5
Infinite Games and Temporal Evolution

All games studied so far were finite. But as we saw in our Introduction, infinite games are equally important. Never-ending processes model a wide range of phenomena, from operating systems to social norms. Indeed, infinite games are the paradigm in evolutionary game theory and much of computer science. A transition to the infinite occurs at various places in this book, starting with the infinite evolutionary games in our Introduction, and continuing with the limit scenarios with iterated announcements of Part II, the infinite evaluation games for fixed point logics in Chapter 14, and the models for branching time that will occur repeatedly. Infinite games can be studied with the tools we have introduced so far, since the modal μ-calculus of Chapters 1 and 4 can describe infinite histories. Still, the major paradigm in this realm is *temporal logic*, including the systems for action and knowledge evolving over time in Chapter 8 (see also Chapter 25). While this book places emphasis on modal and dynamic logics, this is not a rigid choice. In this chapter, we provide some basic background on temporal logics for studying games, while illustrating how these apply to basic issues in games such as information flow and strategic reasoning.

REMARK Names of players
Like most of this book, this chapter has two-player games for its paradigm, and as always, we appreciate diversity in life forms of notation. The two players, or roles, in the games to be discussed will be indicated by either $\boldsymbol{E}$ and $\boldsymbol{A}$, or by i and j, largely depending on customs in the literature of origin.

5.1 Determinacy generalized

We start with a special area close to our original starting point for game logic in Chapter 1. One striking result for finite games was Zermelo's Theorem. Much stronger results exist for infinite games, where players produce infinite runs, marked as winning or losing. Winning conditions for histories in infinite games vary a lot.

Gale-Stewart Theorem An important special case occurs when winning a run for a player depends only on what happens in some finite initial segment of play.

DEFINITION 5.1 Open sets of runs
A set of runs O is *open* if every infinite sequence that belongs to O has a finite initial segment X such that all runs of the game sharing X are also in O. An infinite game is called open if one of the players has an open winning condition. ■

In what follows, the Gale-Stewart Theorem from the 1950s generalizing Zermelo's Theorem is explored.

THEOREM 5.1 All open infinite games are determined.

Proof First we state the following completely general auxiliary fact. It goes under the name of *Weak Determinacy*, for reasons that will be clear from its formulation.

FACT 5.1 If player $\boldsymbol{E}$ has no winning strategy at stage s of an infinite game, then $\boldsymbol{A}$ has a strategy forcing a set of runs from s at all of whose stages $\boldsymbol{E}$ has no winning strategy for the remaining game from then on.

Weak Determinacy is proved as follows. $\boldsymbol{A}$'s strategy arises by the same reasoning as for Zermelo's Theorem. If $\boldsymbol{E}$ is to move, then no successor node can guarantee a win, since $\boldsymbol{E}$ has no winning strategy now, and so $\boldsymbol{A}$ can just wait and see. If $\boldsymbol{A}$ is to move, then there must be at least one possible move leading to a state where $\boldsymbol{E}$ has no winning strategy; otherwise, $\boldsymbol{E}$ has a winning strategy right now after all. By choosing his moves this way, $\boldsymbol{A}$ is bound to produce runs of the kind described.

Now, assume without loss of generality that $\boldsymbol{E}$ is the player who has the open set of winning runs. Then $\boldsymbol{A}$'s nuisance strategy just described is a winning strategy. Consider any run r produced. If r were winning for $\boldsymbol{E}$, then some finite initial segment $r(n)$ would have all its continuations winning, by openness. But then, $\boldsymbol{E}$ would have had a winning strategy by stage $r(n)$: namely, play whatever move. Quod non. ■

Open winning conditions occur, for instance, with our logic games of comparison and construction in Chapters 15 and 16. The Gale-Stewart Theorem is itself a special case of Martin's Theorem stating that all Borel games are determined. In this case, winning conditions lie in the Borel Hierarchy of sets of sequences (cf. Moschovakis 1980, Vervoort 2000). Natural non-open Borel conditions are fairness of operating systems (all requests eventually get answered), and the parity winning conditions for fixed point games in Chapter 14.

Nondeterminacy With non-Borel winning conventions, infinite games can be nondetermined. To define an example we need a basic set-theoretic notion.

Definition 5.2 Ultrafilters
An *ultrafilter* on the natural numbers $\mathbb{N}$ (the only case we will need here) is a family U of non-empty sets of natural numbers with the following three properties:

(a) The family U is closed under taking supersets.

(b) The family U is closed under taking intersections.

(c) $X \in U$ iff $\mathbb{N} - X \notin U$ for all sets $X \subseteq \mathbb{N}$.

A *free* ultrafilter is a family U of this sort that contains no finite sets. ■

Free ultrafilters exist by the Axiom of Choice. Let U^* be one. Here is the game.

Example 5.1 A nondetermined game
Two players, $\boldsymbol{A}$ and $\boldsymbol{E}$, pick successive neighboring closed initial segments of the natural numbers, of arbitrary finite sizes, producing a succession:

$$\boldsymbol{A} : [0, n_1], \text{ with } n_1 > 0, \quad \boldsymbol{E} : [n_1 + 1, n_2], \text{ with } n_2 > n_1 + 1, \quad \text{etc.}$$

We stipulate that player $\boldsymbol{E}$ wins if the union of all intervals chosen by $\boldsymbol{E}$ is in $\boldsymbol{U}^*$, otherwise, $\boldsymbol{A}$ wins. Winning sets in this game are not open for either player, as sets in U^* are not determined through finite initial segments. ■

Fact 5.2 The interval selection game is not determined.

Proof The proof is a "strategy stealing" argument. Player $\boldsymbol{A}$ has no winning strategy. For, if $\boldsymbol{A}$ had one, $\boldsymbol{E}$ could use that strategy with a delay of one step to copy $\boldsymbol{A}$'s responses disguised as $\boldsymbol{E}$'s moves. Both resulting sets of intervals (disjoint up to some finite initial segment) have their unions in $\boldsymbol{U}^*$; however, this cannot be, as $\boldsymbol{U}^*$ is free. But neither does $\boldsymbol{E}$ have a winning strategy, for similar reasons. ■

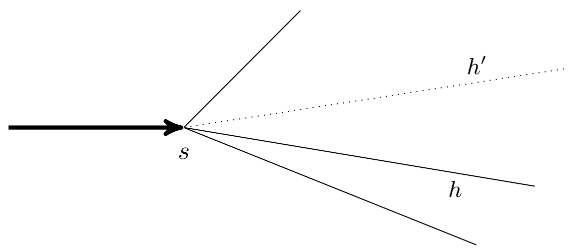
Through copying, modulo initial segments, the players' powers are the same in this game. We will encounter this symmetric situation once more in Chapters 11 and 20, when studying players' powers in possibly nondetermined infinite games. We will analyze the preceding arguments more deeply in Sections 5.2 and 5.5.

The dependence on the Axiom of Choice is significant. Set theorists have studied mathematical universes based on an alternative Axiom of Determinacy, stating that all infinite games are determined.[40]

5.2 Branching time and temporal logic of powers

Now that we have explored infinite games, we make a transition to temporal logic, starting with a simple system. Infinite games remind us of the larger temporal playground of possible histories that processes live in. What sort of logics are appropriate here? A simple point of entry is the proof of the Gale-Stewart Theorem. As we noted several times in Chapter 4, one good design method for logics is to look at the minimal expressive power needed to formalize basic game-theoretic arguments. The proof of the Gale-Stewart Theorem revolved around Weak Determinacy. What is the logical structure of the key argument establishing this principle?

The straightforward formalization is in the world where all games live: that of a *branching time* with forking histories, as in the following familiar picture:



The bold line entering from the left indicates the actual history, only known up to stage s so far. In such models, points can have local properties encoded, while whole branches can have relevant global properties that are stage-independent. Examples of the latter are winning conditions on branches in some infinite logic games (see Chapters 14 and 15), or discounted payoffs in evolutionary games (Osborne &

40 There is a rich literature on determinacy results in descriptive set theory (Moschovakis 1980, Kechris 1994, Vervoort 2000), but it is tangential to our study of games.

Rubinstein 1994). Moreover, in these models, it is possible to distinguish between external properties and game-internal properties, such as the marking of nodes as a turn for one or the other player.

Such structures typically suggest a branching temporal logic, evaluating its formulas at ordered pairs (h, t) of a current branch h and a current point t on it. In this formalism, we have a simple perspicuous formula expressing weak determinacy. Let the two different players be i and j.

FACT 5.3 Weak Determinacy is the validity of the formula $\{i\}\varphi \vee \{j\}A\neg\{i\}\varphi$.

Here $\{i\}\varphi$ is a temporal extension of the forcing modalities of Chapters 1 and 4 stating that there exists a strategy for player i ensuring that only runs result, having the current history h up to point t as an initial segment, that satisfy the temporal logic formula φ. The temporal logic shows in the operator $A\psi$ saying that ψ is always true on the current branch.

REMARK An alternative notation
We can also suppress the temporal operator to obtain a pure forcing notation that may make our main point more clear. The two relevant principles in our discussion so far are then as follows:

Strong Determinacy	$\{i\}\varphi \vee \{j\}\neg\varphi$	SD
Weak Determinacy	$\{i\}\varphi \vee \{j\}\neg\{i\}\varphi$	WD

Strong Determinacy does not always hold in reasoning about games, but Weak Determinacy does. Deriving SD depends on further assumptions that enable the transition from WD, and these, too, often have a simple logical form (see Section 5.3 for more on this, including the relevant forcing notion).

We have turned a game-theoretic lemma into a law of temporal logic with game modalities. Call the set of valid principles on branching models *temporal forcing logic*. This system codifies a temporal base theory of action in infinite games as a companion to earlier modal logics.[41] Its complexity can be analyzed as follows.

FACT 5.4 Temporal forcing logic is decidable.

Proof All of the above modalities, including that for forcing, can be defined in the well-known system MSOL of *monadic second-order logic*, extending first-order logic with quantification over subsets. In particular, on trees with successor relations,

41 For natural merges of modal and temporal logics on trees, see Stirling (1995).

MSOL can quantify over histories, since they are maximal linearly ordered subsets. Crucially, MSOL can also define strategies, since binary subrelations of the *move* relation can be coded uniquely as subsets, too, as shown in the analysis of Backward Induction in Chapter 8. By Rabin's Theorem (Rabin 1968, Walukiewicz 2002), MSOL is decidable on our tree models, and hence so is temporal forcing logic.[42] ■

The preceding result still leaves the following challenge of a complete description.

OPEN PROBLEM Provide a complete axiomatization for temporal forcing logic.

In the next section, we will look closely at the more detailed level of analysis of the preceding chapter, and give further fine structure by scrutinizing the details of the strategic reasoning underlying the earlier results.

5.3 Strategizing temporal logic

To see in more detail what can be done with temporal logics, consider representation of strategies, the theme of Chapter 4. The traditional business of logic is analyzing a given reasoning practice. We apply this perspective to a few set pieces of strategic reasoning (see van Benthem 2013 for further details and motivation). As usual, for convenience, we restrict attention to games with only two players, i and j.

Temporal logic of powers Our starting point is the temporal setting of Section 5.2. Zooming in on the nodes of these branching trees, assuming time to be discrete, we can interpret a branching temporal language of a sort found in many flavors in the literature, in the format

$$\boldsymbol{M}, h, s \models \varphi \qquad \text{formula } \varphi \text{ is true at stage } s \text{ on history } h$$

with formulas constructed using proposition letters, Boolean connectives, temporal operators F (sometimes in the future), G (always in the future), H (always in the past), P (sometimes in the past), and O (at the next moment) referring to the current branch, and modalities $\Diamond$ and $\Box$ over all branches at the current stage. Some typical clauses are as follows.

42 Indeed, many of the MSOL-definable strategy-related modalities on trees that are used in this book are also bisimulation-invariant. Hence, by the Janin-Walukiewicz Theorem (Janin & Walukiewicz 1996), they are also definable in the modal μ-calculus, reflecting our analysis in Chapters 1 and 18.

$\boldsymbol{M}, h, s \models F\varphi$ iff $\boldsymbol{M}, h, t \models \varphi$ for some point $t \geq s$,
$\boldsymbol{M}, h, s \models O\varphi$ iff $\boldsymbol{M}, h, s+1 \models \varphi$ with $s+1$ the direct successor of s on h,
$\boldsymbol{M}, h, s \models \Diamond\varphi$ iff $\boldsymbol{M}, h', s \models \varphi$ for some h' equal to h up to stage s.

We will say more about such logics in Section 5.4, but for now, given our topic, we add a new strategic modality $\{i\}\varphi$ describing the powers of player i at the current stage of the game, a temporal version of the forcing modalities of Chapter 1:

> $\boldsymbol{M}, h, s \models \{i\}\varphi$ Player i has a strategy from s onward which ensures that only histories h' result for which, at each stage $t \geq s$, $\boldsymbol{M}, h', t \models \varphi$.

This looks local to stages s, but φ can also be a global stage-independent property of the histories h'. It is common to speak of "winning conditions" φ in this setting. Sometimes, these are external features of histories, such as having built a partial isomorphism in the model comparison games of Chapter 15, but sometimes also, they refer essentially to game-internal features such as what happened at turns of specific players.

It is also important to note that, although our forcing modality $\{i\}\varphi$ is strong, it does not say that φ must be true on the current branch. We can have a strategy for salvation, and yet be on the road to perdition.[43]

Valid laws Principles of reasoning for this language are a combination of components that occur at several places in this book.

THEOREM 5.2 The following principles are valid in temporal forcing logic:

(a) The standard laws of branching temporal logic.

(b) The standard minimal logic of a monotonic neighborhood modality for $\{i\}\varphi$, plus one axiom stating its modal character: $\{i\}\varphi \to \Box\{i\}\varphi$.

(c) Three game-based principles in the spirit of Chapters 1 and 4:

(i) $\{i\}\varphi \leftrightarrow \big(((\boldsymbol{end} \wedge \varphi) \vee (\boldsymbol{turn}_i \wedge \Diamond O\{i\}\varphi) \vee (\boldsymbol{turn}_j \wedge \Box O\{i\}\varphi))\big)$
(ii) $\big(\alpha \wedge \Box G(((\boldsymbol{turn}_i \wedge \alpha) \to \Diamond O\alpha) \wedge ((\boldsymbol{turn}_j \wedge \alpha) \to \Box O\alpha)\big) \to \{i\}\alpha$
(iii) $(\{i\}\varphi \wedge \{j\}\psi) \to \Diamond(\varphi \wedge \psi)$

43 Our strategy operator is modal, referring to all possible branches. A natural local variation $\{i\}^+\varphi$ would only require the current history h to be played according to the strategy. We have not yet found a need for this further operator.

Proof For (a), references can be found in Section 5.4. For (b), see the modal logic of forcing presented in Chapter 11. As for (c), the first principle (i) is the fixed point recursion that we explored in Chapter 1, while (ii) is a strong introduction law reminiscent of the axiom for the universal iteration modality in PDL, although the antecedent is stronger.[44] The third principle (iii) is distinctly different from the first two. It is a simple form of independence of strategy choices for the two players. This independence is much like the modal logic of forcing, or the STIT-type logics of players' abilities in Chapter 12. ■

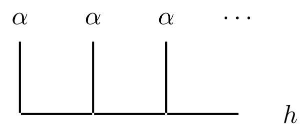
It is easy to derive further valid principles from these laws.

FACT 5.5 The *S4* law $\{i\}\alpha \to \{i\}\{i\}\alpha$ is valid in temporal forcing logic.

This tells us something interesting about strategic powers. If we follow a strategy, we have a guarantee that, in principle, we never have to leave the safe area of having that strategy available. In epistemic logic, this formula would be a principle of introspection. Here, however, it expresses a sort of safety of strategies.

In addition to valid principles, non-validities can be informative, too. A striking example follows from our explanation of the forcing modality: $\{i\}\alpha \to \alpha$ is not valid. More remarkable non-validities only appear on infinite histories.

EXAMPLE 5.2 Informative non-validities
The principle $G\{i\}\alpha \to (\alpha \vee F\alpha)$ is valid on finite games, since on endpoints of the current history, α will be forced. But now consider the following infinite model (an evergreen in many areas), viewed as a one-player game:



On each point of the unique infinite history h, a player has a strategy for ending in an endpoint. But staying on the infinite branch forever will produce a history that fails to satisfy the condition of being finite. ■

---

44 These principles have the most power when applied to genuinely local properties of stages on histories. The reader will find it instructive to see how little the second law says when the predicate α is global winning of infinite branches. This has to do with the difference between finite-stage properties and the potentially more complex winning conditions discussed in connection with the Gale-Stewart Theorem.

Presumably, our logic encodes many further valid and non-valid principles demonstrating the interplay of finite and infinite winning conditions. However, right here, we end with a small illustration going back to an earlier section.

Proving Weak Determinacy Weak Determinacy looks as follows in our system:

$$\{i\}\varphi \vee \{j\}\neg\{i\}\varphi$$

As an illustration of our logic, we now derive this formally, using two clauses from item (c) in Theorem 5.2.

$$\begin{array}{ll} (\boldsymbol{turn}_i \wedge \neg\{i\}\varphi) \rightarrow \Box O \neg\{i\}\varphi & \text{from (i)} \\ (\boldsymbol{turn}_j \wedge \neg\{i\}\varphi) \rightarrow \Diamond O \neg\{i\}\varphi & \text{from (i)} \\ \neg\{i\}\varphi \rightarrow \{j\}\neg\{i\}\varphi & \text{from (ii)} \end{array}$$

Now we can also derive the Gale-Stewart Theorem. Let φ be an open condition:

$$\varphi \rightarrow F\Box G\varphi$$

Using this, it is easy to show formally that $\{j\}\neg\{i\}\varphi \rightarrow \{j\}\neg\varphi$, and in combination with Weak Determinacy, it follows that the game is determined:

$$\{i\}\varphi \vee \{j\}\neg\varphi$$

Finally, Zermelo’s Theorem is a simple corollary, since the fact of having an endpoint is an open property of branches that can be represented as:

$$F\boldsymbol{end} \rightarrow F\Box GF\boldsymbol{end}$$

As for invalid principles, Weak Determinacy does not imply Strong Determinacy, and it turns out to be of interest to analyze our earlier counterexample.

The logic of strategy stealing Within the above general logic of strategies, further interesting properties come to light in special models. To see this, going beyond the Gale-Stewart Theorem, consider the nondetermined interval game of Example 5.1. Analyzing the strategy stealing argument in more detail reveals an interesting logical structure.

Suppose that $\boldsymbol{E}$ starts; the case with $\boldsymbol{A}$ starting is similar. The strategy σ then gives a first move $\sigma(-)$. Now let $\boldsymbol{A}$ play any move e. $\boldsymbol{E}$’s response will be $\sigma(\sigma(-), e)$, after which it is $\boldsymbol{A}$’s turn again. Crucially, this same sequence of events can also be viewed differently, as a move $\sigma(-)$ played by $\boldsymbol{E}$, followed by a move $e : \sigma(\sigma(-), e)$ played by $\boldsymbol{A}$, after which it is $\boldsymbol{E}$'s turn. This shift in perspective is what generated the contradiction in the earlier proof.

This argument presupposes the following special property of games.[45]

> *Composition Closure* Any player can play any concatenation of available successive moves.

The tree for the interval game has an interesting feature that is of independent interest. The two stages described here start the same subgames in terms of available moves, but with the difference that all turn markings are interchanged. Thus, in terms of a notion from general game logic (cf. Chapters 19 and 20), one subgame is a *dual* of the other in that all turns have been interchanged. $\boldsymbol{A}$'s strategy now uses $\boldsymbol{E}$'s strategy in the other game to produce two identical runs in both subgames, except for the inverted turn marking. But then, the winning conditions are in conflict: the union of $\boldsymbol{E}$'s choices should be in the ultrafilter, but so should the union of $\boldsymbol{A}$'s choices on that same history.

More important than this single scenario may be the following positive principle of game logic underneath, that we can state as a "copy law."

FACT 5.6 In games with composition closure, the following principle holds:

$$\{i\}\varphi \to \Diamond OO\{j\}\varphi^d$$

where φ^d is φ with all occurrences of $turn_i$, $turn_j$ interchanged.

Essentially, in games of the sort described here, players only have powers whose duals in the preceding sense are compatible.[46]

---

45 One can define this property in a suitably extended modal-temporal language.

46 More can be said here. The two subgames are not full duals in the sense of our other chapters, since that would also require inverting the sets of winning conditions for the two players. But if we do that, we do not get the above contradiction. In fact, games have two options for dualizing: turns and winning conditions. Both can be done independently. Sometimes, one only wants to dualize winning conditions, as in the games for smallest versus greatest fixed points in Venema (2007). For a logical analogy, compare negating a formula: dualizing all connectives, and then reversing polarity of atoms, versus just doing one or the other. Perhaps the game algebra of both forms of dualization is worth studying.

REMARK Copying, borrowing, and knowledge
One may compare our strategy stealing argument with the use of the copy-cat strategy of Chapter 20, although that involves full dualization. A striking further difference is that here, ***A*** needs to know ***E***'s whole strategy to run the simulation, whereas in copy-cat, it is only necessary to see one move at a time. This reflects a difference between actual parallel play versus playing a "shadow match" in the sense of Venema (2006). What also emerges here is a difference between the rhetoric of our arguments and their formalization. Intuitively, being able to steal or borrow a strategy involves knowledge, but our formalizations in this chapter do not make this feature explicit. This issue will come up again in Chapter 18.

Our conclusion is that temporal logic helps in analyzing strategic reasoning, while also extracting general principles that may not have met the eye before.[47]

5.4 Epistemic and doxastic temporal logics

Having studied specific actions and strategies, we now turn to a broader canvas. Infinite playgrounds are also the domain of temporal logics of information-based agency, and we now consider these on their own. We will present a few systems that add knowledge and belief (and in principle also preference). These form a natural continuation of the process logics for games in earlier chapters.

The grand stage of branching time The appealing structure we have been exploring in this chapter occurs in many fields, from logic, mathematics, and computer science, to philosophy and game theory. Tree models for branching time consist of legal histories h, say, the possible evolutions of a game. At each stage, players are in a node s on some actual history whose past they know completely or partially, but whose future is yet to be revealed. This is a grand stage view of agency, with histories as complete runs of some information-driven process, which is ubiquitous in the literature on interpreted systems (Fagin et al. 1995), epistemic-temporal logic (Parikh & Ramanujam 2003), STIT (Belnap et al. 2001), or game semantics (Abramsky 2008b). Such structures invite languages with temporal operators and other notions important to agency, such as knowledge. These languages describe game structure, but also important computational system properties such

47 There is a growing literature on strategic reasoning in temporal logics of agency, of which we mention Ågotnes et al. (2007) and Pacuit & Simon (2011).

as safety and liveness that underlie the infinite logic games of Part IV of this book. This is also the global playground for the local informational steps that will be studied in Part II of this book, and the connection will return in Chapter 8.

Branching temporal logic When thinking of agency, these structures suggest an action language with knowledge, belief, and added temporal operators. Its models come in two flavors (cf. Hodkinson & Reynolds 2006, van Benthem & Pacuit 2006) that will both be used in this book.

The first kind of model evaluates on pairs of complete histories and stages, which we have already seen in our case study of strategic powers. For instance, with games of perfect information, we evaluated formulas at points on histories. Looking at an extended repertoire, we use the following notations: sa denotes the unique point following s after event a has taken place (we assume that events are unique), and $s < t$ says that point t comes after s.

Definition 5.3 Basic temporal operators
The most important temporal operators are given below, including those in Section 5.3 plus some counterparts to modalities used in earlier chapters:

$\boldsymbol{M}, h, s \models F_a\varphi$	iff	sa lies on h, and $\boldsymbol{M}, h, sa \models \varphi$	(after a next event)
$\boldsymbol{M}, h, s \models O\varphi$	iff	t directly follows s and $\boldsymbol{M}, h, t \models \varphi$	(at the next stage)
$\boldsymbol{M}, h, s \models F\varphi$	iff	for some point $t > s$, $\boldsymbol{M}, h, t \models \varphi$	(in the future)
$\boldsymbol{M}, h, s \models P_a\varphi$	iff	$s = s'a$ for some s', and $\boldsymbol{M}, h, s' \models \varphi$	(at the previous stage)
$\boldsymbol{M}, h, s \models P\varphi$	iff	for some point $t < s$, $\boldsymbol{M}, h, t \models \varphi$	(in the past)

These are purely temporal operators staying on the same history. The next operator is modal, looking at all available histories fanning out from the present stage:

$\boldsymbol{M}, h, s \models \Diamond_i\varphi$ iff $\boldsymbol{M}, h', s \models \varphi$ for some h' equal to h for i up to stage s.[48]

This modality ranges over histories that may yet become realized in the future. ■

Combining operators yields further modalities, such as $\Diamond F\varphi$: φ may become true later on some possible history, but not necessarily on the actual history.

48 A universal version $\Box$ can be defined as usual as the dual operator $\neg\Diamond\neg$. The modality $\Diamond$ may also be viewed as an agent's knowledge about how the process may still unfold. In general, $\Diamond$ can then be agent indexed, since not all agents need to find the same future histories possible. Such modeling options will be discussed in Chapter 6.

We will return to this double-index view of histories in Chapter 6 when discussing various representations of games. But there is also an alternative flavor of temporal logic available to us. We now take a different tack from another broad strand in the literature, interpreting formulas only at finite stages of histories.

A temporal logic for epistemic forests A second widely used perspective focuses on a modal universe of temporal stages. In this approach, we use only finite histories as points of evaluation living in a modalized future of possible histories extending the current one. Take sets $\mathbb{A}$ of agents and $\mathbb{E}$ of events. A *history* will now be just a finite sequence of events, and $\mathbb{E}^*$ denotes the set of all such histories. Looking at the future as events occur, we write he for the unique history after e has happened in h. Also, we will write $h \leq h'$ if h is a prefix of h', and $h \leq_e h'$ if $h' = he$. Models are now as follows, starting from the basic notion of a protocol.

DEFINITION 5.4 Epistemic forest models
A *protocol* is a set of histories $\mathbb{H} \subseteq \mathbb{E}^*$ closed under prefixes. An *ETL frame* is a tuple $(\mathbb{E}, \mathbb{H}, \{\sim_i\}_{i\in\mathbb{A}})$ with a protocol $\mathbb{H}$, and accessibility relations $\sim_i$. An *epistemic forest model* (also called ETL model) is an ETL frame plus a valuation V sending proposition letters to sets of histories in $\mathbb{H}$. ■

In what follows, when we write h, he, etc., we always assume that these histories occur in the relevant protocol.

These models describe how knowledge evolves over time in an informational process. The epistemic relations $\sim_i$ represent uncertainty of agents about how the current history has gone, due to their limited powers of observation or memory. Thus, $h \sim_i h'$ means that from i's point of view, the history h' looks the same as the history h. Importantly, not all of these models are trees. Depending on the scenario, there may be multiple initial points, whence the term "epistemic forest" that we will use occasionally as a more poetic name for these models.

The *epistemic temporal language* L_{ETL} extends the basic epistemic logic of Chapter 3 with event modalities for forest models. It is generated from atomic propositions At by the following syntax:

$$p \mid \neg\varphi \mid \varphi \wedge \psi \mid [i]\varphi \mid \langle e\rangle\varphi$$

where $i \in \mathbb{A}$, $e \in \mathbb{E}$, and $p \in At$. Here the modality $[i]\varphi$ stands for epistemic $K_i\varphi$. Booleans and dual modalities $\langle i\rangle$, $[e]$ are defined in the usual way.

DEFINITION 5.5 Truth of L_{ETL} formulas
Let $\boldsymbol{M} = (\mathbb{E}, \mathbb{H}, \{\sim_i\}_{i\in\mathbb{A}}, V)$ be an epistemic forest model. The truth of a formula φ at a history $h \in \mathbb{H}$ ($\boldsymbol{M}, h \models \varphi$) is defined with the following key clauses:

$$\begin{array}{lll} \boldsymbol{M}, h \models [i]\varphi & \text{iff} & \text{for each } h' \in \mathbb{H}, \text{ if } h \sim_i h', \text{ then } \boldsymbol{M}, h' \models \varphi. \\ \boldsymbol{M}, h \models \langle e\rangle\varphi & \text{iff} & \text{there exists } h' = he \in \mathbb{H} \text{ with } \boldsymbol{M}, h' \models \varphi. \end{array}$$

Further epistemic and temporal operators are easily defined in the same style. ■

We now review a few features of this second epistemic-temporal system that are reminiscent of the modal-epistemic concerns discussed earlier in Chapter 3.

Types of agents Further constraints on models reflect special features of agents, or of the current informational process. These intertwine epistemic and action accessibility, with matching epistemic-temporal axioms in the same style as the axioms in Chapter 3. The following correspondence result shows the analogy.

FACT 5.7 The axiom $K[e]\varphi \to [e]K\varphi$ corresponds to ETL-style perfect recall:
If $he \sim k$, then there is a history h' with $k = h'e$ and $h \sim h'$.

This says that agents' current uncertainties can only come from previous uncertainties.[49] The axiom presupposes perfect observation of the current event e. In the dynamic-epistemic logics of Part II that deal with imperfect information games, this will change to allow for uncertainty about the current event.[50]

Again, as in Chapter 3, epistemic-temporal languages also describe other agents. Take a "memory-free" automaton that only remembers the last-observed event, making any two histories he and ke ending in the same event epistemically accessible. Then, with finitely many events, knowledge of the automaton can be defined in the temporal part of the language. Using the above backward modalities P_e plus a universal modality U over all histories, we have the valid equivalence

$$K\varphi \leftrightarrow \bigvee_e \big(\langle e^{\cup}\rangle\top \wedge U(\langle e^{\cup}\rangle\top \to \varphi)\big).^{51}$$

49 An induction on distance from the root then derives "synchronicity": uncertainties $h \sim k$ only occur between h and k at the same tree level. Weaker forms of perfect recall in games also allow uncertainty links across levels (Bonanno 2004b, Dégremont et al. 2011).

50 The dual $[e]K\varphi \to K[e]\varphi$ corresponds to the no miracles principle: for all ke with $h \sim k$, we have $he \sim ke$. Thus, learning can only take place by observing different signals.

51 Similar ideas work for bounded memory in general (Halpern & Vardi 1989, Liu 2008).

Expressive power and complexity One can vary the expressive power of these languages in the temporal part (one common addition are the temporal expressions Since and Until), but also in the epistemic part, adding operators of common or distributed knowledge (cf. Chapter 3 and 7). But then we meet the balance discussed in Chapters 2 and 3. Increases in expressive power may lead to upward jumps in computational complexity of combined logics of knowledge and time. The first investigation of these phenomena was made in Halpern & Vardi (1989). The table below collects a few observations from this pioneering paper, showing where dangerous thresholds lie for the complexity of validity.

	K, P, F	K, C_G, F_e	K, C_G, F_e, P_e	K, C_G, F
All ETL models	Decidable	Decidable	Decidable	RE
Perfect Recall	RE	RE	RE	Π^1_1-complete
No Miracles	RE	RE	RE	Π^1_1-complete

In this table, complexities run from decidable through axiomatizable (RE) to Π^1_1-complete, which is the complexity of truth for universal second-order statements in arithmetic.[52] The survey of expressive power and complexity for epistemic temporal logics in van Benthem & Pacuit (2006) also cites relevant background on tree logics and modal product logics. As in Chapter 3, complexity comes from defining grid cells and encoding tiling problems, whose feasibility depends on a delicate balance in language design.

Beliefs over time Epistemic temporal models generalize to other attitudes that drive rational agency, in particular, to the notion of belief. Again, this can be done in two ways, either in the earlier (h, s) style, or on finite stages h of histories. In the latter line, *epistemic-doxastic-temporal* models are branching event forests as before, with nodes in epistemic equivalence classes now also ordered by *plausibility relations* for agents. Forest models then interpret belief modalities $B\varphi$ as saying that φ is true in the most plausible accessible histories. A plausibility ordering carries much more information, however, and it can also interpret conditional beliefs in a natural way, as we will see later on. This way of modeling beliefs will be discussed in Chapter 6, and again in Chapter 7. For applications of temporal plausibility models

52 There is a gap between RE and Π^1_1-complete: few epistemic-temporal logics fall in between these classes. This also occurs with extensions of first-order logic, where being able to define a copy of the natural numbers $\mathbb{N}$ is a watershed. If you cannot, as is the case for first-order logic, complexity stays low: if you can, as is the case for first-order fixed point logics or second-order logic, complexity soars.

to game theory, see Bonanno (2001, 2007). Further issues in temporal modeling of games will come up in Chapter 6. In addition, temporal frameworks will play a role in Part II on the dynamics of actions in games.

Representation by underlying mechanisms All temporal models discussed here come with hardwired epistemic accessibility relations or doxastic plausibility orderings. But one can ask, as in Chapter 3, what underlying mechanism or process would produce this pattern of ignorance or preference. Answers will come in the representation theorems of Chapter 8. These show precisely what traces are left by agents updating their knowledge according to general dynamic-epistemic product update (Baltag et al. 1998) and revising their beliefs using a master rule of "priority update" (Baltag & Smets 2008).

5.5 Conclusion

The main points Passing from finite to infinite structures is natural in logic and in computation, and forms no barrier to a study of games. We have shown how temporal logics in two flavors (i.e., history-based or merely stage-based) can analyze basic strategic features of infinite games in an illuminating way. These logics were a natural extension of the modal logics used in earlier chapters for finite games, whose themes returned with new flavors. We also showed how epistemic and doxastic temporal logics are available for a further study of agency in games.

Open problems Still, the move from finite to infinite raises important open problems that will not be addressed in this book. One is a shift in focus for game logics from classical game theory to evolutionary game theory (Hofbauer & Sigmund 1998). Logical systems will then meet the theory of dynamical systems. An interesting program in this direction is the dynamic topological logic of Kremer & Mints (2007). On a much smaller scale, another relevant line are the logics for game matrices in Chapter 12 that can define notions such as evolutionary stability.

The move from finite to infinite may also affect our very understanding of strategies. In much of the preceding, these are finite rules with an inductive character, as in the bottom-up procedure of Backward Induction. However, repeating a point made earlier, a strategy may also be viewed "co-inductively," as a never-ending resource that we invoke as needed. We maintain our health by consulting our doctor, closing the episode, and expecting the doctor to be available as fresh as ever afterward. This sort of intuition fits better with using greatest rather than smallest fixed points in the μ-calculus or LFP(FO), and at a deeper level, with the non-wellfounded structures of co-algebra (Venema 2006). As noted in Chapter 18 on games in computational logic, transposing the theory of this book to the co-algebra setting is an interesting challenge.

5.6 Literature

Our discussion of forcing and strategies is taken from van Benthem (2013), and our general picture of epistemic-temporal frameworks comes from the survey van Benthem & Pacuit (2006).

Descriptive set theory is explained in many modern sources, such as Moschovakis (1980) and Kechris (1994). An interesting connection between solving infinite games and key methods in the foundations of mathematics (realizability, bar induction) is explored in Escardo & Oliva (2010). Epistemic temporal logics exist in great variety, with Fagin et al. (1995) and Parikh & Ramanujam (2003) as major flavors. An important computational source is the game semantics of Abramsky (2008b), a philosophical source is the STIT framework of Belnap et al. (2001). Bonanno (2001, 2007) applies temporal logics to game solution including policies for belief revision. Also relevant to the themes of this chapter are many topics in the logic games of Part IV and systems emanating from these in Part V, in particular, the linear logics of infinite games in Chapter 20.

20
Linear Logic of Parallel Game Operations

Alternative logics of game construction come from another computational tradition: infinite communication and interaction games for program semantics, with historical links to logic games for dialogue. In this chapter, we define operations for playing several games in parallel. Instead of modal logics, these lead to connections with linear logic, showing how game operations are related to intuitions about resources. The inspiration for the latter approach is proof-theoretic, representing a logical paradigm with a vast literature of its own. This chapter will only make a brief connection with the model-theoretic mainstream of this book, although we do provide a window with further references.

20.1 From logic games to game logics, once more

Logic games for argumentation and dialogue were introduced in Chapter 17. These had three main parameters: (a) rules for attack and defense of complex propositions, (b) a format of what constitutes a two-person dialogue, and (c) procedural conventions, including winning conditions and constraints on admissible repetitions.

The settings chosen by Lorenzen linked his games to intuitionistic logic, with intuitionistic proofs matching winning strategies for the proponent $\boldsymbol{P}$ of the initial thesis.[232] These games involve contrived features such as asymmetries in attack and defense rights for the proponent and the counterplayer opponent $\boldsymbol{O}$. This led to important amendments in Blass (1992), with a further modern development in the game semantics of Abramsky (1995).

232 One could get classical logic systems by fiddling with rights and duties under (c).

Again, this involves a move from logic games to game logics. In Chapter 17, atomic propositions were still black boxes: winning was maneuvering your opponent into a contradiction without any fact checking. At this point, however, we make a conceptual change, and think of atomic games as variables for real games. Say, $p \vee q$ may become a compound game, Chess or Cricket. When reaching an atom, we open the box and play the corresponding game. This simple move has many repercussions. For instance, refuting excluded middle will no longer involve tinkering with procedural rules. It suffices to plug in a *nondetermined game* p in the dialogue game for $p \vee \neg p$. Further interesting features of the resulting abstract game logic will unfold in this chapter.

Remark Infinite games and nondeterminacy
Nondetermined games of perfect information must be infinite, by the Gale-Stewart Theorem of Chapter 5. Such games had non-open winning conditions, and their existence involved an appeal to the Axiom of Choice.[233] Infinite games are natural for independent reasons, as we have seen in Chapter 4 and later in this book. The focus in this chapter is on infinite games, although finite ones are included.

20.2 Parallel operations

From sequential to parallel Our first new idea is parallel game operations. Different atoms in a dialogue game stand for different subgames that may become active, just as different atoms were different tasks in the model construction games of Chapter 16. There is no compulsory sequential schedule triggered by one leading formula, as with evaluation games, so we need to go beyond the operations of choice, switch, and sequential composition of the preceding chapter. To do so, the compositional structure of infinite games must also include parallel operations for interaction. In the dialogue games of Chapter 17, switching had to do with repeating attacks or defenses on the same atom. The systems in this chapter re-encode their procedural conventions in a more elegant game format.

Parallel conjunction of games Our pilot example is a new parallel conjunction $A \times B$ stipulating carefully who gets to switch between two games A and B (taking

233 Thus, constructivizing game logic involves games obtained by nonconstructive means. We will see a possible alternative later, in terms of games with imperfect information.

the initiative) and what are the wins. Think of Chess $\times$ Cricket, where we make some moves on a board, hit some balls, return to the board, and so on. We give an informal description here, but the details will come later.

The game $A \times B$ gives the initiative for switching to opponent $\boldsymbol{O}$. We let $\boldsymbol{O}$ win the infinite runs that have at least one infinite projection to moves in the subgame A or B that is a win for $\boldsymbol{O}$. (Thus, $\boldsymbol{P}$ wins a completed run iff both projections are either wins for $\boldsymbol{P}$, or if the run is finite.) A *parallel disjunction game* $A + B$ is like $A \times B$, but with the roles of the players $\boldsymbol{O}$ and $\boldsymbol{P}$ reversed throughout.

Copy-cat strategy A paradigmatic example for these games is simultaneous Chess. If you want to hold your own against any opponent, propose the game

$$Chess + Chess^d$$

Here, the dual game $Chess^d$ is *Chess* with the order for Black and White reversed. In the parallel sum game, you are $\boldsymbol{P}$, and you can play essentially the same Chess game twice, once as $\boldsymbol{P}$ and once as $\boldsymbol{O}$. A simple method works as follows.

Definition 20.1 Copy-cat strategy
Copy-cat works like this. Let the other player open in one game, and copy that opening move across to the other game. Wait for the other player's response there, and play that in the original game, and so on. ∎

This copying strategy produces two identical runs on both sides:

	Chess	$Chess^d$
$\boldsymbol{O}$	m_1	
$\boldsymbol{P}$		m_1
$\boldsymbol{O}$		m_2
$\boldsymbol{P}$	m_2	
	$\dots$	

By the winning convention, one of the projections is a win for $\boldsymbol{P}$, or both are draws, so $\boldsymbol{P}$ never loses the whole game. Note that *communication* between subgames is of the essence here, which makes these games a paradigm for parallel computation. In games without draws, copy-cat is even a winning strategy for $\boldsymbol{P}$.[234]

234 Chess is a finite game, so $\boldsymbol{P}$ could choose the sub game with a Zermelo non-losing strategy directly. But copy-cat is much less complex, and it also works for infinite games.

New operations The above setting supports new operations without classical counterparts. An example is the finite repetition $R(G)$ of Blass (1992), which resembles the DGL iteration G^* of Chapter 19. It lets one player open finitely many copies of G, exploiting what goes on in all of them to win at least one.

20.3 The games defined

The games We now define the structures we will work with in what follows.

DEFINITION 20.2 Infinite games
There are two players $\boldsymbol{P}$ and $\boldsymbol{O}$. *Linear games* (or just games, for brevity) are tuples $G = (M, \delta, Q, W)$ with M a set of moves, δ the turn function assigning a player to each move, Q the set of admissible positions (the legitimate finite sequences of moves), and W the set of infinite (only!) runs all of whose finite initial segments are in Q that we designate as wins for player $\boldsymbol{P}$. ■

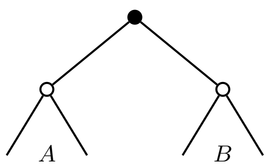
With a few twists, the reader will recognize the infinite models of Chapters 4 and 5. This format is very flexible, by varying admissible positions. Many games are scheduled like the dialogues of Chapter 18: $(\boldsymbol{OP})^*$, but other orders are possible. Also, winning can depend on many types of condition. In addition to their uses in program semantics (see our earlier references), games such as this also occur in the French school of ludics (cf. Danos et al. 1996, Girard 1998a, and Girard 1998b).

Game constructions We now go over the basic operations, some reminiscent of earlier chapters, explained more precisely as constructions on games.

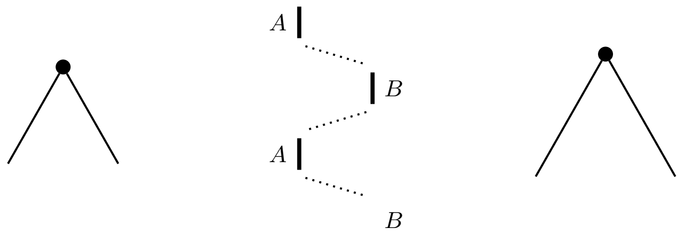
DEFINITION 20.3 Dual, choice, and parallel conjunction

(a) The *dual game* $G^d = (M, \delta^-, Q, M^\infty - W)$ has a turn function δ^- switching the roles of the two players, as well as the winning convention for infinite runs.

(b) *Choice games* $A \vee B$ have an initial choice by $\boldsymbol{P}$, just as in Chapter 19, putting two disjoint copies of the component games side by side under a common root:



(c) Finally, *parallel game conjunction* $A \times B$ is defined as follows. Take the disjoint union of moves in both games, with their original turn functions. Acceptable histories are those whose projections to both A and B are acceptable in those games. That is, one plays in each component always resuming where one left off. Moreover, only $\boldsymbol{O}$ is allowed to switch games. The winning convention is that $\boldsymbol{O}$ wins a run r if at least one projection $r|G$ ($G \in \{A, B\}$) is a win for $\boldsymbol{O}$ in subgame G.



Tree operations like this occurred in Chapter 14, and will return in Chapter 25. ■

One motivation for the rules in the parallel game is this: $\boldsymbol{O}$ attacks the conjunction, and is allowed to choose all the time where $\boldsymbol{P}$ has to defend. But the switching can be viewed just as well as cooperation, initiated by requests from $\boldsymbol{O}$.[235]

Derived game operations We now provide three useful derived game operations:

$\boldsymbol{O}$'s choice	$A \wedge B$	$= (A^d \vee B^d)^d$
parallel sum	$A + B$	$= (A^d \times B^d)^d$
implication	$A \to B$	$= A^d + B$

Parallel sum is the dual of product. Implication has a related flavor: player $\boldsymbol{P}$ can use information obtained in the subgame A to play successfully in game B. This control is reflected in basic logical inference patterns.

EXAMPLE 20.1 Modus ponens
Proponent $\boldsymbol{P}$ has a winning strategy for a modus ponens game:

$$(A \times (A \to B)) \to B$$

The reader should work out the rights and duties for both players here. ■

235 The scheduling $(\boldsymbol{OP})^*$ in the game $A \times B$ is like a finite-state machine (Abramsky 1995) with transitions between states $\langle \textit{player to move in } G_1, \textit{player to move in } G_2 \rangle$.

We will concentrate on the logic of the new parallel operations, since the simpler $\vee$ and $\wedge$ were already studied in Chapter 19.

20.4 Logical validity of game expressions

The new operations show some classical behavior. For instance, parallel product and sum are commutative and associative. But the differences are more striking. Even classical idempotence does not hold. G and choice product $G \wedge G$ amount to the same game, but in general, G is not equivalent to the parallel game $G \times G$.[236] Having one game to play, or various copies of it simultaneously, is not at all the same thing. An example of this phenomenon with parallel sum was the game

$$G + G^d$$

where $\boldsymbol{P}$ had a winning strategy, even when G and G^d are nondetermined. Thus, our earlier discussions about determinacy in the Introduction and Chapter 14 of this book become substantially more sophisticated. Whether excluded middle is valid really depends on the game disjunction used.

Definition 20.4 Validity of game expressions
An expression G is *valid* if there is a winning strategy for player $\boldsymbol{P}$ in each actual game arising from taking concrete games for the atomic expressions in G. ■

The system of game validities looks like propositional logic, but with some super-constructive differences that are even more striking than intuitionistic logic. As in Chapter 18, generic powers of players show nonclassical behavior, but they obey the laws of a natural nonclassical logic proposed independently in the 1980s.

20.5 Linear logic and resources

The non-equivalence of winning powers in games $G, G \times G, G + G$ is reminiscent of linear logic, a system of resource-conscious reasoning from the 1980s Girard 1993 as a fine structure analysis of structural rules in classical proof theory.

236 Compare this with the nondetermined free ultrafilter game G of Chapter 5 and the associated strategy stealing argument. In product play $G \times G$, $\boldsymbol{O}$ can always win, using $\boldsymbol{P}$'s moves in one game one step later against $\boldsymbol{P}$ in the other copy of G.

Occurrences and resources The main intuition is a computational view of logical formulas, say the premises of an inference, as resources that can be used only once. Validity is having exactly the right resources to reach the conclusion.

Example 20.2 Modus ponens and its relatives
An implication $A \to B$ is a function that wants exactly one argument A to produce a value B. This is exemplified perfectly in a modus ponens inference $A, A \to B \Rightarrow B$. More than one copy of A leaves an unused resource. Therefore, in linear logic,

a premise sequence $A, A, A \to B$ implies: not B, but $A \times B$,

with the product $\times$ storing the still available resources. On the other hand, an implication $A \to (A \to B)$ needs two premises A to get the B, not one or three:

$A, A \to (A \to B)$ implies $A \to B$, but not B by itself.

Classical logic ignores resources: many copies of a formula amount to just one. ■

Motivations for the present delicate view of inference come from grammars for natural language (van Benthem 1991), where occurrences of formulas stand for occurrences of linguistic categories. Game semantics (Abramsky 1995) stresses links with communication and interaction, where resources are like calls, moving from a game logic perspective to a wider view of interactive computation. (For further motivations and details for linear logic, cf. Girard 1993.)

Sequent calculus Proof systems for linear logic differ from those for classical logic in two main respects. Let us first fix a widely used proof format, Gentzen sequents $\varphi_1, \ldots, \varphi_n$ (cf. Troelstra & Schwichtenberg 2000). Classical logic reads commas in these sequents as disjunctions, and the sequent expresses that the disjunction of the given formulas is valid. Linear logic reads the commas as parallel sums $+$.

Definition 20.5 Sequent transformation
The classical sequent version of modus ponens $A, A \to B \Rightarrow B$ translates into $\Rightarrow (A \times (A \to B)) \to B$ that becomes the linear sequent

$$A^d, A \times B^d, B$$

after unpacking the definition of the linear implication. ■

A standard sequent calculus for classical logic contains three central ingredients. (a) There are axiomatic sequents $\Sigma, \varphi, \neg\varphi$, with Σ any finite sequence of formulas. (b) Logical rules take valid sequents to new ones, introducing one new logical operator at a time.[237] (c) Structural rules manipulate sequents as sets of formulas: permuting formulas, contracting more occurrences of a formula into one, or expanding one formula into several.

In linear logic, sequents stand for so-called multi-sets or bags of formulas where every occurrence counts. As a consequence, the familiar structural rules of classical logic disappear, such as contraction of identical formulas, monotonicity conjoining formulas to antecedents, or disjunctive weakening of consequents. Only the classical principles of permutation of formulas and the cut rule remain. There is also an initial sequent axiom $A, \neg A$, looking like the classical excluded middle, but as we shall soon see, it says something different. Its validity reflects the above observation about $\boldsymbol{P}$'s having a winning strategy in all games of the form $A + A^d$. In particular, no other formulas may be put alongside A, A^d, as the relevant copy-cat strategy would no longer work.

Weaker base, richer vocabulary Typically, the new paradigm shows a feature found with many weak logics. Moving to a weaker proof-theoretic basis supports a variety of non-equivalent operations beyond standard logic. In particular, classical connectives split: there are multiplicative and additive versions of conjunction and disjunction. Intuitively, multiplicative $A \times B$ is a combination of an A-type and a B-type resource, while additive $A \wedge B$ is both an A-type and a B-type resource. The difference shows clearly in the respective introduction rules, which are as follows.

Example 20.3 Introduction rules for additive and multiplicative conjunction The reader may want to ponder the difference between the following two rules:

$$\frac{X \Rightarrow A \qquad Y \Rightarrow B}{X, Y \Rightarrow A \times B} \qquad\qquad \frac{X \Rightarrow A \qquad X \Rightarrow B}{X \Rightarrow A \wedge B}$$

With the classical structural rules, $\times$ and $\wedge$ are interderivable. ■

In the rest of this chapter, we concentrate on the multiplicatives $\times, +$. For convenience, we push negations inside to atoms via the duality laws for $\times$ and $+$ as in the earlier definitions.

---

237 Examples of such rules are: from Σ, φ to $\Sigma, \varphi \vee \psi$, or from Σ, φ and Σ, ψ to $\Sigma, \varphi \wedge \psi$.

20.6 An axiom system

We now present the proof principles for resource-conscious reasoning.

DEFINITION 20.6 Sequent calculus for linear logic
The *basic proof system* for linear logic has the following principles:

linking axiom	$A, \neg A$
product rule	$\dfrac{\Sigma, A \qquad \Delta, B}{\Sigma, \Delta, A \times B}$
permutation	from Σ, derive any of its permutations
product rule	$\dfrac{\Sigma \qquad \Delta}{\Sigma, \Delta}$

Rules for the defined multiplicative $+$ follow from those for $\times$ and dual ^{d}. ■

Variants of the calculus add rules. For instance, "affine linear logic" has a weakening rule concluding from Σ to Σ, A. If we add in all structural rules of classical logic, the whole repertoire collapses into the standard Boolean operations.

Some simple linear derivations show how the austere basic proof system works.

EXAMPLE 20.4 Derivations in linear logic
(a) As we saw, modus ponens $A \times (A \to B) \Rightarrow B$ becomes $\neg(A \times (\neg A + B)) + B$, which works out to a linear sequent $\neg A, A \times \neg B, B$ that can be proved as follows:

$$\dfrac{\neg A, A \qquad \neg B, B}{\neg A, A \times \neg B, B} \quad \text{(product introduction)}$$

(b) Under the same translation, transitivity $(A \to B) \to ((B \to C) \to (A \to C))$ comes out as $(A \times \neg B), (B \times \neg C), \neg A, C$, and this can be proved as follows:

$$\dfrac{\neg A, A \qquad \dfrac{\neg B, B \quad \neg C, C}{\neg B, B \times \neg C, C}\ \text{(product introduction)}}{A \times \neg B, B \times \neg C, \neg A, C} \quad \text{(product introduction + permutation)}$$

With the many links that exist between $+, \times$ and $\to$ via the dual operation ^{d}, one can often read the same sequent in various equivalent ways. ■

As with the dynamic logic of Chapter 19, there is much more to linear logic than we can discuss here, including a crucial role for cut elimination theorems, and the use of sophisticated representation methods such as proof nets modeling interaction. We refer the reader to textbooks such as Troelstra (1993), while there are also many modern resources.

20.7 Soundness and completeness

The main theoretical results of relevance to our game logic are as follows.

Soundness Unlike the usual soundness arguments for logical calculi, the one for linear logic on its game interpretation is non-trivial and illuminating.

THEOREM 20.1 Every provable sequent of linear logic is valid in our game sense.

Proof All proof rules express significant facts about combining strategies for $\boldsymbol{P}$ in games for the premises into one for the conclusion game. We give two cases.

Product rule Suppose that player $\boldsymbol{P}$ has strategies σ and τ for winning the games Σ, A and Δ, B, respectively. Then $\boldsymbol{P}$ can also win the sequent game for $\Sigma, \Delta, A \times B$ as follows. If a move is made in Σ, let $\boldsymbol{P}$ respond with σ, and if it is in Δ, let $\boldsymbol{P}$ respond with τ. If $\boldsymbol{O}$ chooses a conjunct A or B, then respond with the move prescribed by σ for A, and τ for B. This procedure is a winning strategy, as can be seen by inspecting the resulting histories. Either $\boldsymbol{P}$ obtains an infinite subhistory that is a win in Σ or in Δ, and either is enough to win the parallel disjunction game, or if not, $\boldsymbol{P}$ has winning subhistories in both A and B, making $\boldsymbol{P}$ win the total game for $\Sigma, \Delta, A \times B$ via $A \times B$.

Cut rule Suppose $\boldsymbol{P}$ has strategies σ and τ for winning Σ, A and $\Delta, \neg A$, respectively. The following describes a winning strategy for $\boldsymbol{P}$ in Σ, Δ: play σ in Σ, and play τ in Δ. However, these strategies may prescribe moves that go into the subgames A or $\neg A$. If this happens, $\boldsymbol{P}$ plays a "virtual match," acting as $\boldsymbol{P}$ in A using σ, and also as $\boldsymbol{O}$ using τ from $\neg A$.[238] Now crucially, such a virtual episode cannot go on forever. If it did, some tail of the resulting infinite history would project to a loss for $\boldsymbol{P}$, contradicting the fact that σ and τ were winning strategies for $\boldsymbol{P}$ in Σ, A

238 This is reminiscent of the shadow matches in the strategic reasoning in Chapter 18.

and $\Delta, \neg A$.[239] Therefore, some exit move is produced in the episode, surfacing in one of Σ or Δ again, and this will be $\boldsymbol{P}$'s official response in the game for Σ, Δ. Again, checking some cases for the resulting histories of the game will show that the procedure described is a winning strategy. ■

There are some nasty bookkeeping details when such arguments get fully spelled out. This is a sort of curse of game semantics, at least, to outsiders.

Completeness The next major result is the converse implication.

Theorem 20.2 Every game-valid sequent is provable in multiplicative linear logic.

A proof is beyond the scope of this book, and we refer to Abramsky & Jagadeesan (1994). Moreover, these authors obtain an even stronger result that relates to a persistent theme in this book, namely, the role of strategies as fundamental logical objects in their own right (cf. Chapters 4, 5, and 17). They improve the preceding standard completeness theorem, curing the $\exists$-sickness of Chapter 18, to obtain so-called "full completeness."

Theorem 20.3 There is an effective correspondence between uniform winning strategies for a game sequent and linear proofs for that same sequent.

The proof of this stronger result involves categories of games whose morphisms are strategies, with the crucial cut rule of the system reflecting a natural associative composition of strategies.[240]

An interesting feature of this analysis is also that *history-free* strategies turn out to suffice for establishing completeness of linear logic. Like copy-cat, these only use the last move played, not the full history of the game so far. History-free strategies are like the positional strategies for graph games in Chapter 18, but we leave a comparison between the two settings to the reader.

20.8 From proof theory to program semantics

While our treatment has tied the games in this chapter to logic, this may be misleading. The intuitions behind proof structures in linear logic are linked closely to

239 For motivation, recall the strategy stealing argument analyzed in Chapter 5.

240 The match is between strategies and proof nets. The completeness proof uses a well-chosen category analogous to a logical term model, in which morphisms are proof nets.

multi-agent interaction, giving a computational thrust to the topics discussed here. The game semantics in this chapter is also a model of interactive computation in its own right, with strategies as algorithmic styles of system behavior, and deep uses of proof-theoretic and matching category-theoretic methods. This paradigm has been applied widely in the semantics of programming languages, with connections to denotational semantics (Scott & Strachey 1971) and domain theory (Scott 1976, Abramsky & Jung 2001). The resulting game-based area of interactive system behavior studied by logical techniques is beyond the scope of this book. For some highlights, see Abramsky & McCusker (1999) and Curien (2005, 2006). Abramsky (2012) discusses where this paradigm is leading in terms of natural levels for modeling the realities of modern computation as levels of system behavior.

20.9 Conclusion

The main points Once more, we have followed a natural path from logic games to game logics. The game semantics in this chapter treats logical formulas as game terms, and as in earlier chapters, logical operations then become general game constructions. We have seen how new parallel operations arise in this way, whose origins came from resource-based proof theory, and program semantics for interactive computation. We gave the basic definitions of game semantics, discussed its connections with linear logic, and stated soundness and completeness results showing how this works. But we can also see the system in this chapter as a continuation of the dynamic game logic of Chapter 19, providing a high-level analysis of natural operations on games.

Open problems The logic of parallel games presented here raises quite a few further issues. In particular, one can study many of the topics that came up in Chapter 19, starting with how the two approaches to logic of game constructions compare. A few such lines will be found in Section 20.11 on further directions.

20.10 Literature

We have mentioned some classic sources for linear logic and game semantics in the text, including Blass (1992) and Girard (1993). For the ensuing French School on games in this setting, see Danos et al. (1996), Girard (1998a), Girard (1998b), and the tutorial Curien (2005). For game semantics of programming languages, various papers by Abramsky are recommended, starting from the classic Abramsky & Jagadeesan (1994). A recent tutorial aimed at logicians is Abramsky (2008b). Also relevant is the discussion of information and computation in Abramsky (2008a). An alternative view of logic, games, and computation is found in Japaridze (1997).

20.11 Further directions

As usual, this chapter raises many further issues, of which we only mention a few.

Varieties of parallelism Interaction games have natural choice-points for defining parallel conjunction and disjunction. Here are at least two variants.

One might make things less uniform by letting players switch whose turn it is in both games. Then $\boldsymbol{P}$ can be a switcher after all in $A \times B$, say, when A and B both start with a move for $\boldsymbol{P}$. A more radical approach would introduce switching policies as an additional argument of the operation: $\#(A, B, \pi)$.[241] Netchitailov (2001) is a study of parallel operations with controlled switching in finite games of imperfect information.

Also, our parallel operations interleaved moves. What about games with simultaneous moves, as in Chapter 12? Natural models here would have pair states (s, t) that allow for either componentwise or simultaneous transitions. On such games, many new types of winning conditions make sense beyond the simple Boolean combinations of winning and losing in separate games used in this chapter. For instance, they might now refer to the results of *collective action*, such as saying that the sum of the yields of s and t exceeds some threshold.

Thus, the realm of natural parallel game combinations has by no means been exhausted in this chapter.

Comparisons with dynamic game logic The dynamic game logic of Chapter 19 was about sequential operations, a subset of those studied here. But it described arbitrary outcomes, with a richer language than linear sequents. It is therefore of interest to compare the expressive resources of the two logics. The above linear

241 In a similar vein, process algebra has worked with labeled parallel process operators $A||_c B$ where c is a communication method (cf. Bergstra et al. 2001).

sequents are pure game expressions,[242] and so their logic resembles the earlier game algebra. We might rework linear logic into an equational calculus for game equivalence, and then merge approaches, taking the notion of forcing to linear games, and adding modalities $\{G\}\varphi$ with G a linear game term and φ a statement about infinite runs resulting from it, perhaps in a branching temporal logic.

Adding temporal logics But then, a broader issue merges. The games discussed in this chapter are also extensive games in the sense of Parts I and II of this book. Many of the logics introduced there make sense for linear games, and they form natural combinations with the syntax of this chapter. For instance, Chapter 5 contained a few cases of strategic reasoning when a temporal logic is added that describes the histories of games explicitly. This suggests combining temporal logics and logics with explicit game terms describing the current game. We will discuss this topic briefly in Chapter 25. Such richer description languages also link to our perennial issue of game equivalence, since merging global and local views combines game equivalences at different levels of detail.

Strategy calculus In particular, merging logics allow us to resume a theme from Chapters 4 and 5: making the reasoning about strategies explicit that is supposed to be so central, but often stays inside the metalanguage. Crucial strategies for ***P*** in this chapter are simply definable, with copy-cat as a prime example. We considered definability of strategies in Chapter 4, and a joint logic with the above linear formulas as game terms, and our earlier logics for more game-internal structure would be a good candidate. A good benchmark for such a richer logic would be to formalize the earlier non-trivial soundness proof for linear logic. In a sense, the proof of the full completeness theorem delivers this, but one would also want an analysis in a formalism closer to the mainstream of this book.

More proof theory and category theory While the main thrust of this book is clearly model-theoretic, Chapters 17 and 20 have shown how basic notions and techniques from the area of proof theory, too, can enter into a profitable symbiosis with games. In particular, proof terms witnessing propositions in type theories are very much like strategies in games. This view came up in the concrete samples of strategy calculus in Chapter 4, and it suggests a whole range of analogies between

---

242 They also stood proxy for a statement about these games: viz. that player ***P*** has a winning strategy. We have taken care to distinguish these two perspectives in earlier chapters, and one should also do so here.

proof theory and game analysis that is beyond the scope of our study. Our earlier references on game semantics show how these contacts can be studied in great generality in a category-theoretic framework.

Connections with real game theory The infinite games of this chapter are reminiscent of infinite evolutionary games in game theory. Likewise, identity strategies such as copy-cat play a basic role in the latter area, under names such as strategy stealing, or Tit for Tat. These connections seem obvious, and yet they have not been explored. Likewise, it makes sense to add structure that we have studied in Parts I and II. For example, with imperfect information present, even finite games can be nondetermined. What would imperfect information versions of dynamic and linear game logics be? There may be difficult issues in bringing these worlds together, since in game theory, the interesting games are typically thought of as being noncompositional, whereas compositional analysis is the core methodology in the foundations of computation. But clearly, knowledge and imperfect information make sense for the games of this chapter, and there may well be an interesting interface with game theory here.