A Brief Tour of Game Theory

Luca de Alfaro

January 4, 2007

Abstract

These are the lecture notes of the class *Games in Design and Control* that the author teaches at UC Santa Cruz. These lecture notes present an introduction to game theory that emphasizes repeated, zero-sum games, and their connection to Markov decision process, decision theory, and control theory.

Chapter 1

Introduction

These are the lecture notes of the class *Games in Design and Control* that the author teaches at UC Santa Cruz.

These lecture notes present an introduction to game theory that emphasizes repeated, zero-sum games, and their connection to Markov decision process, decision theory, and control theory. The notes follow the order in which the material was presented in class, except that the description of some standard notation has been relegated to the appendix.

Chapter 2

Two-Player, Concurrent Game Structures

We begin our study of games by defining *two-player, concurrent games.* These games represent the interaction of two players, which play over a set of states. At each state, each player independently chooses a move. The moves chosen by the players, together with the current state, determine a probability distribution for the next state. The game is *two-player* because there are two players, and it is *concurrent* because, as in paper-scissors-stone, the two players choose their moves independently, without knowledge of the move chosen by the other player.

These games can be generalized to n-player games, which will be considered in later sections. These games can be specialized to turn-based games (when the players choose their moves in turns, rather than simultaneously), Markov decision processes (by considering the case where there is only one player with non-trivial choice of moves).

2.1 Probability distributions

For a countable set A, a *probability distribution* on A is a function $p : A \mapsto [0, 1]$ such that $\sum_{a \in A} p(a) = 1$. We denote the set of probability distributions on A by $\mathcal{D}(A)$. Given a distribution $p \in \mathcal{D}(A)$, we denote by $\mathrm{Supp}(p) = \{x \in A \mid p(x) > 0\}$ the *support* of p.

2.2 Game structures

We distinguish between a *game* and a *game structure.* A *game structure* is the structure on which the game is played: the set of states, and the transition rules. A *game* is a game structure together with a goal. We restrict our definitions, for the time being, to *finite games,* where the state space and sets of moves are finite.

A two-player *concurrent game structure* $\mathcal{G} = \langle S, M, \Gamma_1, \Gamma_2, \delta \rangle$ consists of the following components:

- A finite state space S.

- A finite set M of moves.

- Two move assignments $\Gamma_1, \Gamma_2 \colon S \mapsto 2^M \setminus \emptyset$. For $i \in \{1,2\}$, assignment Γ_i associates with each state $s \in S$ the non-empty set $\Gamma_i(s) \subseteq M$ of moves available to player i at state s.

- A probabilistic transition function $\delta : S \times M \times M \mapsto \mathcal{D}(S)$. The quantity $\delta(s, a_1, a_2)(t)$ indicates the probability of a transition from s to t when player 1 plays move a_1, and player 2 plays move a_2.

At every state $s \in S$, player 1 chooses a move $a_1 \in \Gamma_1(s)$, and simultaneously and independently player 2 chooses a move $a_2 \in \Gamma_2(s)$. The game then proceeds to the successor state t with probability $\delta(s, a_1, a_2)(t)$, for all $t \in S$. For all states $s \in S$ and moves $a_1 \in \Gamma_1(s)$ and $a_2 \in \Gamma_2(s)$, we indicate by $\tau(s, a_1, a_2) = \mathrm{Supp}(\delta(s, a_1, a_2))$ the set of possible successors of s when moves a_1, a_2 are selected. A *path* of $\mathcal{G}$ is an infinite sequence $\overline{s} = s_0, s_1, s_2, \ldots$ of states in S such that for all $k \geq 0$, there are moves $a_1^k \in \Gamma_1(s_k)$ and $a_2^k \in \Gamma_2(s_k)$ with $\delta(s_{k+1} \mid s_k, a_1^k, a_2^k) > 0$. We denote by Ω the set of all paths. For a player $i \in \{1,2\}$, we denote by $\neg i = 3 - i$ its *opponent.*

2.3 Special classes of game structures

We distinguish the following special classes of concurrent game structures.

- **Deterministic.** A concurrent game structure $\mathcal{G}$ is *deterministic* if for all $s \in S$ and all $a_1 \in \Gamma_1(s)$, $a_2 \in \Gamma_2(s)$, there is a $t \in S$ such that $\delta(t \mid s, a_1, a_2) = 1$. For added emphasis, we occasionally call *probabilistic* a general game structure (not required to be deterministic).

- **Turn-based.** A concurrent game structure $\mathcal{G}$ is *turn-based* if there is a *turn function* $\gamma : S \mapsto \{1,2\}$ that specifies whose turn it is to play. We require that for all $s \in S$, if $\gamma(s) = i$, then $|\Gamma_{\neg i}(s)| = 1$, so that if at a state, it is not player i's turn to play, player i does not have a choice of moves.

- **One-player.** A concurrent game structure $\mathcal{G}$ is *one-player* if at every state $s \in S$, we have $|\Gamma_2(s)| = 1$. In this case, for the sake of simplicity, we denote the game structure by $\mathcal{G} = \langle S, M, \Gamma, \delta \rangle$, where Γ is Γ_1, and where, for $s \in S$ and $a \in \Gamma(s)$, we let $\delta(s, a) = \delta(s, a, b)$ for the unique $b \in \Gamma_2(s)$.

For brevity, we refer to concurrent turn-based game structures simply as turn-based game structures. Several subclasses of concurrent games correspond to well-known structures in mathematics, and operations research:

- A one-player, deterministic game structure is equivalent to a graph whose edge are labeled by moves.

- A one-player, probabilistic game structure is equivalent to a *Markov decision process* [Der70, Ber95]. Markov decision process are a fundamental model in operations research and decision theory: the moves represent the control actions or decisions, the probabilistic transitions represent the uncertainty about the controlled system or the real world.

- A two-player, deterministic game structure can be considered as a *And-Or* graph: the And-nodes correspond to the states where player 1 has a choice of moves, and the Or-notes correspond to the states where player 2 has a choice of moves [Imm81].

2.4 Size of a game

We define the *size* of the game $\mathcal{G}$ to be equal to the number of entries of the transition function; specifically,

$$|\mathcal{G}| = \sum_{s \in S} \sum_{a_1 \in \Gamma_1(s)} \sum_{a_2 \in \Gamma_2(s)} |\tau(s, a_1, a_2)|.$$

2.5 Strategies

A *strategy* for player $i \in \{1, 2\}$ is a mapping $\pi_i \colon S^+ \mapsto \mathcal{D}(M)$ that associates with every nonempty finite sequence $\sigma \in S^+$ of states, representing the past history of the game, a probability distribution $\pi_1(\sigma)$ used to select the next move. Thus, the choice of the next move can be history-dependent and randomized. The strategy π_i can prescribe only moves that are available to player i; that is, for all sequences $\sigma \in S^*$ and states $s \in S$, we require that $\pi_i(\sigma s)(a) > 0$ implies $a \in \Gamma_i(s)$. We denote by Π_i the set of all strategies for player $i \in \{1, 2\}$.

A *decision rule* for player $i \in \{1, 2\}$ is a mapping $\xi : S \mapsto \mathcal{D}(M)$ that associates with every state $s \in S$ a distribution $\xi(s)$ such that $\xi(s)(a) > 0$ implies $a \in \Gamma_i(s)$. Given a sequence $\xi_0, \xi_1, \xi_2, \ldots$ of decision rules, we denote by $(\xi_0, \xi_1, \xi_2, \ldots)$ the strategy that plays, at round k, according to the decision rule ξ_k: formally, for $\pi = (\xi_0, \xi_1, \xi_2, \ldots)$, $k \geq 0$, and $\sigma \in S^k$, we let $\pi(\sigma) = \xi_k(\mathit{last}(\sigma))$.

2.6 Classes of strategies

We distinguish various classes of strategies:

- **Deterministic.** A strategy π is *deterministic* if for all $\sigma \in S^+$ there exists $a \in M$ such that $\pi(\sigma)(a) = 1$. Thus, deterministic strategies are equivalent to functions $S^+ \mapsto M$.
- **Counting.** A strategy π is *counting* if it can be written in the form $\pi = (\xi_0, \xi_1, \xi_2, \ldots)$ for decision rules $\xi_0, \xi_1, \xi_2, \ldots$.
- **Finite-Memory.** A strategy π is *finite-memory* if the distribution chosen at every state $s \in S$ depends only on s itself, and on a finite number of bits of information about the past history of the game.
- **Memoryless.** A strategy π is *memoryless* if there is a decision rule ξ such that $\pi = (\xi, \xi, \xi, \ldots)$. We call ξ the decision rule *underlying* π.

We indicate with Π^D, Π^M, Π^F the classes of deterministic, memoryless, and finite-memory strategies; we let $\Pi^{DM} = \Pi^D \cap \Pi^M$, and we let Π^H (for *history-dependent*) be the class of all strategies.

2.7 Stochastic process induced by the strategies

Once the starting state s and the strategies π_1 and π_2 for the two players have been chosen, the game is reduced to an ordinary stochastic process. Hence, the probabilities of events are uniquely defined, where an *event* $\mathcal{A} \subseteq \Omega$ is a measurable set of paths[1]. For an event $\mathcal{A} \subseteq \Omega$, we denote by

[1] To be precise, we should define events as measurable sets of paths *sharing the same initial state.* However, our (slightly) improper definition leads to more concise notation.

$\Pr_s^{\pi_1,\pi_2}(\mathcal{A})$ the probability that a path belongs to $\mathcal{A}$ when the game starts from s and the players use the strategies π_1 and π_2. Similarly, for a measurable function f that associates a number in $\mathbb{R} \cup \{\infty\}$ with each path, we denote by $\mathrm{E}_s^{\pi_1,\pi_2}\{f\}$ the expected value of f when the game starts from s and the strategies π_1 and π_2 are used. We denote by Θ_i the random variable representing the i-th state of a path; formally, Θ_i is a variable that assumes value s_i on the path $s_0, s_1, s_2, \ldots$.

The probability space is induced by the probabilities associated with the *cones.* Given a finite sequence of states $\sigma \in S^+$, the *cone* corresponding to σ is defined as:

$$cone(\sigma) = \sigma' \in S^\omega \mid \sigma \prec \sigma'.$$

Intuitively, a cone can be thought of as a finite path σ, followed by a full tree rooted at the end of σ. The probabilities of cones can be defined by induction on the length of the path defining them. We have, for all $s \in S$, and $\pi_1 \in \Pi_1$, $\pi_2 \in \Pi_2$:

$$\Pr_s^{\pi_1,\pi_2}(cone(s)) = 1,$$

and, for $\sigma \in S_s^*$ and $t \in S$,

$$\Pr_s^{\pi_1,\pi_2}\big(cone(\sigma \cdot t)\big) = \Pr_s^{\pi_1,\pi_2}\big(cone(\sigma)\big) \cdot \sum_{a_1 \in M} \sum_{a_2 \in M} \delta\big(last(\sigma), a_1, a_2\big)(t) \cdot \pi_1(\sigma)(a_1) \cdot \pi_2(\sigma)(a_2).$$

♣ complete with part on measurable sets and measure ♣

2.8 Outcomes

In addition to the above probability space, we describe the outcome of a game also by its set of *outcomes.* We say that a path $\overline{s} = s_0, s_1, s_2, \ldots \in S^\omega$ is *possible* from $s \in S$ under $\pi_1 \in \Pi_1$, $\pi_2 \in \Pi_2$, if all prefixes of $\overline{s}$ occur with finite probability: for all $n \geq 0$, we have $\Pr_s^{\pi_1,\pi_2}(cone(\overline{s}_{[0,n]})) > 0$. Equivalently, $\overline{s}$ is possible if $s_0 = s$ and if, for all $n \geq 0$, there are a_1^n, a_2^n such that $\pi_1(s_0, \ldots, s_n)(a_1^n) > 0$, $\pi_2(s_0, \ldots, s_n)(a_2^n) > 0$, and $\delta(s_n, a_1^n, a_2^n)(s_{n+1}) > 0$. We denote the set of possible paths from s under π_1, π_2 by $outcomes(s, \pi_1, \pi_2)$.

Chapter 3

Turn-Based, Deterministic Games over Graphs

We begin our tour of game theory by considering turn-based, deterministic games over graphs, played via deterministic (possibly memory-dependent) strategies.

3.1 Reachability and Safety Games

We consider the two fundamental goals of *reachability* and *safety*. Given a deterministic, turn-based $\mathcal{G} = \langle S, M, \Gamma_1, \Gamma_2, \delta \rangle$, and a set $R \subseteq S$, the reachability goal consists in reaching R, and the safety goal consists in staying always in R. Thus, we define:

$$\Diamond R = \{\overline{s} \in \Omega \mid \exists k \geq 0 \,.\, s_k \in R\}$$
$$\Box R = \{\overline{s} \in \Omega \mid \forall k \geq 0 \,.\, s_k \in R\}.$$

The (turn-based) reachability problem consists, given a game structure $\mathcal{G} = \langle S, M, \Gamma_1, \Gamma_2, \delta \rangle$ and a set $R \subseteq S$, in computing the set of states where player 1 has a strategy that guarantees winning:

$$\langle\langle 1 \rangle\rangle \Diamond R = \{s \in S \mid \exists \pi_1 \in \Pi_1^D \,.\, \forall \pi_2 \in \Pi^D \,.\, \mathit{outcomes}(s, \pi_1, \pi_2) \subseteq \Diamond R\}.$$

Similarly, the (turn-based) safety problem consists in computing the set

$$\langle\langle 1 \rangle\rangle \Box R = \{s \in S \mid \exists \pi_1 \in \Pi_1^D \,.\, \forall \pi_2 \in \Pi^D \,.\, \mathit{outcomes}(s, \pi_1, \pi_2) \subseteq \Box R\}.$$

3.2 Solving the Reachability Problem

We can solve the reachability problem as follows. Define the operator $\mathrm{Pre} : S \mapsto S$ as follows, for $X \subseteq S$:

$$\mathrm{Pre}(X) = \{s \in S \mid \exists a \in \Gamma_1(s) \,.\, \forall b \in \Gamma_2(s) \,.\, \delta(s, a, b) \subseteq X\}.$$

Intuitively, the set $\mathrm{Pre}(X)$ consists of all states from where player 1 can force the game to X in one move. We can then let:

$$X_0 = \emptyset \tag{3.1}$$
$$X_{k+1} = R \cup \mathrm{Pre}(X_k) \qquad \forall k \geq 0. \tag{3.2}$$

Theorem 1 *For $X_\bullet$ defined as in (3.1)–(3.2), we have $\langle\langle 1 \rangle\rangle \Diamond R = \lim_{k\to\infty} X_k$; moreover, the limit is reached for at most $k = |S|$.*

3.3 Solving the Safety Problem

The safety problem can be solved in similar fashion, letting

$$X_0 = S \tag{3.3}$$

$$X_{k+1} = R \cap \mathrm{Pre}(X_k) \qquad \forall k \geq 0. \tag{3.4}$$

Theorem 2 *For $X_\bullet$ defined as in (3.3)–(3.4), we have $\langle\langle 1 \rangle\rangle \Box R = \lim_{k\to\infty} X_k$; moreover, the limit is reached for at most $k = |S|$.*

Chapter 4

Stochastic Games

4.1 The Value of a Game

Valuations. A *valuation* is a function $f : S \mapsto \mathbb{R}$. For valuations f, g, we interpret operators and inequalities in pointwise fashion: for instance, we define $f + g$ to be the valuation defined by $(f + g)(s) = f(s) + g(s)$ for all $s \in S$, and we write $f \leq g$ if $f(s) \leq g(s)$ at all $s \in S$.

Utility functions. We define the value of a game, for a player $i \in \{1, 2\}$, by specifying a measurable function $u_i : \Omega \mapsto \mathbb{R}$ which associates a *utility function,* with each path.

4.2 Zero-sum value of a game

We define the *zero-sum* value of the game with utility function u_1, to player 1, from $s \in S$, by:

$$\langle\langle 1 \rangle\rangle u_1(s) = \sup_{\pi_1 \in \Pi_1} \inf_{\pi_2 \in \Pi_2} \mathrm{E}_s^{\pi_1, \pi_2}(u_1). \tag{4.1}$$

Symmetrically, we let:

$$\langle\langle 2 \rangle\rangle u_2(s) = \sup_{\pi_2 \in \Pi_2} \inf_{\pi_1 \in \Pi_1} \mathrm{E}_s^{\pi_1, \pi_2}(u_2). \tag{4.2}$$

The zero-sum value (4.1) represents the maximum value of u_1 that player 1 can guarantee, regardless of the strategy played by player 2. The zero-sum value of a game is useful in decision theory and control theory. Games are often used as a model of the interaction between a controller and a controlled system. The choice of control action is modeled as the choice of action for player 1; the nondeterministic, randomized responses of the system to control actions are modeled by the moves of player 2, and by the randomization in the transition function. The control synthesis problem consists then in devising a strategy for player 1 that ensures that the control goal is met, regardless of the strategy played by the system, or player 2. If the utility function u_1 represents the quality with which the control goal is realized, then the zero-sum value of the game (4.1) represents the best control quality that the controller can guarantee.

A game is called *zero-sum* if $u_1 = -u_2$: player 1's gain is player 2's loss. In zero-sum games, it suffices to specify just one utility function $u = u_1$. We say that a zero-sum game is *determined* if

$$\sup_{\pi_1 \in \Pi_1} \inf_{\pi_2 \in \Pi_2} \mathrm{E}_s^{\pi_1, \pi_2}(u_1) = \inf_{\pi_2 \in \Pi_2} \sup_{\pi_1 \in \Pi_1} \mathrm{E}_s^{\pi_1, \pi_2}(u_1). \tag{4.3}$$

For zero-sum, determined games, we have:

$$\begin{aligned}\langle\!\langle 1 \rangle\!\rangle u_1 &= \sup_{\pi_1 \in \Pi_1} \inf_{\pi_2 \in \Pi_2} \mathrm{E}_s^{\pi_1,\pi_2}(u_1) \\ &= \inf_{\pi_2 \in \Pi_2} \sup_{\pi_1 \in \Pi_1} \mathrm{E}_s^{\pi_1,\pi_2}(-u_2) \\ &= -\sup_{\pi_2 \in \Pi_2} \inf_{\pi_1 \in \Pi_1} \mathrm{E}_s^{\pi_1,\pi_2}(u_2) = -\langle\!\langle 2 \rangle\!\rangle u_2,\end{aligned}$$

so that

$$\langle\!\langle 1 \rangle\!\rangle u_1 + \langle\!\langle 2 \rangle\!\rangle u_2 = 0. \tag{4.4}$$

Problem 1 Give an example of a function $f : \mathbb{R} \times \mathbb{R} \mapsto \mathbb{R}$ such that $\sup_x \inf_y f(x,y) \neq \inf_y \sup_x f(x,y)$. ∎

Problem 2 Consider a function $f : A \times B \mapsto \mathbb{R}$, for any two sets A, B. Prove, or disprove with a counterexample, each of the following assertions:

$$\begin{aligned}\sup_{a \in A} \inf_{b \in B} f(a,b) &\leq \inf_{b \in B} \sup_{a \in A} f(a,b) \\ \sup_{a \in A} \inf_{b \in B} f(a,b) &\geq \inf_{b \in B} \sup_{a \in A} f(a,b)\end{aligned}$$ ∎

4.3 Games with language goals

Often, rather than specifying a utility function, we specify a *measurable language,* or a measurable subset $A \subseteq \Omega$ of paths. This is equivalent to specifying the utility function $u = [A]$ corresponding to the characteristic function of A (recall that $\Pr_s^{\pi_1,\pi_2}(A) = \mathrm{E}_s^{\pi_1,\pi_2}([A])$). We also write $\langle\!\langle i \rangle\!\rangle A for \langle\!\langle i \rangle\!\rangle [A]$, for $i \in \{1,2\}$.

For games with language conditions, by slight abuse of notation, we say that the game is *zero-sum* if the goal for player 1 is A, and the goal of player 2 is $\overline{A} = S^\omega \setminus A$, for some measurable $A \subseteq \Omega$. This implies that $u_1 + u_2 = \nVdash$, rather than $u_1 + u_2 = \nvdash$ as usual for zero-sum games. However, this slight difference in definitions has little implications, since adding a constant to a utility function does not change the relative preference of a player for the game paths. We should remark, however, that under these definitions, (4.4) becomes, for all measurable $A \subseteq \Omega$:

$$\langle\!\langle 1 \rangle\!\rangle A + \langle\!\langle 2 \rangle\!\rangle \overline{A} = 1. \tag{4.5}$$

The above equation has the following intuitive interpretation: in a determined game, the maximal probability with which player 1 can ensure the goal is equal to 1 minus the maximal probability with which player 2 can ensure the complementary goal. A fundamental result by Martin ensures that for any set A in the Borel hierarchy, (4.5) holds [Mar98]. The proof by Martin is non-constructive, and rather advanced; we will not present it here. In the following, we will provide constructive proofs of (4.5) for several classes of games. The proofs will be based on deriving algorithms for computing $\langle\!\langle 1 \rangle\!\rangle A$ and $\langle\!\langle 2 \rangle\!\rangle \overline{A}$, and in showing, on the basis of the algorithms, that (4.5) holds.

4.4 Optimal strategies

An *optimal strategy* for player 1 is a strategy π_1^* such that:

$$\inf_{\pi_2 \in \Pi_2} \mathrm{E}_s^{\pi_1^*, \pi_2}(u_1) = \sup_{\pi_1 \in \Pi_1} \inf_{\pi_2 \in \Pi_2} \mathrm{E}_s^{\pi_1, \pi_2}(u_1). \tag{4.6}$$

Optimal strategies for player 2 are defined in an analogous fashion. In general, optimal strategies are not guaranteed to exist; all we can ask for is ε-optimality. For $\varepsilon > 0$, a strategy π_1^* is ε-optimal if

$$\inf_{\pi_2 \in \Pi_2} \mathrm{E}_s^{\pi_1^*, \pi_2}(u_1) \geq \sup_{\pi_1 \in \Pi_1} \inf_{\pi_2 \in \Pi_2} \mathrm{E}_s^{\pi_1, \pi_2}(u_1) - \varepsilon.$$

From the definition (4.1), it is immediate to check that ε-optimal strategies for both players exist for all $\varepsilon > 0$.

♣ give here the three examples of games, penny-matching, hide-or-run, etc ♣

Chapter 5

Reachability and Safety in Markov Decision Processes

We begin our analysis of algorithms for Markov decision processes and games by considering safety and reachability goals. Let $\mathcal{G} = \langle S, M, \Gamma, \delta \rangle$ be an MDP. A safety goal asks for staying forever in a given set of states; the goal of a reachability game consists in reaching a given set of states. Safety and reachability goals are dual: for $R \subseteq S$, the game reaches R iff it does not forever stay in $\overline{R}$. Given $R \subseteq S$, we define the safety goal $\Box R$, and the reachability goal $\Diamond R$, by:

$$\Box R = \{\overline{s} \in \Omega \mid \forall k . \overline{s}_k \in R\}$$
$$\Diamond R = \{\overline{s} \in \Omega \mid \exists k . \overline{s}_k \in R\}.$$

We can also consider *bounded* versions of these goals: for $n \geq 0$, we define

$$\Box_{\leq n} R = \{\overline{s} \in \Omega \mid \forall k \leq n . \overline{s}_k \in R\}$$
$$\Diamond_{\leq n} R = \{\overline{s} \in \Omega \mid \exists k \leq n . \overline{s}_k \in R\}.$$

Thus, $\Diamond_{\leq n} R$ is the event corresponding to reaching R in at most n steps, and $\Box_{\leq n} R$ is the event corresponding to staying in R for at least n steps. Like $\Box$ and $\Diamond$, the events $\Diamond_{\leq n} R$ and $\Box_{\leq n} R$ are dual, in the sense that $\Diamond_{\leq n} R$ holds iff $\Box_{\leq n}(S \setminus R)$ does not hold.

We begin our exposition of algorithms for solving games by considering *bounded reachability* problems on Markov decision process.

5.1 A dynamic programming algorithm for bounded reachability

Our goal is to compute the maximal probability of reaching a subset R of states in at most $n \geq 0$ steps. As a shorthand, we let

$$v_\pi^n(s) = \mathrm{Pr}_s^\pi(\Diamond_{\leq n} R) \tag{5.1}$$
$$v_*^n(s) = \sup_{\pi \in \Pi_1} \mathrm{Pr}_s^\pi(\Diamond_{\leq n} R). \tag{5.2}$$

We compute v_*^n via a *dynamic programming* approach. We compute here a sequence of valuations $X_0, X_1, X_2, \ldots$; our aim will be to show that $X_n = v_*^n$ for all $n \geq 0$. For $n = 0$, we take $X_0 = [R]$.

For $n = 1$, we take:

$$X_1(s) = [R](s) \sqcup \max_{a \in \Gamma(s)} \sum_{t \in S} \delta(s,a)(t) \cdot [R](t) \tag{5.3}$$

for all $s \in S$. This equation can be justified as follows. For $s \in R$, the term $[R](s)$ ensures that $X_1(s) = 1$, as desired. For $s \notin R$, if we are to reach R in at most one step, we have to choose the action a that maximizes the expectation of $[R]$ at the next step — which is equal to the probability of being in R at the next step. In similar fashion, we can write:

$$n = 0: \qquad X_0(s) = [R] \tag{5.4}$$

$$n > 0: \qquad X_{n+1}(s) = [R](s) \sqcup \max_{a \in \Gamma(s)} \sum_{t \in S} \delta(s,a)(t) \cdot X_n(t) \tag{5.5}$$

for all $s \in S$.

Problem 3 Give an example of an MDP $\mathcal{G} = \langle S, M, \Gamma, \delta \rangle$ together with a set $R \subseteq S$ such that, for all $n \geq 0$, there is $s \in S$ where $X_n(s) < X_{n+1}(s)$, for X_n defined as in (5.4), (5.5). Intuitively, this indicates that the iteration scheme defined in (5.4), (5.5) does not need to converge in finitely many iterations. ∎

5.2 The predecessor operator Pre

In order to prove that $X_n = v_*^n$ for all $n \geq 0$, we first rewrite (5.4) and (5.5) in a more pleasant notation that will serve us well in the future. We define the *predecessor operator Pre* on valuations as follows. Given a valuation $f : S \mapsto \mathbb{R}$, we let:

$$\mathrm{Pre}(f)(s) = \max_{a \in \Gamma(s)} \sum_{t \in S} \delta(s,a)(t) \cdot f(t) \tag{5.6}$$

for all $s \in S$. Intuitively, $\mathrm{Pre}(f)(s)$ represents the maximum expectation of f we can guarantee from s. We also denote by $\arg \mathrm{Pre}(f)$ an arbitrary, but fixed, decision rule which associates with each $s \in S$ the action

$$\arg \max_{a \in \Gamma(s)} \sum_{t \in S} \delta(s,a)(t) \cdot f(t).$$

With this notation, we can rewrite (5.5) as:

$$n = 0: \qquad X_0(s) = [R] \tag{5.7}$$

$$n > 0: \qquad X_{n+1} = [R] \sqcup \mathrm{Pre}(X_n). \tag{5.8}$$

Correctness proof

We now present the proof that $X_n = v_*^n$ for all $n \geq 0$. The proof follows the standard lines of proofs of dynamic-programming algorithms for Markov decision processes; see for instance [Der70, Ber95, Put94, FV97] for similar proofs. In the proof, we use the following shorthand: for a decision rule ξ, and $s, t \in S$, we denote:

$$\mathrm{Pr}_s^\xi(\Theta_1 = t) = \sum_{a \in \Gamma(s)} \xi(a) \cdot \delta(s,a)(t).$$

Theorem 3 *For all $n \geq 0$, and X_n defined as in (5.7), (5.8), we have $X_n = \sup_{\pi \in \Pi_1} \Pr_s^\pi(\Diamond_{\leq n} R)$.*

Proof. To prove that $v_*^n = X_n$ for all $n \geq 0$, we prove that inequality holds in both directions.

1. Proof of $v_*^n \geq X_n$. For $n \geq 0$, let $\xi_n = \arg \operatorname{Pre}(X_n)$, and let $\pi^n = (\xi_{n-1}, \xi_{n-2}, \ldots, \xi_1, \xi_0, \xi_0, \ldots, \xi_0, \ldots)$. We prove by induction on n that $v_{\pi^n}^n \geq X_n$, which implies what is to be proved. The base case is obvious, and so is the case for $s \in R$. For $s \in S \setminus R$, we have:

$$\Pr_s^{\pi^n}(\Diamond_{\leq n} R) = \sum_{t \in S} \Pr_s^{\xi_{n-1}}(\Theta_1 = t) \cdot \Pr_t^{\pi^{n-1}}(\Diamond_{\leq n-1} R)$$

using the induction hypothesis,

$$\geq \sum_{t \in S} \Pr_s^{\xi_{n-1}}(\Theta_1 = t) \cdot X_{n-1}(t)$$

and, by the optimality of $\xi_{n-1} = \arg \operatorname{Pre}(X_{n-1})$,

$$\Pr_s^{\pi^n}(\Diamond_{\leq n} R) \geq \operatorname{Pre}(X_{n-1})(s) = X_n(s).$$

2. Proof of $v_*^n \leq X_n$. We prove by induction on n the following statement: for all $\pi \in \Pi$, we have $v_\pi^n \leq X_n$. Again, the base case is obvious. For the induction step, we have, for all $s \in S \setminus R$:

$$\Pr_s^\pi(\Diamond_{\leq n} R) = \sum_{t \in S} \Pr_s^\pi(\Theta_1 = t) \cdot \Pr_t^{\pi'}(\Diamond_{\leq n-1} R),$$

where π' is like π after s (formally, for all $\sigma \in S^+$, $\pi'(\sigma) = \pi(s \cdot \sigma)$). Using the inductive hypothesis, and then the definition of $\operatorname{Pre}(X_{n-1})$ as the maximum achievable expectation of X_{n-1} in one step, we obtain:

$$\begin{aligned} \Pr_s^\pi(\Diamond_{\leq n} R) &\leq \sum_{t \in S} \Pr_s^\pi(\Theta_1 = t) \cdot X_{n-1}(t) \\ &\leq \operatorname{Pre}(X_{n-1}) = X_n. \quad \blacksquare \end{aligned}$$

Chapter 6

Lattices and Fixpoints

The algorithm (5.7), (5.8) has an iterative form typical of many algorithms for games and Markov decision processes. This form can be best brought out as follows. We can consider the space $\mathcal{F} = (S \mapsto [0,1])$ of functions from S to the interval $[0,1]$, and define an operator $B : \mathcal{F} \mapsto \mathcal{F}$[1] by

$$B(X) = [R] \sqcup \mathrm{Pre}(X). \tag{6.1}$$

We have then $\langle\langle 1 \rangle\rangle \diamond_{\leq n} R = B^{n+1}(\mathbf{0})$, where $\mathbf{0} \in \mathcal{F}$ is the identically-zero valuation, and B^n indicates the operator B applied n times. Furthermore, we will prove in the following that

$$\sup_{\pi \in \Pi_1} \mathrm{Pr}^{\pi}(\diamond R) = \lim_{n \to \infty} B^n(\mathbf{0}). \tag{6.2}$$

In the following, we will encounter many similar operators and solution formulas. Thus, we seek a notation for writing solution formulas such as (5.7), (5.8) concisely, and we seek a theory that enables us to easily reason about the existence of limits such as (6.2). The notation is provided by μ-calculus, and the theory by results about fixpoints in lattices. We begin by presenting the latter.

6.1 Lattices

A *partially ordered set* $(L, \leq)$ is a set L together with a relation $\leq$ that is:

- *Reflexive:* for all $a \in L$, we have $a \leq a$.
- *Transitive:* for all $a, b, c \in L$, $a \leq b$ and $b \leq c$ imply $a \leq c$.
- *Antisymmetric:* for all $a, b \in L$, $a \leq b$ and $b \leq a$ imply $a = b$.

A *lattice* is a partially ordered set $(L, \leq)$ such that, for every $a, b \in L$, the *least upper bound* $a \sqcup b$ and the *greatest lower bound* $a \sqcap b$ always exist. A lattice is *complete* if, for every $H \subseteq L$, both the least upper bound $\sqcup H$ and the greatest lower bound $\sqcap H$ always exist. We denote by $\bot = \sqcap L$ and $\top = \sqcup L$ the bottom and top elements of a complete lattice $(L, \leq)$.

Example 1 The following are examples of lattices:

[1] B is a particular example of a *Bellman* operator; see [Bel54, Bel57].

1. $(\mathbb{N}, \leq)$
2. $(\mathbb{R}, \leq)$
3. $([0, 1], \leq)$
4. $(2^S, \subseteq)$ for a set S
5. $(\{1, 2, \ldots, 100\}, |\ )$ where $a|b$ if there is $k \in \mathbb{N}$ such that $b = ka$ ∎

Problem 4 Which of the above lattices are complete? ∎

Example 2 A lattice we will use frequently is $(\mathcal{F}, \leq)$, where $\mathcal{F} : S \mapsto [0, 1]$ for a set S, and where $f \leq g$ for $f, g \in \mathcal{F}$ iff $f(s) \leq g(s)$ for all $s \in S$. ∎

A *monotonic* function over a lattice $(L, \leq)$ is a function $f : L \mapsto L$ such that, for all $a, b \in L$, $a \leq b$ implies $f(a) \leq f(b)$.

Problem 5 Define monotonic functions on each of the lattices of Example 1. ∎

6.2 Fixpoints of monotonic functions over lattices

A *fixpoint* of a function $f : L \mapsto L$ is a value $a \in L$ such that $a = f(a)$. Fixpoints are a very important notion in many contexts: for instance, we will see that the limit in (6.2) computes a fixpoint of B. Functions over a lattice do not always have a fixpoint: for instance, if A is a set, the function f over the lattice $(2^A, \subseteq)$ defined by $f(a) = A \setminus a$ for all $a \subseteq A$ does not have any fixpoint. If a function f has at least one fixpoint, then the set $F = \{a \in L \mid a = f(a)\}$ of fixpoints of f is non-empty. If $\sqcap F = f(\sqcap F)$, then we say that f has a *least fixpoint* $\sqcap F$. Similarly, if $\sqcup F = f(\sqcup F)$, then we say that f has a *greatest fixpoint* $\sqcup F$. A very important theorem, known as the Knaster-Tarski theorem, ensures the existence of least and greatest fixpoints of monotonic functions over complete lattices.

Theorem 4 (Knaster-Tarski) *Let $(L, \leq)$ be a complete lattice, and let $f : L \mapsto L$ be a monotonic function over $(L, \leq)$. Let $F = \{a \in L \mid a = f(a)\}$ be the set of fixpoints of f. Then, $F \neq \emptyset$, $\sqcap F \in F$ and $\sqcup F \in F$.*

Proof. Let $H = \{x \in L \mid f(x) \leq x\}$ and let $a = \sqcap H$.
We first prove that $a \in H$. For all $x \in H$, we have $a \leq x$, and thus, $f(a) \leq f(x) \leq x$. So for all $x \in H$, we have $f(a) \leq x$, and hence $f(a) \leq \sqcap H = a$. From $f(a) \leq a$ we conclude $a \in H$.
Using the fact that $a \in H$, we prove $a = f(a)$. As $a \in H$, we have $f(a) \leq a$, and $f(f(a)) \leq f(a)$. So $f(a) \in H$, and $a = \sqcap H \leq f(a)$. From $f(a) \leq a$ and $a \leq f(a)$ we conclude $a = f(a)$, and $a \in F$. If f has another fixpoint, say b, then from $f(b) \leq b$ we derive $a \leq b$, showing that $a = \sqcap F$.
The statement $\sqcup F \in F$ can be proved analogously. ∎

Problem 6 Prove that, under the assumptions of the Knaster-Tarski theorem, we have $\sqcup F = f(\sqcup F)$. ∎

If a function $f : L \mapsto L$ over a lattice $(L, \leq)$ has a least fixpoint, we denote it by $\mu x.f(x)$; similarly, if f has a greatest fixpoint, we denote it by $\nu x.f(x)$. This notation is borrowed from μ-calculus; the full calculus is presented later in this section.

6.3 Iterative computation of fixpoints

The Knaster-Tarski theorem guarantees the existence of least and greatest fixpoints, but does not provide an algorithmic way for their computation or approximation. We now show that, if the function f is not only monotonic, but also *left-continuous,* then its least fixed point can be computed as the limit of an increasing sequence. A function $f : L \mapsto L$ over a lattice $(L, \leq)$ is *left-continuous* (resp. *right-continuous*) if, for $A \subseteq L$, we have $f(\sqcup A) = \sqcup f(A)$. Similarly, if f is monotonic and *right-continuous,* that is, $f(\sqcap(A)) = \sqcap f(A)$ for all $A \subseteq L$, then its greatest fixpoint can be computed as the limit of a decreasing sequence.

Theorem 5 (iterative computation of fixpoints) *Let $f : L \mapsto L$ be a monotonic function over a lattice $(L, \leq)$. The following assertions hold:*

1. *If f is left-continuous, we have*

$$\mu x.f(x) = \bigsqcup_{n \in \mathbb{N}} f^n(\bot).$$

2. *Similarly, if f is right-continuous, we have*

$$\nu x.f(x) = \sqcap_{n \in \mathbb{N}} f^n(\top).$$

Proof. We prove the first assertion; the proof of the second one is analogous. Let $z = \mu x.f(x)$ be the least fixpoint of f, and for $n \geq 0$, let $x_n = f^n(\bot)$. From $\bot \leq z$, we get $f(\bot) \leq f(z) = z$, and for all $n \geq 0$, $f^n(\bot) \leq z$, so that $x_n \leq z$ for all $n \geq 0$. Note that the sequence $x_0, x_1, x_2, \ldots$ is monotonically increasing: in fact, $\bot \leq f(\bot)$, and by monotonicity of f, it follows that $f^n(\bot) \leq f^{n+1}(\bot)$ for all $n \geq 0$. Let $x^* = \bigsqcup_{n \in \mathbb{N}} x_n$, and since $x_n \leq z$ for all $n \in \mathbb{N}$, we have also $x^* \leq z$.

To see that $z \leq x^*$, write:

$$f(x^*) = f\big(\textstyle\bigsqcup_{n \in \mathbb{N}} x_n\big) = \bigsqcup_{n \in \mathbb{N}} f(x_n) = \bigsqcup_{n \in \mathbb{N}} x_n = x^*.$$

This shows that x^* is a fixpoint of f, and since z is the least such fixpoint, we have $z \leq x^*$. ∎

Problem 7 Consider the lattice $(\mathcal{F}, \leq)$ of Example 2.

1. For $u, v \in \mathcal{F}$, what is $u \sqcup v$ and $u \sqcap v$?
2. Show that $(\mathcal{F}, \leq)$ is a complete lattice.
3. Show that the function Pre defined in (5.6) is monotonic on $(\mathcal{F}, \leq)$.
4. Show that the function Pre defined in (5.6) is left-continuous on $(\mathcal{F}, \leq)$.
5. Show that the functional B of (6.1) is monotonic, and left-continuous, on $(\mathcal{F}, \leq)$. ∎

Problem 8 Give an example of a lattice $(L, \leq)$ and of a function $f : L \mapsto L$ that is left-continuous but not right-continuous. ∎

Problem 9 Let $f : L \mapsto L$ be a monotonic function over a lattice $(L, \leq)$. Consider $a \in L$ such that $\mu x.f(x) \leq a \leq \nu x.f(x)$. Prove, or disprove with a counterexample, that $f(a) = a$. ∎

Problem 10 Consider the lattice $(L, \leq)$, where $L = (S \mapsto [0,1])$, and for $u, v \in L$, we have $u \leq v$ if for all $s \in S$, we have $u(s) \leq v(s)$. For $\mathrm{Pre} : L \mapsto L$ defined in (5.6), what are $\mu x.\mathrm{Pre}(x)$ and $\nu x.\mathrm{Pre}(x)$? ∎

Problem 11 Show that if f, g are left-continuous on a lattice $(L, \leq)$, so are $f \sqcup g$ and $f \sqcap g$. ∎

6.4 Unbounded Reachability in Markov Decision Processes

6.4.1 Maximal reachability probability

We now return to the analysis of Markov decision processes with reachability goals, and we provide a proof of (6.2). We consider the metric on $(\mathcal{F}, \leq)$ defined, for $f, g \in \mathcal{F}$, by $|f - g| = \sup_{s \in S} |f(s) - g(s)|$ (this choice is not critical, and other metrics would work equally well).

Theorem 6 *For all MDPs $\mathcal{G} = \langle S, M, \Gamma, \delta \rangle$ and all $R \subseteq S$, define the operator B as in (6.1). Then,* $\sup_{\pi \in \Pi_1} \Pr_s^\pi(\Diamond R) = \lim_{n \to \infty} B^n(\mathbf{0})$.

Proof. First, note that the statement makes sense: Since B is monotonic on the lattice $(\mathcal{F}, \leq)$, the limit $\lim_{n \to \infty} B^n(\mathbf{0})$ exists. We prove that inequality holds in both directions.

1. Proof of $\sup_{\pi \in \Pi_1} \Pr^\pi(\Diamond R) \geq \lim_{n \to \infty} B^n(\mathbf{0})$. Choose an arbitrary $\varepsilon > 0$, and let $v^* = \lim_{n \to \infty} B^n(\mathbf{0})$. Pick any $s \in S$. Since $\lim_{n \to \infty} \big(B^n(\mathbf{0})\big)(s) = v^*(s)$, there is $n \in \mathbb{N}$ such that $\big(B^n(\mathbf{0})\big)(s) \geq v^*(s) - \varepsilon$. By constructing a strategy π^n as in the proof of Theorem 3, we have, following the same argument, that $\Pr_s^{\pi^n}(\Diamond_{\leq n} R) \geq v^*(s) - \varepsilon$, which implies $\Pr_s^{\pi^n}(\Diamond R) \geq v^*(s) - \varepsilon$. As $\varepsilon > 0$ is arbitrary, we conclude $\sup_{\pi \in \Pi_1} \Pr_s^\pi(\Diamond R) \geq v^*(s)$.

2. Proof of $\sup_{\pi \in \Pi_1} \Pr^\pi(\Diamond R) \leq \lim_{n \to \infty} B^n(\mathbf{0})$. Assume, towards the contradiction, that there are $\pi \in \Pi_1$ and $s \in S$ such that
$$\Pr_s^\pi(\Diamond R) > \lim_{n \to \infty} \big(B^n(\mathbf{0})\big)(s).$$
Then, since $\Pr_s^\pi(\Diamond R) = \lim_{k \to \infty} (\Diamond_{\leq k} R)$, there is $k \in \mathbb{N}$ such that
$$\Pr_s^\pi(\Diamond_{\leq k} R) > \lim_{n \to \infty} \big(B^n(\mathbf{0})\big)(s) \geq \big(B^{n+1}(\mathbf{0})\big)(s),$$
where the last inequality is due to the monotonicity of B. The above inequality contradicts Theorem 3, concluding the proof by contradiction. ∎

It is important to note what the theorem above does *not* state: it does not state that the maximum value can be attained by a strategy, nor does it say anything about the properties of optimal, or ε-optimal, strategies. In a later section, we will prove that MDPs, and games, with reachability goals admit optimal memoryless strategies.

By comparing the above theorem with Theorem 5, we have the following corollary.

Corollary 1 *For all MDPs $\mathcal{G} = \langle S, M, \Gamma, \delta \rangle$ and all $R \subseteq S$, define the operator B as in (6.1). Then,* $\sup_{\pi \in \Pi_1} \Pr_s^\pi(\Diamond R) = \mu x.B(x)$.

6.4.2 Existence of deterministic memoryless optimal strategies

We now prove the existence of memoryless optimal strategies. The argument is taken from [CdAH04].

Theorem 7 *For all MDPs $\mathcal{G} = \langle S, M, \Gamma, \delta \rangle$ and all $R \subseteq S$, there is a deterministic and memoryless $\pi^* \in \Pi_1$ such that $\Pr_s^{\pi^*}(\Diamond R) = \sup_{\pi \in \Pi_1} \Pr_s^{\pi}(\Diamond R)$.*

Proof. We prove this result for a special class of MDPs: the ones where probabilistic and non-deterministic choice are present in different states. Specifically, we postulate that for every $s \in S$, one of the two following cases holds:

1. either $|\Gamma(s)| = 1$;
2. or for each $a \in \Gamma(s)$ there is $t_a \in S$ such that $\delta(s, a)(t) = 1$.

We can represent the MDP in an alternative fashion, as a structure $((S, E), S_1, S_p, p)$, where:

- $E = \{(s, t) \in S \times S \mid \exists a \in \Gamma(s) \,.\, \delta(s, a)(t) \geq 0\}$; for $s \in S$, we also let $E(s) = \{t \mid (s, t) \in E\}$;
- $S_p = \{s \in S \mid |\Gamma(s)| = 1\}$;
- $S_1 = S \setminus S_p$;
- $p : V_p \mapsto \mathcal{D}(S)$ is defined by $p(s)(t) = \delta(s, a)(t)$ for the unique $a \in \Gamma(s)$. We often write $p(s)(t)$ as $p(s, t)$.

With this notation, we proceed as follows. Let $T_1 = R$, and let $T_0 \subseteq S$ be the set of vertices that cannot reach R in the graph (S, E); let also $T = T_0 \cup T_1$ and $U = S \setminus T$. In T, the probability of winning is either 0 or 1 (it can be equal to 1 also outside T). From T, all strategies are optimal with respect to the objective $\Diamond R$, so we can fix a pure memoryless strategy on T arbitrarily. Consider the following fixpoint equation for x: $S \to [0, 1]$, where for all $s \in S$:

$$x(s) \;=\; \begin{cases} 0 & \text{if } s \in T_0; \\ 1 & \text{if } s \in T_1; \\ \max_{t \in E(s)} x(t) & \text{if } s \in S_1 \setminus T; \\ \sum_{t \in E(s)} x(t) \cdot p(s, t) & \text{if } s \in S_p \setminus T. \end{cases} \tag{6.3}$$

This system of equations in general has many fixpoints, and its least fixpoint x^* equals $\langle\!\langle 1 \rangle\!\rangle \Diamond R$; see, e.g., [dAM04]. For $s \in U \cap S_1$, define the set of *optimal successors* of s by $d(s) = \{t \in E(s) \mid x^*(t) = x^*(s)\}$. Clearly, an optimal strategy must select only optimal successors of states in $U \cap S_1$. Thus, we cut from the MDP all the edges $(s, t) \in E$ with $s \in S_1 \cap U$ and $t \notin d(s)$. It is immediate to check that x^* is still a fixpoint of (6.3) in the resulting MDP.

To obtain a memoryless strategy, we can choose all optimal successors of states in $U \cap S_1$ uniformly at random. To obtain a memoryless pure strategy, we observe that if a state $s \in U \cap S_1$ has multiple optimal successors, i.e., $|d(s)| > 1$, and we cut an edge (s, t) with $t \in d(s)$, then x^* is still a fixpoint of (6.3) in the resulting MDP. However, we cannot arbitrarily fix one optimal successor for each state in $U \cap S_1$ and cut the edges to all other successors: doing so could create new fixpoints *below* x^*. This occurs, for instance, whenever there are mutually reachable states with equal x^*, and the selected successors create a cycle that prevents reaching T. Our goal is to pick optimal successors, and cut the edges to other successors, so that x^* is the *only* fixpoint of (6.3) in the resulting MDP. This will guarantee that $x^* = \langle\langle 1 \rangle\rangle \Diamond R$ for the resulting pure memoryless strategy.

To ensure the uniqueness of the fixpoint, we cut edges from $S_1 \cap U$ while maintaining the invariant that every state in U can reach T_1 in the graph $(S \setminus T_0, E)$. Note that this invariant holds initially by the definition of T_0. Let $E' \subseteq E$ be a subset of edges that consists of shortest paths from U to T_1 such that every state has only one outgoing edge, i.e., for all $s, t_1, t_2 \in S$, if $(s, t_1), (s, t_2) \in E'$, then $t_1 = t_2$. Then, prune from player-1 states all edges that are not in E'; precisely, for all $s \in U \cap S_1$ and $(s, t) \in E$, keep (s, t) if $(s, t) \in E'$, and prune it otherwise. The MDP corresponds thus to a Markov chain. We define the transition probability matrix $[P_{s,t}]_{s,t \in U}$ and the vector $[q_s]_{s \in U}$ as follows, for all $s, t \in U$:

$$P_{s,t} = \begin{cases} 1 & \text{if } s \in S_1 \text{ and } (s,t) \in E; \\ 0 & \text{if } s \in S_1 \text{ and } (s,t) \notin E; \\ p(s,t) & \text{if } v \in S_p; \end{cases}$$

$$q_v = \begin{cases} 1 & \text{if } s \in S_1 \text{ and } \exists t \in T_1.\, (s,t) \in E; \\ 0 & \text{if } s \in S_1 \text{ and } \forall t \in T_1.\, (s,t) \notin E; \\ \sum_{u \in T} p(s,t) & \text{if } v \in S_p. \end{cases}$$

Then x^*, as a fixpoint of (6.3), is a solution of $x = Px + q$. Since every state in U has a path to T_1 in the graph $(S \setminus T_0, E)$, the matrix P corresponds to a transient chain, and $\det(I - P) \neq 0$ [KSK66]. Hence, $x^* = (I - P)^{-1} q$ is the unique fixpoint of (6.3), showing the optimality of the pure memoryless strategy thus constructed. ∎

Chapter 7

Safety and Reachability in Concurrent Games

In this chapter, we provide the solution of concurrent games with respect to reachability and safety goals.

7.1 Reachability

To compute the value of a concurrent game with reachability goal $\diamond R$, for a subset R of states, consider once more the solution formulas (5.7), (5.8) for MDPs. The intuition used to justify these formulas is still valid: however, to adapt them to games, we need to provide a new definition for the operator Pre, to take into account the presence of two players. Given a game $\mathcal{G} = \langle S, M, \Gamma, \delta \rangle$, let as usual $\mathcal{F} = (S \mapsto [0,1])$ be the set of all valuations. We define two operators, $\mathrm{Pre}_1, \mathrm{Pre}_2 : \mathcal{F} \mapsto \mathcal{F}$. Intuitively, given a valuation $f \in \mathcal{F}$, $\mathrm{Pre}_1(f)$ will associate with every state $s \in S$ the maximal expectation $\mathrm{Pre}_1(f)(s)$ of f which player 1 can ensure in one step, and symmetrically, $\mathrm{Pre}_2(f)$ will associate with every state $s \in S$ the maximal expectation $\mathrm{Pre}_2(f)(s)$ of f which player 1 can ensure in one step.

To define these operators, let $\Xi_1(s) = \mathcal{D}(\Gamma_1(s))$ be the set of distributions over moves available to player 1 at s, and symmetrically, let $\Xi_2(s) = \mathcal{D}(\Gamma_2(s))$. The operator Pre_1 is defined as follows, for $s \in S$ and $f \in \mathcal{F}$:

$$\mathrm{Pre}_1(f)(s) = \sup_{\xi_1 \in \Xi_1(s)} \inf_{\xi_2 \in \Xi_2(s)} \mathrm{E}_s^{\xi_1,\xi_2}(f), \tag{7.1}$$

where:

$$\mathrm{E}_s^{\xi_1,\xi_2}(f) = \sum_{t \in S} \sum_{a_1 \in \Gamma_1(s)} \sum_{a_2 \in \Gamma_2(s)} \delta(s, a_1, a_2)(t)\, \xi_1(a_1)\, \xi_2(a_2)\, f(f). \tag{7.2}$$

In fact, a result that goes back to von Neumann [vN28] will ensure that the sup and inf in (7.1) can be replaced by max and min, and it will also ensure that the max and min can be permuted. It is also not difficult to see that the operators Pre_1 and Pre_2 are monotonic. We will look into both of these facts in detail later. For now, let us state and prove part of the result about reachability games. Given a subset $R \subseteq S$ of states write, in analogy to the formula for MDPs:

$$n = 0: \qquad X_0(s) = [R] \tag{7.3}$$

$$n > 0: \qquad X_{n+1} = [R] \sqcup \mathrm{Pre}_1(X_n). \tag{7.4}$$

Again, we have $\langle\!\langle 1 \rangle\!\rangle \Diamond R = \lim_{n\to\infty} X_n$, as stated by the following theorem.

Theorem 8

$$\langle\!\langle 1 \rangle\!\rangle \Diamond R = \mu X.([R] \sqcup \mathit{Pre}_1(X)). \tag{7.5}$$

In order to prove this theorem, we prove separately the two inequalities:

$$\langle\!\langle 1 \rangle\!\rangle \Diamond R \geq \mu X.([R] \sqcup \mathrm{Pre}_1(X)) \tag{7.6}$$

$$\langle\!\langle 1 \rangle\!\rangle \Diamond R \leq \mu X.([R] \sqcup \mathrm{Pre}_1(X)). \tag{7.7}$$

The first inequality is a consequence of the following lemma; the second inequality will be proved in the next subsection.

Lemma 1 *Let $w = \mu X.([R] \sqcup \mathit{Pre}_1(X))$. For all $\varepsilon \geq 0$, there is a strategy $\pi_1 \in \Pi_1$ for player 1, such that for all $\pi_2 \in \Pi_2$ and all $s \in S$, we have:* $\mathrm{Pr}_s^{\pi_1,\pi_2}(\Diamond R) \geq w(s) - \varepsilon$.

Proof. The proof follows a classical argument (see, e.g., [Eve57, FV97]). For $n \geq 0$, consider the n-step version of the game, whose winning condition $\Diamond_n R$ requires reaching R in at most n steps. We construct inductively a sequence $\{\pi_1^n\}_{n\geq 0}$ of strategies for player 1. Let $X_0 = \mathbf{0}$ and $X_{k+1} = [R] \sqcup \mathrm{Pre}_1(X_k)$ for $k \geq 0$. The proof uses the fact, which will be proved later, that for $X \in \mathcal{F}$, there is a distribution $\xi \in \mathcal{D}(M)$ which realizes the sup in (7.1); we denote such distribution by arg $\mathrm{Pre}_1(X)$. Proceeding as in the proof of Theorem 3, for $n \geq 0$ we let $\xi_n = \arg \mathrm{Pre}(X_n)$, and let $\pi^n = (\xi_{n-1}, \xi_{n-2}, \ldots, \xi_1, \xi_0, \xi_0, \ldots, \xi_0, \ldots)$. We show by induction on n that for all strategies π_2 for player 2, and for all $s \in S$, we have $\mathrm{Pr}_s^{\pi_1^n,\pi_2}(\Diamond_{\leq n} R) \geq X_n$. For $n = 0$, the result is immediate; the result is also immediate for $s \in R$. For $n \geq 0$ and $s \notin R$, we have

$$\begin{aligned}
\mathrm{Pr}_s^{\pi^n}(\Diamond_{\leq n} R) &= \sum_{t\in S} \mathrm{Pr}_s^{\xi_{n-1}}(\Theta_1 = t) \cdot \mathrm{Pr}_t^{\pi^{n-1}}(\Diamond_{\leq n-1} R) \\
&\geq \sum_{t\in S} \mathrm{Pr}_s^{\xi_{n-1}}(\Theta_1 = t) \cdot X_{n-1}(t) \\
&\geq \mathrm{Pre}_1(X_{n-1})(s) = X_n(s).
\end{aligned}$$

where $\pi_2[t]$ is the strategy that behaves like π_2 after a transition to t has occurred. The lemma then follows from $w = \lim_{n\to\infty} X_n$, and from the fact that $\Diamond_{\leq n} R$ implies $\Diamond R$ for all $n \geq 0$. In fact, given any $\varepsilon > 0$, there is $n > 0$ such that $\max\{X(s) - X_n(s) \mid s \in S\} < \varepsilon$. When player 1 uses strategy π_1^n we have, for all strategies π_2 of player 2, $\mathrm{Pr}^{\pi_1^n,\pi_2}(\Diamond R) \geq \mathrm{Pr}^{\pi_1^n,\pi_2}(\Diamond_{\leq n} R) \geq X_n \geq w - \varepsilon$.
∎

To prove (7.6), we constructed strategies for player 1 that approximate $w = \mu X.([R] \sqcup \mathrm{Pre}_1(X))$ within an arbitrary amount. The construction of the strategies takes advantage of the fact that (7.6) is written in terms of the Pre_1 predecessor operator: the strategies are constructed by considering the distributions that realize the maximum for Pre_1.

A direct proof of (7.7) cannot exploit the same technique. One approach to proving (7.7) consists in showing that, for all strategies of player 1, player 2 has a strategy that ensures that R is reached with probability at most w. The difficulty of this approach lies in the necessity of considering arbitrary strategies for player 1. A second approach to proving (7.7) consists in constructing a strategy for player 2 such that, for all strategies of player 1, R is reached with probability at most

w. However, the solution formula, written in terms of Pre_1, offers no help in the construction of a strategy for player 2. Indeed, the classical proof of (7.7) relies on the analysis of the limit behavior of discounted versions of the reachability games, for values of the discount parameter in the neighbourhood of 1; the argument uses some complex facts from analysis (see, e.g., [FV97]).

We pursue here a different approach to the proof of (7.7). The approach was first used by [EJ91] for deterministic, turn-based games, and later extended in [dAM04] to probabilistic games. The approach consists in complementing the solution formula (7.5), obtaining the formula:

$$1 - \mu X.([R] \sqcup \mathrm{Pre}_1(X)) = \nu X.([S \setminus R] \sqcap \mathrm{Pre}_2(X)).$$

This complementary formula is written in terms of Pre_2. By considering the distributions that realize the maximum in Pre_2, we can construct a strategy for player 2, and we can show that the strategy guarantees a probability of staying always in $S \setminus R$ (the complementary goal to reaching R) of at least $1 - w$. This yields a proof of (7.7).

7.2 Complementation

For $f \in \mathcal{F}$, let $\neg f = (1 - f) \in \mathcal{F}$ denote the valuation defined by $(1 - f)(s) = 1 - f(s)$ for all $s \in S$. The complementation laws are as follows, for valuations $f, g \in \mathcal{F}$:

$$\begin{aligned} \neg(f \sqcup g) &= (\neg f) \sqcap (\neg g) \\ \neg(f \sqcap g) &= (\neg f) \sqcup (\neg g) \\ \neg(\mu X.f(X)) &= \nu X.\neg f(\neg X) \\ \neg(\nu X.f(X)) &= \mu X.\neg f(\neg X) \\ \neg\mathrm{Pre}_1(\neg X) &= \mathrm{Pre}_2(X). \end{aligned}$$

7.3 Safety

By complementing (7.5), swapping the roles of player 1 and player 2, and replacing R with $S \setminus R$, we obtain $\nu X.([R] \sqcap \mathrm{Pre}_1(X))$, which bears a close similarity to the solution formula for safety games in MDPs. The following theorem makes this connection precise.

Theorem 9

$$\langle\langle 1 \rangle\rangle \Box R = \nu X.([R] \sqcap \mathit{Pre}_1(X)). \tag{7.8}$$

The proof of this theorem, and of Theorem 8, proceeds via the following lemma, which proves (7.7).

Lemma 2 *Let $w = \nu X.([R] \sqcap \mathit{Pre}_1(X))$. Player 1 has a strategy π_1 such that $\mathrm{Pr}_s^{\pi_1,\pi_2}(\Box R) \geq w(s)$ for all $\pi_2 \in \Pi_2$ and all $s \in S$.*

The lemma can be proved using standard arguments about positive reward games [FV97]. We present here a simpler argument, taken from [dAM04].

Proof. Let π_1 be a strategy for player 1 that at all $s \in R$ plays according to an optimal distribution for $\mathrm{Pre}_1(w)(s)$, and at all $s \in S \setminus R$ plays arbitrarily. Fix a state $s_0 \in S$ and an arbitrary strategy

$\pi_2 \in \Pi_2$. The process $\{H_n\}_{n \geq 0}$ defined by $H_n = w(\Theta_n)$ is a submartingale [Wil91]: in fact, from $w(s) = \mathrm{Pre}_1(w)(s)$ for $s \in R$ and from the choice of π_1 follows that

$$\mathrm{E}_{s_0}^{\pi_1,\pi_2}\{H_{n+1} \mid H_0, H_1, \ldots, H_n\} \geq H_n$$

for all $n \geq 0$. Hence, we have $\mathrm{E}_{s_0}^{\pi_1,\pi_2}\{H_n\} \geq H_0 = w(s_0)$. Moreover, since $w(s) \leq 1$ at all $s \in S$, by inspection we have $\mathrm{E}_{s_0}^{\pi_1,\pi_2}\{H_n\} \leq \mathrm{Pr}_{s_0}^{\pi_1,\pi_2}(\Box_{\leq n} R)$, where $\Box_{\leq n} R$ is the event of staying in R for at least n steps. Combining these two inequalities we obtain $w(s_0) \leq \mathrm{Pr}_{s_0}^{\pi_1,\pi_2}(\Box_{\leq n} R)$, and the result follows from $\mathrm{Pr}_{s_0}^{\pi_1,\pi_2}(\Box R) = \lim_{n\to\infty} \mathrm{Pr}_{s_0}^{\pi_1,\pi_2}(\Box_{\leq n} R)$. ∎

Appendix A

Notation

A.1 Sequences

Given a set S, we denote with S^n the Cartesian product of n copies of S, for $i \geq 0$. We also let:

$$S^* = \bigcup_{n=0}^{\infty} S^n \qquad S^+ = \bigcup_{n=1}^{\infty} S^n \qquad S^\omega = [\mathbb{N} \mapsto S],$$

so that S^* is the set of finite strings over S, S^+ is the set of non-empty finite strings over S, and S^ω is the set of infinite strings over S. We also let $S^\bullet = S^* \cup S^\omega$. For a string $\sigma \in S^n$, we say that the *length* of σ, denoted $|\sigma|$, is n; for $\sigma \in S^\omega$, we say that $|\sigma| = \infty$. For $\sigma = s_0, s_1, s_2, \ldots,$, and $n \leq m < |\sigma|$, we let $\sigma_n = s_n$ and $\sigma_{[n,m]} = s_n, s_{n+1}, \ldots, s_m$; furthermore, we $\mathit{last}(\sigma) = \sigma_{|\sigma|-1}$. We often indicate a sequence $s_0, s_1, s_2, \ldots$ by $s_\bullet$.

For $\sigma, \sigma' \in S^\bullet$, we denote by $\sigma \cdot \sigma'$, or simply $\sigma\,\sigma'$, the concatenation of σ and σ'; note that if $|\sigma| = \infty$, then for all $\sigma' \in S^\bullet$ we have $\sigma \cdot \sigma' = \sigma$. For $\sigma, \sigma' \in S^\bullet$, if there is $\sigma'' \in S^\bullet$ such that $\sigma \cdot \sigma'' = \sigma'$, then we say that σ is a *prefix* of σ', and we indicate this by $\sigma \preceq \sigma'$; if $\sigma \preceq \sigma'$ and $\sigma \neq \sigma'$, we write $\sigma \prec \sigma'$.

The concatenation notion is propagated to sets: for $A, B \subseteq S^\bullet$, we let $A \cdot B = \{a \cdot b \mid a \in A, b \in B\}$. For $t \in S$, we let $S_t^+ = \{t\} \cdot S^*$ be the set of sequences beginning with t.

Bibliography

[Bel54] R.E. Bellman. The theory of dynamic programming. *Bull. Amer. Math. Soc.*, 60:503–516, 1954.

[Bel57] R.E. Bellman. *Dynamic Programming*. Princeton University Press, 1957.

[Ber95] D.P. Bertsekas. *Dynamic Programming and Optimal Control*. Athena Scientific, 1995. Volumes I and II.

[CdAH04] K. Chatterjee, L. de Alfaro, , and T.A. Henzinger. Trading memory for randomness. In *In* QEST 04: *Proceedings of the First International Conference on Quantitative Evaluation of Systems*. IEEE Computer Society Press, 2004.

[dAM04] L. de Alfaro and R. Majumdar. Quantitative solution of omega-regular games. *Journal of Computer and System Sciences*, 68:374–397, 2004.

[Der70] C. Derman. *Finite State Markovian Decision Processes*. Academic Press, 1970.

[EJ91] E.A. Emerson and C.S. Jutla. Tree automata, mu-calculus and determinacy (extended abstract). In *Proc. 32nd IEEE Symp. Found. of Comp. Sci.*, pages 368–377. IEEE Computer Society Press, 1991.

[Eve57] H. Everett. Recursive games. In *Contributions to the Theory of Games III*, volume 39 of *Annals of Mathematical Studies*, pages 47–78, 1957.

[FV97] J. Filar and K. Vrieze. *Competitive Markov Decision Processes*. Springer-Verlag, 1997.

[Imm81] N. Immerman. Number of quantifiers is better than number of tape cells. *J. Computer and System Sciences*, 22:384–406, 1981.

[KSK66] J.G. Kemeny, J.L. Snell, and A.W. Knapp. *Denumerable Markov Chains*. D. Van Nostrand Company, 1966.

[Mar98] D.A. Martin. The determinacy of Blackwell games. *The Journal of Symbolic Logic*, 63(4):1565–1581, 1998.

[Put94] M.L. Puterman. *Markov Decision Processes*. John Wiley and Sons, 1994.

[vN28] J. von Neumann. Zur Theorie der Gesellschaftsspiele. *Math. Annal*, 100:295–320, 1928.

[Wil91] D. Williams. *Probability With Martingales*. Cambridge University Press, 1991.