex games are then constructed out of these by means of the following operations: (i) choice between two games, (ii) role switch between the players of the game, and (iii) "sequential composition": first playing one game, and then another. Show that all evaluation games for predicate logical formulas can be defined in this manner. Conversely, can you give a game of this more abstract sort that does not correspond to a predicate-logical formula?

7.3 Zermelo's Theorem and Winning Strategies

Logic games involve broader game-theoretical features. Here is a striking one. Our evaluation games have a simple, but striking feature:

> Either Verifier or Falsifier must have a winning strategy!

The reason is simply the logical law of Excluded Middle. In any semantic model, either the given formula φ is true, or its negation is true. By the Truth Lemma then, either **V** has a winning strategy in the game for φ, or **V** has a winning strategy in the game for $\neg\varphi$: i.e.,after a role switch, a winning strategy for **F** in the game for φ. Two-player games in which some player has a winning strategy are called *determined*. The general game-theoretic background of our observation is due to the German set theorist Ernst Zermelo, though it was rediscovered independently by Max Euwe, the Dutch world-champion in Chess (1935–1937).

Ernst Zermelo

Max Euwe

We state it here for two-person "zero-sum" games whose players I, II can only win or lose, and where there is a fixed finite bound on the length of all runs.

Theorem 7.9 All zero-sum two-player games of fixed finite depth are determined.

Proof. Here is a simple algorithm determining the player having the winning strategy at any given node of a game tree of this finite sort. It works bottom-up through the game tree. First, colour those end nodes black that are wins for player I, and colour the other end nodes white, being the wins for II. Then extend this colouring stepwise as follows:

If all children of node n have been coloured already, do one of the following:

(1) if player I is to move, and at least one child is black: colour n black; if all children are white, colour n white;

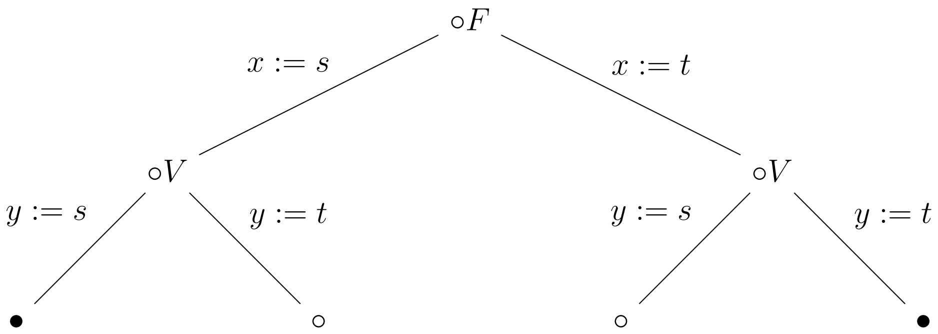
(2) if player II is to move, and at least one child is white: colour n white; if all children are black, colour n black.

This procedure eventually colours all nodes black where player I has a winning strategy, making those where II has a winning strategy white. Here is the reason:

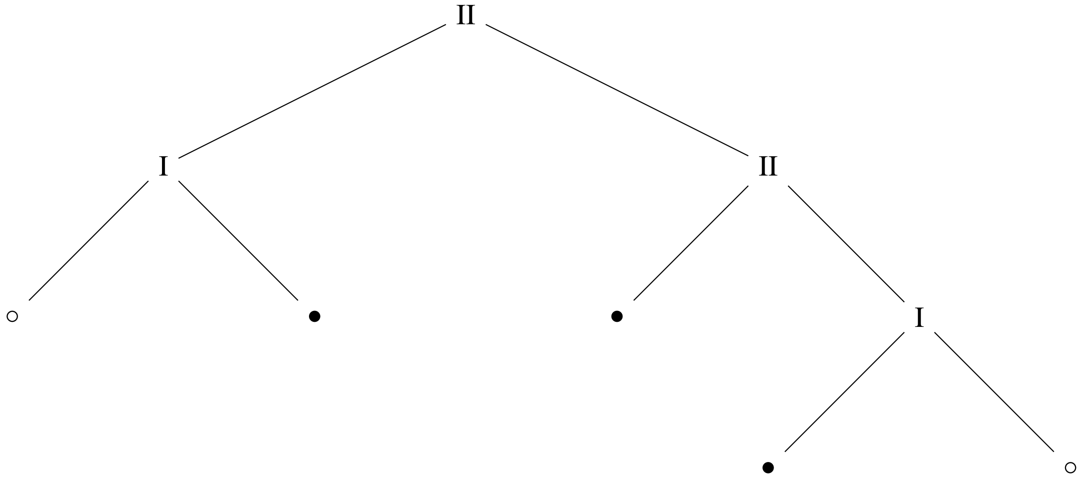
> A player has a winning strategy at one of his turns iff he can make a move to at least one daughter node where he has a winning strategy.

□

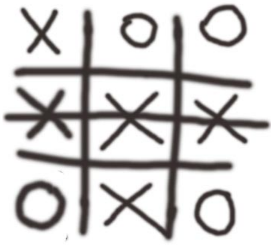
Here is the correct colouring for the simple game tree of our first example:



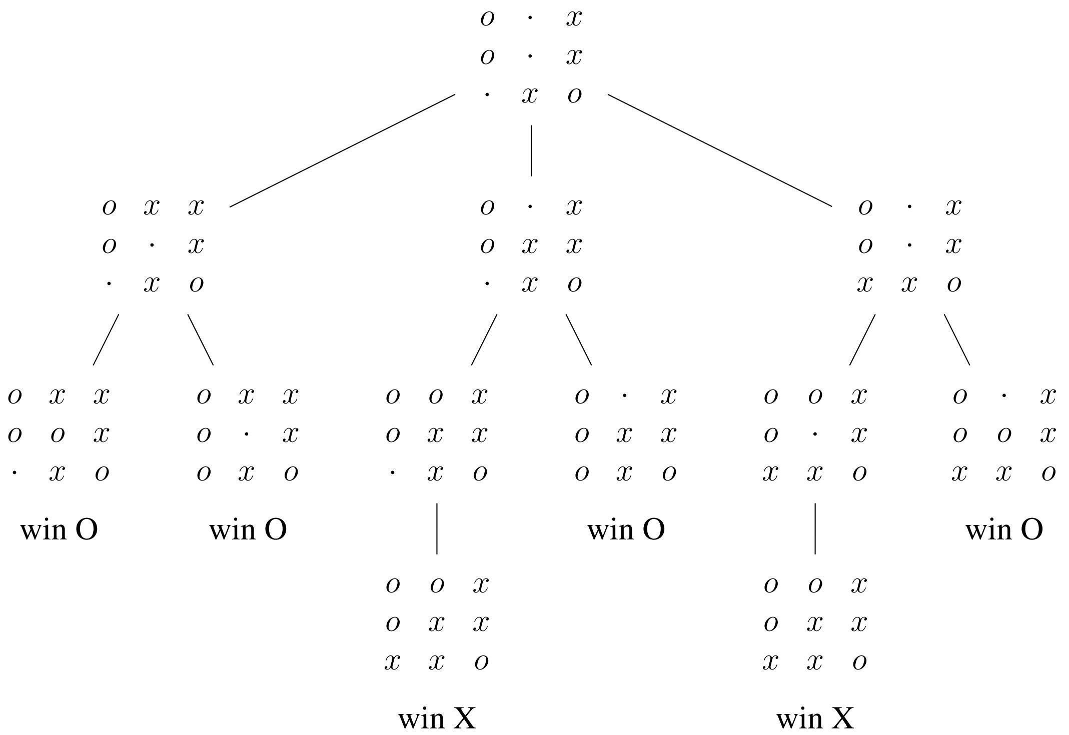
Exercise 7.10 Give the right colouring for the following game, whose black end nodes are wins for player I and white end nodes for player II:



Note how the evaluation games that we defined above satisfy all conditions of Zermelo's Theorem: two players, zero-sum, and finite depth. But its range is much broader. Recursive algorithms like this, in much more sophisticated optimized versions, are widely used in solving real board games, and even general AI search problems.

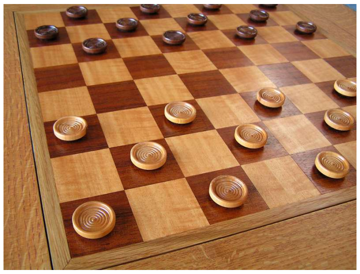


Example 7.11 Here is part of the game tree for the common game of Noughts and Crosses, indicating all possible moves from the configuration given at the top. It is easy to see that the Nought-player O has a winning strategy there by the colouring algorithm:



Exercise 7.12 Compute all the appropriate colours for the players in this game tree according to the Zermelo algorithm.

Zermelo was mainly concerned with games like chess, which also allow draws. Here the above method implies that one of the two players has a non-losing strategy. The difference between theory and practice is shown by the following. A century after the original result, it is still unknown which player has a non-losing strategy! But for other highly non-trivial board games, such as Checkers, the Zermelo solution has been found (2007).



Exercise 7.13 Actually, the very proof of Zermelo's Theorem may be cast as a form of Excluded Middle $\varphi \vee \neg\varphi$. Consider a game with 3 moves, and show how the statement of Determinacy can be derived using a suitable first-order formula about players and moves.

Not all two-player games of winning and losing are determined. Counter-examples are games where players need not be able to observe every move made by their opponents, of infinite games, where runs can go on forever.

Exercise 7.14 Consider an infinite game between two players, where histories may go on forever. Using the same style of reasoning as for Zermelo's Theorem, prove the following fact. If player II has no winning strategy at some stage s of the game, then I has a strategy for achieving a set of runs from s during all of which II never has a winning strategy for the remaining game from then on. Explain why this statement is not the same as determinacy for such games.

We hope that we have shown sufficiently how games can be close to the logics that you have learnt, and that thereby, familiar logical laws may acquire striking game-theoretic import. There are many further examples of this interplay, but for that, you will have to go to the literature. For now, we just note that many logical systems have corresponding evaluation games, that are used both as a technical tool, and as a attractive "dynamic" perspective on what logical tasks really are.

Exercise 7.15 Define an evaluation game for the epistemic language of Chapter 5. Hint: Positions of the game will be pointed models (M, s), and the new idea is that modalities move the world s to an accessible successor t. Now specify winning conditions for Verifier and Falsifier in such a way that the Truth Lemma stated above holds for this game with respect to the epistemic language.

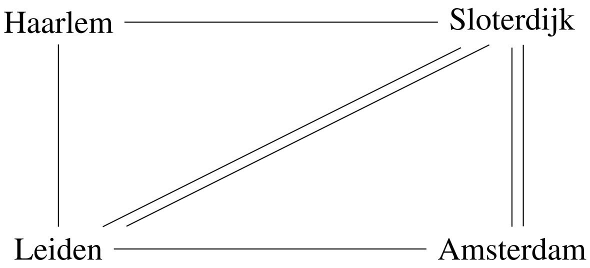
7.4 Sabotage Games: From Simple Actions to Games

This section is a digression. We give one more example of a logic-related game where Zermelo's Theorem plays a role. Our main purpose is to show you how game design is still going on, and you yourself may want to try your hand at it.

The following "sabotage game" was designed, tongue-in-cheek, as a model for railway travel in The Netherlands in periods of strikes and disruptions. Normally, traveling involves solving a search problem "from A to B" along a sequence of links in some fixed network. But what if things get out of hand?

Consider a network consisting of nodes representing cities and links representing ways to travel between them. There are two players: 'Runner' and 'Blocker'. Runner moves from a given starting node A and tries to reach some specified goal node B along existing connections. In each round of the game, Blocker first removes one link from the current network, Runner then moves along one available connection where he is. The game ends if Runner has reached the goal node (he wins then), or if Runner cannot move any more (Blocker then wins).

Example 7.16 In the following railway network, each line is a possible connection for Runner to take:



Runner starts in Haarlem, and wants to reach Amsterdam. Suppose that Blocker first removes the link Haarlem-Sloterdijk. Then Runner can go to Leiden. Now Blocker must remove Leiden-Amsterdam, leaving Runner a link from Leiden to Sloterdijk. Now Blocker is too late: whichever link he cuts between Sloterdijk and Amsterdam, Runner can then use the remaining one to arrive. Does this mean that Runner has a winning strategy in this game? The answer is "No": Blocker has a winning strategy, but it goes as follows.

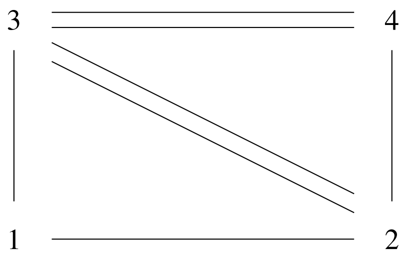
First cut a link between Sloterdijk and Amsterdam, Then see what Runner does. If he goes to Sloterdijk, cut the second link, and whatever he does next, cut the link Leiden-Amsterdam. If Runner goes to Leiden as his first step, cut the Leiden-Amsterdam link first, then cut the second Sloterdijk-Amsterdam link. Now Amsterdam has become isolated: Runner will never get there.

We have been talking as if the Sabotage Game is determined. And it is, since the conditions of Zermelo's Theorem apply. There are two players, there is just winning and losing as outcomes, and moreover, the game cannot last longer than it takes to cut all links in the given finite graph.

Actually, the Sabotage Game is even closely related to the evaluation games that you have seen before. You can also see it as an evaluation game for a first-order formula on the given graph, which is of the form "For every first move by Blocker, there is a move by Runner in the accessibility relation minus Blockers move such that, for every second move by Blocker . . . etc." Thus, again, logic and games remain close.

Exercise 7.17 Suppose we change the preceding game as follows: Blocker want to *force* Runner to go to Amsterdam, by making it impossible for him to stay anywhere else, assuming that Runner has to move as long as he has open links where he is. By the way, this version has been used to model situations of learning where Teachers are pushing unwilling Students to goal states where they should be. Who has the winning strategy in this new scenario?

Exercise 7.18 Consider the sabotage game with the following initial configuration:



This time, the task for Runner is to start from position 1 and then visit all nodes of the network. Blocker wins if she can somehow prevent this. Who has the winning strategy? How does it work?

You can view the Sabotage Game as a typical multi-agent game version of a standard algorithm for graph search. This is just one instance where the computational perspective of Chapter 6, too, meets with game-theoretic ideas.

7.5 Model Comparison as a Logic Game

Logic games can perform evaluation or argumentation. But they can also be used to perform other basic tasks that you have not learnt about yet in this course. Let us look at one of these, the issue of comparing models. One of the main functions of a language is distinguishing between different situations, represented by models. And one vivid way of measuring the expressive power of a language is through the following game of spotting differences.

Playing the game Consider any two models $\mathcal{M}$, $\mathcal{N}$. Player D (Duplicator) claims that $\mathcal{M}$, $\mathcal{N}$ are similar, while S (Spoiler) maintains that they are different. Players agree on some finite number k of rounds for the game, 'the severity of the probe'.

Definition 7.19 (Comparison games) A *comparison game* works as follows, packing two moves into one round:

S chooses one of the models, and picks an object d in its domain.

D then chooses an object e in the other model, and the pair (d, e) is added to the current list of matched objects.

At the end of the k rounds, the total object matching obtained is inspected. If this is a 'partial isomorphism', D wins; otherwise, S has won the game.

Here, a *partial isomorphism* is an injective partial function f between models $\mathcal{M}$, $\mathcal{N}$, which is an isomorphism between its own domain and range seen as submodels. This sounds complicated but it is really very easy: a partial isomorphism links finite sets of objects one-to-one in such a way that all their structure is preserved.

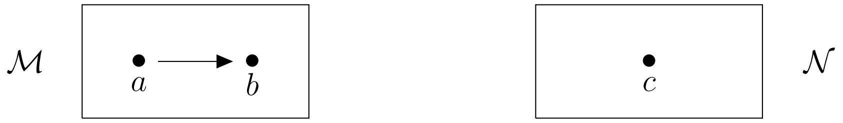
Example 7.20 Let $\mathbb{R}$ be the real numbers with the relation 'less than', and let $\mathbb{Q}$ be the rational numbers (the set of all numbers that can be written as p/q, where p and q are integer numbers, with $q \neq 0$). Both of these sets are ordered by $\leq$ ('less than or equal'). Note that 0 can be written as $\frac{0}{1}$, so 0 is a rational number.

For an injective function f between a finite set of reals and a finite set of rationals 'to preserve the structure' in this case means: $x \leq y$ iff $f(x) \leq f(y)$.

The number $\sqrt{2} \in \mathbb{R}$ is not a fraction. Consider the set of pairs $\{(0, 0), (\sqrt{2}, 1.4142)\}$. This is a partial isomorphism, for it preserves the $\leq$ relation.

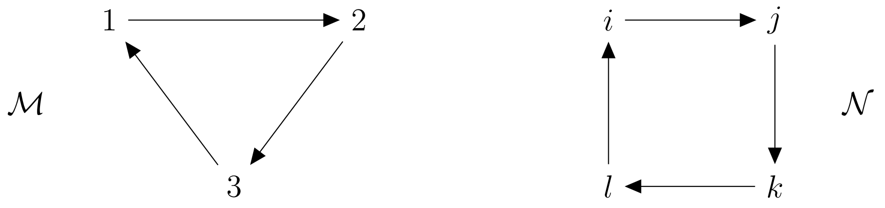
Here are some possible runs for models with relations only, as we have often used in Chapter 4, illustrating players' strategies. As before, the game character shows in that players may play badly and lose, but it is their winning strategies that are most important to us. We look at a first-order language with a binary relation symbol R only, mostly disregarding identity atoms with $=$ for the sake of illustration.

Example 7.21 (Playing between graphs: Pin versus Dot) We discuss one run and its implications.



In the first round, **S** chooses a in $\mathcal{M}$, and **D** must choose c in $\mathcal{N}$. If we stopped after one round, **D** would win. There is no detectable difference between single objects in these models. They are all irreflexive, and that's it. But now take a second round. Let **S** choose b in $\mathcal{M}$. Then **D** must again choose c in $\mathcal{N}$. Now **S** wins, as the map $\{(a, c), (b, c)\}$ is not a partial isomorphism. On the lefthand side, there is an R link between a and b, on the righthand side there is none between c and c. Clearly, the structure does not match.

Example 7.22 ('3-Cycle' vs '4-Cycle') Our next example is a match between a '3-Cycle' and a '4-Cycle':



We just display a little table of one possible 'intelligent run':

Round 1 **S** chooses 1 in $\mathcal{M}$, **D** chooses i in $\mathcal{N}$.

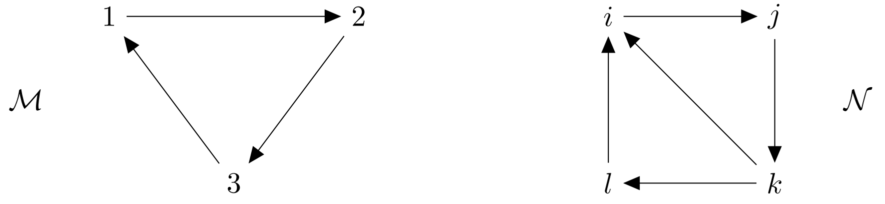
Round 2 **S** chooses 2 in $\mathcal{M}$, **D** chooses j in $\mathcal{N}$.

Round 3 **S** chooses 3 in $\mathcal{M}$, **D** chooses k in $\mathcal{N}$.

S wins, as $\{(1, i), (2, j), (3, k)\}$ is not a partial isomorphism. But he can do even better:

> **S** has a winning strategy in two rounds, first picking i in $\mathcal{N}$, and then taking k in the next round. No such pattern occurs in $\mathcal{M}$, so **D** is bound to lose.

Exercise 7.23 Consider the following variation on the last example.



Which of the two players has a winning strategy in the partial isomorphism game?

Example 7.24 The final example match is 'Integers' $\mathbb{Z}$ versus 'Rationals' **Q**. These two linear orders have obviously different first-order properties: the latter is dense, the former discrete. Discreteness intuitively means that there are pairs of different numbers with ‘nothing in between’. Denseness intuitively means the negation of this: for every pair of different numbers, there is always a number in between. Here is the formal version of density:

$$\forall x \forall y(x < y \to \exists z(x < z \wedge z < y)).$$

And the formal version of discreteness:

$$\exists n \exists m(n < m \wedge \neg\exists k(n < k \wedge k < m)).$$

So this difference between $\mathbb{Q}$ and $\mathbb{Z}$ can be expressed by means of predicate logical formulas. The only question is how soon this will surface in the game.

By choosing his objects well, **D** has a winning strategy here for the game over two rounds. But **S** can always win the game in three rounds. Here is a typical play:

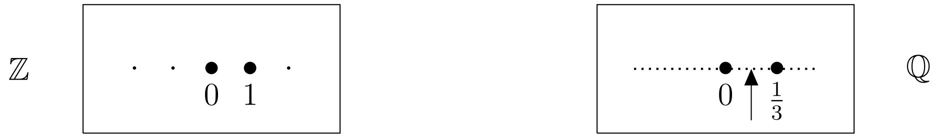
Round 1 **S** chooses 0 in $\mathbb{Z}$, **D** chooses 0 in $\mathbb{Q}$.

$\mathbb{Z}$
0
0
$\mathbb{Q}$

Round 2 **S** chooses 1 in $\mathbb{Z}$, **D** chooses $\frac{1}{3}$ in $\mathbb{Q}$.

$\mathbb{Z}$
0 1
0 $\frac{1}{3}$
$\mathbb{Q}$

Round 3 **S** chooses $\frac{1}{5}$ in $\mathbb{Q}$, any response for **D** is losing.



7.6 Different Formulas in Model Comparison Games

Example 7.24 suggests a connection between strategies in model comparison games and formulas of predicate logic. In actual play of model comparison games, you will notice this connection yourself. We discuss it here because it may give you a different perspective on the predicate logic that you have learnt in Chapter 4. In fact, model comparison games throw a lot of new light on predicate logic, as we will explain now.

Winning strategies for **S** are correlated with specific first-order formulas φ that bring out a *difference* between the models. And this correlation is tight. The quantifier syntax of φ triggers the moves for **S**.

Example 7.25 (Continuation of Example 7.21) Exploiting definable differences: 'Pin versus Point'. An obvious difference between the two in first-order logic is

$$\exists x \exists y Rxy$$

Two moves were used by **S** to exploit this, staying inside the model where it holds.

Example 7.26 (Continuation of Example 7.22, '3-Cycle versus 4-Cycle') The first **S**-play exploited the formula

$$\exists x \exists y \exists z (Rxy \wedge Ryz \wedge Rxz)$$

which is true only in $\mathcal{M}$, taking three rounds. The second play, which had only two rounds, used the following first-order formula, which is true only in the model $\mathcal{N}$:

$$\exists x \exists y (\neg Rxy \wedge \neg Ryx \wedge x \neq y).$$

Example 7.27 (Continuation of Example 7.24, 'Integers versus Rationals') **S** might use the definition of density for a binary order that was given above,

$$\forall x \forall y (x < y \to \exists z (x < z \wedge z < y)),$$

to distinguish $\mathbb{Q}$ from $\mathbb{Z}$. Let us spell this out, to show how the earlier spontaneous play for this example has an almost algorithmic derivation from a first-order difference formula. For convenience, we use density in a form with existential quantifiers only. The idea is for **S** to maintain a difference between the two models, of stepwise decreasing syntactic depth. **S** starts by observing that the negation of density, i.e., the property of discreteness, is true in $\mathbb{Z}$, but false in $\mathbb{Q}$:

$$\exists x \exists y (x < y \wedge \neg \exists z (x < z \wedge z < y)). \qquad (*)$$

He then chooses an integer witness d for x, making

$$\exists y (d < y \wedge \neg \exists z (d < z \wedge z < y))$$

true in $\mathbb{Z}$. **D** can then take any object d' she likes in $\mathbb{Q}$:

$$\exists y (d' < y \wedge \neg \exists z (d' < z \wedge z < y))$$

will always be false for it, by the fact that (*) is false in $\mathbb{Q}$. We have:

$$\mathbb{Z} \models \exists y (d < y \wedge \neg \exists z (d < z \wedge z < y)), \; \mathbb{Q} \not\models \exists y (d' < y \wedge \neg \exists z (d' < z \wedge z < y)).$$

In the second round, **S** continues with a witness e for the new outermost quantifier $\exists y$ in the true existential formula in $\mathbb{Z}$: making $(d < e \wedge \neg \exists z (d < z \wedge z < e))$ true there. Again, whatever object e' **D** now picks in $\mathbb{Q}$, the formula $(d' < e' \wedge \neg \exists z (d' < z \wedge z < e'))$ is false there. In the third round, **S** analyzes the mismatch in truth value. If **D** kept $d' <' e$ true in $\mathbb{Q}$, then, as $\neg \exists z (d < z \wedge z < e))$ holds in $\mathbb{Z}$, $\exists z (d' < z \wedge z < e'))$ holds in $\mathbb{Q}$. **S** then switches to $\mathbb{Q}$, chooses a witness for the existential formula, and wins.

Thus, even the model switches for **S** are encoded in the difference formulas. These are mandatory whenever there is a switch in type from one outermost quantifier to a lower one. Thus, you see how the game is tightly correlated with the structure of the logical language.

Adequacy in terms of quantifier depth Our examples may have suggested the following correlation to you:

> 'winning strategy for **S** over n rounds' versus 'difference formula with n quantifiers'.

But that is not quite the right measure, if you think of Spoiler's reasoning in the above examples. The correct syntactic correlation for the number of rounds needed to win is *syntactic quantifier depth*, being the *maximum length of a quantifier nesting in a formula*. Here is the result that ties it all together.

> Let us write
>
> $$\text{WIN}(\mathbf{D}, \mathcal{M}, \mathcal{N}, k)$$
>
> for: **D** has a *winning strategy* against **S** in the k-round comparison game between the models $\mathcal{M}$ and $\mathcal{N}$.

Comparison games can start from any given 'handicap', i.e., an initial matching of objects in $\mathcal{M}$ and $\mathcal{N}$. In particular, if models have distinguished objects named by individual constants, then these must be matched automatically at the start of the game. In the proofs to follow, for convenience, we will think of all 'initial matches' in the latter way. Now here is the analogue of the Success Lemma for our earlier evaluation games:

Theorem 7.28 (Adequacy Theorem) For all models $\mathcal{M}$, $\mathcal{N}$, all $k \in \mathbb{N}$, the following two assertions are equivalent:

(1) $\text{WIN}(\mathbf{D}, \mathcal{M}, \mathcal{N}, k)$: **D** has a winning strategy in the k-round game.

(2) $\mathcal{M}, \mathcal{N}$ agree on all first-order sentences up to quantifier depth k.

Again, a proof is not hard, but it goes beyond this course. If you go through such a proof (there is of course no harm in trying, it works by induction on the number k), you will find that the situation is even more interesting. There is an explicit correspondence between

(1) winning strategies for **S** in the k-round comparison game for $\mathcal{M}, \mathcal{N}$,

(2) first-order sentences φ of quantifier depth k with $\mathcal{M} \models \varphi$ and $\mathcal{N} \not\models \varphi$.

A similar match exists for Duplicator, whose winning strategies correspond to well-known mathematical notions of similarity between models, in some cases: "isomorphism".

Determinacy, and further theoretical issues As long as we fix a finite duration k, Zermelo's Theorem still applies to model comparison games: either Duplicator or Spoiler must have a winning strategy. But actually, it is easy to imagine comparison games that go on forever: in that case, we say that Duplicator wins if she loses at no finite stage of such an infinite history. (This is a very natural feature game-theoretically, since infinitely repeated games are central in in "evolutionary games" modeling what happens in large communities of players over time.) It can be shown that infinite comparison games are still determined, since their structure is still quite simple. This abstract similarity from a game-theoretic perspective goes even further. It can be shown that comparison games are evaluation games at a deeper level, for logical languages that define structural similarities.

7.7 Bisimulation Games

Comparison games do not just apply to predicate logic. They are also widely used for the logics of information and action that you have seen in Chapters 5 and 6. Here is an important example, for the notion of bisimulation that was defined in Chapter 6 (Definition 6.44).

A straightforward modification of the above game is by restricted selection of objects, say, to relational successors of objects already matched. This leads to comparison games for epistemic and dynamic logic. The definition of bisimulation is repeated here, adapted to the present context of models $\mathcal{M}, \mathcal{N}$ for multi-modal logic.

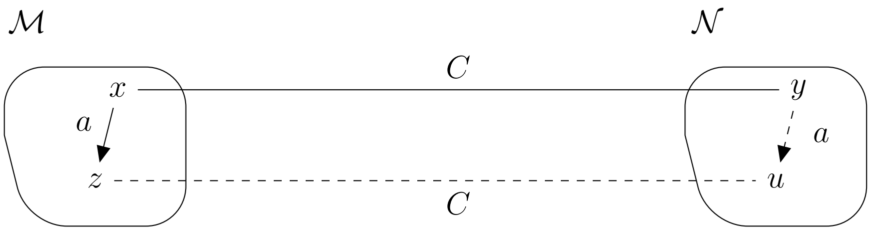
Definition 7.29 (Bisimulation) A bisimulation is a binary relation C between states of models $\mathcal{M}, \mathcal{N}$ with binary transition relations R_a, such that, whenever xCy, then we have 'atomic harmony' or 'invariance' (x, y satisfy the same proposition letters), plus two-way zigzag clauses for all relations a:

Invariance x and y verify the same proposition letters,

Zig If xR_az, then there exists u in $\mathcal{N}$ with yR_au and zCu.

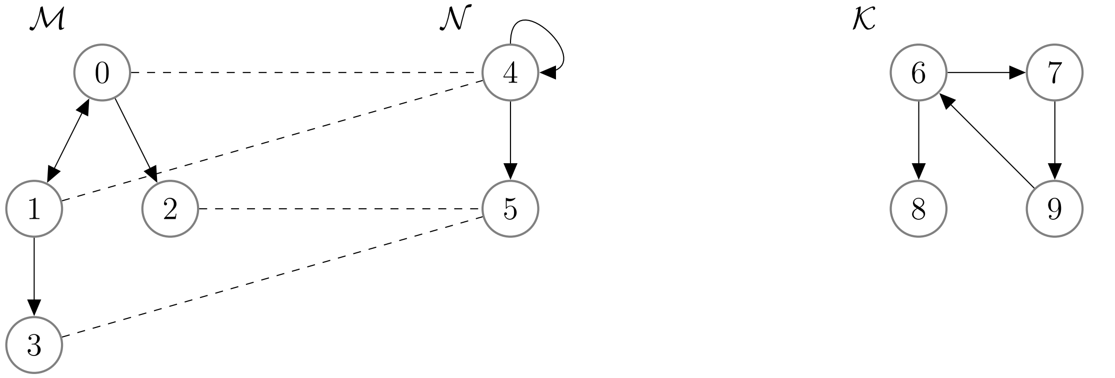
Zag vice versa.

The **Zig** condition in a picture:



This definition was already illustrated by some examples in Chapter 6. Here are some more examples.

Example 7.30 (Bisimulation between process graphs) State 0 in $\mathcal{M}$ and state 4 in $\mathcal{N}$ are connected by the bisimulation given by the dotted lines – but no bisimulation includes a match between world 4 in $\mathcal{N}$ and world 6 in $\mathcal{K}$:



We recall from Chapter 6 that modal formulas are invariant for bisimulation:

Proposition 7.31 (Invariance Lemma) If C is a bisimulation between two graphs $\mathcal{M}$ and $\mathcal{N}$, and mCn, then $\mathcal{M}, m$ and $\mathcal{N}, n$ satisfy the same modal formulas.

The fine-structure of bisimulation suggests a game comparing epistemic or dynamic models between Duplicator and Spoiler, comparing successive pairs (m, n) in two models $\mathcal{M}$, $\mathcal{N}$:

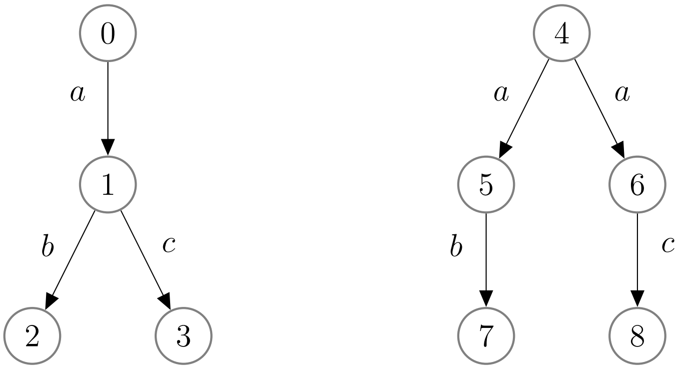
> In each round Spoiler chooses a state x in one model which is a successor of the current m or n, and Duplicator responds with a matching successor y in the other model. If x, y differ in their atomic properties, Spoiler wins. If Duplicator cannot find a matching successor: likewise.

Again one can show that this fits precisely:

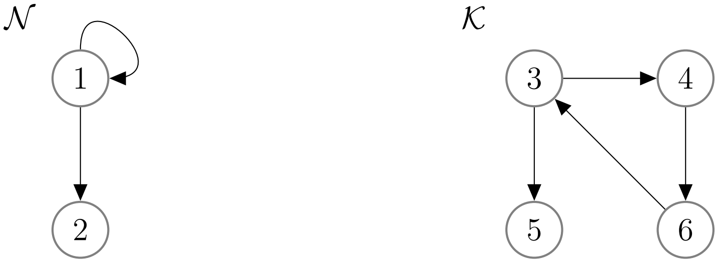
(1) Spoiler's winning strategies in a k-round game between $(\mathcal{M}, s)$, $(\mathcal{N}, t)$ match the modal formulas of operator depth k on which s, t disagree.

(2) Duplicator's winning strategies over an infinite round game between $(\mathcal{M}, s)$, $(\mathcal{N}, t)$ match the bisimulations between them linking s to t.

Example 7.32 Spoiler can win the game between the following models from their roots. He needs two rounds – and different strategies do the job. One stays on the left, exploiting the difference formula $\Diamond_a(\Diamond_b\top \lor \Diamond_c\top)$ of depth 2, with three existential modalities. Another winning strategy switches models, but it needs a smaller formula $\Box_a\Diamond_b\top$.



In the non-bisimulation pair $\mathcal{N}$, $\mathcal{K}$ from above, repeated here, starting from a match between worlds 1 and 3, Spoiler needs three rounds to win.



Spoiler forces Duplicator in two rounds into a match where one world has no successor, while the other does. One winning strategy for this exploits the modal difference formula $\Diamond\Diamond\Box\bot$.

Exercise 7.33 Give a winning strategy for Spoiler in the game about the two process graphs in Exercise 6.22 from Chapter 6.

7.8 Preference, Equilibrium, and Backward Induction

Now we turn to real game theory. The games that we considered so far are trees of nodes (the stages of the game) and moves, that is, labeled transition relations. Moreover, the endpoints of the game were marked for winning or losing by the relevant players. Compare the trees for the game of Noughts and Crosses in Example 7.11. But this is not enough for real games. What is typical there is that players may have finer *preferences* between outcomes, that can lead to much better predictions of what rational players would achieve.

Definition 7.34 (Extensive Games with Perfect Information) An *extensive game with perfect information* consists of

(1) a set N of players,

(2) a set H of (finite) sequences of successive actions by players closed under taking prefixes

(3) a function P mapping each non-terminal history (i.e., one having a proper extension in H) to the player whose turn it is,

(4) for each player $i \in N$ a preference relation $\geq_i$ on the set of terminal histories (histories having no proper extension in H).

Without the preference relation, one has an *extensive game form*.

Example 7.35 (Donation, Extensive Game)

(1) There are two players I and II.

(2) There are two actions, giving a donation to the other player (d) and failing to do so (n). Distinguishing between players, these become D, d, N, n (capital letters for the first player).

The rules of the game are as follows. Each of the two players is given 10 euros. Each player is informed that a donation of 5 euros to the other player will be doubled. Next, the players are asked in turn whether they want to make the donation or not.

Assuming I plays first, the terminal histories are Dd, Dn, Nn, Nd. The set H consists of the terminal histories plus all proper prefixes:

$$\{\lambda, D, N, Dd, Dn, Nn, Nd\}$$

where λ is the empty list. The turn function P is given by $P(\lambda) = \text{I}, P(D) = \text{II}, P(N) = \text{II}$.

(3) To see what the preferences for I are, note that receiving a donation without giving one is better than receiving a donation and giving one, which is in turn better than not receiving a donation and not giving one, while giving a donation while receiving nothing is worst of all. So we get:

$$Nd >_1 Dd >_1 Nn >_1 Dn$$

The preferences for II are:

$$Dn >_2 Dd >_2 Nn >_2 Nd.$$

Definition 7.36 (Preferences and Payoff Functions) A payoff function (or: utility function) for a player i is a function u_i from game outcomes (terminal histories) to integers. A payoff function u_i *represents* the preference ordering $\leq_i$ of player i if $p \leq_i q$ iff $u_i(p) \leq u_i(q)$, for all game outcomes p, q.

Example 7.37 For I's preferences in the Donation game, we need a utility function u_1 with

$$u_1(Nd) > u_1(Dd) > u_1(Nn) > u_1(Dn).$$

The most obvious candidate for this is the function that gives the payoff for I in euros:

$$u_1(Nd) = 20, u_1(Dd) = 15, u_1(Nn) = 10, u_1(Dn) = 5.$$

For II this gives:

$$u_2(Dn) = 20, u_2(Dd) = 15, u_2(Nn) = 10, u_2(Nd) = 5.$$

Combining these payoff functions, we get:

$$u(Nd) = (20, 5), u(Dd) = (15, 15), u(Nn) = (10, 10), u(Dn) = (5, 20).$$

But there are other possible candidates for this. Here is an example (one of many):

$$u_1(Nd) = 3, u_1(Dd) = 2, u_1(Nn) = 1, u_1(Dn) = 0.$$

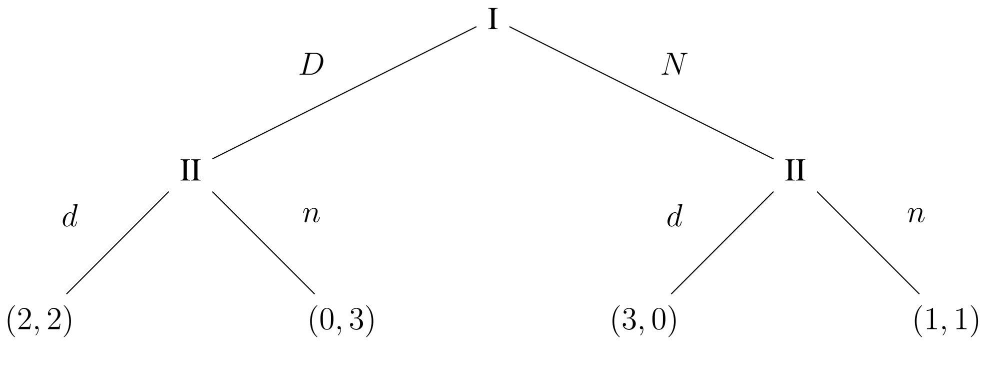
Similarly, for u_2, we can choose:

$$u_2(Dn) = 3, u_2(Dd) = 2, u_2(Nn) = 1, u_2(Nd) = 0.$$

Combining these payoff functions, we get:

$$u(Nd) = (3, 0), u(Dd) = (2, 2), u(Nn) = (1, 1), u(Dn) = (0, 3).$$

Such a combined payoff function can be used in the game tree for the Donation game, as follows:

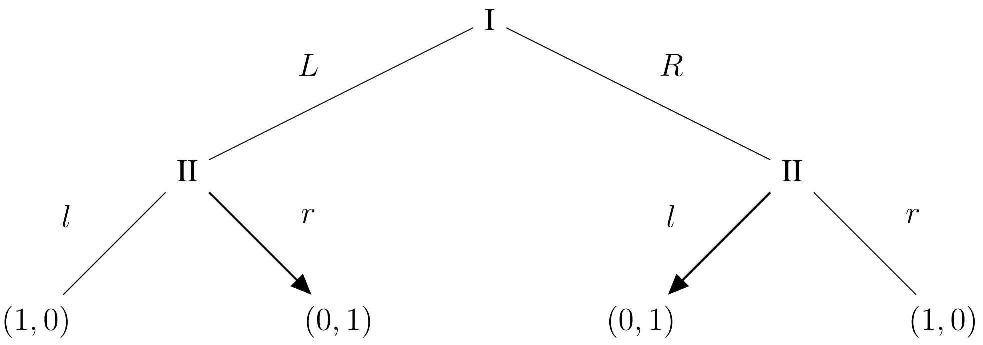


The definition of extensive games can easily be generalized to infinite games, where the action histories need not be finite. If we allow H to contain countably infinite histories, then we need to impose the extra condition of *closure under countable limits*. What this means is that if an infinite sequence of histories $h, h', \ldots$ is present in H where each history h_i extends the previous one, then the infinite history that has each h_i as a prefix is also in H. We can generalize still further to infinite games of higher cardinalities. Mathematicians love to do this, not bothered at all by the fact that this breaks the link to real-world-games that people can play.

Pictorially, extensive games are finite (or infinite) mathematical trees, whose branches are the possible runs or histories of the game. You may already have recognized them as a structure that you have learnt about in Chapter 6: extensive game trees are obviously models for a dynamic logic with various sorts of labeled actions (the moves), and special atomic predicates given by a valuation (say, the marking for preference values of players, or indications whose players turn it is at some intermediate node). We will return to this "process perspective" on games later, since it is the starting point for their general logical analysis.

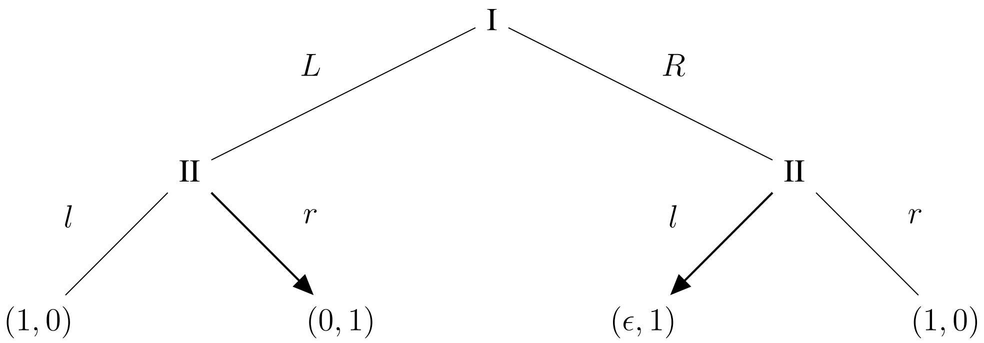
But for now, let us just look a bit closer at what preferences do. We start with a simple case where preferences may make a Zermelo-style analysis more interesting.

Example 7.38 (Losing with a little twist) Recall our very first example, where we now indicate players' evaluation of the outcomes in pairs (I-value, II-value):



A Zermelo computation tells us that II has a winning strategy indicated by the black arrows, and it does not matter what I does in equilibrium.

But now suppose that I has a slight preference between the two sites for his defeat, being the end nodes with values $(0, 1)$. Say, the one to the left takes place on a boring beach, where the sea will wash out all traces by tomorrow. But the one to the right is a picturesque mountain top, and bards might sings ballads about I's last stand for centuries. The new preferences might be indicated as follows:



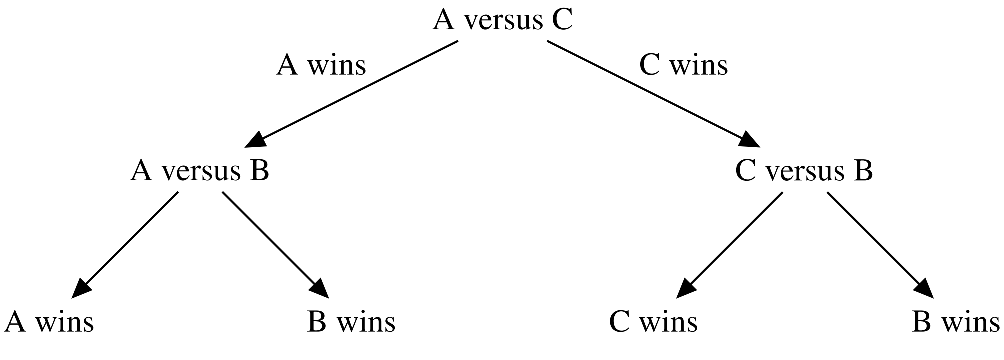
Intuitively, with these preferences, I goes 'right' at the start, and then II goes 'left'.

With preferences present, however, examples can quickly get highly non-trivial:

Example 7.39 (Tiered Voting) Three political parties 1, 2, 3 have the following preferences concerning issues A, B, C, indicated in order from top to bottom:

1	2	3
A	C	B
C	B	A
B	A	C

Moreover, the following schedule of voting has been agreed upon. First, there will be a majority vote between A and C, eliminating one of these. The winner will be paired against B. The game tree for this is as follows, where players move simultaneously casting a vote. We just record the outcomes, without their vote patterns:



Here is what will happen if everyone votes according to their true preferences. A will win against C, after which A will lose against B. But now 1 might reason as follows. "If I had voted for C against A in the first round (against my real preference), the last round would have been between B and C, which would have been won by C – which I prefer to outcome B." But other players can reason in the same way.

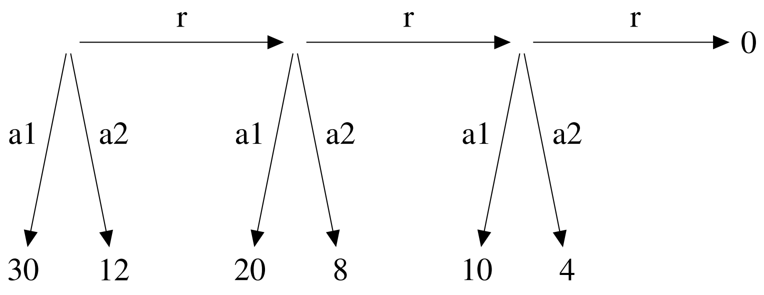
What is a stable outcome, representing rational behaviour of people in such a situation? Well, players cannot do better at pre-final nodes than state their true preferences. Any deviation will harm their favourite outcome. So players know the outcomes at the middle nodes of the procedure. Therefore, in the first round, players will vote according to their true preferences between those outcomes.

Backward Induction This brings us to the key notion of this section: **backward induction** is a general algorithm for computing a 'most rational course of action' by finding values for each node in the game tree for each player, representing the best outcome value she can guarantee through best possible further play (as far as within her power). Here is a formal definition.

Definition 7.40 (Backward Induction Algorithm) Suppose II is to move, and all values for daughter nodes are known. The II-value is the maximum of all the II-values on the daughters, the I-value the minimum of the I-values at all II-best daughters. The dual case for I's turns is completely analogous.

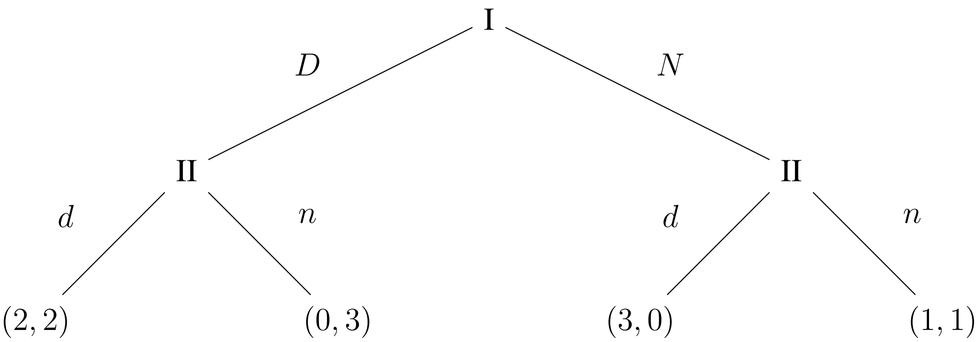
Backward induction is a useful tool for decision making.

Example 7.41 (Decision Making by Backward Induction) Consider a lady who decides that she is in need of a husband. She is a high flyer, with no time to spare for hanging around in bars and discos. So she agrees that a dating agency presents her with three candidates. She decides beforehand that she will marry one of these three, or no-one at all. She distinguishes three categories: a guy can be (1) great, (2) halfway decent, or (3) completely hopeless. She has confidence that the agency will be able to come up with category (1) and (2) candidates, and she estimates that the two kinds are equally likely. She puts the value of being married to a great guy at 10 per year, and the value of being married to a halfway decent guy (snores, drinks too much, but still decent enough to put the garbage out on Mondays) at 4 per year. The value of being single is 0. Taking a time horizon of 3 years, and given that the beginning of every year has one candidate in store for her, what should she do? Clearly, given her utilities, she should grab the first great guy that comes along. But suppose the agency offers her only a halfway decent guy? Then what? This decision problem can be pictured as follows:



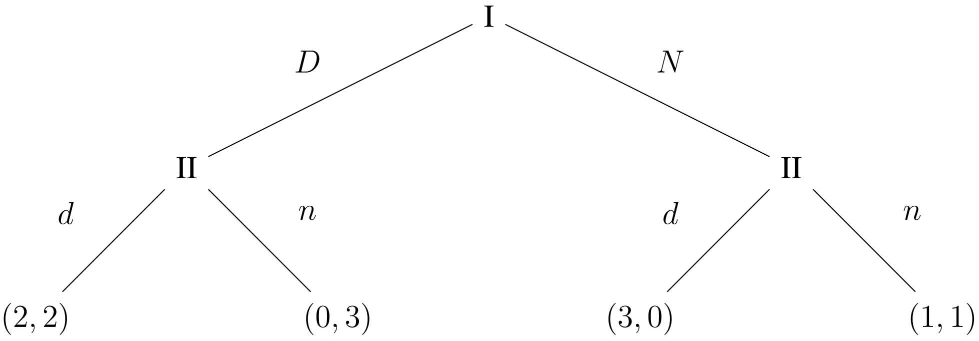
Reasoning backward, she first asks herself what she should do with the third candidate, supposing she has refused the earlier two. Clearly, given her utilities, she should accept him no matter what. Accepting a so-so guy would give her 4, and refusing him will give her 0. Now how about accepting a so-so guy as second candidate? Refusing him will give on average a payoff of 7, and accepting him gives her twice 4, which is 8. This is better, so it is rational for her to accept. But for the first candidate she can afford to be picky. Accepting a so-so guy would give her 12, and refusing him will give her on average 14, given that she accepts the second candidate no matter what, which she should, for the reasons we we have just seen.

Example 7.42 (BI Solution to the Donation game) Let's compute the 'rational course of action' in the Donation game (Example 7.35) by Backward Induction (BI):

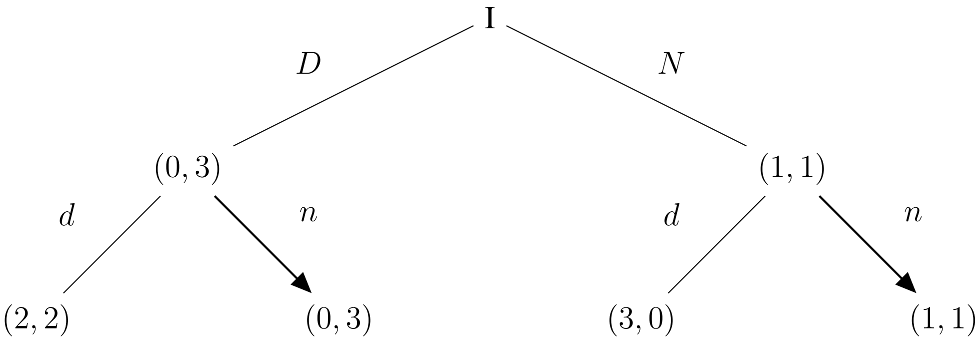


The II value for D is that of Dd, which is 2, the II value for N is that of Nn, which is 1. The I value for D is the I value of Dd, which is 2. The I value for N is the I value of Nn, which is 1. So BI dictates that I plays N, to which II will respond with n.

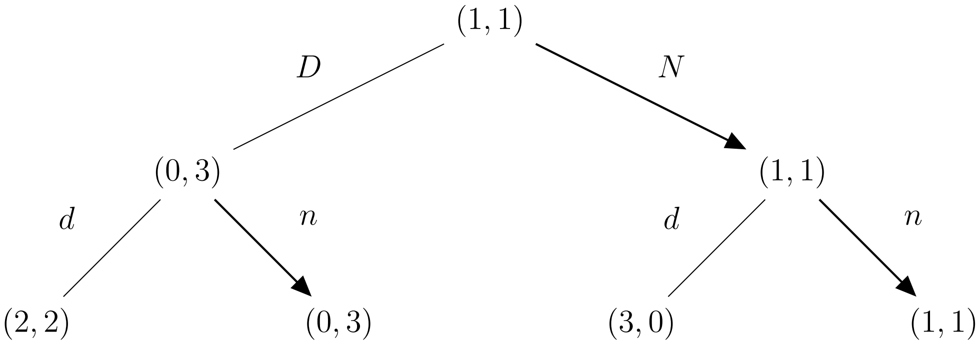
Backward Induction clearly generalizes the Zermelo algorithm that we have seen before. Instead of propagating win/lose information up the tree, we now propagate payoffs: from



to



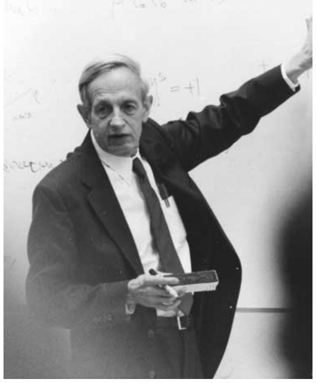
and next to



One can see it as a *maximin* procedure: players *maxi*mize their *min*imal gain. Again, its scope goes further than you might think. Algorithms computing numerical values like this also occur in AI, under the name '$\alpha\beta$ search'. In that case, the values at nodes indicate heuristic potentials for finding some desired goal.

Exercise 7.43 Using Backward Induction, compute how the parties should vote in the scenario of Example 7.39.

Strategies and equilibrium behaviour If you look at arrows moving from nodes to subsequent nodes where the current player gets her maximum value, Backward Induction computes a pair of *strategies* in our earlier sense, one for each player. Why do game theorists think that Backward Induction computes a "best" or "rational" behaviour in this manner? This has to do with the fact that these strategies are in *equilibrium*: no player has an incentive to deviate. To define this precisely, first note that any strategies σ, τ for two players determines a unique outcome $[\sigma, \tau]$ of the game, obtained by playing the two strategies against each other.

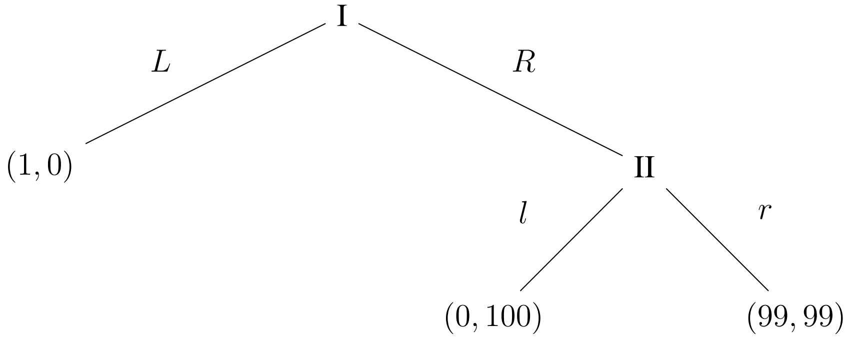


John Nash

Definition 7.44 (Nash equilibrium) A pair of strategies σ, τ in a two-player game is a *Nash equilibrium* if, for no $\sigma' \neq \sigma$, $[\sigma', \tau] \geq_1 [\sigma, \tau]$, and similarly for player II with respect to τ: for no $\tau' \neq \tau$, $[\sigma, \tau'] \geq_2 [\sigma, \tau]$. Here $[\sigma, \tau]$ signifies the outcome of the game when I plays σ and II plays τ. In other words, neither player can improve his outcome by deviating from his strategy while it is given that the other player sticks to hers.

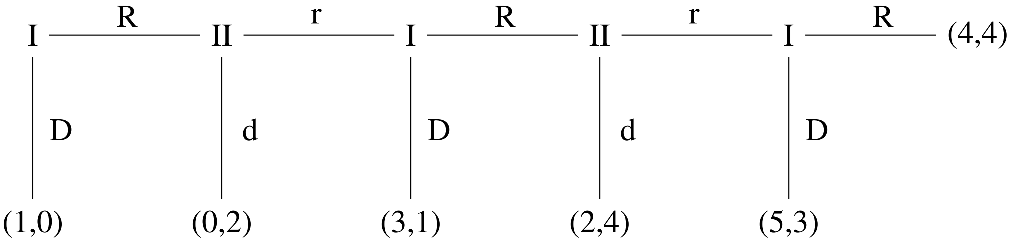
In our earlier logic games, any pair of a winning strategy plus any strategy for the other player is a Nash equilibrium. This shows that equilibria of a game need not be unique, and indeed, there can be one, more, or none at all. Backward Induction at least produces equilibria of a very special kind: they are "subgame-perfect". What this says is that the computed best strategies at nodes remain best when restricted to lower nodes heading subgames underneath. This property is not guaranteed by Nash equilibrium per se: think again of my playing badly in a logic game against an opponent playing a winning strategy. This is not perfect in subgames at lower nodes that are not reached, where I could have won after all by playing better.

Criticism Despite its appealing features, Backward Induction has also been criticized for being at odds with intuition in some cases. In the following simple game, Backward Induction computes an equilibrium with outcome $(1, 0)$, making both players hugely worse off than the outcome $(99, 99)$ that is also a possible outcome of the game:



This has been a starting point for analyzing the reasoning underlying this “obvious” algorithm in much more detail, and game theorists have found it useful to employ techniques from logic for this purpose. We will return to this issue later on. The following example is another case where Backward Induction yields an unintuitive result.

Example 7.45 (Centipede Games) A centipede game is a game where two players take turns and where each player can decide to opt out or play on. Opting out gives a better payoff than the opponent, but playing on raises the stakes: the sum of the payoffs increases. Here is an example.



Analyzing this, one sees that the players together would be best off by staying in the game until the very end, for then they will each receive 4. But will player I play R in his last move? Clearly not, for playing D will give a better payoff of 5 rather than 4. So player II realizes that staying in the game at the pre-final stage, will yield payoff 3. So opting out at this stage is better for her, so she plays d. Player I is aware of this, and to avoid this outcome, will play D, with outcome $(3, 1)$. So II realizes that this will be the result of playing r. She will therefore play d, with result $(0, 2)$. But this is bad for player I, who will therefore play D on his very first move, and the game ends with outcome $(1, 0)$.

Please note that we have now left the “game of logic” here, that is, the games of winning and losing that we used for logical tasks. We will now take a look at “logic of games”: what is there for a logically trained mind to see in them?

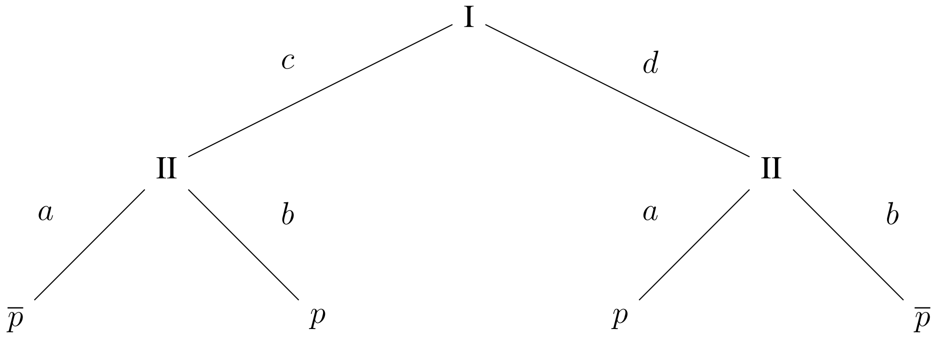
7.9 Game logics

We have seen that not all is well in finding game 'solutions' by the usual methods. To remedy that, we need to analyze the reasoning of the players more carefully. For that we introduce **game logics**.

Logics of game structure and strategies For a start, extensive games are processes in the same sense as Chapter 6. This means that everything you have learnt there applies at once.

Describing moves What would a logic for describing moves look like?

Example 7.46 (An extensive game tree) . Consider a game tree for two players I, II with four possible actions c, d, a, b, and some special property p holding at two of the four possible end states:



Here is a typical dynamic formula which is true at the root of his model:

$$[c \cup d]\langle a \cup b\rangle p.$$

Each of the actions c and d leads to a state where either a or b can be executed to get to a final state where p holds. In our earlier terms, this says that player II has a strategy ensuring that the outcome of the game satisfies p. Here, p might just be the property that II wins, in which case the modal formula expresses that II has a winning strategy.

Describing strategies The preceding style of description does not yet define the strategies themselves. But that, too, can be done with the techniques of Chapter 6, namely programs viewed as defining sets of transitions. The total move relation of a game is clearly a union of atomic transitions, and strategies are subrelations of the move relation, namely, transition functions defined on players' turns. (Arbitrary subrelations would be more like more loosely specified "plans".) Thus, on top of the 'hard-wired' moves in a game, complex PDL-style relations can define strategies in terms of players options at a current node (*IF THEN ELSE*), sequential composition, and even iteration (as in a rule "always repeat the previous move by the other player until you have won").

Example 7.47 (Broken Record) As an example, here is a PDL version of the well-known 'broken record' strategy: whatever player I says (does), player II keeps repeating her message (action) b until I gives up:

$$(\text{move}_1; b)^*; ?\text{win}_2.$$

Example 7.48 (Match Removal Game) This is played between I and II. The player that is next to move removes 1, 2 or 3 matches from a pile. The player that can take the last match(es) has won. If the number of matches on the table is a four-fold, and I is next to move, the following is a winning strategy for player II:

$$((\text{one}_1; \text{three}_2) \cup (\text{two}_1; \text{two}_2) \cup (\text{three}_1; \text{one}_2))^*; ?\text{stack-empty}.$$

Exercise 7.49 Consider a finite game tree. Using the language of propositional dynamic logic, define the following assertion about players powers in the game:

> σ is a strategy for player i forcing the game, against any play of the others, to pass only through states satisfying φ.

Describing preferences Game trees are not only models for a dynamic logic of moves and strategies, but also for players preferences. In this course, we have not told you how to reason about preferences, even though this is an upcoming topic in studies of agency, since our behaviour is clearly not just driven by pure information, but just as much by what we want and prefer. Evidently, games involve information, action and preferences all intertwined. Indeed, a solution procedure like the above Backward Induction really depends on mixing these notions in a very specific way, that game theorists call "rationality": players only choose moves whose outcome they consider best for them, given what they know and believe about the game and the other players.

It would take us too far in this course to analyze Backward Induction in full logical style, but here is one typical fact about it. Let us add an operator

$$\langle \leq_i \rangle \varphi$$

to our logical language with the following intended meaning:

> There is some outcome of the game that player i finds at least as good as the present stage where the formula φ is true.

Then the key fact about the Backward Induction strategy σ, viewed as a program in our dynamic logic, can be stated as follows in logical terms:

Fact 7.50 The backward induction solution of a finite game is the unique binary relation bi on the game tree satisfying the following modal preference-action law:

$$[\text{bi}^*](\text{end} \to \varphi) \to [\text{move}]\langle\text{bi}^*\rangle(\text{end} \wedge \langle\leq_i\rangle\varphi)$$

for all formulas φ.

This looks extremely intimidating. But you may find it a useful exercise in reading logical formulas to see that it essentially says the following:

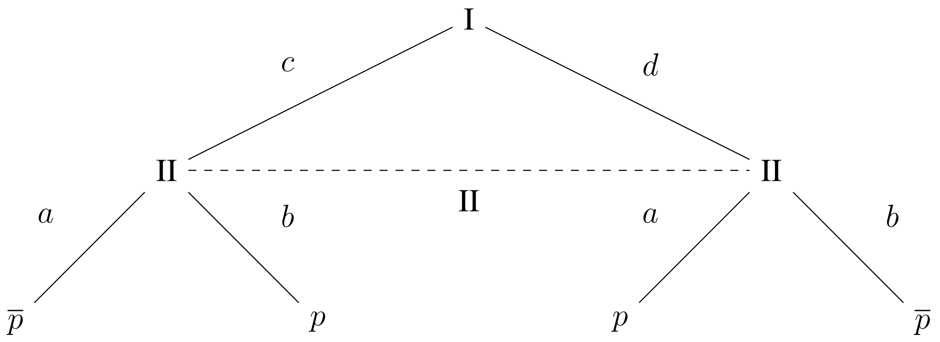
> There is no alternative move to the BI-prescription at the current node all of whose outcomes would be better than following the BI-solution.

7.10 Games with imperfect information

Logical analysis extends beyond the kinds of games that we have seen in this chapter so far. For instance, the ideas of Chapter 5 come into play with the extended class of games with imperfect information: that is, the players need not know exactly where they are in a game tree. This happens in many settings, for instance, when playing at cards where many things are not publicly known – and in this sense, our card examples of Chapter 5 were entirely appropriate.

Here we just show how logics of the sort you have studied apply to this broader setting.

Example 7.51 (An extensive game with imperfect information) Consider a game given earlier, in Example 7.46. But now assume we want to add an extra touch: player II is uncertain about the first move played by I. (Perhaps, I put it in an envelope, or perhaps this is a version of the donation game where there is no communication between the participants). This models a combined dynamic-epistemic language using ideas that you have seen in Chapters 5 and 6:



The modal formula $[c \cup d]\langle a \cup b\rangle p$ is still true at the root. But we can make more subtle assertions now, using the dotted line as an accessibility relation for knowledge. At stage

s, a player knows those propositions true throughout the 'information set' to which s belongs. Thus, after I plays move c in the root, in the left middle state, II knows that playing either a or b will give her p – the disjunction $\langle a\rangle p \vee \langle b\rangle p$ is true at both middle states:

$$\Box_2(\langle a\rangle p \vee \langle b\rangle p).$$

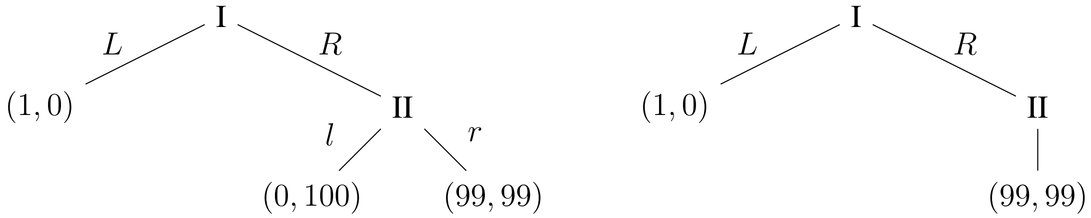
Nevertheless, there is no specific move of which II knows that it guarantees an outcome satisfying p – which shows in the leftmost middle state the truth of the formula

$$\neg\Box_2\langle a\rangle p \wedge \neg\Box_2\langle b\rangle p.$$

Think of a tragic person who knows the right partner is walking around right in this city, but does not know of any particular person whether (s)he is that partner.

Information dynamics Our final example is the information dynamics of Chapter 5, which again mixed information that agents have with changes in that information as events happen. Games typically have this dynamic flavour. As you play on, you learn more about what your opponent has done. But also, you can even change the whole game by exchanging information, as shown in the following scenario.

Example 7.52 (Making a promise) One can sometimes break a bad Backward Induction solution by changing the game. In our earlier game, the Nash equilibrium (1, 0) can be avoided by E's promise that she will not go left. This may be seen as a public announcement that some histories will not occur (E actually gives up some of her freedom) and the new equilibrium (99, 99) results, making both players better off:

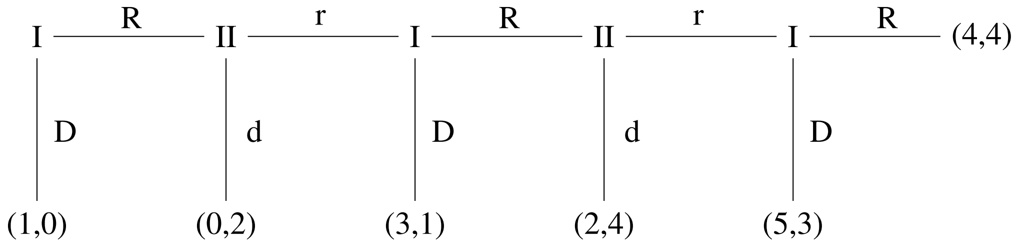


Another use of such dynamic actions is Backward Induction itself. We can view this procedure as a process of 'internal deliberation' via repeated announcements of 'rationality' that prunes the initial game tree:

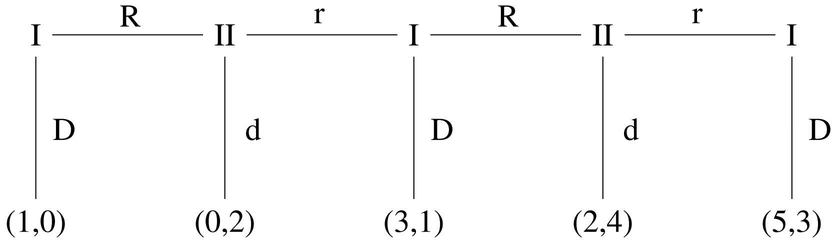
Theorem 7.53 The Backward Induction solution for extensive games is obtained through repeated announcement of the assertion "no player chooses a move all of whose further histories end worse than all histories after some other available move".

Instead of giving a proof, we show how the procedure works out for an example.

Example 7.54 (The Centipede Again) Consider the game from Example 7.45 again:



This has five turns, with I moving first and last. Stage 1 of the announcement procedure: I announces that he will not play R at the end. This rules out the branch leading to $(4,4)$:



Next, stage 2. II announces that she will not play r. This rules out the state with payoff $(5,3)$.

Stage 3: I announces that he will not play R. This rules out the state with payoff $(2,4)$.

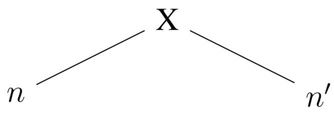
Stage 4: II announces that she will not play r. This rules out the state with payoff $(3,1)$.

Stage 5: I announces that he will not play R. This rules out the state with payoff $(0,2)$.

So I plays D and the game ends with payoff $(1,0)$.

This scenario, in terms of repeated events of public announcements to the effect "I will act rationally, i.e., in my own best interest" removes nodes from the tree that are *strictly dominated* by siblings as long as this can be done.

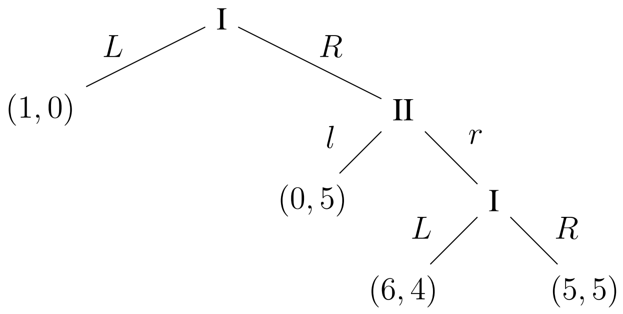
Definition 7.55 (Strict Domination) A node n in a game tree is strictly dominated by a sibling node n' if the player who is first to move (X in the picture) is better off by playing n' than playing n, no matter what the other players do.



Clearly, a rational player should never play a strictly dominated node. Technically, the iterated announcement procedure for extensive games ends in largest sub-models in which players have common knowledge of rationality in the above sense. This is one of the central notions in the foundations of game theory.

Other ways of reasoning about games We end with one more issue where logic meets the foundations of game theory today. Backward Induction is just one scenario for creating plausibility in a game. To see alternatives, consider what has been called a paradox in its reasoning. Assuming the above analysis, we expect a player to follow the BI path. So, if she does not, we must revise our beliefs about her reasoning. But then, why would we assume at all that she will play BI later on? BI seems to bite itself in the tail. Consider a concrete example:

Example 7.56 ('Irrational Behaviour' of Players) Backward Induction tells us that I will play L at the start in the following game:



So, if I plays R instead, what should II conclude? There are many different options, such as 'it was just an error, and I will go back to being rational', 'I is trying to tell me that he wants me to go right, and I will surely be rewarded for that', 'I is an automaton with a general rightward tendency', and so on.

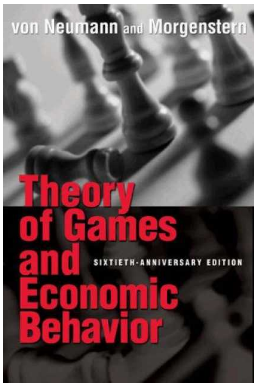
Our logical analysis so far chooses for the interpretation that agents will always play rationally from the current stage onward. But this can be doubted, and in that case, logical analysis of games also needs an account of belief revision: the way in which we change our earlier beliefs about the game, and about the other players, as we proceed.

7.11 Logic and Game Theory

Game theory emerged in the course of the 20th century as the formal study of interactive decision making. Until recently, this field was perceived as rather far removed from logic. Key figures in its early development were John von Neumann and Oskar Morgenstern, who published their influential *Theory of games and economic behavior* in 1944, starting off a new scientific discipline.

John von Neumann

Oskar Morgenstern


von Neumann and Morgenstern
Theory of Games and Economic Behavior
SIXTIETH-ANNIVERSARY EDITION

This section gives a brief introduction to the game theoretic way of thinking, in order to identify the many points of connection with logic.

First, here is a key definition that presents a different perspective on the extensive games we encountered before:

Definition 7.57 (Strategic Games) A *strategic game* consists of

(1) a finite set N of players,

(2) for each player $i \in N$ a non-empty set A_i of actions available to the player,

(3) for each player $i \in N$ a preference relation $\geq_i$ on $A = \prod_{j \in N} A_j$.

The members of $A = \prod_{j \in N} A_j$ are tuples $(a_1, \ldots, a_n)$, where a_1 is an action of the first player, a_2 an action of the second player, and so on. In a strategic game, there is no notion of temporal progression. The strategy of each player is viewed as condensed in a single action. So in a strategic game we consider the moves of all players simultaneously. The tuples in A are the possible global outcomes of the game, which can be evaluated by the players. The preference relation may also be encoded in terms of numerical utilities for players over outcomes, as explained before.

Many key notions and results in game theory work on strategic games, disregarding individual moves. This is the habitat of the matrixes for two-player games which most people probably associate with game theory.

Example 7.58 (Matching Pennies, Strategic Form) Players I and II both choose whether to show the head or tail of a coin. If the sides match, I gets 1 euro from II, if the sides are different, II gets 1 euro from I.

The matrix indicates all actions, possible outcomes, and their numerical utilities with that of I stated first:

	h	t
H	$1, -1$	$-1, 1$
T	$-1, 1$	$1, -1$

This game may be viewed as an analog of the one in 7.3. There, I (Falsifier) chose one object out of two, and then II (Verifier) chose one, with equality of the objects chosen being the criterion for winning or losing. In Matching Pennies, players choose simultaneously, or in ignorance of what the other did. This changes their powers considerably. E.g., unlike in 7.3, no one has a clear winning strategy here. The game has no Nash equilibrium in Ht, Hh, Th, Tt. The strategy pair Hh is not Nash, for Ht is better for II. Ht is not Nash, for Tt is better for I. Th is not Nash, for Hh is better for I, Tt is not Nash, for Th is better for II.

Still, we can ask ourselves what is the best one can do if one is forced to play the game of Matching Pennies repeatedly. Clearly, the answer to this is that randomly choosing between showing heads or showing tails, with equal probability, ensures that neither of the players will lose money. This motivates the following definition.

Definition 7.59 (Mixed Strategies) A mixed strategy for a player in a strategic game is a probability distribution over the player's possible actions.

Example 7.60 (Mixed Strategies for Matching Pennies) The mixed strategies $(\frac{1}{2}, \frac{1}{2})$ for both players in the Matching Pennies game form a Nash equilibrium: if one player plays this strategy, then deviation from the probability distribution $(\frac{1}{2}, \frac{1}{2})$ for II will make no difference for the outcome. For let $(p, 1-p)$ be a probability distribution for II, and assume $\frac{1}{2} < p \leq 1$. Then the good outcome Th for II will occur with probability $\frac{1}{2}p$, and the good outcome Ht with probability $\frac{1}{2}(1-p)$. The probability of a good outcome for II is $\frac{1}{2}p + \frac{1}{2}(1-p) = \frac{1}{2}$. In other words, as long as I plays $(\frac{1}{2}, \frac{1}{2})$, it makes no difference which mix II plays. This shows that $(\frac{1}{2}, \frac{1}{2})$ versus $(\frac{1}{2}, \frac{1}{2})$ is indeed a Nash equilibrium.

Exercise 7.61 Show that no other pair of mixed strategies is a Nash equilibrium for Matching Pennies. In particular, if one player plays a particular action with probability $p > \frac{1}{2}$, then the other player can exploit this by playing a pure strategy. But the resulting pair of strategies is not Nash.

Notice that the game of Matching Pennies is zero-sum: one player's gain is the other player's loss. This is not the case in the Donation game, or in the famous Prisoner's Dilemma.

The Dilemma of the Prisoners is probably the most famous example of game theory. For those who have never seen it, here is an informal description. Two players I and II are in prison, both accused of a serious crime. The prison authorities try to lure each of them into making a statement against the other. They are each promised a light sentence as a reward for getting their partner in crime convicted. If the prisoners both keep silent, they will get off with a light sentence because of lack of evidence. If one of them keeps silent but the other starts talking, the one who keeps silent is going to serve a considerable time in prison and the other is set free. If both of them talk they will both get a medium term sentence.

Example 7.62 (Prisoner's Dilemma, Strategic Form) Here is the Prisoner's Dilemma in matrix form:

	s	b
S	$2, 2$	$0, 3$
B	$3, 0$	$1, 1$

Note that the payoff function is the same as in the Donation game (Example 7.35) The difference is that the Prisoner's Dilemma game is not played sequentially.

Why is this non-zero-sum game an evergreen of game theory? Because it is a top-level description of the plight of two people, or countries, who can either act trustfully or not, with the worst outcome that of being a sucker. For an armament race version, read the two options as 'arm' or 'disarm'.

The pair of strategies Bb is the only Nash equilibrium of the game: if the other one betrays me, there is nothing better I can do than also betray. For all other strategy pairs, one of the players is better off by changing his action.

In the Prisoner's Dilemma, the players have no rational incentive to coordinate their actions, and they end up in a situation that is worse than what would have resulted from their collaboration. This notion of being 'worse of' is made precise in the following definition.

Definition 7.63 (Pareto optimum) A *Pareto optimum* of a game is an outcome that cannot be improved without hurting at least one player.

Example 7.64 (The Prisoner's Dilemma Again)

	s	b
S	$2, 2$	$0, 3$
B	$3, 0$	$1, 1$

The Pareto optima are Ss, Sb, Bs. The Nash equilibrium Bb is *not* a Pareto optimum.

Example 7.65 (Tragedy of the Commons) This was made famous by Garrett Hardin in his classic essay, still available on internet, and very much recommended:

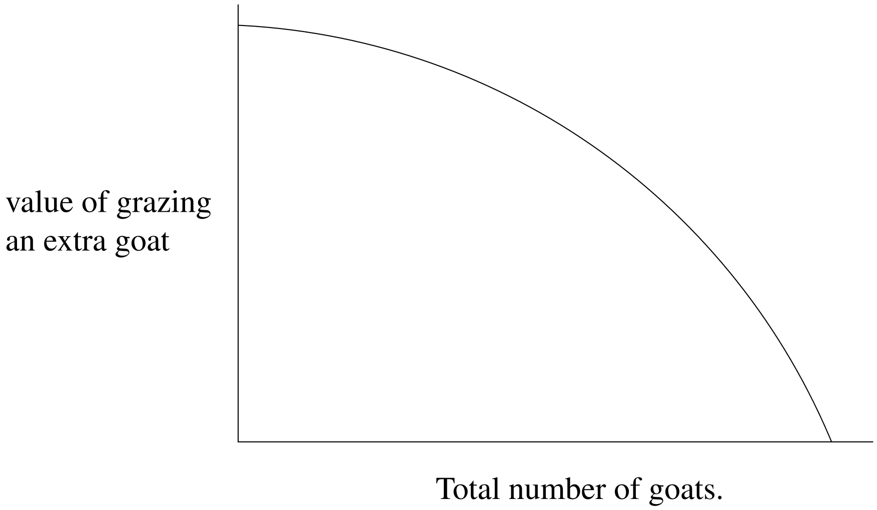
```
www.garretthardinsociety.org/articles/art_tragedy_of_
the_commons.html
```

Essentially, the Tragedy of the Commons is a multi-agent version of the Prisoner's Dilemma.

> The tragedy of the commons develops in this way. Picture a pasture open to all. It is to be expected that each herdsman will try to keep as many cattle as possible on the commons. Such an arrangement may work reasonably satisfactorily for centuries because tribal wars, poaching, and disease keep the numbers of both man and beast well below the carrying capacity of the land. Finally, however, comes the day of reckoning, that is, the day when the long-desired goal of social stability becomes a reality. At this point, the inherent logic of the commons remorselessly generates tragedy. [Har68]

Bringing more and more goats to the pasture will in the end destroy the commodity for all. Still, from the perspective of an individual herdsman it is profitable until almost the very end to bring an extra goat.

The following picture illustrates the dilemma:



Or view this as a game of an individual herdsman II against the collective I. Then the matrix is:

	m	g
M	$2, 2$	$0, 3$
G	$3, 0$	$-1, -1$

Each player has a choice between g (adding goats) and m (being moderate). Assuming that the collective is well-behaved, it pays off to be a free rider. But if everyone acts like this, system breakdown will result.

Of course, the general import of the matrices of strategic games is not the particular story *per se*, but rather their standing proxy for frequent types of social situation.

Example 7.66 ('Tragedy of the Commons' Scenario) The Tragedy of the Commons game describes a general mechanism that is rational for the one and disastrous for the many. Such mechanisms abound in the world around us:

- Citizens of Amsterdam who want cheap parking in the inner city;
- prosperous families wanting to drive bigger and bigger SUVs;
- airport hubs wanting to attract ever more air traffic;
- fishermen roaming the oceans in ever bigger fishing trawlers;
- logging companies cutting down ever more tropical forest;
- developed countries exporting their industrial waste to developing countries;
- US citizens defending the Second Amendment right to keep and bear firearms (“NRA: The largest civil-rights group ever”).

It should be noted that slight differences in payoff function result in strikingly different scenarios.

Example 7.67 (Hawk versus Dove) Being aggressive against someone who is passive is advantageous. Being passive against someone who is also passive is so-so. Being aggressive against an aggressor can be disastrous. This gives the following matrix for the ‘Hawk’ versus ‘Dove’ game, where two players have the choice between aggressive and meek behaviour:

	h	d
H	$0, 0$	$1, 3$
D	$3, 1$	$2, 2$

This example also occurs frequently in biology. What is the best behaviour for two people or animals in a single encounter? And in the long run, what will be stable populations of predators playing Hawk and prey playing Dove?

‘Hawk versus Dove’ has two Nash equilibria, viz. Hd and Dh. In neither situation can anyone better himself by unilaterally switching strategies, while in the other two, both players can.

Exercise 7.68 What are pure strategy Nash equilibria for Hawk versus Dove? (Note: ‘pure’ means that actions are either played with probability 1 or with probability 0.)

Example 7.69 (Vos Savant’s Library Game) The following story is from a column by Marilyn Vos Savant, San Francisco Chronicle, March 2002.

> A stranger walks up to you in the library and offers to play a game. You both show heads or tails. If both show heads, she pays you 1 dollar, if both tails, then she pays 3 dollars, while you must pay her 2 dollars in the two other cases. Is this game fair?

Let's put this in matrix form, with the stranger as the row player:

	h	t
H	$-1, 1$	$2, -2$
T	$2, -2$	$-3, 3$

You may think it is fair, for you can reason that your expected value equals

$$\frac{1}{4} \cdot (+1) + \frac{1}{4} \cdot (+3) + \frac{1}{2} \cdot (-2) = 0.$$

Vos Savant said the game was unfair to you with repeated play. The stranger can then play heads two-thirds of the time, which would give you an average pay-off of

$$\frac{2}{3}(\frac{1}{2} \cdot (+1) + \frac{1}{2} \cdot (-2)) + \frac{1}{3}(\frac{1}{2} \cdot (+3) + \frac{1}{2} \cdot (-2)) = -\frac{1}{6}.$$

But what if I play a different counter-strategy against this, viz. "Heads all the time"? Then my expected value would be

$$\frac{2}{3} \cdot (+1) + \frac{1}{3} \cdot (-2) = 0.$$

So, what is the fair value of this game – and should you engage in it? We will take this up again in Example 7.73 below.

Example 7.70 (Making sense) Linguistic expressions may be ambiguous, referring to more than one situation. This helps keep code short in communication, whereas unambiguous expressions tend to be elaborate and costly to process. Let A have two meanings: it can refer to situation X or Y. B is unambiguous, referring only to X, and C only to Y. The complexity of B, C is greater than that of A, in some intuitive sense. A speaker strategy is a choice of expression for each of the situations X, Y, while a hearer strategy decodes expressions into situations. Here are the possible strategies for both, in matrix form:

Speaker:

	X	Y
S_1	A	C
S_2	A	A
S_3	B	A
S_1	B	C

Hearer:

	A	B	C
H_1	X	X	Y
H_2	Y	X	Y

Let there be a known chance that situation X obtains versus Y: say $\frac{2}{3}$. First, Speaker says something, then Hearer interprets it. As for players' utilities, both prefer correct decodings to incorrect ones, and given that, less complex expressions to more complex ones. Linguistic behaviour amounts to pairs of strategies (S_i, H_j). This setting is called a *signalling game*. Is this enough to predict the observed behaviour of language users, which is that the ambiguous expression is used for the most frequent situation, whereas the less frequent situation is referred to by its unambiguous code?

'Making Sense' has two Nash equilibria, viz. (S_1, H_1) and (S_3, H_2). The first of these represents the intended outcome. The second describes a situation where the ambiguous expression is used for the less frequent situation.

The notion of a Nash equilibrium remains the same in the larger strategy space where mixed strategies are allowed. Of course, outcomes will now be computed as expected values in the obvious sense. E.g., as we have seen, to do the best they can in Matching Pennies, players should play each action 'Heads', 'Tails' with probability 0.5. This will guarantee an optimal expected value 0 for both. Here is perhaps the most celebrated result from game theory.

Theorem 7.71 (von Neuman, Nash) All finite strategic games have equilibria in mixed strategies.

Rather than give a proof, we will illustrate this for the case of games with 2×2 matrices. For a strategy pair σ, τ in equilibrium yielding value $[\sigma, \tau]$, we call ϵ a best response for I to τ if $[\epsilon, \tau] = [\sigma, \tau]$. In other words, playing ϵ rather than σ against τ does not change the payoff. Now here is a useful fact.

Fact 7.72 If the strategy pair σ, τ is in equilibrium, then each pure strategy occurring in the mixed strategy σ is also a best response for player I to τ.

Proof. If some component pure strategy S gave a lower outcome against τ then we could improve the outcome of σ itself by decreasing its probability of playing S. □

We can use this to analyze Vos Savant's library game.

Example 7.73 (Library Game, Continued from Example 7.69) In equilibrium, suppose the stranger plays Heads with probability p and Tails with $1 - p$. You play heads with probability q and Tails with probability $1 - q$. By Fact 7.72, your expected outcome against the p-strategy should be the same whether you play Heads all the time, or Tails all the time. Therefore, the following equality should hold:

$$p \cdot 1 + (1 - p) \cdot (-2) = p \cdot (-2) + (1 - p) \cdot 3.$$

Working this out yields: $p = \frac{5}{8}$. By a similar computation, q equals $\frac{5}{8}$ as well. The expected value for you is

$$\frac{5}{8} \cdot \frac{5}{8} \cdot 1 + \frac{5}{8} \cdot \frac{3}{8} \cdot (-2) + \frac{5}{8} \cdot \frac{3}{8} \cdot (-2) + \frac{3}{8} \cdot \frac{3}{8} \cdot (3) = -\frac{1}{8}.$$

Thus, the game is indeed unfavourable to you – though not for exactly the reason given by Vos Savant.

Note that probabilistic solutions, for games like Matching Pennies or the Library Game, make most sense when we think of repeated games where you can switch between Heads and Tails.

But there are also other interpretations of what it means to play a mixed strategy. For instance, by a similar computation, besides its two equilibria in pure strategies, the Hawk versus Dove game has an equilibrium with each player choosing Hawk and Dove 50% of the time. This can be interpreted biologically in terms of stable populations having this mixture of types of individual.

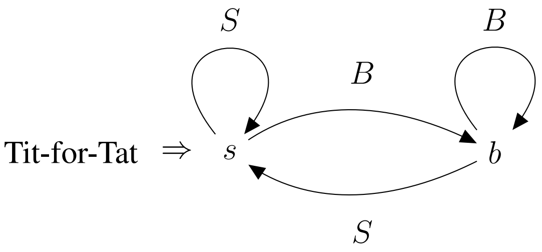
Conclusion As we said at the beginning of this chapter, our aim has not been to develop one more standard logical system. Instead, we have shown how logic and games are a natural meeting ground, where the themes of earlier chapters all return. We showed how predicate logic can be analyzed using special "logic games" of evaluation and model comparison. But we also showed how our logics of information and action apply to games in the general sense of game theory. These two directions are related: logics can be used to analyze games, but conversely games can also be used to analyze logics. This intriguing duality is far from being completely understood – but at least, you now know what it is about.

7.12 Outlook — Iterated Game Playing

Infinite games One striking trend in modern game theory is the evolutionary theory of infinitely repeated games. One source for this is the Prisoner's Dilemma game. This has only one Nash equilibrium, in which both players choose 'betray', even though both keeping silent would make both players better-off. Many amendments have been proposed since the problem was first proposed in the 1950s. In particular, it has become clear that one needs to look at repetitions of games like this, allowing for reactions to observed behaviour in the past. For then, we can punish or reward our opponents' previous behaviour. Now, fixed finite repetitions of games like Prisoner's Dilemma are of no help. A backward induction argument shows that, working back from the final play where retaliation is impossible, the 'bad' equilibrium Bb comes out best after all. But with infinite repetitions, and some natural 'discounting' of utilities of games further in the future, new equilibria emerge. An example made famous by Axelrod [Axe84] is:

> **Tit-for-Tat**: Copy one's opponents last choice, thereby giving immediate, and rancour-free, rewards and punishments.

Here is a process picture of this strategy (for player II, against player I):



As long as I sticks to S, respond with s, as soon as I plays B, respond with b, keep playing b as long as I plays B, and as soon as I plays S again, be forgiving and switch back to s.

It can be shown that (Tit-for-Tat,Tit-for-Tat) is a Nash equilibrium in the infinite Prisoner's Dilemma. Hence, cooperation is at least a stable option in the long run. The backdrop for this result are the 'folk theorems' of game theory showing that repeated versions of a game have a huge strategy space with many new equilibria. There is also a flourishing literature on showing when such a cooperative equilibrium will emerge in a population. One relevant line of research here is the learning theory of infinite games, where certain equilibria are learnt under plausible assumptions.

A complete analysis of infinite games in this sense requires the mathematics of dynamical systems with special leads from biology for setting up plausible systems equations. Such considerations over time are very rare in logic, at least so far. Sometimes, though, these considerations can be pushed back to simple scenarios that also make sense in logical analysis. Here is a nice illustration that makes sense when thinking about the stability of rules, say of some logical or linguistic practice.

Example 7.74 (Mutant Invasion) Consider a population playing some strategy S in an infinitely repeated game of encounters between 2 agents, with (S, S) a Nash equilibrium. E.g., S could be some logico-linguistic convention, like 'speaking the truth'. Now suppose that a small group of mutants enters, playing strategy F in every encounter. Let the probability that a mutant encounters another mutant be ϵ, typically a small number. Then the expected utility of any encounter for a mutant can be computed as follows:

$$\epsilon \cdot \text{utility}_M(F, F) + (1 - \epsilon) \cdot \text{utility}_M(F, S) \qquad \text{(mutant value)}$$

For members of the original population, the expectation lies symmetrically:

$$\epsilon \cdot \text{utility}_P(S, F) + (1 - \epsilon) \cdot \text{utility}_P(S, S) \qquad \text{(normal value)}$$

Here is an attractive notion of biology describing stability of a situation. A population is 'evolutionarily stable' if mutant invasions fizzle out. That is,

$$\text{for every strategy } F \neq S, \text{mutant value} < \text{normal value}.$$

By a simple calculation, this condition can be simplified, at least for 'symmetric' games where

$$\text{utility}_M(F, S) = \text{utility}_P(S, F).$$

It then becomes this qualitative notion.

Definition 7.75 A strategy S in a game is evolutionarily stable if we have both

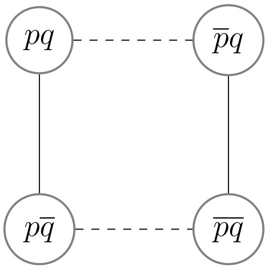
(1) (S, S) is a Nash equilibrium, and

(2) for every different best response S' to S, utility$(S', S') <$ utility(S, S').

This notion has additional bite. E.g., Tit-for-Tat, though a Nash equilibrium against itself, is not evolutionarily stable in the repeated Prisoner's Dilemma. Of course, other scenarios are possible. E.g., if mutants stick together, increasing their chances of mutual encounters, the above computations fail, and invasion may be possible after all.

7.13 Outlook — Knowledge Games

Information exchange in the sense of Chapter 5 can also be viewed as a game. We give a brief sketch. Players I and II both have a secret: I secretly knows about p and II secretly knows about q. Here is a picture of the situation (solid lines for I accessibilities, dashed lines for II accessibilities):



Both players would like to learn the other's secret, but are reluctant to tell their own. They don't want to tell their own secret without learning the secret of the other. They both have a choice: telling their own secret or not. The choice for I is that between $\pm p$ (telling whether p is the case) and $\top$ (uttering a triviality). II can choose between $\pm q$ and $\top$. The preferences for I are:

$$\top \pm q >_1 \pm p \pm q >_1 \top\top >_1 \pm p \top.$$

The preferences for II are:

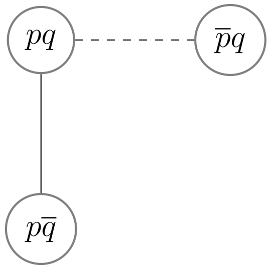
$$\pm p \top >_2 \pm p \pm q >_2 \top\top >_2 \top \pm q.$$

This situation is represented by the following strategic game:

	$\pm q$	$\top$
$\pm p$	$2, 2$	$0, 3$
$\top$	$3, 0$	$1, 1$

This game has one pure Nash equilibrium, $\top\top$.

Now consider a variation of this, where both players know that $p \vee q$. The model becomes:



Now more complex strategies make sense. Consider the following I-strategy: "If I knows that p then I keeps silent, otherwise I reveals $\neg p$". Formally:

$$(?\Box_1 p; !\top) \cup (?\neg\Box_1 p; !\neg p).$$

Exercise 7.76 Show that playing this strategy against $\top$ is an equilibrium.

7.14 Outlook — Games and Foundations

As we have seen, a game in which one of the two players has a winning strategy is called determined. Now, are all games determined? With this simple question, we are right in the foundations of set theory. Examples have been found of infinite non-determined games, but their construction turned out to depend strongly on the mathematical axioms one assumes for sets, in particular, the famous 'Axiom of Choice'. Therefore, in 1962 it has been proposed (by Jan Mycielski and Hugo Steinhaus) to turn the tables, and just *postulate* that all games are determined. This 'Axiom of Determinacy' states that all two-player games of length ω with perfect information are determined.

Since the Axiom of Choice allows the construction of non-determined two-player games of length ω, the two axioms are incompatible. From a logical point of view, the Axiom of Determinacy has a certain appeal. We have in finitary logic

$$\forall x_1 \exists x_2 \forall x_3 \exists x_4 \varphi(x_1, x_2, x_3, x_4) \vee \exists x_1 \forall x_2 \exists x_3 \forall x_4 \neg\varphi(x_1, x_2, x_3, x_4),$$

and so on, for longer $\forall\exists$ alternations of quantifiers. The infinitary version of this runs:

$$\forall G \subseteq S^{\infty} :$$

$$\forall x_1 \in S \exists x_2 \in S \forall y_1 \in S \exists y_2 \in S \forall z_1 \in S \exists z_2 \in S \ldots : (x_1, x_2, y_1, y_2, z_1, z_2, \ldots) \in G$$

iff

$$\exists x_1 \in S \forall x_2 \in S \exists y_1 \in S \forall y_2 \in S \exists z_1 \in S \forall z_2 \in S \ldots : (x_1, x_2, y_1, y_2, z_1, z_2, \ldots) \notin G.$$

But this *is* a formulation of the Axiom of Determinacy. Indeed, the Axiom of Determinacy might be viewed as 'Excluded Middle run wild', but then a gallop with beautiful mathematical consequences. There is a broad consensus today that set theory needs new axioms, but much less: which ones, and Determinacy is just one option. In any case, it may be said that games are an important source of intuitions here.

7.15 Outlook — Games, Logic and Cognition

The arena of game theory has many intriguing examples where there is a mismatch between what game theoretical considerations would predict and what actually happens if the games are played. Here is one much-discussed example:

Example 7.77 (The Ultimatum Game) Player I is shown a substantial amount of money, say 1000 euros. He is asked to propose a split of the money between himself and player II. If player II accepts the deal, they may both keep their shares, otherwise they both receive nothing. If this game is played once, a split $(999, 1)$ should be acceptable for II. After all, receiving 1 euro is better than receiving nothing. But this is not what we observe when this game is played. What we see is that II rejects the deal, often with great indignation.

Considerations about repeated play, reputation mechanisms, psychological factors, have been called to the rescue to explain what happens.

Other examples where what we observe in reality seems at odds with game theoretical rationality are the centipede games, discussed above in Examples 7.45 and 7.54. Instead of the first player immediately opting out of the game, players often show partial cooperation. Maybe they reason that it is better to cooperate for a while, and then defect later, when there is a better reward for the evil deed? It has also been suggested that this has something to do with limitations in our cognitive processing. We all can do 'first order theory of mind': imagine how other people think about reality. Some of us do 'second order theory of mind': imagine how other people think about how *we* think about reality. Very few people take the trouble to move to higher orders. But in a backward induction argument for a centipede game this is what seems to be going on, and on and on …

Summary of Things You Have Learnt in This Chapter *You have become aware of the natural fit between games and logic, in a number of areas. You have learnt to see reasoning about logical consequence (argumentation) as a game. You know how to play an evaluation game. You know the concept of a winning strategy, and you understand Zermelo's theorem and the algorithm behind it. You have learnt how to apply Zermelo's procedure to find winning strategies for finite zero-sum two-player games such as Sabotage. You know how to play model comparison games, and you know what a difference formula is. You are able to find winning strategies in bisimulation games. You understand* *the concept of a Nash equilibrium, and you are able to solve games by backward induction. Finally, you understand the basic game-theoretic notions of a strategic game and of a mixed strategy solution, and you understand how well-known strategic games like the Prisoner's Dilemma and Hawk versus Dove stand proxy for social situations.*

Further Reading A recent textbook that explores and explains the connections between games and logic is [VÏ1]. An illuminating paper on the importance of the game-theoretic perspective in logic is [Fag97].

The book that started game theory, [NM44], was already mentioned above. There are many excellent textbooks on game theory: [Str93] and [Osb04] are among our favourites. A more light-hearted introduction is [Bin92].

The use of game theory to investigate the games we play when conversing with each other in natural language is demonstrated in [Cla12].