Chapter 10  DECISIONS, ACTIONS, AND GAMES

We have now developed separate logics for knowledge update, inference, belief revision, and preference change. But concrete agency has all these entangled. A concrete intuitive setting where this happens is in games, and this chapter will explore such interactive scenarios. Our second reason for studying games here is as a concrete model of mid-term and long-term interaction over time (think of conversation or other forms of shared agency), beyond the single steps that were the main focus in our logical systems so far.

This chapter is a ‘mini-treatise’ on logic and games introducing the reader to a lively new area that draws on two traditions: computational and philosophical logic. We discuss both statics, viewing games as encoding all possible runs of some process, and the dynamics when events change games. We start with examples. Then we introduce logics for static game structure, from moves and strategies to preferences and uncertainty. About half-way, we make a turn and start exploring what our dynamic logics add in terms of update and revision steps that change game models as new information arrives. ¹ Our technical treatment is not exhaustive (we refer to further literature in many places, and van Benthem to appearA will go into more depth), but we do hope to convey the main picture.

10.1   Decisions, practical reasoning, and games

Action and preference  Even the simplest scenarios of practical reasoning involve different logical notions at the same time. Recall this example from Chapter 8:

Example   One single decision.

An agent has two alternative courses of action, but prefers one outcome to the other:

\[
\begin{array}{c}
a \\
x \leq y \\
b \\
\end{array}
\]

¹ This chapter is not a full survey of current interfaces between logic and game theory. For further material, cf. Chapter 15, van der Hoek & Pauly 2006, van Benthem 1999A, to appearA.
A typical form of reasoning here is the so-called Practical Syllogism: (i) the agent can choose either action $a$ or $b$, (ii) the agent prefers the result of $a$ over the result of $b$, and therefore, (iii) the agent will do [or maybe: should do?] $b$.

Choice of a best available option given one’s beliefs is the basic notion of rationality in philosophy, economics, and many other fields. It can help predict agents’ behaviour beforehand, or rationalize observed behaviour afterwards. And it intertwines all our notions so far: preferences, actions, and beliefs about what agents are going to do.

In decision theory, we see further roles of beliefs. Decisions may involve uncertainty about states of nature, and we choose an action with highest expected value, a weighted sum of utilities for the outcomes (cf. Chapters 8, 9). The probability distribution over states of nature represents our beliefs about the world. Here is a qualitative version:

**Example**  Deciding with an external influence.

Nature has two moves $c, d$, and an agent has moves $a, b$. Now we get combined moves:

Suppose the agent thinks Nature’s move $c$ is more likely than $d$. This turns the outcomes into an epistemic-doxastic model (Chapter 7): the epistemic range has 4 worlds, but the most plausible ones are $x, y$, and the agent’s beliefs only refer to the latter.

Thus we see entanglement of preference and belief as in Chapter 9, while also, as in Chapter 7, beliefs guide our actions. This mix is even more pronounced in games:

**Solving games by backward induction** In a multi-agent setting, behaviour is locked in place by mutual expectations. This requires an interactive decision dynamics, and standard game solution procedures like Backward Induction do exactly that. We will take this particular procedure as a running example in this chapter – not because we fully endorse it, but because it demonstrates many logical issues so beautifully:
Example  Reasoning about interaction.

In the following game (cf. Chapters 1, 9), players’ preferences are encoded in utility values, as pairs (value of $A$, value for $E$). Backward Induction tells player $E$ to turn left at her turn, just as in our single decision case, which gives $A$ a belief that this will happen, and so, based on this belief about his counter-player, $A$ should turn left at the start:

```
A
/  \      
0, 100 99, 99
\  /    |
1, 0    E
```

But why should players act this way? The reasoning is again a mixture of all notions so far. $A$ turns left since she believes that $E$ will turn left, and then her preference is for grabbing the value 1. Once more, practical reasoning intertwines action, preference, and belief.

Here is the rule that drives all this, at least when preferences are encoded numerically:

**Definition**  Backward Induction algorithm.

Starting from the leaves, one assigns values for each player to each node, using the rule

Suppose $E$ is to move at a node, and all values for daughters are known. The $E$-value is the maximum of all the $E$-values on the daughters, the $A$-value is the minimum of the $A$-values at all $E$-best daughters. The dual calculation for $A$’s turns is completely analogous.

This seems obvious and easy to apply, telling us players’ best course of action (Osborne & Rubinstein 1994). And yet, it is packed with assumptions. We will perform a logical deconstruction later on, but for now, note that (a) the rule assumes the same reasoning by both players, (b) one makes worst-case assumptions about opponents, taking a minimum value when it is not our turn, (c) the rule changes its interpretation of values: at leaves they encode plain utilities, higher up in the game tree, they represent expected utilities. Thus, in terms of earlier chapters, Backward Induction is a mechanism for generating a plausibility order among histories, and hence, it relates our models in Chapter 7 to those of Chapter 9: betterness and plausibility order become intertwined in a systematic manner.
We will look at this reasoning in much more detail using our dynamic logics. But for now, we step back, and look at static logic of games \textit{ab initio}. We first consider pure action structure, adding preference and epistemic structure for realistic games in due course.

\section*{10.2 Basic modal action logic of extensive games}

In this chapter, we will view extensive games as multi-agent processes. Technically, such structures are models for a modal logic of computation in a straightforward sense:

\textit{Definition} Extensive game forms.

An \textit{extensive game form} is a tree \( M = (\text{NODES}, \text{MOVES}, \text{turn}, \text{end}, V) \) with binary transition relations from the set \text{MOVES} pointing from parent to daughter nodes. Also, non-final nodes have unary proposition letters \text{turn}, indicating the player whose turn it is, while \text{end} marks end nodes. The valuation \( V \) can also interpret other local predicates at nodes, such as utility values for players or more ad-hoc properties of game states.

Henceforth, we will restrict attention to extensive games in the modal style.

\textit{Basic modal logic} Extensive game trees support a standard modal language:

\textit{Definition} Modal game language and semantics.

\textit{Modal formulas} are interpreted at nodes \( s \) in game trees \( M \). Labeled modalities \(<a>\varphi\) express that some move \( a \) is available leading to a next node in the game tree satisfying \( \varphi \). Proposition letters true at nodes may be special-purpose constants for game structure, such as indications for turns and end-points, but also arbitrary local properties.

In particular, modal operator combinations now describe potential interaction:

\textit{Example} Modal operators and strategic powers.

Consider a simple 2-step game like the following, between two players \( A, E \):

\begin{center}
\begin{tikzpicture}
  \node (A) at (0,1) {A}
  child {node (E) {E} edge from parent node [left] {a}}
  child {node (E1) {E} edge from parent node [left] {b}}

  \node (2) at (-1,-2) {2}
  child {node (E2) {E} edge from parent node [left] {c}}
  child {node (E3) {E} edge from parent node [left] {d}}

  \node (1) at (1,-2) {1}
  child {node (E4) {E} edge from parent node [left] {c}}
  child {node (E5) {E} edge from parent node [left] {d}}

  \node (p) at (2,-4) {p}
  child {node (p1) {p} edge from parent node [left] {p}}
\end{tikzpicture}
\end{center}
Player $E$ clearly has a strategy for making sure that a state is reached where $p$ holds. This feature of the game is directly expressed by the modal formula $[a]<d>p \land [b]<c>p$.

More generally, letting move stand for the union of all relations available to players, in the preceding game, the modal operator combination

$$[\text{move-A}]<\text{move-E}>\varphi$$

says that, at the current node, player $E$ has a strategy for responding to $A$’s initial move which ensures that the property expressed by $\varphi$ results after two steps of play.  

**Excluded middle and determinacy** Extending this observation to extensive games up to some finite depth $k$, and using alternations $\Box\Diamond\Box\Diamond\ldots$ of modal operators up to length $k$, we can express the existence of winning strategies in fixed finite games. Indeed, given this connection, with finite depth, standard logical laws have immediate game-theoretic import. In particular, consider the valid law of excluded middle in the following modal form

$$\Box\Diamond\Diamond\ldots\varphi \lor \neg\Box\Diamond\Diamond\ldots\neg\varphi$$

or after some logical equivalences, pushing the negation inside:

$$\Box\Diamond\Diamond\ldots\varphi \lor \Diamond\Box\Diamond\Diamond\ldots\neg\varphi,$$

where the dots indicate the depth of the tree. Here is its game-theoretic content:

**Fact** Modal excluded middle expresses the determinacy of finite games.

Here, determinacy is the fundamental property of many games that one of the two players has a winning strategy. This need not be true in infinite games (players cannot both have one, but maybe neither has), and Descriptive Set Theory has deep results in this realm.

**Zermelo’s theorem** This brings us to perhaps the oldest game-theoretic result, predating Backward Induction, proved by Ernst Zermelo in 1913 for zero-sum games, where what one players wins is lost by the other (‘win’ versus ‘lose’ is the typical example):

**Theorem** Every finite zero-sum 2-player game is determined.

---

2 Thus, one can express having winning strategies, losing strategies, and so on. Links between such powers and logical operator combinations are crucial to *logic games* (van Benthem 2007D).
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 A, and colour the other end nodes white, being wins for E. Then extend this colouring stepwise as follows:

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

(a) if player $A$ is to move, and at least one child is black:

\[
\text{colour } s \text{ black; if all children are white, colour } s \text{ white}
\]

(b) if player $E$ is to move, and at least one child is white:

\[
\text{colour } s \text{ white; if all children are black, colour } s \text{ black.}
\]

This simplified Backward Induction eventually colours all nodes black where player $A$ has a winning strategy, while colouring those where $E$ has a winning strategy white. And the reason for its correctness is that a player has a winning strategy at one of her turns iff she can make a move to at least one daughter node where she has a winning strategy.

Zermelo's Theorem is widely applicable. Recall the Teaching Game from Chapter 1, our first example demonstrating the logical flavour of multi-move interaction:

Example Teaching, the grim realities.

A Student located at position $S$ in the next diagram wants to reach the escape $E$ below, while the Teacher wants to prevent him from getting there. Each line segment is a path that can be traveled. In each round of the game, the Teacher first cuts one connection, anywhere, and the Student must then travel one link still open at his current position:

![Diagram of the Teaching Game](attachment:teaching_game.png)

General education games like this arise on any graph with single or multiple lines.  

We now have a principled explanation why either Student or Teacher has a winning strategy, since this game is two-player zero sum and of finite depth.

---

3 Gierasimczuk, Kurzen & Velazquez 2009 give connections with real teaching scenarios.
10.3 Fixed-point languages for equilibrium notions

A good test for logical languages is their power of representing basic proofs. Our modal language cannot express the generic Zermelo argument. Starting from atomic predicates \( \text{win}_i \) at end nodes marking which player has won, we inductively defined new predicates \( \text{WIN}_i \) (‘player \( i \) has a winning strategy at the current node’ – here we use \( i, j \) to indicate the opposing players) through a recursion

\[
\text{WIN}_i \iff (\text{end} \land \text{win}_i) \lor (\text{turn}_i \land \langle \text{move}-i \rangle \text{WIN}_i) \lor (\text{turn}_j \land [\text{move}-j] \text{WIN}_i)
\]

Here \( \text{move-}x \) is the union of all moves for player \( x \). This inductive definition for \( \text{WIN}_i \) is definable as a fixed-point expression in a richer system that we saw already in Chapter 4.

The modal \( \mu \)-calculus extends basic modal logic with operators for smallest and greatest fixed-points (cf. Bradfield & Stirling 2006, Blackburn, de Rijke & Venema 2000):

Fact The Zermelo solution is definable as follows in the modal \( \mu \)-calculus:

\[
\text{WIN}_i = \mu p^* (\text{end} \land \text{win}_i) \lor (\text{turn}_i \land \langle \text{move}-i \rangle p) \lor (\text{turn}_j \land [\text{move}-j]p)^5
\]

The \( \mu \)-calculus has many uses in games: see below, and also Chapter 15 below. 6 7

Other notions: forcing Winning is just one aspect. Games are all about control over outcomes that players have via their strategies. This suggests further logical notions:

---

4 Zermelo’s Theorem implies that in Chess, one player has a winning strategy, or the other a non-losing strategy, but almost a century later, we do not know which: the game tree is just too large. But the clock is ticking for Chess. Recently, for the game of Checkers, 15 years of computer verification yielded the Zermelo answer: the starting player has a non-losing strategy.

5 Crucially, the defining schema has only positive occurrences of the predicate \( p \).

6 Our smallest fixed-point definition reflects the iterative equilibrium character of game solution (Osborne & Rubinstein 1994). In infinite games, we would switch to greatest fixed-points defining a largest predicate satisfying the recursion. This is also an intuitive view of strategies: they are not built from below, but can be used as needed, and remain at our service as fresh as ever next time we need them – the way we think of doctors. This is the perspective of co-algebra (Venema 2007). Greatest fixed-points seem the best match to the equilibrium theorems in game theory.

7 As we shall see later, there are also other inductive formats for defining game solutions.
**Definition** Forcing modalities \( \{i\} \varphi \).

\( M, s \models \{i\} \varphi \) iff player \( i \) has a strategy for the subgame starting at \( s \) which guarantees that only end nodes will be reached where \( \varphi \) holds, whatever the other player does.

\[ \square \]

**Fact** The modal \( \mu \)-calculus can define forcing modalities for games.

**Proof** The modal fixed-point formula \( \{i\} \varphi = \mu p \cdot (\text{end} \land \varphi) \lor (\text{turn}_i \land <\text{move}-i>p) \lor (\text{turn}_j \land [\text{move}-j]p) \) defines the existence of a strategy for \( i \) making proposition \( \varphi \) hold at the end of the game, whatever the other player does.\(^8\)

\[ \square \]

Analogously, \( \text{COOP} \varphi \leftrightarrow \mu p \cdot (\text{end} \land \varphi) \lor (\text{turn}_i \land <\text{move}-i>p) \lor (\text{turn}_j \land <\text{move}-j>p) \) defines the existence of a cooperative outcome \( \varphi \), just by shifting some modalities.\(^9\)

### 10.4 Explicit dynamic logic of strategies

So far, we left out a protagonist in our story. Strategies with successive interactive moves are what drives rational agency over time. Thus, it makes sense to move them explicitly into our logics, to state their properties and reason with long-term behaviour. Our discussion follows van Benthem 2007C, whose main tool is *propositional dynamic logic* (PDL), used in analyzing conversation in Chapter 3 and common knowledge in Chapter 4. PDL is an extension of basic modal logic designed to study effects of imperative computer programs constructed using (a) sequential composition \( ; \), (b) guarded choice \( \text{IF} \ldots \text{THEN} \ldots \text{ELSE} \ldots \), and (c) guarded iterations \( \text{WHILE} \ldots \text{DO} \ldots \). We recall some earlier notions:

**Definition** Propositional dynamic logic.

The *language of PDL* defines formulas and programs in a mutual recursion, with formulas denoting sets of worlds (local conditions on states of the process), while programs denote binary transition relations between worlds, recording pairs of input and output states for their successful terminating computations. Programs are created from

atomic actions (‘moves’) \( a, b, \ldots \) and tests \( ? \varphi \) for arbitrary formulas \( \varphi \).\(^{10}\)

---

\(^8\) We can also change the definition of \( \{i\} \varphi \) to enforce truth of \( \varphi \) at all intermediate nodes.

\(^9\) This is also definable in PDL by \( <(?\text{turn}_i \cdot \text{move}-i) \cup (?\text{turn}_j \cdot \text{move}-j)^* > (\text{end} \land \varphi) \).

\(^{10}\) Please note: these PDL tests inside one model are not the question actions of Chapter 6.
using the three operations of ; (interpreted as sequential composition),
∪ (non-deterministic choice) and * (non-deterministic finite iteration).

Formulas are as in our basic modal language, but with modalities [π]ϕ saying that ϕ is true after every successful execution of the program π starting at the current world.

The logic PDL is decidable, and it has a simple complete set of axioms. This system can say much more about games. In our first example, the move relation was a union of atomic relations, and the pattern for the winning strategy was [a∪b]<c∪d>p. Admittedly, PDL focuses on terminating programs, and once more, we restrict attention to finite games.

**Strategies as transition relations** Strategies in game theory are partial functions on players’ turns, given by instructions of the form “if she plays this, then I play that”. More general strategies are binary transition relations with more than one best move. This view is like plans that agents have in interactive settings. A plan can be very useful when it constrains my moves, without fixing a unique course of action. Thus, on top of the hard-wired move relations in a game, we now get defined further relations, corresponding to players’ strategies, and these definitions can often be given explicitly in a PDL-format.

In particular, in finite games, we can define an explicit version of the earlier forcing modality, indicating the strategy involved – without recourse to the modal µ-calculus:

**Fact** For any game program expression σ, PDL can define an explicit forcing modality {σ, i}ϕ stating that σ is a strategy for player i forcing the game, against any play of the others, to pass only through states satisfying ϕ.

**Proof** The formula [((?turn_E ; σ) ∪ (?turn_A ; move-A))]* ϕ defines the forcing.

Here is a related observation (cf. van Benthem 2002A). Given relational strategies for two players A, E, we get to a substructure of the game described like this:

**Fact** Outcomes of running joint strategies σ, τ can be described in PDL.

**Proof** The formula [((?turn_E ; σ) ∪ (?turn_A ; τ))]* (end → ϕ) does the job.

On a model-by-model basis, the expressive power of PDL is high (Rodenhäuser 2001). Consider any finite game M with strategy σ for player i. As a relation, σ is a finite set of ordered pairs (s, t). Assume that we have an ‘expressive model’ M, where states s are
definable in our modal language by formulas \( \text{def}_s \). \(^{11}\) Then we define pairs \((s, t)\) by formulas \( \text{def}_s; a; \text{def}_t \), with \(a\) the relevant move, and take the relevant union:

Fact  In expressive finite extensive games, all strategies are \(PDL\)-definable. \(^{12}\)

The operations of \(PDL\) can also describe combination of strategies (van Benthem 2002A). In all, propositional dynamic logic does a good job in defining explicit strategies in simple extensive games. In what follows, we extend it to deal with more realistic game structures, such as preferences and imperfect information. But there are alternatives. Van Benthem 2007C is a survey and defense of many kinds of logic with explicit strategies. \(^{13}\)

### 10.5 Preference logic and defining backward induction

Real games go beyond game forms by adding preferences or numerical utilities over outcomes. Defining the earlier Backward Induction procedure for solving extensive games has become a benchmark for game logics — and many solutions exist:

Fact  The Backward Induction path is definable in modal preference logic.

Many solutions have been published by logicians and game-theorists, cf. de Bruin 2004, Harrenstein 2004, van der Hoek & Pauly 2006. We do not state an explicit \(PDL\) solution here, but we give one version involving the modal preference language of Chapter 9:

\[
<pref_i> \varphi: \text{ player } i \text{ prefers some node where } \varphi \text{ holds to the current one.}
\]

\(^{11}\) This can be achieved using temporal past modalities to describe the history up to \(s\).

\(^{12}\) As for infinite games, the modal \(\mu\)-calculus is an extension of \(PDL\) that can express existence of infinite computations — but it has no explicit programs. One would like to add infinite strategies like ‘keep playing \(a\)’ that give infinite \(a\)-branches for greatest fixed-point formulas \(\nu p \cdot <a> p\).

\(^{13}\) For a temporal \(STIT\)-type logic of strategies in games with simultaneous actions, cf. Broersen 2009. (See also Herzig & Lorini 2010 on recent \(STIT\) logics of agents’ powers.) Concurrent action without explicit strategies occurs in Alternating Temporal Logic (Alur, Henzinger & Kupferman 1997) with an epistemic version \(ATEL\) in van der Hoek & Wooldridge 2003. Ågotnes, Goranko & Jamroga 2007 discuss \(ATL\) with strategies, van Otterloo 2005 high-lights strategies in \(ATEL\). An alternative might be versions of \(PDL\) with structured simultaneous actions, as suggested earlier.
The following result from van Benthem, van Otterloo & Roy 2006 defines the backward induction path as a unique relation $\sigma$ by a frame correspondence for a modal axiom on finite structures. The original analysis assumed that all moves are unique:

**Fact** The BI strategy is the unique relation $\sigma$ satisfying the following modal axiom for all propositions $p$ – viewed as sets of nodes – for all players $i$:

$$(\text{turn}_i \land <\sigma>(\text{end} \land p)) \rightarrow [\text{move-}i]<\sigma>(\text{end} \land <\text{pref}_i> p).$$

**Proof** The argument is by induction on the depth of finite game trees. The crux here is that the given modal axiom expresses the following form of ‘maximin’ Rationality:

No alternative move for the current player $i$ guarantees a set of outcomes – via further play using $\sigma$ – with a higher minimal value for $i$ than the outcomes that result from playing $\sigma$ all the way down the tree.

The typical picture to keep in mind here, and also in some later analyses, is this:

```
            x
           / \   
          /   \  
         /     \ 
        /       
       y       z
```

via $\sigma$

via $\sigma$

More precisely, the stated modal axiom is equivalent by a modal frame correspondence (cf. Chapter 2) to the following *confluence property* for action and preference:

$$\forall x \forall y ((\text{Turn}(x) \land x \sigma y) \rightarrow \forall z (x \text{ move } z \rightarrow \forall u ((\text{end}(u) \land y \sigma^+ u) \rightarrow \exists v (\text{end}(v) \land z \sigma^+ v \land v \leq_i u))))$$

This $\forall\forall\forall\exists$ format expresses precisely the stated property about minimum values. \(\square\)

---

14 Confluence is reminiscent of the grid cells in Chapter 2, that could give logics high complexity. Little is known about the computational effects of rationality principles on our game logics.
The analysis also works for the relational version of Backward Induction, where the strategy connects a node to all daughters (one or more) with maximal values for the active player. Then $\sigma$ is the largest subrelation of the move relation with the stated property.

**Alternatives** Our logical analysis can also deal with other relational versions of Backward Induction, that make alternative assumptions about players’ behaviour. In particular, a more standard (though weaker) form of Rationality is ‘avoiding dominance’:

No alternative move for the current player $i$ guarantees outcomes via further play using $\sigma$ that are all strictly better for $i$ than the outcomes resulting from starting at the current move and then playing $\sigma$ all the way down the tree.

This time, the modal axiom is

$$(\text{turn}_i \land <\sigma> [\text{end} \rightarrow p]) \rightarrow [\text{move-}i]<\sigma^*>(\text{end} \land <\text{pref}>p).$$

and the corresponding formula is the $\forall\forall\exists\exists$ form

$$\forall x \forall y ((\text{Turn}_i(x) \land x \sigma y) \rightarrow \forall z (x \text{ move } z \rightarrow \exists u \exists v (\text{end}(u) \land \text{end}(v) \\
\land y \sigma^* v \land z \sigma^* u \land u \leq_i v)))$$

This version is technically a bit more convenient (van Benthem & Gheerbrant 2010).

We will return to this scenario for game solution later on, with a new dynamic analysis.

### 10.6 Epistemic logic of games with imperfect information

The next level of static structure gives up perfect information, a presupposition so far. In games with *imperfect information*, players need not know where they are in the tree. Think of card games, private communication, or real life with bounds on memory or observation. Such games have ‘information sets’: equivalence classes of epistemic relations $\sim_i$ between nodes that players $i$ cannot distinguish, as in Chapters 2–5. Van Benthem 2001 points out how these games model an epistemic action language with knowledge operators $K_i\varphi$ interpreted in the usual manner as ‘$\varphi$ is true at all nodes $\sim_i$-related to the current one’.

**Example** Partial observation in games.

In the following imperfect information game, the dotted line marks player $E$’s uncertainty about her position when her turn comes. She does not know the initial move played by $A$. 

---

12
Maybe \( A \) put his move in an envelope, or \( E \) was otherwise prevented from seeing: \(^{15}\)

Structures like this interpret a combined dynamic-epistemic language. For instance, after \( A \) plays move \( c \) in the root, in both middle states, \( E \) knows that playing \( a \) or \( b \) will give her \( p \) – as the disjunction \( <a>p \lor <b>p \) is true at both middle states:

\[
K_E(<a>p \lor <b>p)
\]

On the other hand, there is no specific move of which \( E \) knows at this stage that it will guarantee a \( p \)-outcome – and this shows in the truth of the formula

\[
\neg K_E<a>p \land \neg K_E<b>p
\]

Thus, \( E \) knows de dicto that she has a strategy which guarantees \( p \), but she does not know, de re, of any specific strategy that it guarantees \( p \). Such finer distinctions are typical for a language with both actions and knowledge for agents. \(^{16}\)

**Special games and modal axioms** We can analyze special kinds of imperfect information game using modal frame correspondences. Recall the axiom \( K_i[a]q \rightarrow [a]K_iq \) of Perfect Recall in Chapters 3, 4, that allowed an interchange of knowledge and action modalities. Again, it expresses a semantic confluence – this time, between knowledge and action:

**Fact** The axiom \( K_i[a]p \rightarrow [a]K_ip \) holds for all propositions \( p \) iff

\[
M \text{ satisfies } \forall xyz: ((x R_ay \land y \sim_i z) \rightarrow \exists u (x \sim_i u \land u R_ay))
\]


\(^{15}\) An imperfect information game comes without an explicit informational scenario attached. We will return later to the natural issue how given epistemic game trees may arise.

\(^{16}\) You may know that the ideal partner for you is around on the streets, but you might never convert this \( K3 \) combination into \( 3K \) knowledge that some particular person is right for you.
The reader may want to check in a diagram how the given Confluence property guarantees the truth of $K_i[a]φ \rightarrow [a]K_iφ$, starting from an upper-left world $x$ verifying $K_i[a]φ$.

Perfect Recall says that, if the last observation has not provided differential information, present uncertainties must have come from past ones. (Chapter 11 has more general versions than the one we show here.) Similar analyses work for memory bounds, and further observational powers (van Benthem 2001). For instance, as a converse to Perfect Recall, agents satisfy No Miracles when their current epistemic uncertainty between worlds $x$, $y$ can only disappear by observing subsequent events on $x$ and on $y$ that they can distinguish. Incidentally, the game we just gave satisfies Perfect Recall, but No Miracles fails, since $E$ suddenly knows where she is after she played her move. We will discuss these properties in greater generality in the epistemic-temporal logics of Chapter 11.

**Uniform strategies** Another striking feature of the above game with imperfect information is its non-determinacy. $E$’s playing the opposite of player $A$ was a strategy guaranteeing outcome $p$ in the underlying perfect information game without an epistemic uncertainty in the middle – but it is useless now, since $E$ cannot tell what $A$ played. Game theorists allow only *uniform strategies* here, prescribing the same move at indistinguishable nodes. But then no player has a winning strategy in our game, when we interpret $p$ as ‘$E$ wins’ (and hence $¬p$ as a win for $A$): $A$ did not have one to begin with, and $E$ loses hers.

As for explicit strategies, we can add PDL-style programs to the epistemic setting. But there is a twist. We need the knowledge programs of Fagin et al. 1995, whose only test conditions for actions are knowledge statements. In such programs, moves for an agent are guarded by conditions that the agent knows to be true or false. Thus, knowledge programs

---

17 Our analysis restates that of Halpern & Vardi in epistemic-temporal logic (Fagin et al. 1995).
18 The game does have probabilistic solutions in mixed strategies: it is like Matching Pennies. Both players should play both moves with probability $\frac{1}{2}$, for an optimal outcome 0.
define uniform strategies, where a player always chooses the same move at game nodes that she cannot distinguish epistemically. A converse also holds, given some conditions on expressive power of models for defining nodes in the tree (van Benthem 2001):

**Fact** On expressive finite games of imperfect information, the uniform strategies are precisely those definable by knowledge programs in epistemic PDL.

**Several kinds of knowledge in games** Imperfect information reflects processing limitations of players, resulting in ignorance of the past. But even perfect information games leave players ignorant of the future, perhaps because of ignorance about the kind of player one is up against. These richer views of knowledge will return later on in this chapter.

### 10.7 From statics to dynamics: DEL-representable games

Now we make a switch. So far, we have used static modal-preferential-epistemic logics to describe given game trees. Now we move to dynamic logics in the sense of this book, where games, or their associated models, can change because of triggering events. As a first step, we analyze how a given game might have come about through some dynamic process – the way we see a dormant volcano but imagine the tectonic forces that shaped it. We provide two illustrations, linking games of imperfect information first to DEL (cf. Chapter 4) and then to epistemic-temporal logics (Chapter 11). After that, we will explore many further dynamic scenarios in the analysis of games, based on earlier chapters.

**Imperfect information games and dynamic-epistemic logic** The reader will long have seen a link with the logic DEL of Chapter 4. Which imperfect information games make sense in some underlying informational process, as opposed to mere sprinkling of uncertainty links over trees? Consider any finite game as the domain of an event model, with preconditions on occurrence of moves encoded in the tree structure. Also assume that we are given observational limitations of players over these moves. We can then decorate the game tree with epistemic links through iterated product update, as in DEL update evolution:

**Example** Updates during play: propagating ignorance along a game tree.

When moving in the following game tree, players can distinguish their own moves, but not all moves of their opponents – as described in the accompanying event model:
Here are the successive $DEL$ updates that create the uncertainty links in the tree:

1. **tree level 1**
   - Node A
   - Node E
   - Node E

2. **tree level 2**
   - Node A
   - Node E
   - Node E
   - Node E

3. **tree level 3**
   - Node A
   - Node E
   - Node E
   - Node E

The resulting annotated tree is the following imperfect information game:

In Chapter 11, we will characterize the trees that arise from $DEL$-style update evolutions, in terms of some properties whose definitions we postpone until there:

**Theorem** An extensive game is isomorphic to an iterated update model $Tree(M, E)$ over some epistemic event model $E$ iff it satisfies, for all players, (a) Perfect Recall, (b) No Miracles, (c) Bisimulation Invariance for the domains of moves.

**10.8 Future uncertainty, procedural information, and temporal logic**

As we have said, knowledge or ignorance in games has several senses. One is *observation uncertainty*: players may not be able to observe all moves completely, and hence need not know where they are in the game. This is the past-oriented view of $DEL$. But there is also future-oriented *expectation uncertainty*. In general, players have only limited procedural information about what will happen, and Chapters 5, 11 show that this, too, is a basic
notion of information about the process agents are in. Future-oriented knowledge need not reduce to uncertainty between local nodes. It suggests uncertainty between whole histories, or players’ strategies: i.e., whole ways in which the game might evolve.  

**Branching epistemic temporal models** The following structure is common to many fields (cf. Chapter 11 below for details). In tree models for branching time, legal histories \( h \) are possible evolutions of a game. At each stage, players are in a node \( s \) on some history whose past they know completely or partially, but whose future is yet to be revealed:

\[
\text{This can be described in an action language with knowledge, belief, and added temporal operators. We first describe games of perfect information (about the past):}
\]

(a) \( M, h, s \models F_a \phi \iff s^<a> \text{ lies on } h, \) and \( M, h, s^<a> \models \phi \)

(b) \( M, h, s \models P_a \phi \iff s = s' \cap <a>, \) and \( M, h, s' \models \phi \)

(c) \( M, h, s \models \Diamond_i \phi \iff M, h', s \models \phi \text{ for some } h' \text{ equal for } i \text{ to } h \text{ up to stage } s. \)

Now, as moves are played publicly, players make public observations of them, leading to an epistemic-temporal version of our system **PAL** in Chapter 3:

**Fact** The following valid principle is the temporal equivalent of the key **PAL** recursion axiom for public announcement: \( F_a \Diamond \phi \leftrightarrow (F_a T \land \Diamond F_a \phi). \)

This principle will return as a recursion law for **PAL** with protocols in Chapter 11.

**Excursion: trading future for current uncertainty** Again, there is a reconstruction closer to the local dynamics of **PAL** and **DEL**. Intuitively, each move by a player is a public announcement that changes the current game. Here is a folklore observation converting global uncertainty about the future into local uncertainty about the present:

---

19 This section is from van Benthem 2004A on update and revision in game trees.

20 As in our earlier modal-epistemic analysis, this expresses a form of Perfect Recall.
Fact Trees with future uncertainty are isomorphic to trees with current uncertainties.

Proof Given any game tree $G$, assign epistemic models $M_s$ to each node $s$ whose domain is the set of histories passing through $s$ (all share the same past up to $s$), letting the agent be uncertain about all of them. Worlds in these models may be seen as pairs $(h, s)$ with $h$ any history passing through $s$. This will cut down the current set of histories in just the right manner. The above epistemic-temporal language matches this construction.

10.9 Intermezzo: three levels of game analysis, from ‘thin’ to ‘thick’

At this point, it may be useful to distinguish three levels at which games give rise to models for logics. All three come with their own intuitions, both static and dynamic.

Level One is the most immediate: extensive game trees are models for modal languages, with nodes as worlds, and accessibility relations for actions, preferences, and uncertainty.

Level Two looks at extensive games as branching tree models with nodes and histories, supporting richer epistemic-temporal(-preferential) languages. The difference with Level One is slight in finite games, where histories correspond to end-points. But the intuitive step is clear, and Level Two cannot be reduced when game trees are infinite. But even this is not enough. Consider hypotheses about the future, involving procedural information about strategies. I may know that I am playing against either a simple automaton, or a sophisticated learner. Modeling this may go beyond epistemic-temporal models:

Example Strategic uncertainty.

In the following simple game, let $A$ know that $E$ will play the same move throughout:

![Diagram](attachment:image.png)

Then all four histories are still possible. But $A$ only considers two future trees, viz.

![Diagram](attachment:image.png)
In longer games, this difference in modeling can be highly important, since observing only one move by \( E \) will tell \( A \) exactly what \( E \)’s strategy will be in the whole game.

To model these richer settings, one needs full-fledged Level Three game models.

**Definition** Epistemic game models.

*Epistemic game models* for an extensive game \( G \) are epistemic models \( M = (W, \sim, V) \) whose worlds carry local information about all nodes in \( G \), plus strategy profiles: total specifications of each player’s behaviour throughout the game. Players’ information about structure and procedure is encoded by uncertainty relations \( \sim \), between the worlds.

Partial knowledge about strategies is encoded in the set of profiles represented in such a model. And observing moves telling me which strategy you are following leads to dynamic update of the model, in the sense of our earlier chapters. Level-Three models are a natural limit for agency, and they are the usual models in the epistemic foundations of game theory (cf. Geanakoplos 1992, Bonanno & Sinischalci 1999, Stalnaker 1999, Halpern 2003). Also, they are congenial to modal logicians because of their abstract worlds flavour.

Even so, our preference throughout this book has been for ‘thin’ epistemic models plus explicit dynamics over ‘thick’ models encoding events and much else inside worlds (cf. Chapters 2, 4, 9). This lightness is indeed a major attraction of our dynamic framework for information flow. Hence, in the rest of this chapter we keep things simple, discussing issues at the thinnest level where they make sense: histories usually suffice.

### 10.10 Game change: public announcements, promises and solving games

Now let us look at actual transformations of given games, and explicit triggers for them.

**Promises and intentions** One can break the impasse of a bad Backward Induction solution by changing the game through making promises (van Benthem 2007G).

**Example** Promises and game change.

In the following game, discussed before, the bad Nash equilibrium \((1, 0)\) can be avoided by \( E \)’s *promise* that she will not go left. This public announcement eliminates histories (we
can make this binding by a fine on infractions)\(^\text{21}\) – and the new equilibrium \((99, 99)\) results, making both players better off by restricting the freedom of one of them:

\[
\begin{array}{ccc}
A & \rightarrow & A \\
1, 0 & & 1, 0 \\
0, 100 & & 99, 99 \\
& & 99, 99 \\
\end{array}
\]

Van Otterloo 2005 has a dynamic logic of strategic enforceability, where games change by announcing intentions or preferences. Game theory has much more sophisticated analyses of such scenarios, including the study of ‘cheap talk’ (Osborne & Rubinstein 1994). One can also add new moves when trying to change a game in some desired direction, and indeed, there is a whole family of game transformations that make sense.\(^\text{22}\)

**Modal action logic and strategies** Our methods from Chapters 3, 4 apply:

**Theorem** The modal action logic of games plus public announcement is completely axiomatized by the modal game logic chosen, the recursion axioms of PAL for atoms and Booleans, plus the following law for the move modality:

\[
<!P><a>\varphi \leftrightarrow (P \land <a>(P \land <!P>\varphi)).
\]

Using PDL for strategies in games as before, this leads to a logic PDL+PAL adding public announcements \([!P]\). The following result uses the closure of PDL under relativization to definable submodels, both in its propositional and its program parts, with a recursive operation \(\pi|P\) for programs \(\pi\) that wraps every atomic move \(a\) in tests to \(?P; a; ?P\).

**Theorem** PDL+ PAL is axiomatized by merging their separate laws while adding the following reduction axiom: \([!P](\sigma)|\varphi \leftrightarrow (P \rightarrow (\sigma|P)[!P]\varphi)\).\(^\text{23}\)

\(^{21}\) Alternatively, this changes utilities and preferences to give Backward Induction a new chance.

\(^{22}\) One might put all game changes beforehand in one grand *Super Game*, like the Supermodel of Chapters 3, 4 – but that would lose the flavour of what happens in a stepwise manner.

\(^{23}\) The axiom derives what old plan I should have had in \(G\) to run a given plan in the new game \(G’\). But usually, I have a plan \(\sigma\) in \(G\) to get effect \(\varphi\). Now \(G\) changes to \(G’\) with more or fewer moves. How should I revise \(\sigma\) to get some related effect \(\psi\) in \(G’\)? This seems harder, and even the special
There are also extended versions with epistemic preference languages.

**Solving games by announcements of rationality** Here is a more foundational use of logical dynamics, where public announcements act as ‘reminders’ driving a process of deliberation. Van Benthem 2007F makes the solution process of extensive games itself the focus of a PAL style analysis. Let us say that, at a turn for player \( i \), a move \( a \) is dominated by a sibling \( b \) (a move available at the same decision point) if every history through \( a \) ends worse, in terms of \( i \)’s preference, than every history through \( b \). Now define:

**Rationality** No player chooses a strictly dominated move.

This makes an assertion about nodes in a game tree, viz. that they did not arise through a dominated move. Some nodes will satisfy this, others may not. Thus, announcing this formula as a fact about the players is informative, and it will in general make the current game tree smaller. But then we get a dynamics similar to that with the Muddy Children in Chapter 3. In the new smaller game tree, new nodes may become dominated, and hence announcing Rationality makes sense again, and so on. This process must reach a limit, a smallest subtree where no move is dominated any more. Here is how this works:

**Example** Solving games through iterated assertions of Rationality.

Consider a game with three turns, four branches, and pay-offs for \( A, E \) in that order:

```
        A
       / \  
      x   E
     / \  /  
    1, 0 y  A
   /  \ /  
  0, 5 z u  
   \   \  
    6, 4 5, 5
```

Stage 0 of the procedure rules out point \( u \) (the only point where Rationality fails), Stage 1 rules out \( z \) and the node above it (the new points where Rationality fails), and Stage 2 rules out \( y \) and the node above it. In the remaining game, Rationality holds throughout:

---

case where \( G' \) is a subgame of \( G \) might be related to open technical problems like finding a complete syntactic characterization for the PDL formulas that are preserved under submodels.
When Backward Induction assigns unique moves, we get the following connection:

**Theorem**  The actual Backward Induction path for extensive games is obtained by repeated announcement of the assertion of Rationality to its limit.

**Proof** This can be proved by a simple induction on finite game trees.  

With general relational strategies, this particular iterated announcement scenario produces the earlier $\forall \forall \exists \exists$ version of Backward Induction stated in Section 10.5.

**Dynamic instead of static foundations** An important feature of our new scenario is this.

In the terminology of Chapter 3, the above iterated announcement procedure for game trees (or game models) is *self-fulfilling*: it ends in non-empty largest submodels where players have *common knowledge of rationality*.  Thus, this dynamic style of game analysis is a big change away from the usual static characterizations of Backward Induction in the epistemic foundations of game theory (van Benthem 2007F):

Common knowledge or common belief of rationality is not assumed, but *produced* by the logic.

In Chapter 15, we will look at similar iterated announcement procedures that can solve strategic games, working on game matrices or Level-Three models for games.

---

24 The repeated announcement limit yields only the actual BI path. To get the BI moves at nodes $x$ not reached by the latter, we must remember the stage for the subgame $T_x$ with $x$ as its root.

25 We do not pursue such technical details here: the main point is the new scenario itself.

26 We forego the issue of logical languages for explicitly *defining* the limit submodel.

27 There are other ways to go. Baltag, Smets & Zvesper 2009 show how strong belief in ‘dynamic rationality’ defines epistemic-doxastic models whose plausibility relations encode the Backward
The PAL style of game analysis also works with other assertions. Van Benthem 2004A, 2007F consider alternatives to future-oriented rationality where players steer actions by considering the legitimate rights of other players because of their past merits.  

10.11 Belief, update and revision in extensive games

So far, we have studied players’ knowledge in games, observation-based or procedural. Next, consider the equally fundamental notion of belief, that has the same two aspects. Indeed, many foundational studies in game theory use belief rather than knowledge.

To deal with this, our earlier Level-One game models would add plausibility orders as in Chapter 7. A game tree annotated in this way records steps of knowledge update and belief revision as the game is played, and we will use this view in Chapter 11. For the moment, we look at Level-Two branching trees, a vivid model of beliefs about the future:

Beliefs over time We now add binary relations $\leq_{i,s}$ of state-dependent relative plausibility between histories to branching models. As in Chapter 7, we then get a doxastic modality (an existential one here, for convenience), with absolute and conditional versions:

**Definition** Absolute and conditional belief.

We set $M, h, s \models <B, i>\phi$ iff $M, h', s \models \phi$ for some history $h'$ coinciding with $h$ up to stage $s$ and most plausible for $i$ according to the given relation $\leq_{i,s}$. As an extension, $M, h, s \models <B, i>^s\phi$ iff $M, h', s \models \phi$ for some history $h'$ most plausible for $i$ according to the given $\leq_{i,s}$ among all histories coinciding with $h$ up to stage $s$ and satisfying $M, h', s \models \psi$. ■

Suppose we are at node $s$ in the game, and move $a$ is played in public. Epistemically, this just eliminates some histories from the current set. But there is now also belief revision, as we move to a new plausibility relation $\leq_{i',s'}$ describing the updated beliefs:

**Hard belief update** With hard information, as in Chapter 7, the new plausibility relation is the old one, restricted to a smaller set of histories. Here are matching recursion laws, where

---

Induction strategy, in a way that avoids the usual ‘Paradox of Backward Induction’ (see below) by assuming ‘incurable optimism’ about players’ reverting to Rationality later on in the game.

28 Cf. the contrast in social philosophy between fairness as desired future income distribution and entitlement views calling a society fair if it arose from a history of fair transactions.
the temporal operator $F_a \varphi$ says that $a$ is the next event on the current branch, and that $\varphi$ is true on the current branch immediately after $a$ has taken place:

**Fact** The following temporal principles are valid for hard revision along a tree:

$$F_a <B, i> \varphi \leftrightarrow (F_a T \land <B, i>^F \varphi)$$

$$F_a <B, i>^s \varphi \leftrightarrow (F_a T \land <B, i>^F \varphi)$$

This is like the dynamic doxastic recursion axioms for hard information in Chapter 7. There are also analogies with temporal protocol axioms for belief revision (Chapter 11, and Dégrémont 2010). As for concrete temporal scenarios driven by hard update, Dégrémont & Roy 2009 present a beautiful analysis of disagreement between agents (cf. Aumann 1976) via iterated announcements of conflicts in belief: Chapter 15 has more details.

**Soft game update** So far, our dynamics is driven by hard information from observed moves. An analogue for soft information and plausibility changes, the more exciting theme in Chapter 7, calls for new events as soft triggers – say, signals that players receive in the course of a game. Recursion axioms will then reflect revision policies.

Our final illustration is such a soft reconstruction of our running example. Again we give just one illustration, leaving deeper logical theory for another occasion.

**Backward induction in a soft light** So far we gave two analyses of Backward Induction. The first in Section 10.5 was a modal frame correspondence with a rationality assertion. The second in Section 10.10 was dynamic, capturing the BI-path as a limit of successive hard announcements $!RAT$ of Rationality. But perhaps the most appealing take on the BI strategy uses soft update (van Benthem 2009B). All information produced by the algorithm is in the binary plausibility relations that it creates inductively for players among end nodes in the game. For concreteness, consider our running example once more:

---

29 This is a principle of Coherence: the most plausible histories for $i$ at $h'$, $s'$ are the intersection of $B$, with all continuations of $s'$. Cf. Bonanno 2007 on temporal AGM theory.

30 In Chapter 11, we study belief revision under soft information on nodes in branching trees.

31 There is an easy correspondence between strategies as subrelations of the total move-relation and ‘move-respecting’ total orders of endpoints. Cf. van Benthem & Gheerbrant 2010.
Example  The bad Nash equilibrium, hard and soft.
The hard scenario, in terms of events $!RAT$ removes nodes from the tree that are strictly
dominated by siblings as long as this can be done, resulting in the following stages:

By contrast, a soft scenario does not eliminate nodes but modifies the plausibility relation.
We start with all endpoints of the game tree incomparable (other versions would have them
equiplausible). Next, at each stage, we compare sibling nodes, using the following notion:

A turn $x$ for player $i$ dominates its sibling $y$ in beliefs if the most plausible
end nodes reachable after $x$ are all better for the active player than all the
most plausible end nodes reachable after $y$.

$Rationality^*$ ($RAT^*$) is the assertion that no player plays a move that is dominated in beliefs.
Now we perform a relation change that is like a radical upgrade $⇑RAT^*$ as in Chapter 7:

If $x$ dominates $y$ in beliefs, we make all end nodes from $x$ more plausible
than those reachable from $y$, keeping the old order inside these zones.

This changes the plausibility order, and hence the dominance pattern, so that an iteration
can start. Here are the stages for this procedure in the above example:

In the first game tree, going right is not yet dominated in beliefs for $A$ by going left. $RAT^*$
only has bite at $E$’s turn, and an upgrade takes place that makes $(0, 100)$ more plausible
than $(99, 99)$. After this upgrade, however, going right has now become dominated in
beliefs, and a new upgrade takes place, making $A$’s going left most plausible. ■
Theorem  On finite trees, the Backward Induction strategy is encoded in the plausibility order for end nodes created by iterated radical upgrade with rationality-in-belief.

At the end of this procedure, players have acquired common belief in rationality.

Other revision policies 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, so why would we assume that she will play BI later on? BI seems to bite itself in the tail. This is not a real inconsistency: the dynamic scenarios that we have given make sense. Even so, consider a concrete example:

![Game Tree Diagram]

Backward Induction tells us that A will go left at the start. So, if A plays right, what should E conclude? There are many different options, such as ‘it was just an error, and A will go back to being rational’, ‘A is trying to tell me that he wants me to go right, and I will surely be rewarded for that’, ‘A is an automaton with a general rightward tendency’, and so on. Our dynamic logics do not choose for the agent, and support many policies, in line with Stalnaker’s view on rationality in games as tied to belief revision (Stalnaker 1999).

10.12 Conclusion
We have seen how games invite logics of action, knowledge, belief, and preference. In particular, we showed how, once the static base has been chosen, our dynamic logics can deal with entanglement of all these notions, as they play in interactive agency, not just locally, but also through time. To make these points, we gave pilot studies rather than grand theory – but some of our themes will return in Chapters 11, 15.

---

32 I admit that one natural reaction to these surprise events is a switch to an entirely new reasoning style about agents. That might require more finely-grained syntax-based views of revision.
Even at this stage, we hope to have shown that logical dynamics and games are a natural match. And as a spin-off, we found many interfaces between logic and game theory.

10.13 Coda: game logics and logic games

What is the point of logical definitions for game-theoretic notions? The way I see it, logic has the same virtues as always. By formalizing a practice, we see more clearly what makes it tick. In particular, logical syntax is a perspicuous high-level way of capturing the gist of complex game-theoretic constructions. For instance, our modal Zermelo formula gave the essence of the recursive solution process in a nutshell. Zvesper 2010 has other examples where simple modal fixed-point rules summarize sophisticated complex game-theoretic proofs. Of course, there is no unique best game logic, and first-order logic makes sense, too. Moreover, a logical stance also suggests new notions about games, as it goes far beyond specific examples (cf. Chapter 15). Finally, logical results about expressive power, completeness, and complexity can be used for model checking, proof search, and other ways of analyzing players. Still, in this book, games are only one example of rational agency, and our logic style of analysis has broader aims than serving game theory.

But there is also another connection: less central to our study of information dynamics, but quite important in general. As we saw in Chapter 2, basic notions of logic themselves have a game character, with logic games of argumentation, evaluation, or model comparison. Thus, logic does not just describe games, it also embodies games. Only in this dual perspective, the true grit of the logic and games connection becomes clear. 33

10.14 Further directions and open problems

We list a few further topics broadening the interface of logical dynamics and games.

Designing concrete knowledge games The games studied in this chapter are all abstract. But many epistemic and doxastic scenarios in earlier chapters also suggest concrete games, say, of being the first to know certain important facts. It would be of interest to have such ‘gamifications’ of our dynamic logics: Ågotnes & van Ditmarsch 2009 is a first study.

33 This duality is the main thrust of the companion volume van Benthem, to appearA.
**Dynamics of rationalization** We are often better at rationalizing behaviour afterwards than in predicting it. Given minimally rational strategies, there is an algorithm for creating preferences that make these the Backward Induction outcome – and even with given preferences, one can postulate beliefs that make the behaviour rational (van Benthem 2007G). One would like to understand this dynamics, too, in terms of our logics.

**Dynamics in games with imperfect information** This chapter distinguished two kinds of knowledge: one linked to past observation in imperfect information games, the other to uncertainty about how a game will proceed. We did not bring the two views together in our dynamic perspective. Now, imperfect information games are a difficult area of study. We just illustrate the task ahead with two simple scenarios (with the order ‘A-value, E-value’):

The game to the left seems a straightforward extension of our techniques with dominance, but the one to the right raises tricky issues of what A would be telling E by moving right. We leave the question what should or will happen in both games to the reader. Dégrémont 2010, Zvesper 2010 have more extensive discussion of what DEL might say here.

**Groups and parallel action** We have not discussed coalitions and group agency in games. A very simple desideratum here would be a treatment of games with simultaneous moves. Nothing in our dynamic logics prevents either development. In fact, it is easy to extend DEL and PDL with compound actions that are tuples of acts for each player, adding some appropriate vocabulary accessing this structure. It just has not been done yet.

**Bounded rationality** We characterized the DEL-like games of imperfect information in terms of players with Perfect Recall observing public moves. But there are many other types of player, and the treatment of this chapter needs to be extended to cope with agent

---

34 The tree to right is slightly adapted from an invited lecture by Robert Stalnaker at the Gloriclass Farewell Event, ILLC Amsterdam, January 2010.
diversity in powers of memory (cf. Chapters 3, 4) and inferential resources (cf. the syntax-based dynamics of Chapter 5) for choosing their strategies toward others.

**Infinite games** All games in this chapter are finite. Can we extend our dynamic logics to deal with infinite games, the paradigm in evolutionary game theory (Hofbauer & Sigmund 1998) and much of computer science? A transition to the infinite was implicit in the iterated announcement scenarios of Chapter 3, and it also arises in modal process theories of computation (Bradfield & Stirling 2006). See also Chapter 11, where we will fit DEL into Grand Stage views of temporally evolving actions, knowledge, and belief.

**Games equivalence and other logics** Beyond extensive games, there are strategic games with only strategy profiles and their outcomes. A deeper study would raise the question when two games are the same (van Benthem 2002). Consider the following two games:

![Game Diagram](image)

Are these the same? As with computational processes, the answer depends on our level of interest. If we focus on turns and local moves, then the games are not equivalent – and some sort of bisimulation would be the right invariance, showing a difference in some matching modal language. But if we focus on players’ powers, the two games are the same. In both, using their strategies players can force the same sets of outcomes. A can force the outcome to fall in the sets \( \{p\} \) or \( \{q, r\} \), E can force the outcome to fall in the sets \( \{p, q\} \) or \( \{p, r\} \). This time, the right invariance is power bisimulation, and the matching language uses the global forcing modalities defined earlier in this chapter (van Benthem 2001). All these issues become more delicate in the presence of preferences. Van Benthem 2002A looks at intuitions of preference-equivalence, or supporting the same strategic equilibria. Thus, this chapter has only told part of a much larger story of logic and games. Chapter 15 is a case study of logical dynamics on games in strategic form.

**Model theory, fixed-point logics, and complexity** There are many questions about the games in this chapter that invite standard mathematical logic. For instance, little is known
about preservation behaviour of assertions about players and strategies across various
*game transformations*, including just going to subgames. Say, if we know what equilibria
exist in one game, what can we say systematically about those in related games?

Another source of logical questions are solution procedures. For instance, our hard and soft
dynamic scenarios for Backward Induction raise the issue how their limits are definable.
Working on finite games, van Benthem & Gheerbrant 2010 define the update steps by
means of suitable formulas, connect our various construals, and show that sometimes,
*monotone fixed-point logic* $\text{LFP}(\text{FO})$ can define limits, while for most limits of iterated
hard or soft update, *inflationary fixed-point logic* $\text{IFP}(\text{FO})$ is more natural. (Gurevich &
Shelah 1986, Kreutzer 2004 show that these logics have the same expressive power – but
the arity of fixed-points can differ. Cf. Chapter 15 for related issues with strategic games.)
But they also give an alternative analysis, pointing out how the notions of rational game
solution in this chapter define unique relations by recursion on a well-founded order on
finite trees, namely, the composition of the *sibling* and *tree dominance* relations.

One important issue is *which fragments* of such strong logics for inductive definition are
needed in the theory of game solution. One intriguing issue here concerns the role of
*Rationality*, usually seen as a way of simplifying theory. Its logical syntax seems
significant, since it gives a Confluence diagram with moves and preference. This suggests
a similarity with the Tiling problems of Chapter 2 that generated high *complexity* of logics.
Are logics for games with rational players essentially complex?

**The gist of it all: modal logics of best action** Our final open problem is about a modal
logic, just a fragment of sophisticated fixed-point logics. In the spirit of mathematical
abstraction, it would be good to extract a simple surface logic for reasoning with the ideas
in this chapter, while hiding most of the machinery. Now, when all is said and done, game
solution procedures take a structure with actions and preferences, and then compute a new
relation that may be called *best action*. And the latter, of course, is the only thing that we
agents are really after (cf. the ‘logic of solved games’ in van Otterloo 2005):

**Open problem** Can we axiomatize the modal logic of finite game trees with a *move*
relation and its transitive closure, turns and *preference* relations for players,
and a new relation *best* as computed by Backward Induction?
My guess would be that we get a simple modal logic for the moves (these exist) plus a preference logic as in Chapter 9, while the modality $<\text{best}>$ would satisfy some obvious simple laws plus one major bridge axiom that we have encountered earlier:

$$(\text{turn} \land <\text{best}>[\text{best}^*(\text{end} \rightarrow p)) \rightarrow [\text{move-i}]<\text{best}^*(\text{end} \land <\text{pref}>p).$$

10.15 Literature