Logic and Games
SS 2009

Prof. Dr. Erich Grädel
Łukasz Kaiser, Tobias Ganzow

Mathematische Grundlagen der Informatik
RWTH Aachen

Contents

1 Finite Games and First-Order Logic

An important problem in the field of logics is the question for a given logic L, a structure $\mathfrak{A}$ and a formula $\psi \in L$, whether $\mathfrak{A}$ is a model of ψ. In this chapter we will discuss an approach to the solution of this *model checking problem* via games for some logics. Our goal is to reduce the problem $\mathfrak{A} \models \psi$ to a strategy problem for a *model checking game* $\mathcal{G}(\mathfrak{A}, \psi)$ played by two players called *Verifier* (or *Player* 0) and *Falsifier* (or *Player* 1). We want to have the following relation between these two problems:

$$\mathfrak{A} \models \psi \text{ iff Verifier has a winning strategy for } \mathcal{G}(\mathfrak{A}, \psi).$$

We can then do model checking by constructing or proving the existence of winning strategies.

1.1 Model Checking Games for Modal Logic

The first logic to be considered is propositional modal logic (ML). Let us first briefly review its syntax and semantics:

Definition 1.1. For a given set of actions A and atomic properties $\{P_i : i \in I\}$, the syntax of ML is inductively defined:

- All propositional logic formulae with propositional variables P_i are in ML.
- If $\psi, \varphi \in$ ML, then also $\neg\psi$, $(\psi \wedge \varphi)$ and $(\psi \vee \varphi) \in$ ML.
- If $\psi \in$ ML and $a \in A$, then $\langle a \rangle \psi$ and $[a]\psi \in$ ML.

Remark 1.2. If there is only one action $a \in A$, we write $\Diamond\psi$ and $\Box\psi$ instead of $\langle a \rangle \psi$ and $[a]\psi$, respectively.

Definition 1.3. A *transition system* or *Kripke structure* with actions from a set A and atomic properties $\{P_i : i \in I\}$ is a structure

$$\mathcal{K} = (V, (E_a)_{a\in A}, (P_i)_{i\in I})$$

with a universe V of states, binary relations $E_a \subseteq V \times V$ describing transitions between the states, and unary relations $P_i \subseteq V$ describing the atomic properties of states.

A transition system can be seen as a labelled graph where the nodes are the states of $\mathcal{K}$, the unary relations are labels of the states, and the binary transition relations are the labelled edges.

Definition 1.4. Let $\mathcal{K} = (V, (E_a)_{a\in A}, (P_i)_{i\in I})$ be a transition system, $\psi \in$ ML a formula and v a state of $\mathcal{K}$. The *model relationship* $\mathcal{K}, v \models \psi$, i.e. ψ holds at state v of $\mathcal{K}$, is inductively defined:

- $\mathcal{K}, v \models P_i$ if and only if $v \in P_i$.
- $\mathcal{K}, v \models \neg\psi$ if and only if $\mathcal{K}, v \not\models \psi$.
- $\mathcal{K}, v \models \psi \vee \varphi$ if and only if $\mathcal{K}, v \models \psi$ or $\mathcal{K}, v \models \varphi$.
- $\mathcal{K}, v \models \psi \wedge \varphi$ if and only if $\mathcal{K}, v \models \psi$ and $\mathcal{K}, v \models \varphi$.
- $\mathcal{K}, v \models \langle a\rangle\psi$ if and only if there exists w such that $(v, w) \in E_a$ and $\mathcal{K}, w \models \psi$.
- $\mathcal{K}, v \models [a]\,\psi$ if and only if $\mathcal{K}, w \models \psi$ holds for all w with $(v, w) \in E_a$.

Definition 1.5. For a transition system $\mathcal{K}$ and a formula ψ we define the *extension*

$$[\![\psi]\!]^{\mathcal{K}} := \{v : \mathcal{K}, v \models \psi\}$$

as the set of states of $\mathcal{K}$ where ψ holds.

Remark 1.6. In order to keep the following propositions short and easier to understand, we assume that all modal logic formulae are given in negation normal form, i.e. negations occur only at atoms. This does not change the expressiveness of modal logic as for every formula an equivalent one in negation normal form can be constructed. We omit a proof here, but the transformation can be easily achieved by applying

DeMorgan's laws and the duality of $\Box$ and $\Diamond$ (i.e. $\neg\langle a\rangle\psi \equiv [a]\neg\psi$ and $\neg[a]\psi \equiv \langle a\rangle\neg\psi$) to shift negations to the atomic subformulae.

We will now describe model checking games for ML. Given a transition system $\mathcal{K}$ and a formula $\psi \in$ ML, we define a game $\mathcal{G}$ that contains positions (φ, v) for every subformula φ of ψ and every $v \in V$. In this game, starting from position (φ, v), Verifier's goal is to show that $\mathcal{K}, v \models \varphi$, while Falsifier tries to prove $\mathcal{K}, v \not\models \varphi$.

In the game, Verifier is allowed to move at positions $(\varphi \vee \vartheta, v)$, where she can choose to move to position (φ, v) or (ϑ, v), and at positions $(\langle a\rangle\varphi, v)$, where she can move to position (φ, w) for a $w \in vE_a$. Analogously, Falsifier can move from $(\varphi \wedge \vartheta, v)$ to (φ, v) or (ϑ, v) and from $([a]\varphi, v)$ to (φ, w) for a $w \in vE_a$. Finally, there are the terminal positions (P_i, v) and $(\neg P_i, v)$, which are won by Verifier if $\mathcal{K}, v \models P_i$ and $\mathcal{K}, v \models \neg P_i$, respectively, otherwise they are winning positions for Falsifier.

The intuitive idea of this construction is to let the Verifier make the existential choices. To win from one of her positions, a disjunction or diamond subformula, she either needs to prove that one of the disjuncts is true, or that there exists a successor at which the subformula holds. Falsifier, on the other hand, in order to win from his positions, can choose a conjunct that is false or, if at a box formula, choose a successor at which the subformula does not hold.

The idea behind this construction is that at disjunctions and diamonds, Verifier can choose a subformula that is satisfied by the structure or a successor position at which the subformula is satisfied, while at conjunctions and boxes, Falsifier can choose a subformula or position that is not. So it is easy to see that the following lemma holds.

Lemma 1.7. Let $\mathcal{K}$ be a Kripke structure, $v \in V$ and φ a formula in ML. Then we have

$$\mathcal{K}, v \models \varphi \quad \Leftrightarrow \quad \text{Verifier has a winning strategy from } (\varphi, v).$$

To assess the efficiency of games as a solution for model checking problems, we have to consider the complexity of the resulting model checking games based on the following criteria:

- Are all plays necessarily finite?
- If not, what are the winning conditions for infinite plays?
- Do the players always have perfect information?
- What is the structural complexity of the game graphs?
- How does the size of the graph depend on different parameters of the input structure and the formula?

For first-order logic (FO) and modal logic (ML) we have only finite plays with positional winning conditions, and, as we will see, the winning regions are computable in linear time with respect to the size of the game graph (for finite structures of course).

Model checking games for fixed-point logics however admit infinite plays, and we use so called *parity conditions* to determine the winner of such plays. It is still an open question whether winning regions and winning strategies in parity games are computable in polynomial time.

1.2 Finite Games

In the following section we want to deal with two-player games with perfect information and positional winning conditions, given by a *game graph* (or *arena*)

$$\mathcal{G} = (V, E)$$

where the set V of positions is partitioned into sets of positions V_0 and V_1 belonging to Player 0 and Player 1, respectively. Player 0, also called *Ego*, moves from positions $v \in V_0$, while Player 1, called *Alter*, moves from positions $v \in V_1$. All moves are along edges, and we use the term *play* to describe a (finite or infinite) sequence $v_0 v_1 v_2 \dots$ with $(v_i, v_{i+1}) \in E$ for all i. We use a simple positional winning condition: Move or lose! Player σ wins at position v if $v \in V_{1-\sigma}$ and $vE = \emptyset$, i.e., if the position belongs to his opponent and there are no moves possible from that position. Note that this winning condition only applies to finite plays, infinite plays are considered to be a draw.

We define a *strategy* (for Player σ) as a mapping

$$f : \{v \in V_\sigma : vE \neq \emptyset\} \to V$$

with $(v, f(v)) \in E$ for all $v \in V$. We call f *winning* from position v if Player σ wins all plays that start at v and are consistent with f.

We now can define *winning regions* W_0 and W_1:

$$W_\sigma = \{v \in V : \text{Player } \sigma \text{ has a winning strategy from position } v\}.$$

This proposes several algorithmic problems for a given game $\mathcal{G}$: The computation of winning regions W_0 and W_1, the computation of winning strategies, and the associated decision problem

$$\textsc{Game} := \{(\mathcal{G}, v) : \text{Player 0 has a winning strategy for } \mathcal{G} \text{ from } v\}.$$

Theorem 1.8. Game is P-complete and decidable in time $O(|V| + |E|)$.

Note that this remains true for *strictly alternating games*.

A simple polynomial-time approach to solve Game is to compute the winning regions inductively: $W_\sigma = \bigcup_{n \in \mathbb{N}} W_\sigma^n$, where

$$W_\sigma^0 = \{v \in V_{1-\sigma} : vE = \emptyset\}$$

is the set of terminal positions which are winning for Player σ, and

$$W_\sigma^{n+1} = \{v \in V_\sigma : vE \cap W_\sigma^n \neq \emptyset\} \cup \{v \in V_{1-\sigma} : vE \subseteq W_\sigma^n\}$$

is the set of positions from which Player σ can win in at most $n+1$ moves.

After $n \leq |V|$ steps, we have that $W_\sigma^{n+1} = W_\sigma^n$, and we can stop the computation here.

To solve Game in linear time, we have to use the slightly more involved Algorithm 1.1. Procedure Propagate will be called once for every edge in the game graph, so the running time of this algorithm is linear with respect to the number of edges in $\mathcal{G}$.

Furthermore, we can show that the decision problem Game is equivalent to the satisfiability problem for propositional Horn formulae.

Algorithm 1.1. A linear time algorithm for GAME

```
Input: A game 𝒢 = (V, V0, V1, E)
output: Winning regions W0 and W1

for all v ∈ V do                  (* 1: Initialisation *)
    win[v] := ⊥
    P[v] := ∅
    n[v] := 0
end do

for all (u, v) ∈ E do             (* 2: Calculate P and n *)
    P[v] := P[v] ∪ {u}
    n[u] := n[u] + 1
end do

for all v ∈ V0                    (* 3: Calculate win *)
    if n[v] = 0 then Propagate(v, 1)
for all v ∈ V \ V0
    if n[v] = 0 then Propagate(v, 0)
return win

procedure Propagate(v, σ)
    if win[v] ≠ ⊥ then return
    win[v] := σ                   (* 4: Mark v as winning for player σ *)
    for all u ∈ P[v] do           (* 5: Propagate change to predecessors *)
        n[u] := n[u] − 1
        if u ∈ Vσ or n[u] = 0 then Propagate(u, σ)
    end do
end
```


We recall that propositional Horn formulae are finite conjunctions $\bigwedge_{i\in I} C_i$ of clauses C_i of the form

$$\begin{aligned} X_1 \wedge \ldots \wedge X_n &\rightarrow X \quad \text{or} \\ \underbrace{X_1 \wedge \ldots \wedge X_n}_{\text{body}(C_i)} &\rightarrow \underbrace{0}_{\text{head}(C_i)} \quad . \end{aligned}$$

A clause of the form X or $1 \rightarrow X$ has an empty body.

We will show that SAT-HORN and GAME are mutually reducible via logspace and linear-time reductions.

(1) GAME $\leq_{\text{log-lin}}$ SAT-HORN
For a game $\mathcal{G} = (V, V_0, V_1, E)$, we construct a Horn formula $\psi_{\mathcal{G}}$ with clauses

$$\begin{aligned} v &\rightarrow u && \text{for all } u \in V_0 \text{ and } (u,v) \in E, \text{ and} \\ v_1 \wedge \ldots \wedge v_m &\rightarrow u && \text{for all } u \in V_1 \text{ and } uE = \{v_1, \ldots, v_m\}. \end{aligned}$$

The minimal model of $\psi_{\mathcal{G}}$ is precisely the winning region of Player 0, so

$$(\mathcal{G}, v) \in \text{GAME} \iff \psi_{\mathcal{G}} \wedge (v \rightarrow 0) \text{ is unsatisfiable.}$$

(2) SAT-HORN $\leq_{\text{log-lin}}$ GAME
For a Horn formula $\psi(X_1, \ldots, X_n) = \bigwedge_{i\in I} C_i$, we define a game $\mathcal{G}_\psi = (V, V_0, V_1, E)$ as follows:

$$\begin{aligned} V &= \underbrace{\{0\} \cup \{X_1, \ldots, X_n\}}_{V_0} \cup \underbrace{\{C_i : i \in I\}}_{V_1} \quad \text{and} \\ E &= \{X_j \rightarrow C_i : X_j = \text{head}(C_i)\} \cup \{C_i \rightarrow X_j : X_j \in \text{body}(C_i)\}, \end{aligned}$$

i.e., Player 0 moves from a variable to some clause containing the variable as its head, and Player 1 moves from a clause to some variable in its body. Player 0 wins a play if, and only if, the play reaches a clause C with $\text{body}(C) = \emptyset$. Furthermore, Player 0 has a winning strategy from position X if, and only if, $\psi \models X$, so we have

$$\text{Player 0 wins from position 0} \iff \psi \text{ is unsatisfiable.}$$

These reductions show that Sat-Horn is also P-complete and, in particular, also decidable in linear time.

1.3 Alternating Algorithms

Alternating algorithms are algorithms whose set of configurations is divided into *accepting*, *rejecting*, *existential* and *universal* configurations. The acceptance condition of an alternating algorithm A is defined by a game played by two players $\exists$ and $\forall$ on the computation tree $\mathcal{T}_{A,x}$ of A on input x. The positions in this game are the configurations of A, and we allow moves $C \to C'$ from a configuration C to any of its successor configurations C'. Player $\exists$ moves at existential configurations and wins at accepting configurations, while Player $\forall$ moves at universal configurations and wins at rejecting configurations. By definition, A accepts some input x if and only if Player $\exists$ has a winning strategy for the game played on $\mathcal{T}_{A,x}$.

We will introduce the concept of alternating algorithms formally, using the model of a Turing machine, and we prove certain relationships between the resulting alternating complexity classes and usual deterministic complexity classes.

1.3.1 Turing Machines

The notion of an alternating Turing machine extends the usual model of a (deterministic) Turing machine which we introduce first. We consider Turing machines with a separate input tape and multiple linear work tapes which are divided into basic units, called cells or fields. Informally, the Turing machine has a reading head on the input tape and a combined reading and writing head on each of its work tapes. Each of the heads is at one particular cell of the corresponding tape during each point of a computation. Moreover, the Turing machine is in a certain state. Depending on this state and the symbols the machine

is currently reading on the input and work tapes, it manipulates the current fields of the work tapes, moves its heads and changes to a new state.

Formally, a *(deterministic) Turing machine with separate input tape and k linear work tapes* is given by a tuple $M = (Q, \Gamma, \Sigma, q_0, F_{\text{acc}}, F_{\text{rej}}, \delta)$, where Q is a finite set of *states*, Σ is the *work alphabet* containing a designated symbol $\square$ (*blank*), Γ is the *input alphabet*, $q_0 \in Q$ is the *initial state*, $F := F_{\text{acc}} \cup F_{\text{rej}} \subseteq Q$ is the set of *final states* (with F_{acc} the *accepting states*, F_{rej} the *rejecting states* and $F_{\text{acc}} \cap F_{\text{rej}} = \emptyset$), and $\delta : (Q \setminus F) \times \Gamma \times \Sigma^k \to Q \times \{-1,0,1\} \times \Sigma^k \times \{-1,0,1\}^k$ is the *transition function*.

A *configuration* of M is a complete description of all relevant facts about the machine at some point during a computation, so it is a tuple $C = (q, w_1, \dots, w_k, x, p_0, p_1, \dots, p_k) \in Q \times (\Sigma^*)^k \times \Gamma^* \times \mathbb{N}^{k+1}$ where q is the recent state, w_i is the contents of work tape number i, x is the contents of the input tape, p_0 is the position on the input tape and p_i is the position on work tape number i. The contents of each of the tapes is represented as a finite word over the corresponding alphabet[, i.e., a finite sequence of symbols from the alphabet]. The contents of each of the fields with numbers $j > |w_i|$ on work tape number i is the blank symbol (we think of the tape as being infinite). A configuration where x is omitted is called a *partial configuration*. The configuration C is called *final* if $q \in F$. It is called *accepting* if $q \in F_{\text{acc}}$ and *rejecting* if $q \in F_{\text{rej}}$.

The *successor configuration* of C is determined by the recent state and the $k+1$ symbols on the recent cells of the tapes, using the transition function: If $\delta(q, x_{p_0}, (w_1)_{p_1}, \dots, (w_k)_{p_k}) = (q', m_0, a_1, \dots, a_k, m_1, \dots, m_k, b)$, then the successor configuration of C is $\Delta(C) = (q', \overline{w}', \overline{p}', x)$, where for any i, w'_i is obtained from w_i by replacing symbol number p_i by a_i and $p'_i = p_i + m_i$. We write $C \vdash_M C'$ if, and only if, $C' = \Delta(C)$.

The *initial configuration* $C_0(x) = C_0(M, x)$ of M on input $x \in \Gamma^*$ is given by the initial state q_0, the blank-padded memory, i.e., $w_i = \varepsilon$ and $p_i = 0$ for any $i \geq 1$, $p_0 = 0$, and the contents x on the input tape.

A *computation* of M on input x is a sequence $C_0, C_1, \dots$ of configurations of M, such that $C_0 = C_0(x)$ and $C_i \vdash_M C_{i+1}$ for all $i \geq 0$. The computation is called *complete* if it is infinite or ends in some final configuration. A complete finite computation is called *accepting* if the last configuration is accepting, and the computation is called *rejecting* if the last configuration is rejecting. M *accepts* input x if the (unique) complete computation of M on x is finite and accepting. M *rejects* input x if the (unique) complete computation of M on x is finite and rejecting. The machine M *decides* a language $L \subseteq \Gamma^*$ if M accepts all $x \in L$ and rejects all $x \in \Gamma^* \setminus L$.

1.3.2 Alternating Turing Machines

Now we shall extend deterministic Turing machines to nondeterministic Turing machines from which the concept of alternating Turing machines is obtained in a very natural way, given our game theoretical framework.

A *nondeterministic Turing machine* is nondeterministic in the sense that a given configuration C may have several possible successor configurations instead of at most one. Intuitively, this can be described as the ability to *guess*. This is formalised by replacing the transition function $\delta : (Q \setminus F) \times \Gamma \times \Sigma^k \to Q \times \{-1,0,1\} \times \Sigma^k \times \{-1,0,1\}^k$ by a transition relation $\Delta \subseteq ((Q \setminus F) \times \Gamma \times \Sigma^k) \times (Q \times \{-1,0,1\} \times \Sigma^k \times \{-1,0,1\}^k)$. The notion of successor configurations is defined as in the deterministic case, except that the successor configuration of a configuration C may not be uniquely determined. Computations and all related notions carry over from deterministic machines in the obvious way. However, on a fixed input x, a nondeterministic machine now has several possible computations, which form a (possibly infinite) finitely branching computation tree $\mathcal{T}_{M,x}$. A nondeterministic Turing machine M *accepts* an input x if there *exists* a computation of M on x which is accepting, i.e., if there exists a path from the root $C_0(x)$ of $\mathcal{T}_{M,x}$ to some accepting configuration. The language of M is $L(M) = \{x \in \Gamma^* \mid M \text{ accepts } x\}$. Notice that for a nondeterministic machine M to decide a language $L \subseteq \Gamma^*$ it is not necessary, that all computations of M are finite. (In a sense, we count infinite computations as rejecting.)

From a game-theoretical perspective, the computation of a nondeterministic machine can be viewed as a solitaire game on the computation tree in which the only player (the machine) chooses a path

through the tree starting from the initial configuration. The player wins the game (and hence, the machine accepts its input) if the chosen path finally reaches an accepting configuration.

An obvious generalisation of this game is to turn it into a two-player game by assigning the nodes to the two players who are called $\exists$ and $\forall$, following the intuition that Player $\exists$ tries to show the existence of a *good* path, whereas Player $\forall$ tries to show that all selected paths are *bad*. As before, Player $\exists$ wins a play of the resulting game if, and only if, the play is finite and ends in an accepting leaf of the game tree. Hence, we call a computation tree accepting if, and only if, Player $\exists$ has a winning strategy for this game.

It is important to note that the partition of the nodes in the tree should not depend on the input x but is supposed to be inherent to the machine. Actually, it is even independent of the contents of the work tapes, and thus, whether a configuration belongs to Player $\exists$ or to Player $\forall$ merely depends on the current state.

Formally, an *alternating Turing machine* is a nondeterministic Turing machine $M = (Q, \Gamma, \Sigma, q_0, F_{\text{acc}}, F_{\text{rej}}, \Delta)$ whose set of states $Q = Q_\exists \cup Q_\forall \cup F_{\text{acc}} \cup F_{\text{rej}}$ is partitioned into *existential, universal, accepting*, and *rejecting* states. The semantics of these machines is given by means of the game described above.

Now, if we let accepting configurations belong to player $\forall$ and rejecting configurations belong to player $\exists$, then we have the usual winning condition that a player loses if it is his turn but he cannot move. We can solve such games by determining the winner at leaf nodes and propagating the winner successively to parent nodes. If at some node, the winner at all of its child nodes is determined, the winner at this node can be determined as well. This method is sometimes referred to as backwards induction and it basically coincides with our method for solving GAME on trees (with possibly infinite plays). This gives the following equivalent semantics of alternating Turing machines:

The subtree $\mathcal{T}_C$ of the computation tree of M on x with root C is called *accepting*, if

- C is accepting

- C is existential and there is a successor configuration C' of C such that $\mathcal{T}_{C'}$ is accepting or
- C is universal and $\mathcal{T}_{C'}$ is accepting for all successor configurations C' of C.

M accepts an input x, if $\mathcal{T}_{C_0(x)} = \mathcal{T}_{M,x}$ is accepting.

For functions $T, S : \mathbb{N} \to \mathbb{N}$, an alternating Turing machine M is called *T-time bounded* if, and only if, for any input x, each computation of M on x has length less or equal $T(|x|)$. The machine is called *S-space bounded* if, and only if, for any input x, during any computation of M on x, at most $S(|x|)$ cells of the work tapes are used. Notice that time boundedness implies finiteness of all computations which is not the case for space boundedness. The same definitions apply for deterministic and nondeterministic Turing machines as well since these are just special cases of alternating Turing machines. These notions of resource bounds induce the complexity classes ATIME containing precisely those languages L such that there is an alternating T-time bounded Turing machine deciding L and ASPACE containing precisely those languages L such that there is an alternating S-space bounded Turing machine deciding L. Similarly, these classes can be defined for nondeterministic and deterministic Turing machines.

We are especially interested in the following alternating complexity classes:

- ALOGSPACE $= \bigcup_{d \in \mathbb{N}}$ ASPACE$(d \cdot \log n)$,
- APTIME $= \bigcup_{d \in \mathbb{N}}$ ATIME(n^d),
- APSPACE $= \bigcup_{d \in \mathbb{N}}$ ASPACE(n^d).

Observe that GAME $\in$ ALOGSPACE. An alternating algorithm which decides GAME with logarithmic space just plays the game. The algorithm only has to store the *current* position in memory, and this can be done with logarithmic space. We shall now consider a slightly more involved example.

Example 1.9. QBF $\in$ ATIME(O(n)). W.l.o.g we assume that negation appears only at literals. We describe an alternating procedure $Eval(\varphi, \mathcal{I})$ which computes, given a quantified Boolean formula ψ and a valuation $\mathcal{I} : \text{free}(\psi) \to \{0, 1\}$ of the free variables of ψ, the value $[\![\psi]\!]^{\mathcal{I}}$.

Algorithm 1.2. Alternating algorithm deciding QBF.

Input: $(\psi, \mathcal{I})$	where $\psi \in$ QAL and $\mathcal{I}$: free$(\psi) \to \{0, 1\}$	
if $\psi = Y$	**then**	
	if $\mathcal{I}(Y) = 1$ **then** accept	
	else reject	
if $\psi = \varphi_1 \vee \varphi_2$	**then** „∃“ guesses $i \in \{1, 2\}$, $Eval(\varphi_i, \mathcal{I})$	
if $\psi = \varphi_1 \wedge \varphi_2$	**then** „∀“ chooses $i \in \{1, 2\}$, $Eval(\varphi_i, \mathcal{I})$	
if $\psi = \exists X \varphi$	**then** „∃“ guesses $j \in \{0, 1\}$, $Eval(\varphi, \mathcal{I}[X = j])$	
if $\psi = \forall X \varphi$	**then** „∀“ chooses $j \in \{0, 1\}$, $Eval(\varphi, \mathcal{I}[X = j])$	

The main results we want to establish in this section concern the relationship between alternating complexity classes and deterministic complexity classes. We will see that alternating time corresponds to deterministic space, while by translating deterministic time into alternating space, we can reduce the complexity by one exponential. Here, we consider the special case of alternating polynomial time and polynomial space. We should mention, however, that these results can be generalised to arbitrary large complexity bounds which are well behaved in a certain sense.

Lemma 1.10. NPSPACE $\subseteq$ APTIME.

Proof. Let $L \in$ NPSPACE and let M be a nondeterministic n^l-space bounded Turing machine which recognises L for some $l \in \mathbb{N}$. The machine M accepts some input x if, and only if, some accepting configuration is reachable from the initial configuration $C_0(x)$ in the configuration tree of M on x in at most $k := 2^{cn^l}$ steps for some $c \in \mathbb{N}$. This is due to the fact that there are most k different configurations of M on input x which use at most n^l cells of the memory which can be seen using a simple combinatorial argument. So if there is some accepting configuration reachable from the initial configuration $C_0(x)$, then there is some accepting configuration reachable from $C_0(x)$ in at most k steps. This is equivalent to the existence of some intermediate configuration C' that is reachable from $C_0(x)$ in at most $k/2$ steps and from which some accepting configuration is reachable in at most $k/2$ steps.

So the alternating algorithm deciding L proceeds as follows. The existential player guesses such a configuration C' and the universal player chooses whether to check that C' is reachable from $C_0(x)$ in at most $k/2$ steps or whether to check that some accepting configuration is reachable from C' in at most $k/2$ steps. Then the algorithm (or equivalently, the game) proceeds with the subproblem chosen by the universal player, and continues in this binary search like fashion. Obviously, the number of steps which have to be performed by this procedure to decide whether x is accepted by M is logarithmic in k. Since k is exponential in n^l, the time bound of M is dn^l for some $d \in \mathbb{N}$, so M decides L in polynomial time. Q.E.D.

Lemma 1.11. APTIME $\subseteq$ PSPACE.

Proof. Let $L \in$ APTIME and let A be an alternating n^l-time bounded Turing machine that decides L for some $l \in \mathbb{N}$. Then there is some $r \in \mathbb{N}$ such that any configuration of A on any input x has at most r successor configurations and w.l.o.g. we can assume that any non-final configuration has precisely r successor configurations. We can think of the successor configurations of some non-final configuration C as being enumerated as $C_1, \dots, C_r$. Clearly, for given C and i we can compute C_i. The idea for a deterministic Turing machine M to check whether some input x is in L is to perform a depth-first search on the computation tree $\mathcal{T}_{A,x}$ of A on x. The crucial point is, that we cannot construct and keep the whole configuration tree $\mathcal{T}_{A,x}$ in memory since its size is exponential in $|x|$ which exceeds our desired space bound. However, since the length of each computation is polynomially bounded, it is possible to keep a single computation path in memory and to construct the successor configurations of the configuration under consideration on the fly.

Roughly, the procedure M can be described as follows. We start with the initial configuration $C_0(x)$. Given any configuration C under consideration, we propagate 0 to the predecessor configuration if C is rejecting and we propagate 1 to the predecessor configuration if C is accepting. If C is neither accepting nor rejecting, then we construct,

for $i = 1, \dots, r$ the successor configuration C_i of C and proceed with checking C_i. If C is existential, then as soon as we receive 1 for some i, we propagate 1 to the predecessor. If we encounter 0 for all i, then we propagate 0. Analogously, if C is universal, then as soon as we receive a 0 for some i, we propagate 0. If we receive only 1 for all i, then we propagate 1. Then x is in L if, and only if, we finally receive 1 at $C_0(x)$. Now, at any point during such a computation we have to store at most one complete computation of A on x. Since A is n^l-time bounded, each such computation has length at most n^l and each configuration has size at most $c \cdot n^l$ for some $c \in \mathbb{N}$. So M needs at most $c \cdot n^{2l}$ memory cells which is polynomial in n. Q.E.D.

So we obtain the following result.

Theorem 1.12. (Parallel time complexity = sequential space complexity)

(1) APTIME $=$ PSPACE.

(2) AEXPTIME $=$ EXPSPACE.

Proposition (2) of this theorem is proved exactly the same way as we have done it for proposition (1). Now we prove that by translating sequential *time* into alternating *space*, we can reduce the complexity by one exponential.

Lemma 1.13. EXPTIME $\subseteq$ APSPACE

Proof. Let $L \in$ EXPTIME. Using a standard argument from complexity theory, there is a deterministic Turing machine $M = (Q, \Sigma, q_0, \delta)$ with time bound $m := 2^{c \cdot n^k}$ for some $c, k \in \mathbb{N}$ with only a single tape (serving as both input and work tape) which decides L. (The time bound of the machine with only a single tape is quadratic in that of the original machine with k work tapes and a separate input tape, which, however, does not matter in the case of an exponential time bound.) Now if $\Gamma = \Sigma \uplus (Q \times \Sigma) \uplus \{\#\}$, then we can describe each configuration C of M by a word

$$\underline{C} = \#w_0 \dots w_{i-1}(qw_i)w_{i+1} \dots w_t\# \in \Gamma^*.$$

Since M has time bound m and only one single tape, it has space bound m. So, w.l.o.g., we can assume that $|\underline{C}| = m + 2$ for all configurations C of M on inputs of length n. (We just use a representation of the tape which has a priori the maximum length that will occur during a computation on an input of length n.) Now the crucial point in the argumentation is the following. If $C \vdash C'$ and $1 \leq i \leq m$, symbol number i of the word $\underline{C'}$ only depends on the symbols number $i - 1$, i and $i + 1$ of $\underline{C}$. This allows us, to decide whether $x \in L(M)$ with the following alternating procedure which uses only polynomial space.

Player $\exists$ guesses some number $s \leq m$ of steps of which he claims that it is precisely the length of the computation of M on input x. Furthermore, $\exists$ guesses some state $q \in F_{\text{acc}}$, a Symbol $a \in \Sigma$ and a number $i \in \{0, \ldots, s\}$, and he claims that the i-th symbol of the configuration $\underline{C}$ of M after the computation on x is (qa). (So players start inspecting the computation of M on x from the final configuration.) If M accepts input x, then obviously player $\exists$ has a possibility to choose all these objects such that his claims can be validated. Player $\forall$ wants to disprove the claims of $\exists$. Now, player $\exists$ guesses symbols $a_{-1}, a_0, a_1 \in \Gamma$ of which he claims that these are the symbols number $i - 1$, i and $i + 1$ of the predecessor configuration of the final configuration $\underline{C}$. Now, $\forall$ can choose any of these symbols and demand, that $\exists$ validates his claim for this particular symbol. This symbol is now the symbol under consideration, while i is updated according to the movement of the (unique) head of M. Now, these actions of the players take place for each of the s computation steps of M on x. After s such steps, we check whether the recent symbol and the recent position are consistent with the initial configuration $C_0(x)$. The only information that has to be stored in the memory is the position i on the tape, the number s which $\exists$ has initially guessed and the current number of steps. Therefore, the algorithm uses space at most $\mathrm{O}(\log(m)) = \mathrm{O}(n^k)$ which is polynomial in n. Moreover, if M accepts input x then obviously, player $\exists$ has a winning strategy for the computation game. If, conversely, M rejects input x, then the combination of all claims of player $\exists$ cannot be consistent and player $\forall$ has a strategy to spoil any (cheating) strategy of player $\exists$ by choosing the appropriate symbol at the appropriate

computation step. Q.E.D.

Finally, we make the simple observation that it is not possible to gain more than one exponential when translating from sequential time to alternating space. (Notice that EXPTIME is a proper subclass of 2EXPTIME.)

Lemma 1.14. APSPACE $\subseteq$ EXPTIME

Proof. Let $L \in$ APSPACE, and let A be an alternating n^k-space bounded Turing machine which decides L for some $k \in \mathbb{N}$. Moreover, for an input x of A, let $\mathrm{Conf}(A, x)$ be the set of all configurations of A on input x. Due to the polynomial space bound of A, this set is finite and its size is at most exponential in $|x|$. So we can construct the graph $G = (\mathrm{Conf}(A, x), \vdash)$ in time exponential in $|x|$. Moreover, a configuration C is reachable from $C_0(x)$ in $\mathcal{T}_{A,x}$ if and only if C is reachable from $C_0(x)$ in G. So to check whether A accepts input x we simply decide whether player $\exists$ has a winning strategy for the game played on G from $C_0(x)$. This can be done in time linear in the size of G, so altogether we can decide whether $x \in L(A)$ in time exponential in $|x|$. Q.E.D.

Theorem 1.15. (Translating sequential time into alternating space)

(1) ALOGSPACE = P.
(2) APSPACE = EXPTIME.

Proposition (1) of this theorem is proved using exactly the same arguments as we have used for proving proposition (2). An overview over the relationship between deterministic and alternating complexity classes is given in Figure 1.1.

LOGSPACE	$\subseteq$	PTIME	$\subseteq$	PSPACE	$\subseteq$	EXPTIME	$\subseteq$	EXPSPACE
		$\|\|$		$\|\|$		$\|\|$		$\|\|$
		ALOGSPACE	$\subseteq$	APTIME	$\subseteq$	APSPACE	$\subseteq$	AEXPTIME

Figure 1.1. Relation between deterministic and alternating complexity classes

1.4 Model Checking Games for First-Order Logic

Let us first recall the syntax of FO formulae on relational structures. We have that $R_i(\bar{x})$, $\neg R_i(\bar{x})$, $x = y$ and $x \neq y$ are well-formed valid FO formulae, and inductively for FO formulae φ and ψ, we have that $\varphi \vee \psi$, $\varphi \wedge \psi$, $\exists x \varphi$ and $\forall x \varphi$ are well-formed FO formulae. This way, we allow only formulae in *negation normal form* where negations occur only at atomic subformulae and all junctions except $\vee$ and $\wedge$ are eliminated. These constraints do not limit the expressiveness of the logic, but the resulting games are easier to handle.

For a structure $\mathfrak{A} = (A, R_1, \dots, R_m)$ with $R_i \subseteq A^{r_i}$, we define the evaluation game $\mathcal{G}(\mathfrak{A}, \psi)$ as follows:

We have positions $\varphi(\bar{a})$ for every subformula $\varphi(\bar{x})$ of ψ and every $\bar{a} \in A^k$.

At a position $\varphi \vee \vartheta$, Verifier can choose to move either to φ or to ϑ, while at positions $\exists x \varphi(x, \bar{b})$, he can choose an instantiation $a \in A$ of x and move to $\varphi(a, \bar{b})$. Analogously, Falsifier can move from positions $\varphi \wedge \vartheta$ to either φ or ϑ and from positions $\forall x \varphi(x, \bar{b})$ to $\varphi(a, \bar{b})$ for an $a \in A$.

The winning condition is evaluated at positions with atomic or negated atomic formulae φ, and we define that Verifier wins at $\varphi(\bar{a})$ if, and only if, $\mathfrak{A} \models \varphi(\bar{a})$, and Falsifier wins if, and only if, $\mathfrak{A} \not\models \varphi(\bar{a})$.

In order to determine the complexity of FO model checking, we have to consider the process of determining whether $\mathfrak{A} \models \psi$. To decide this question, we have to construct the game $\mathcal{G}(\mathfrak{A}, \psi)$ and check whether Verifier has a winning strategy from position ψ. The size of the game graph is bound by $|\mathcal{G}(\mathfrak{A}, \psi)| \leq |\psi| \cdot |A|^{\mathrm{width}(\psi)}$, where $\mathrm{width}(\psi)$ is the maximal number of free variables in the subformulae of ψ. So the game graph can be exponential, and therefore we can get only exponential time complexity for Game. In particular, we have the following complexities for the general case:

- alternating time: $O(|\psi| + \mathrm{qd}(\psi) \log |A|)$
  where $\mathrm{qd}(\psi)$ is the quantifier-depth of ψ,
- alternating space: $O(\mathrm{width}(\psi) \cdot \log |A| + \log |\psi|)$,
- deterministic time: $O(|\psi| \cdot |A|^{\mathrm{width}(\psi)})$ and

- deterministic space: $O(|\psi| + \mathrm{qd}(\psi) \log |A|)$.

Efficient implementations of model checking algorithms will construct the game graph on the fly while solving the game.

There are several possibilities of how to reason about the complexity of FO model checking. We can consider the *structural complexity*, i.e., we fix a formula and measure the complexity of the model checking algorithm in terms of the size of the structure only. On the other hand, the *expression complexity* measures the complexity in terms of the size of a given formula while the structure is considered to be fixed. Finally, the *combined complexity* is determined by considering both, the formula and the structure, as input parameters.

We obtain that the structural complexity of FO model checking is ALogtime, and both the expression complexity and the combined complexity is PSpace.

1.4.1 *Fragments of* FO *with Efficient Model Checking*

We have just seen that in the general case the complexity of FO model checking is exponential with respect to the width of the formula. In this section, we will see that some restrictions made to the underlying logic will also reduce the complexity of the associated model checking problem.

We will start by considering the *k-variable fragment of* FO :

$$\mathrm{FO}^k := \{\psi \in \mathrm{FO} : \mathrm{width}(\psi) \leq k\}.$$

In this fragment, we have an upper bound for the width of the formulae, and we get polynomial time complexity:

ModCheck(FO^k) is P-complete and solvable in time $O(|\psi| \cdot |A|^k)$.

There are other fragments of FO that have model checking complexity $O(|\psi| \cdot \|\mathfrak{A}\|)$:

- ML: propositional modal logic,
- FO^2: formulae of width two,
- GF: the guarded fragment of first-order logic.

We will have a closer look at the last one, GF.

GF is a fragment of first-order logic which allows only guarded quantification

$$\exists \bar{y}(\alpha(\bar{x},\bar{y}) \land \varphi(\bar{x},\bar{y})) \text{ and } \forall \bar{y}(\alpha(\bar{x},\bar{y}) \rightarrow \varphi(\bar{x},\bar{y}))$$

where the *guards* α are atomic formulae containing all free variables of φ.

GF is a generalisation of modal logics, and we have that ML $\subseteq$ GF $\subseteq$ FO. In particular, the modal logic quantifiers $\Diamond$ and $\Box$ can be expressed as

$$\langle a \rangle \varphi \equiv \exists y(E_a xy \land \varphi(y)) \text{ and } [a]\varphi \equiv \forall y(E_a xy \rightarrow \varphi(y)).$$

Since guarded logics have small model checking games of size $\|\mathcal{G}(\mathfrak{A},\psi)\| = O(|\psi| \cdot \|\mathfrak{A}\|)$, there exist efficient game-based model checking algorithms for them.

2 Parity Games and Fixed-Point Logics

2.1 Parity Games

In the previous section we presented model checking games for first-order logic and modal logic. These games admit only finite plays and their winning conditions are specified just by sets of positions. Winning regions in these games can be computed in linear time with respect to the size of the game graph.

However, in many computer science applications, more expressive logics like temporal logics, dynamic logics, fixed-point logics and others are needed. Model checking games for these logics admit infinite plays and their winning conditions must be specified in a more elaborate way. As a consequence, we have to consider the theory of infinite games.

For fixed-point logics, such as LFP or the modal μ-calculus, the appropriate evaluation games are *parity games*. These are games of possibly infinite duration where to each position a natural number is assigned. This number is called the *priority* of the position, and the winner of an infinite play is determined according to whether the least priority seen infinitely often during the play is even or odd.

Definition 2.1. We describe a *parity game* by a labelled graph $\mathcal{G} = (V, V_0, V_1, E, \Omega)$ where (V, V_0, V_1, E) is a game graph and $\Omega : V \rightarrow \mathbb{N}$, with $|\Omega(V)|$ finite, assigns a *priority* to each position. The set V of positions may be finite or infinite, but the number of different priorities, called the *index* of $\mathcal{G}$, must be finite. Recall that a finite play of a game is lost by the player who gets stuck, i.e. cannot move. For infinite plays $v_0 v_1 v_2 \ldots$, we have a special winning condition: If the least number appearing infinitely often in the sequence $\Omega(v_0)\Omega(v_1)\ldots$ of priorities is even, then Player 0 wins the play, otherwise Player 1 wins.

Definition 2.2. A strategy (for Player σ) is a function

$$f : V^* V_\sigma \to V$$

such that $f(v_0 v_1 \dots v_n) \in v_n E$.

We say that a play $\pi = v_0 v_1 \dots$ is *consistent* with the strategy f of Player σ if for each $v_i \in V_\sigma$ it holds that $v_{i+1} = f(v_i)$. The strategy f is *winning* for Player σ from (or on) a set $W \subseteq V$ if each play starting in W that is consistent with f is winning for Player σ.

In general, a strategy depends on the whole history of the game. However, in this chapter, we are interested in simple strategies that depend only on the current position.

Definition 2.3. A strategy (of Player σ) is called *positional* (or *memoryless*) if it only depends on the current position, but not on the history of the game, i.e. $f(hv) = f(h'v)$ for all $h, h' \in V^*, v \in V$. We often view positional strategies simply as functions $f : V \to V$.

We will see that such positional strategies suffice to solve parity games by proving the following theorem.

Theorem 2.4 (Forgetful Determinacy)**.** In any parity game, the set of positions can be partitioned into two sets W_0 and W_1 such that Player 0 has a positional strategy that is winning on W_0 and Player 1 has a positional strategy that is winning on W_1.

Before proving the theorem, we give two general examples of positional strategies, namely attractor and trap strategies, and show how positional winning strategies on parts of the game graph may be combined to positional winning strategies on larger regions.

Remark 2.5. Let f and f' be positional strategies for Player σ that are winning on the sets W, W', respectively. Let $f + f'$ be the positional strategy defined by

$$(f + f')(x) := \begin{cases} f(x) & \text{if } x \in W \\ f'(x) & \text{otherwise.} \end{cases}$$

Then $f + f'$ is a winning strategy on $W \cup W'$.

Definition 2.6. Let $\mathcal{G} = (V, V_0, V_1, E)$ be a game and $X \subseteq V$. We define the *attractor of X for Player σ* as

$$\begin{aligned} \mathrm{Attr}_\sigma(X) = \{v \in V : {}& \text{Player } \sigma \text{ has a (w.l.o.g. positional) strategy} \\ & \text{to reach some position } x \in X \cup T_\sigma \\ & \text{in finitely many steps}\} \end{aligned}$$

where $T_\sigma = \{v \in V_{1-\sigma} : vE = \emptyset\}$ denotes the set of terminal positions in which Player σ has won.

A set $X \subseteq V$ is called a *trap* for Player σ if Player $1 - \sigma$ has a (w.l.o.g. positional) strategy that avoids leaving X from every $x \in X$.

We can now turn to the proof of the Forgetful Determinacy Theorem.

Proof. Let $\mathcal{G} = (V, V_0, V_1, E, \Omega)$ be a parity game with $|\Omega(V)| = m$. Without loss of generality we can assume that $\Omega(V) = \{0, \dots, m-1\}$ or $\Omega(V) = \{1, \dots, m\}$. We prove the statement by induction over $|\Omega(V)|$.

In the case of $|\Omega(V)| = 1$, i.e., $\Omega(V) = \{0\}$ or $\Omega(V) = \{1\}$, the theorem clearly holds as either Player 0 or Player 1 wins every infinite play. Her opponent can only win by reaching a terminal position that does not belong to him. So we have, for $\Omega(V) = \{\sigma\}$,

$$\begin{aligned} W_{1-\sigma} &= \mathrm{Attr}_{1-\sigma}(T_{1-\sigma}) \text{ and} \\ W_\sigma &= V \setminus W_{1-\sigma}. \end{aligned}$$

Computing $W_{1-\sigma}$ as the attractor of $T_{1-\sigma}$ is a simple reachability problem, and thus it can be solved with a positional strategy. Concerning W_σ, it can be seen that there is a positional strategy that avoids leaving this $(1-\sigma)$-trap.

Let $|\Omega(v)| = m > 1$. We only consider the case $0 \in \Omega(V)$, i.e., $\Omega(V) = \{0, \dots, m-1\}$ since otherwise we can use the same argumentation with switched roles of the players. We define

$$X_1 := \{v \in V : \text{Player 1 has positional winning strategy from } v\},$$

and let g be a positional winning strategy for Player 1 on X_1.

Our goal is to provide a positional winning strategy f^* for Player 0 on $V \setminus X_1$, so in particular we have $W_1 = X_1$ and $W_0 = V \setminus X_1$.

First of all, observe that $V \setminus X_1$ is a trap for Player 1. Indeed, if Player 1 could move to X_1 from a $v \in V_1 \setminus X_1$, then v would also be in X_1. Thus, there exists a positional *trap strategy* f for Player 0 that guarantees to stay in $V \setminus X_1$.

Let $Y = \Omega^{-1}(0) \setminus X_1$, $Z = \text{Attr}_0(Y)$ and let a be an *attractor strategy* for Player 0 which guarantees that Y (or a terminal winning position $y \in T_0$) can be reached from every $z \in Z \setminus Y$. Moreover, let $V' = V \setminus (X_1 \cup Z)$.

The restricted game $\mathcal{G}' = \mathcal{G}|_{V'}$ has less priorities than $\mathcal{G}$ (since at least all positions with priority 0 have been removed). Thus, by induction hypothesis, the Forgetful Determinacy Theorem holds for $\mathcal{G}'$: $V' = W_0' \cup W_1'$ and there exist positional winning strategies f' for Player 0 on W_0' and g' for Player 1 on W_1' in $\mathcal{G}'$.

We have that $W_1' = \emptyset$, as the strategy

$$g + g' : x \mapsto \begin{cases} g(x) & x \in X_1 \\ g'(x) & x \in W_1' \end{cases}$$

is a positional winning strategy for Player 1 on $X_1 \cup W_1'$. Indeed, every play consistent with $g + g'$ either stays in W_1' and is consistent with g' or reaches X_1' and is from this point on consistent with g. But X_1, by definition, already contains *all* positions from which Player 1 can win with a positional strategy, so $W_1' = \emptyset$.

Knowing that $W_1' = \emptyset$, let $f^* = f' + a + f$, i.e.

$$f^*(x) = \begin{cases} f'(x) & \text{if } x \in W_0' \\ a(x) & \text{if } x \in Z \setminus Y \\ f(x) & \text{if } x \in Y \end{cases}$$

We claim that f^* is a positional winning strategy for Player 0 from $V \setminus X_1$. If π is a play consistent with f^*, then π stays in $V \setminus X_1$.

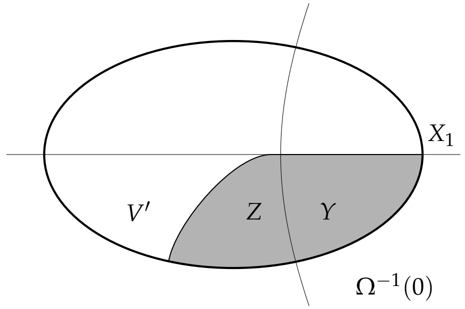


Figure 2.1. Construction of a winning strategy

Case (a): π hits Z only finitely often. Then π eventually stays in W_0' and is consistent with f' from this point, so Player 0 wins π.
Case (b): π hits Z infinitely often. Then π also hits Y infinitely often, which implies that priority 0 is seen infinitely often. Thus, Player 0 wins π. Q.E.D.

The following theorem is a consequence of positional determinacy.

Theorem 2.7. It can be decided in NP $\cap$ coNP whether a given position in a parity game is a winning position for Player 0.

Proof. A node v in a parity game $\mathcal{G} = (V, V_0, V_1, E, \Omega)$ is a winning position for Player σ if there exists a positional strategy $f : V_\sigma \to V$ which is winning from position v. It therefore suffices to show that the question whether a given strategy $f : V_\sigma \to V$ is a winning strategy for Player σ from position v can be decided in polynomial time. We prove this for Player 0; the argument for Player 1 is analogous.

Given $\mathcal{G}$ and $f : V_0 \to V$, we obtain a reduced game graph $\mathcal{G}_f = (W, F)$ by retaining only those moves that are consistent with f, i.e.,

$$F = \{(v, w) : (v \in W \cap V_\sigma \wedge w = f(v)) \vee \\ (v \in W \cap V_{1-\sigma} \wedge (v, w) \in E)\}.$$

In this reduced game, only the opponent, Player 1, makes non-trivial moves. We call a cycle in (W, F) odd if the least priority of its nodes is odd. Clearly, Player 0 wins $\mathcal{G}$ from position v via strategy f if, and only if, in $\mathcal{G}_f$ no odd cycle and no terminal position $w \in V_0$ is reachable from v. Since the reachability problem is solvable in polynomial time, the claim follows. Q.E.D.

2.1.1 Algorithms for parity games

It is an open question whether winning sets and winning strategies for parity games can be computed in polynomial time. The best algorithms known today are polynomial in the size of the game, but exponential with respect to the number of priorities. Such algorithms run in polynomial time when the number of priorities in the input parity game is bounded.

One way to intuitively understand an algorithm solving a parity game is to imagine a judge who watches the players playing the game. At some point, the judge is supposed to say "Player 0 wins", and indeed, whenever the judge does so, there should be no question that Player 0 wins. Note that we have no condition in case that Player 1 wins. We will first give a formal definition of a certain kind of judge with bounded memory, and later use this notion to construct algorithms for parity games.

Definition 2.8. A *judge* $\mathcal{M} = (M, m_0, \delta, F)$ for a parity game $\mathcal{G} = (V, V_0, V_1, E, \Omega)$ consists of a set of states M with a distinguished initial state $m_0 \in M$, a set of final states $F \subseteq M$, and a transition function $\delta : V \times M \to M$. Note that a judge is thus formally the same as an automaton reading words over the alphabet V. But to be called a judge, two special properties must be fulfilled. Let $v_0 v_1 \dots$ be a play of $\mathcal{G}$ and $m_0 m_1 \dots$ the corresponding sequence of states of $\mathcal{M}$, i.e., m_0 is the initial state of $\mathcal{M}$ and $m_{i+1} = \delta(v_i, m_i)$. Then the following holds:

(1) if $v_0 \dots$ is winning for Player 0, then there is a k such that $m_k \in F$,
(2) if, for some k, $m_k \in F$, then there exist $i < j \leq k$ such that $v_i = v_j$ and $\min\{\Omega(v_{i+1}), \Omega(v_{i+2}), \dots, \Omega(v_j)\}$ is even.

To illustrate the second condition in the above definition, note that in the play $v_0 v_1 \dots$ the sequence $v_i v_{i+1} \dots v_j$ forms a cycle. The judge is

indeed truthful, because both players can use a positional strategy in a parity game, so if a cycle with even priority appears, then Player 0 can be declared as the winner. To capture this intuition formally, we define the following reachability game, which emerges as the product of the original game $\mathcal{G}$ and the judge $\mathcal{M}$.

Definition 2.9. Let $\mathcal{G} = (V, V_0, V_1, E, \Omega)$ be a parity game and $\mathcal{M} = (M, m_0, \delta, F)$ an automaton reading words over V. The reachability game $\mathcal{G} \times \mathcal{M}$ is defined as follows:

$$\mathcal{G} \times \mathcal{M} = (V \times M, V_0 \times M, V_1 \times M, E', V \times F),$$

where $((v, m),\ (v', m')) \in E'$ iff $(v, v') \in E$ and $m' = \delta(v, m)$, and the last component $V \times F$ denotes positions which are immediately winning for Player 0 (the goal of Player 0 is to reach such a position).

Note that $\mathcal{M}$ in the definition above is a deterministic automaton, i.e., δ is a function. Therefore, in $\mathcal{G}$ and in $\mathcal{G} \times \mathcal{M}$ the players have the same choices, and thus it is possible to translate strategies between $\mathcal{G}$ and $\mathcal{G} \times \mathcal{M}$. Formally, for a strategy σ in $\mathcal{G}$ we define the strategy $\overline{\sigma}$ in $\mathcal{G} \times \mathcal{M}$ as

$$\overline{\sigma}((v_0, m_0)(v_1, m_1) \dots (v_n, m_n)) = (\sigma(v_0 v_1 \dots v_n), \delta(v_n, m_n)).$$

Conversely, given a strategy σ in $\mathcal{G} \times \mathcal{M}$ we define the strategy $\underline{\sigma}$ in $\mathcal{G}$ such that $\underline{\sigma}(v_0 v_1 \dots v_n) = v_{n+1}$ if and only if

$$\sigma((v_0, m_0)(v_1, m_1) \dots (v_n, m_n)) = (v_{n+1}, m_{n+1}),$$

where $m_0 m_1 \dots$ is the unique sequence corresponding to $v_0 v_1 \dots$.

Having defined $\mathcal{G} \times \mathcal{M}$, we are ready to formally prove that the above definition of a judge indeed makes sense for parity games.

Theorem 2.10. Let $\mathcal{G}$ be a parity game and $\mathcal{M}$ a judge for $\mathcal{G}$. Then Player 0 wins $\mathcal{G}$ from v_0 if and only if he wins $\mathcal{G} \times \mathcal{M}$ from (v_0, m_0).

Proof. ($\Rightarrow$) By contradiction, let σ be the winning strategy for Player 0 in $\mathcal{G}$ from v_0, and assume that there exists a winning strategy ρ for

Player 1 in $\mathcal{G} \times \mathcal{M}$ from (v_0, m_0). (Note that we just used determinacy of reachability games.) Consider the unique plays

$$\pi_{\mathcal{G}} = v_0 v_1 \dots \quad \text{and} \quad \pi_{\mathcal{G}\times\mathcal{M}} = (v_0, m_0)(v_1, m_1)\dots$$

in $\mathcal{G}$ and $\mathcal{G} \times \mathcal{M}$, respectively, which are consistent with both σ and $\underline{\rho}$ (the play $\pi_{\mathcal{G}}$) and with $\overline{\sigma}$ and ρ ($\pi_{\mathcal{G}\times\mathcal{M}}$). Observe that the positions of $\mathcal{G}$ appearing in both plays are indeed the same due to the way $\overline{\sigma}$ and $\underline{\rho}$ are defined. Since Player 0 wins $\pi_{\mathcal{G}}$, by Property (1) in the definition of a judge there must be an $m_k \in F$. But this contradicts the fact that Player 1 wins $\pi_{\mathcal{G}\times\mathcal{M}}$.

($\Leftarrow$) Let σ be a winning strategy for Player 0 in $\mathcal{G} \times \mathcal{M}$, and let ρ be a *positional* winning strategy for Player 1 in $\mathcal{G}$. Again, we consider the unique plays

$$\pi_{\mathcal{G}} = v_0 v_1 \dots \quad \pi_{\mathcal{G}\times\mathcal{M}} = (v_0, m_0)(v_1, m_1)\dots$$

such that $\pi_{\mathcal{G}}$ is consistent with $\underline{\sigma}$ and ρ, and $\pi_{\mathcal{G}\times\mathcal{M}}$ is consistent with σ and $\overline{\rho}$. Since $\pi_{\mathcal{G}\times\mathcal{M}}$ is won by Player 0, there is an $m_k \in F$ appearing in this play.

By Property (2) in the definition of a judge, there exist two indices $i < j$ such that $v_i = v_j$ and the minimum priority appearing between v_i and v_j is even. Let us now consider the following strategy σ' for Player 0 in $\mathcal{G}$:

$$\sigma'(w_0 w_1 \dots w_n) = \begin{cases} \underline{\sigma}(w_0 w_1 \dots w_n) & \text{if } n < j, \\ \underline{\sigma}(w_0 w_1 \dots w_m) & \text{otherwise,} \end{cases}$$

where $m = i + [(n - i) \mod (j - i)]$. Intuitively, the strategy σ' makes the same choices as $\underline{\sigma}$ up to the $(j-1)$st step, and then repeats the choices of $\underline{\sigma}$ from steps $i, i+1, \dots, j-1$.

We will now show that the unique play π' in $\mathcal{G}$ that is consistent with both σ' and ρ is won by Player 0. Since in the first j steps σ' is the same as $\underline{\sigma}$, we have that $\pi[n] = v_n$ for all $n \leq j$. Now observe that $\pi[j+1] = v_{i+1}$. Since ρ is positional, if v_j is a position of Player 1, then $\pi[j+1] = v_{i+1}$, and if v_j is a position of Player 0, then $\pi[j+1] = v_{i+1}$

because we defined $\sigma'(v_0 \dots v_j) = \sigma(v_0 \dots v_i)$. Inductively repeating this reasoning, we get that the play π repeats the cycle $v_i v_{i+1} \dots v_j$ infinitely often, i.e.

$$\pi = v_0 \dots v_{i-1}(v_i v_{i+1} \dots v_{j-1})^{\omega}.$$

Thus, the minimal priority occurring infinitely often in π is the same as $\min\{\Omega(v_i), \Omega(v_{i+1}), \dots \Omega(v_{j-1})\}$, and thus is even. Therefore Player 0 wins π, which contradicts the fact that ρ was a winning strategy for Player 1. Q.E.D.

The above theorem allows us, if only a judge is known, to reduce the problem of solving a parity game to the problem of solving a reachability game, which we already tackled with the Game algorithm. But to make use of it, we first need to construct a judge for an input parity game.

The most naïve way to build a judge for a *finite* parity game $\mathcal{G}$ is to just remember, for each position v visited during the play, what is the minimal priority seen in the play since the last occurrence of v. If it happens that a position v is repeated and the minimal priority since v last occurred is even, then the judge decides that Player 0 won the play.

It is easy to check that an automaton defined in this way indeed is a judge for any finite parity game $\mathcal{G}$, but such judge can be very big. Since for each of the $|V| = n$ positions we need to store one of $|\Omega(V)| = d$ colours, the size of the judge is in the order of $O(d^n)$. We will present a judge that is much better for small d.

Definition 2.11. A *progress-measuring judge* $\mathcal{M}_P = (M_P, m_0, \delta_P, F_P)$ for a parity game $\mathcal{G} = (V, V_0, V_1, E, \Omega)$ is constructed as follows. If $n_i = |\Omega^{-1}(i)|$ is the number of positions with priority i, then

$$M_P = \{0, 1, \dots, n_0 + 1\} \times \{0\} \times \{0, 1, \dots, n_2 + 1\} \times \{0\} \times \dots$$

and this product ends in $\dots \times \{0, 1, \dots, n_m + 1\}$ if the maximal priority m is even, or in $\dots \times \{0\}$ if it is odd. The initial state is $m_0 = (0, \dots, 0)$, and the transition function $\delta(v, \overline{c})$ with $\overline{c} = (c_0, 0, c_2, 0, \dots, c_m)$ is given by

$$\delta(v,\bar{c}) = \begin{cases} (c_0, 0, c_2, 0, \dots, c_{\Omega(v)} + 1, 0, \dots, 0) & \text{if } \Omega(v) \text{ is even,} \\ (c_0, 0, c_2, 0, \dots, c_{\Omega(v)-1}, 0, 0, \dots, 0) & \text{otherwise.} \end{cases}$$

The set F_P contains all tuples $(c_0, 0, c_2, \dots, c_m)$ in which some counter $c_j = n_j + 1$ reached the maximum possible value.

The intuition behind $\mathcal{M}_P$ is that it counts, for each even priority p, how many positions with priority p were seen without any lower priority in between. If more than n_p such positions are seen, then at least one must have been repeated, which guarantees that $\mathcal{M}_P$ is a judge.

Lemma 2.12. For each finite parity game $\mathcal{G}$ the automaton $\mathcal{M}_P$ constructed above is a judge for $\mathcal{G}$.

Proof. We need to show that $\mathcal{M}_P$ exhibits the two properties characterising a judge:

(1) if $v_0 \dots$ is winning for Player 0, then there is a k such that $m_k \in F$,
(2) if, for some k, $m_k \in F$, then there exist $i < j \leq k$ such that $v_i = v_j$ and $\min\{\Omega(v_{i+1}), \Omega(v_{i+2}), \dots, \Omega(v_j)\}$ is even.

To see (1), assume that $v_0 v_1 \dots$ is a play winning for Player 0. Let k be such an index that $\Omega(v_k)$ is even, appears infinitely often in $\Omega(v_k)\Omega(v_{k+1})\dots$, and no priority higher than $\Omega(v_k)$ appears in this play suffix. Then, starting from v_k, the counter $c_{\Omega(v_k)}$ will never be decremented, but it will be incremented infinitely often. Thus, for a finite game $\mathcal{G}$, it will reach $n_{\Omega(v_k)} + 1$ at some point, i.e. a state in F_P.

To prove (2), let $v_0 v_1 \dots v_k$ be such a prefix of a play that after v_k some counter c_p is set to $n_p + 1$ for an even priority p. Let v_{i_0} be the last position at which this counter was 0, and v_{i_m} the subsequent positions at which it was incremented, up to $i_{n_p} = k$. All positions $v_{i_0}, v_{i_1}, \dots, v_{i_{n_p}}$ have priority p, but since there are only n_p different positions with priority p, we get that, for some $k < l$, $v_{i_k} = v_{i_l}$. Now i_k and i_l are the positions required to witness (2), because indeed the minimum priority between i_k and i_l is p since c_p was not reset in between. Q.E.D.

For a parity game $\mathcal{G}$ with an even number of priorities d, the above presented judge has size $n_0 \cdot n_2 \cdots n_d$, which is at most $(\frac{n}{d/2})^{d/2}$. We get the following corollary.

Corollary 2.13. Parity games can be solved in time $O((\frac{n}{d/2})^{d/2})$.

Notice that the algorithm using a judge has high space demand: Since the product game $\mathcal{G} \times \mathcal{M}_P$ must be explicitly constructed, the space complexity of this algorithm is the same as its time complexity. There is a method to improve the space complexity by storing the maximal counters the judge $\mathcal{M}_P$ uses in each position and lifting such annotations. This method is called *game progress measures* for parity games. We will not define it here, but the equivalence to modal μ-calculus proven in the next chapter will provide another algorithm for solving parity games with polynomial space complexity.

2.2 Fixed-Point Logics

We will define two fixed-point logics, the modal μ-calculus, L_μ, and the first-order least fixed-point logic, LFP, which extend modal logic and first-order logic, respectively, with the operators for least and greatest fixed-points.

The syntax of L_μ is analogous to modal logic, with two additional rules for building least and greatest fixed-point formulas:

$$\mu X.\varphi(X) \text{ and } \nu X.\varphi(X)$$

are L_μ formulas if $\varphi(X)$ is, where X is a variable that can be used in φ the same way as predicates are used, but must *occur positively in* φ, i.e. under an even number of negations (or, if φ is in negation normal form, simply non-negated).

The syntax of LFP is analogous to first-order logic, again with two additional rules for building fixed-points, which are now syntactically more elaborate. Let $\varphi(T, x_1, x_2, \dots x_n)$ be a LFP formula where T stands for an n-ary relation and occurs only positively in φ. Then both

$$[\mathrm{lfp}\, T\bar{x}.\varphi(T,\bar{x})](\bar{a}) \text{ and } [\mathrm{gfp}\, T\bar{x}.\varphi(T,\bar{x})](\bar{a})$$

are LFP formulas, where $\overline{a} = a_1 \dots a_n$.

To define the semantics of L_μ and LFP, observe that each formula $\varphi(X)$ of L_μ or $\varphi(T, \bar{x})$ of LFP defines an operator $[\![\varphi(X)]\!] : \mathcal{P}(V) \to \mathcal{P}(V)$ on states V of a Kripke structure $\mathcal{K}$ and $[\![\varphi(T, \bar{x})]\!] : \mathcal{P}(A^n) \to \mathcal{P}(A^n)$ on tuples from the universe of a structure $\mathfrak{A}$. The operators are defined in the natural way, mapping a set (or relation) to a set or relation of all these elements, which satisfy φ with the former set taken as argument:

$$[\![\varphi(X)]\!](B) = \{v \in \mathcal{K} : \mathcal{K}, v \models \varphi(B)\}, \text{ and}$$

$$[\![\varphi(T, \bar{x})]\!](R) = \{\bar{a} \in \mathfrak{A} : \mathfrak{A} \models \varphi(R, \bar{a})\}.$$

An argument B is a fixed-point of an operator f if $f(X) = X$, and to complete the definition of the semantics, we say that $\mu X.\varphi(X)$ defines the *smallest* set B that is a fixed-point of $[\![\varphi(X)]\!]$, and $\nu X.\varphi(X)$ defines the *largest* such set. Analogously, $[\text{lfp}\, T\bar{x}.\varphi(T, \bar{x})](\bar{x})$ and $[\text{gfp}\, T\bar{x}.\varphi(T, \bar{x})](\bar{x})$ define the smallest and largest relations being a fixed-point of $[\![\varphi(T, \bar{x})]\!]$, respectively. In a few paragraphs, we will give an alternative characterisation of least and greatest fixed-points, which is better tailored towards an algorithmic computation.

To justify this definition, we have to assure that all notions are well-defined, i.e., in particular, we have to show that the operators actually have fixed-points, and that least and greatest fixed-points always exist. In fact, this relies on the monotonicity of the operators used.

Definition 2.14. An operator F is *monotone* if

$$X \subseteq Y \implies F(X) \subseteq F(Y).$$

The operators $[\![\varphi(X)]\!]$ and $[\![\varphi(T, \bar{x})]\!]$ are monotone because we assumed that X (or T) occurs only positively in φ, and, except for negation, all other logical operators are monotone (the fixed-point operators as well, as we will see). Each monotone operator not only has unique least and greatest fixed-points, but these can be calculated iteratively, as stated in the following theorem.

Definition 2.15. Let A be a set, and $F : \mathcal{P}(A^k) \to \mathcal{P}(A^k)$ be a monotone

operator. We define the stages X_α of an inductive fixed-point process:

$$\begin{aligned} X_0 &:= \emptyset \\ X_{\alpha+1} &:= f(X_\alpha) \\ X_\lambda &:= \bigcup_{\alpha<\lambda} X_\alpha \qquad \text{for limit ordinals } \lambda. \end{aligned}$$

Due to the monotonicity of F, the sequence of stages is increasing, i.e. $X_\alpha \subseteq X_\beta$ for $\alpha < \beta$, and hence for some γ, called the *closure ordinal*, we have $X_\gamma = X_{\gamma+1} = F(X_\gamma)$. This fixed-point is called the *inductive fixed-point* and denoted by X_∞.

Analogously, we can define the stages of a similar process:

$$\begin{aligned} X^0 &:= A^k \\ X^{\alpha+1} &:= F(X^\alpha) \\ X^\lambda &:= \bigcap_{\alpha<\lambda} X^\alpha \qquad \text{for limit ordinals } \lambda. \end{aligned}$$

which yields a decreasing sequence of stages X^α that leads to the inductive fixed-point $X^\infty := X^\gamma$ for the smallest γ such that $X^\gamma = X^{\gamma+1}$.

Theorem 2.16 (Knaster, Tarski)**.** Let F be a monotone operator. Then the least fixed-point $\text{lfp}(F)$ and the greatest fixed-point $\text{gfp}(F)$ of F exist, they are unique and correspond to the inductive fixed-points, i.e. $\text{lfp}(F) = X_\infty$, and $\text{gfp}(F) = X^\infty$.

To understand the inductive evaluation let us consider an example. We will evaluate the formula $\mu X.(P \lor \Diamond X)$ on the following Kripke structure:

$$\mathcal{K} = (\{0, \dots, n\}, \{(i, i+1) \mid i < n\}, \{n\}).$$

The structure $\mathcal{K}$ represents a path of length $n+1$ ending in a position marked by the predicate P. The evaluation of this least fixed-point formula starts with $X_0 = \emptyset$ and $X_1 = P = \{n\}$, and in step $i+1$ all nodes having a successor in X_i are added. Therefore, $X_2 = \{n-1, n\}$, $X_3 = \{n-2, n-1, n\}$, and in general $X_k = \{n-k+1, \dots, n\}$. Finally, $X_{n+1} = X_{n+2} = \{0, \dots, n\}$. As you can see, the formula $\mu X.(P \lor \Diamond X)$

describes the set of nodes from which P is reachable. This example shows one motivation for the study of fixed-point logics: It is possible to express transitive closures of various relations in such logics.

2.3 Model Checking Games for Fixed-Point Logics

In this section we will see that parity games are the model checking games for LFP and L_μ.

We will construct a parity game $\mathcal{G}(\mathfrak{A}, \Psi(\bar{a}))$ from a formula $\Psi(\bar{x}) \in$ LFP, a structure $\mathfrak{A}$ and a tuple $\bar{a}$ by extending the FO game with the moves

$$[\text{fp}\, T\bar{x}.\varphi(T,\bar{x})](\bar{a}) \to \varphi(T,\bar{a})$$

and

$$T\bar{b} \to \varphi(T,\bar{b}).$$

We assign priorities $\Omega(\varphi(\bar{a})) \in \mathbb{N}$ to every instantiation of a subformula $\varphi(\bar{x})$. Therefore, we need to make some assumptions on Ψ:

- Ψ is given in negation normal form, i.e. negations occur only in front of atoms.
- Every fixed-point variable T is bound only once in a formula $[\text{fp}\, T\bar{x}.\varphi(T,\bar{x})]$.
- In a formula $[\text{fp}\, T\bar{x}.\varphi(T,\bar{x})]$ there are no other free variables besides $\bar{x}$ in φ.

Then we can assign the priorities using the following schema:

- $\Omega(T\bar{a})$ is even if T is a gfp-variable, and $\Omega(T\bar{a})$ is odd if T is an lfp-variable.
- If T' depends on T (i.e. T occurs freely in $[\text{fp}\, T'\bar{x}.\varphi(T,T',\bar{x})]$), then $\Omega(T\bar{a}) \le \Omega(T'\bar{b})$ for all $\bar{a},\bar{b}$.
- $\Omega(\varphi(\bar{a}))$ is maximal if $\varphi(\bar{a})$ is not of the form $T\bar{a}$.

Remark 2.17. The minimal number of different priorities in the game $\mathcal{G}(\mathfrak{A}, \Psi(\bar{a}))$ corresponds to the alternation depth of Ψ.

Before we provide the proof that parity games are in fact the appropriate model checking games for LFP and L_μ, we introduce the notion of an *unfolding* of a parity game.

Let $\mathcal{G} = (V, V_0, V_1, E, \Omega)$ be a parity game. We assume that the lowest priority $m = \min_{v \in V} \Omega(v)$ is even and that for all positions $v \in V$ with minimal priority $\Omega(v) = m$ we have a unique successor $vE = \{s(v)\}$. This assumption can be easily satisfied by changing the game slightly.

We define the set

$$T = \{v \in V : \Omega(v) = m\}$$

of positions with minimal priority. For any such set T we get a modified game $\mathcal{G}^- = (V, V_0, V_1, E^-, \Omega)$ with $E^- = E \setminus (T \times V)$, i.e., positions in T are rendered terminal positions.

Additionally, we define a sequence of games $\mathcal{G}^\alpha = (V, V_0^\alpha, V_1^\alpha, E^-, \Omega)$ that only differ in the assignment of the terminal positions in T to the players. For this purpose, we use a sequence of disjoint pairs of sets T_0^α and T_1^α such that each pair partitions the set T, and let $V_\sigma^\alpha = (V_\sigma \setminus T) \cup T_{1-\sigma}^\alpha$, i.e., Player σ wins at final positions $v \in T_\sigma^\alpha$. The sequence of partitions is inductively defined depending on the winning regions of the players in the games $\mathcal{G}^\alpha$ as follows:

- $T_0^0 := T$,
- $T_0^{\alpha+1} := \{v \in T : s(v) \in W_0^\alpha\}$ for any ordinal α,
- $T_0^\lambda := \bigcup_{\alpha<\lambda} T_0^\alpha$ if λ is a limit ordinal,
- $T_1^\alpha = T \setminus T_0^\alpha$ for any ordinal α.

We have

- $W_0^0 \supseteq W_0^1 \supseteq W_0^2 \supseteq \ldots \supseteq W_0^\alpha \supseteq W_0^{\alpha+1} \ldots$
- $W_1^0 \subseteq W_1^1 \subseteq W_1^2 \subseteq \ldots \subseteq W_1^\alpha \subseteq W_1^{\alpha+1} \ldots$

So there exists an ordinal $\alpha \le |V|$ such that $W_0^\alpha = W_0^{\alpha+1} = W_0^\infty$ and $W_1^\alpha = W_1^{\alpha+1} = W_1^\infty$.

Lemma 2.18 (Unfolding Lemma)**.**

$$W_0 = W_0^\infty \quad \text{and} \quad W_1 = W_1^\infty.$$

Proof. Let α be such that $W_0^\infty = W_0^\alpha$ and let f^α be a positional winning strategy for Player 0 from W_0^α in $\mathcal{G}$. Define:

$$f \ : \ V_0 \to V \ : \ v \mapsto \begin{cases} f^\alpha(v) & \text{if } v \in V_0 \setminus T, \\ s(v) & \text{if } v \in V_0 \cap T. \end{cases}$$

A play π consistent with f that starts in W_0^∞ never leaves W_0^∞:

- If $\pi(i) \in W_0^\infty \setminus T$, then $\pi(i+1) = f^\alpha(\pi(i)) \in W_0^\alpha = W_0^\infty$ (f_α is a winning strategy in $\mathcal{G}^\alpha$).
- If $\pi(i) \in W_0^\infty \cap T = W_0^\alpha \cap T = W_0^{\alpha+1} \cap T$, then $\pi(i) \in T_0^{\alpha+1}$, i.e. $\pi(i)$ is a terminal position in $\mathcal{G}^\alpha$ from which Player 0 wins, so by the definition of $T_0^{\alpha+1}$ we have $\pi(i+1) = s(v) \in W_0^\alpha = W_0^\infty$.

Thus, we can conclude that Player 0 wins π:

- If π hits T only finitely often, then from some point onwards π is consistent with f^α and stays in W_0^α which results in a winning play for Player 0.
- Otherwise, $\pi(i) \in T$ for infinitely many i. Since we had $\Omega(t) = m \leq \Omega(v)$ for all $v \in V, t \in T$, the lowest priority seen infinitely often is m, which we have assumed to be even, so Player 0 wins π.

For $v \in W_1^\infty$, we define $\rho(v) = \min\{\beta : v \in W_1^\beta\}$ and let g^β be a positional winning strategy for Player 1 on W_1^β in $\mathcal{G}^\beta$. We define a positional strategy g of Player 1 in $\mathcal{G}^\infty$ by:

$$g : V_1 \to V, \quad v \mapsto \begin{cases} g^{\rho(v)}(v) & \text{if } v \in W_1^\infty \setminus T \cap V_1 \\ s(v) & \text{if } v \in T \cap V_1 \\ \text{arbitrary} & \text{otherwise} \end{cases}$$

Let $\pi = \pi(0)\pi(1)\dots$ be a play consistent with g and $\pi(0) \in W_1^\infty$.

Claim 2.19. Let $\pi(i) \in W_1^\infty$. Then

(1) $\pi(i+1) \in W_1^\infty$,
(2) $\rho(\pi(i+1)) \leq \rho(\pi(i))$
(3) $\pi(i) \in T \Rightarrow \rho(\pi(i+1)) < \rho(\pi(i))$.

Proof. Case (1): $\pi(i) \in W_1^\infty \setminus T$, $\rho(\pi(i)) = \beta$ (so $\pi(i) \in W_1^\beta$). We have $\pi(i+1) = g(\pi(i)) = g^\beta(\pi(i))$, so $\pi(i+1) \in W_1^\beta \subseteq W_1^\infty$ and $\rho(\pi(i+1)) \leq \beta = \rho(\pi(i))$.
Case (2): $\pi(i) \in W_1^\infty \cap T$, $\rho(\pi(i)) = \beta$. Then we have $\pi(i) \in W_1^\infty$, $\beta = \gamma + 1$ for some ordinal γ, and $\pi(i+1) = s(\pi(i)) \in W_1^\gamma$, so $\pi(i+1) \in W_1^\infty$ and $\rho(\pi(i+1)) \leq \gamma < \beta = \rho(\pi(i))$. Q.E.D.

As there is no infinite descending chain of ordinals, there exists an ordinal β such that $\rho(\pi(i)) = \rho(\pi(k)) = \beta$ for all $i \geq k$, which means that $\pi(i) \notin T$ for all $i \geq k$. As $\pi(k)\pi(k+1)\dots$ is consistent with g^β and $\pi(k) \in W_1^\beta$, so π is won by Player 1.

Therefore we have shown that Player 0 has a winning strategy from all vertices in W_0^∞ and Player 1 has a winning strategy from all vertices in W_1^∞. As $V = W_0^\infty \cup W_1^\infty$, this shows that $W_0 = W_0^\infty$ and $W_1 = W_1^\infty$. Q.E.D.

We can now give the proof that parity games are indeed appropriate model checking games for LFP and L_μ.

Theorem 2.20. If $\mathfrak{A} \models \Psi(\bar{a})$, then Player 0 has a winning strategy in the game $\mathcal{G}(\mathfrak{A}, \Psi(\bar{a}))$ starting at position $\Psi(\bar{a})$.

Proof. By structural induction over $\Psi(\bar{a})$. We will only consider the interesting cases $\Psi(\bar{a}) = [\text{gfp}\, T\bar{x}.\varphi(T,\bar{x})](\bar{a})$ and $\Psi(\bar{a}) = [\text{lfp}\, T\bar{x}.\varphi(T,\bar{x})](\bar{a})$.

Let $\Psi(\bar{a}) = [\text{gfp}\, T\bar{x}.\varphi(T,\bar{x})](\bar{a})$. In the game $\mathcal{G}(\mathfrak{A}, \Psi(\bar{a}))$, the positions $T\bar{b}$ have priority 0. Every such position has a unique successor $\varphi(T,\bar{b})$, so the unfoldings $\mathcal{G}^\alpha(\mathfrak{A}, \Psi(\bar{a}))$ are well defined.

Let us take the chain of steps of the gfp-induction of $\varphi(\bar{x})$ on $\mathfrak{A}$.

$$X^0 \supseteq X^1 \supseteq \dots \supseteq X^\alpha \supseteq X^{\alpha+1} \supseteq \dots$$

We have

$$\begin{aligned} \mathfrak{A} \models \Psi(\bar{a}) \quad &\Leftrightarrow \quad \bar{a} \in \text{gfp}(\varphi^{\mathfrak{A}}) \\ &\Leftrightarrow \quad \bar{a} \in X^\alpha \text{ for all ordinals } \alpha \\ &\Leftrightarrow \quad \bar{a} \in X^{\alpha+1} \text{ for all ordinals } \alpha \\ &\Leftrightarrow \quad (\mathfrak{A}, X^\alpha) \models \varphi(\bar{a}) \text{ for all ordinals } \alpha. \end{aligned}$$

Induction hypothesis: For every $X \subset A^k$

$$(\mathfrak{A}, X) \models \varphi(\bar{b}) \text{ iff Player 0 has a winning strategy in } \mathcal{G}((\mathfrak{A}, x^\alpha), \varphi(\bar{a})) \text{ from } \varphi(\bar{a}).$$

We show: If Player 0 has a winning strategy in $\mathcal{G}((\mathfrak{A}, x^\alpha), \varphi(\bar{a}))$ starting at position $\varphi(\bar{a})$, then Player 0 has a winning strategy in $\mathcal{G}^\alpha(\mathfrak{A}, \Psi(\bar{a}))$ starting at position $\varphi(\bar{a})$.

By the unfolding lemma, the second statement is true for all ordinals α if and only if Player 0 has a winning strategy in $\mathcal{G}(\mathfrak{A}, \Psi(\bar{a})$ starting at $\varphi(\bar{a})$.

As $\varphi(\bar{a})$ is the only successor of $\Psi(\bar{a}) = [\text{gfp}\, T\bar{x}.\varphi(T, \bar{x})](\bar{a})$, this holds exactly if Player 0 has a winning strategy in $\mathcal{G}(\mathfrak{A}, \Psi(\bar{a}))$ starting at $\Psi(\bar{a})$.

It remains to show that Player 0 has indeed a winning strategy in the game $\mathcal{G}((\mathfrak{A}, x^\alpha), \varphi(\bar{a}))$ starting at the position $\varphi(\bar{a})$.

There are few differences between $\mathcal{G}((\mathfrak{A}, x^\alpha), \varphi(\bar{a}))$ and the unfolding $\mathcal{G}^\alpha(\mathfrak{A}, \Psi(\bar{a}))$:

- In $\mathcal{G}^\alpha(\mathfrak{A}, \Psi(\bar{a}))$, there is an additional position $\Psi(\bar{a})$, but this position is not reachable.
- The assignment of the atomic propositions $T\bar{b}$:
  - Player 0 wins at position $T\bar{b}$ in $\mathcal{G}((\mathfrak{A}, x^\alpha), \varphi(\bar{a}))$ if and only if $\bar{b} \in X^\alpha$.
  - Player 0 wins at position $T\bar{b}$ in $\mathcal{G}^\alpha(\mathfrak{A}, \Psi(\bar{a}))$ if and only if $T\bar{b} \in T_0^\alpha$.

So we need to show using an induction over α that

$$\bar{b} \in X^\alpha \text{ iff } T\bar{b} \in T_0^\alpha.$$

Base case $\alpha = 0$: $X^0 = A^k$ and $T_\alpha^0 = T = \{T\bar{b} : \bar{b} \in A^k\}$.
Induction step $\alpha = \gamma + 1$: Then $\bar{b} \in X^\alpha = X^{\alpha+1}$ if and only if $(\mathfrak{A}, X^\gamma) \models \varphi(\bar{b})$, which in turn holds if Player 0 wins $\mathcal{G}((\mathfrak{A}, X^\gamma), \varphi(\bar{b}))$ starting at $\varphi(\bar{b})$. By induction hypothesis, this holds if and only if Player 0 wins the unfolding $\mathcal{G}^\gamma(\mathfrak{A}, \Psi(\bar{a}))$ starting at $\varphi(\bar{b}) = s(T\bar{b})$ if and only if $T\bar{b} \in T_0^{\gamma+1} = T_0^\alpha$.

Induction step with α being a limit ordinal: We have that $\bar{b} \in X^\alpha$ if $\bar{b} \in X^\gamma$ for all ordinals $\gamma < \alpha$, which holds, by induction hypothesis, if and only if $T\bar{b} \in T_0^\gamma$ for all $\gamma < \alpha$, which is equivalent to $T\bar{b} \in T_0^\alpha$.

The proof for $\Psi(\bar{a}) = [\text{gfp}\, T\bar{x}.\varphi(T, \bar{x})](\bar{a})$ is analogous. Q.E.D.

2.3.1 *Defining Winning Regions in Parity Games*

To conclude, we consider the converse question—whether winning regions in a parity game can be defined in fixed-point logic—and show that, given an appropriate representation of parity games as structures, winning regions are definable in the μ-calculus.

To represent a parity game $\mathcal{G} = (V, V_0, V_1, E, \Omega)$ with priorities $\Omega(V) = \{0, 1, \ldots, d-1\}$, we use the Kripke structure $\mathcal{K}_\mathcal{G} = (V, V_0, V_1, E, P_0, \ldots, P_{d-1})$. The universe and edge relation of this Kripke structure are the same as in the parity game, and so are the predicates V_0 and V_1 assigning positions to players. The only difference is in the predicates P_j, which are used to explicitly represent positions with priority j, i.e. we define $P_j = \{v \in V : \Omega(v) = j\}$.

Given the above representation, the μ-calculus formula

$$\varphi_d^{\text{Win}} = \nu X_0.\mu X_1.\nu X_2.\ldots\lambda X_{d-1} \bigvee_{j=0}^{d-1} \big((V_0 \wedge P_j \wedge \Diamond X_j) \vee (V_1 \wedge P_j \wedge \Box X_j)\big),$$

where $\lambda = \nu$ if d is odd, and $\lambda = \mu$ otherwise, defines the winning region of Player 0 in the sense of the following theorem.

Theorem 2.21. $\mathcal{K}_\mathcal{G}, v \models \varphi_d^{\text{Win}}$ if and only if Player 0 has a winning strategy from v_0 in $\mathcal{G}$.

Proof (Idea). The model checking game for φ_d^{Win} on $\mathcal{K}_\mathcal{G}$ is essentially the same as the game $\mathcal{G}$ itself, up to some negligible modifications:

- eliminate moves after which the opponent wins in at most two steps (e.g. Verifier would never move to a position $(V_0 \wedge P_j \wedge \Diamond X_j, v)$ if v was not a vertex of Player 0 or did not have priority j),
- contract sequences of trivial moves and remove the intermediate positions.

A schematic view of a model checking game for φ_d^{Win} is sketched in Figure 2.2. Q.E.D.

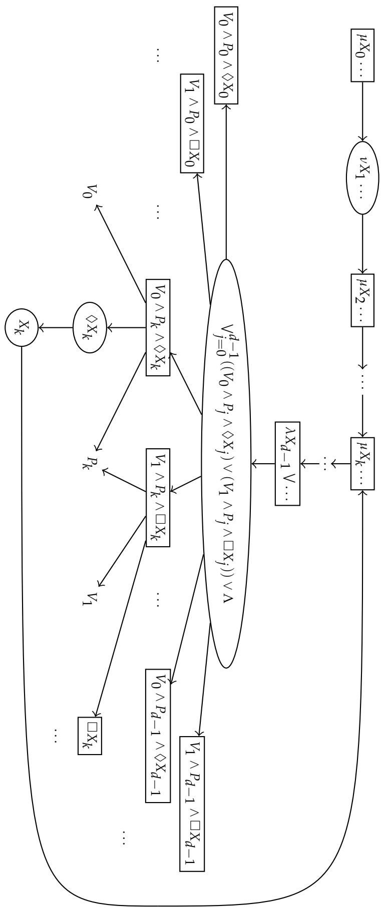


Figure 2.2. Part of a model checking game for φ_d^{Win}.

3 Infinite Games

In this chapter we want to discuss a special kind of *two-player zero-sum games of perfect information*. These games are played by two players, and one player's gain is compensated by the other player's loss, hence the name zero-sum games. Chess and Go are examples of zero-sum games: a win for one player is a loss for the other.

We will start with formal definitions of the basic notions that are used throughout this chapter.

A *game* is a pair $\mathcal{G} = (G, \mathrm{Win})$ where $G = (V, V_0, V_1, E, \Omega)$ is a directed graph with $V = V_0 \cup V_1$ and $\Omega : V \to C$ for a finite set C of *priorities* and $\mathrm{Win} \subseteq C^\omega$. We call G the *arena* of $\mathcal{G}$ and Win the *winning condition* of $\mathcal{G}$.

We will often use the identity function for Ω if we want to define winning conditions depending on the visited vertices of a play. Note that this violates the assumption that the set of priorities is finite if $\mathcal{G}$ itself is infinite.

A *play* of $\mathcal{G}$ is a finite or infinite sequence $\pi = v_0 v_1 v_2 \ldots \in V^{\leq\omega}$ such that $(v_i, v_{i+1}) \in E$ for all i. A finite play is lost by the player who cannot move any more. An infinite play π is won by Player 0 if $\Omega(\pi) = \Omega(v_0)\Omega(v_1) \ldots \in \mathrm{Win}$, otherwise Player 1 wins (there are no draws).

A *strategy* for Player σ is a function $f : V^* V_\sigma \to V$ such that $(v, f(xv)) \in E$ for all $x \in V^*$ and $v \in V_\sigma$. Thus, a strategy maps prefixes of plays which end in a position in V_σ to legal moves of Player σ.

A play $\pi = v_0 v_1 \ldots$ is *consistent with a strategy* f for Player σ if for all proper prefixes $v_0 \ldots v_n$ of π such that $v_n \in V_\sigma$ we have $v_{n+1} = f(v_0 \ldots v_n)$. We say that f is a *winning strategy* from position v_0 if every play starting in v_0 that is consistent with f is won by Player σ.

The set

$$W_\sigma = \{v \in V : \text{Player } \sigma \text{ has a winning strategy from } v\}$$

is the *winning region* of Player σ. In zero-sum games it always holds that $W_0 \cap W_1 = \emptyset$.

We call a game $\mathcal{G}$ *determined* if $W_0 \cup W_1 = V$, i.e. if from each position one player has a winning strategy.

As shown in the first chapter, games where Win is a reachability condition are determined. Recall that Win is a reachability condition if there exists a subset $D \subseteq C$ such that each play that reaches D is won by Player 0, i.e. $\pi \in \text{Win}$ iff $\pi[i] \in D$ for some i.

In the previous chapter, we learnt that parity games are determined as well. But what are the properties of Win that guarantee determinacy? Are there non-determined games at all? To answer these questions, we need topological arguments.

3.1 Topology

Definition 3.1. A *topology* on a set S is defined by a collection of *open* subsets of S. It is required that

- $\emptyset$, and S are open;
- if X and Y are open, then $X \cap Y$ is open;
- if $\{X_i : i \in I\}$ is a family of open sets, then $\bigcup_{i \in I} X_i$ is open.

If $\mathcal{O} \subseteq \mathcal{P}(S)$ is a collection of open sets, we call the pair $(S, \mathcal{O})$ a *topological space*.

Often, a topology is defined by its *base*: A set B of open subsets of S such that every open set can be represented as a union of sets in B.

Example 3.2. The standard topology on $\mathbb{R}$ is defined by the base consisting of all open intervals $(a, b) \subseteq \mathbb{R}$.

In our setting, we will only be concerned with the following topology on B^ω, which is due to Cantor. Its base consists of all sets of the form $z\!\uparrow := z \cdot B^\omega$ for $z \in B^*$. Consequently, a set $X \subseteq B^\omega$ is *open* if it is the union of sets $z\!\uparrow$, i.e. if there exists a set $W \subseteq B^*$ such that

$X = W \cdot B^\omega$. Moreover, a set $X \subseteq B^\omega$ is *closed* if its complement $B^\omega \setminus X$ is open. For $B = \{0, 1\}$, this topology is called the *Cantor space*, and for $B = \omega$ it is called the *Baire space*.

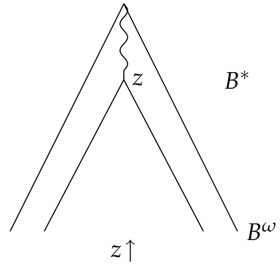


Figure 3.1. Base sets in the Cantor space

Example 3.3.

- The base sets $z\!\uparrow$ are both open and closed (*clopen*) since we have $B^\omega \setminus z\!\uparrow = W_z \cdot B^\omega$ where $W_z = \{y \in B^* \mid y \not\leq z \text{ and } z \not\leq y\}$. (Here, $u \leq v$ means that u is a prefix of v.)
- $0^*1\{0,1\}^\omega$ is open. The complement $\{0^\omega\}$ is closed, but not open.
- $L_d = \{x \in \omega^\omega : x \text{ contains } d \text{ infinitely often}\} = \bigcap_{n \in \omega} (\omega^* \cdot d)^n \cdot \omega^\omega$ is a countable union of open sets.

Next, we will give another useful characterisation of closed sets. A *tree* $T \subseteq B^*$ is a prefix-closed set of finite words, i.e., $z \in T$ and $y \leq z$ implies $y \in T$. For a tree T let $[T]$ be the set of infinite paths through T (note: $T \subseteq B^*$, but $[T] \subseteq B^\omega$).

Example 3.4. Let $T = 0^* = \{0^n : n \in \omega\}$. Then $[T] = \{0^\omega\}$.

Lemma 3.5. $X \subseteq B^\omega$ is closed if and only if there exists a tree $T \subseteq B^*$ such that $X = [T]$.

Proof.
($\Rightarrow$) Let X be closed. Then there is a $W \subseteq B^*$ such that $B^\omega \setminus X = W \cdot B^\omega$. Let $T := \{w \in B^* \mid \forall z (z \leq w \Rightarrow z \notin W)\}$. T is closed under prefixes and $[T] = X$.

($\Leftarrow$) Let $X = [T]$. For every $x \notin [T]$ there exists a smallest prefix $w_x \leq x$ such that $w_x \notin T$. Let $W := \{w_x : x \notin X\}$. Then $B^\omega \setminus X = W \cdot B^\omega$ is open, thus X is closed. Q.E.D.

We call a set $W \subseteq B^*$ *prefix-free* if there is no pair $x, y \in W$ such that $x < y$.

Lemma 3.6.

(1) For every open set $A \subseteq B^\omega$ there is a prefix-free set $W \subseteq B^*$ such that $A = W \cdot B^\omega$.
(2) Let B be a finite alphabet. $A \subseteq B^\omega$ is clopen if and only if there is a finite set $W \subseteq B^*$ such that $A = W \cdot B^\omega$.

Proof.

(1) Let $A = U \cdot B^\omega$ for some open $U \subseteq B^*$. Define

$$W := \{w \in U : U \text{ contains no proper prefix of } w\}.$$

W is prefix-free and $W \cdot B^\omega = U \cdot B^\omega = A$.

(2) ($\Rightarrow$) Let $A \subseteq B^\omega$ be clopen. Thus there exist prefix-free sets $U, V \subseteq B^*$ such that $A = U \cdot B^\omega$ and $B^\omega \setminus A = V \cdot B^\omega$. We will show that $U \cup V$ is finite. Let $T = \{w \in B^* \mid w \text{ has no prefix in } U \cup V\}$. If T is finite, then $U \cup V$ is also finite. If U (or V) is infinite, then T is also infinite since it contains all prefixes of elements of U (respectively V). T is a finitely branching tree (since B is finite) that contains no infinite path, since otherwise there exists an infinite word $x \in B^\omega$ corresponding to this path with $x \notin U \cdot B^\omega \cup V \cdot B^\omega = A \cup (B^\omega \setminus A) = B^\omega$. By König's Lemma, this implies that T is finite.
($\Leftarrow$) Let $A = W \cdot B^\omega$ where $W \subseteq B^*$ is finite. Let $l = \max\{|w| : w \in W\}$. Then $B^\omega \setminus A = Z \cdot B^\omega$ where

$$Z = \{z \in B^* : |z| = l \text{ and no prefix of } z \text{ is in } W\}.$$

Thus, A is clopen. Q.E.D.

Remark 3.7. Lemma 3.6 (2) does not hold for infinite alphabets B.

Since we are investigating games on graphs, the topological space that interests us is the space of all sequences in V^ω (or C^ω) that are plays of a game $\mathcal{G}$. As not all such sequences correspond to feasible plays in $\mathcal{G}$, it is not directly clear that the topological notions we defined for V^ω can be used for the space of all plays of $\mathcal{G}$. But this is indeed the case, as stated by the following lemma (which immediately follows from Lemma 3.5 by considering the unravelling of $\mathcal{G}$).

Lemma 3.8. Let $\mathcal{G}$ be a game with positions V. The set of all plays of $\mathcal{G}$ is a closed subset of V^ω.

Definition 3.9. Let $T = (S, \mathcal{O})$ be a topological space. The class of *Borel sets* is the smallest class $\mathcal{B} \subseteq \mathcal{P}(S)$ that contains all open sets and is closed under countable unions and complementation:

- $\mathcal{O} \subseteq \mathcal{B}$;
- If $X \in \mathcal{B}$ then $S \setminus X \in \mathcal{B}$;
- If $\{X_n : n \in \omega\} \subseteq \mathcal{B}$ then $\bigcup_{n\in\omega} X_n \in \mathcal{B}$.

Most of the ω-languages $L \subseteq B^\omega$ occurring in Computer Science are Borel sets. Borel sets form a natural hierarchy of sets Σ^0_α and Π^0_α for $0 \leq \alpha < \omega_1$, where ω_1 is the first uncountable ordinal number.

- $\Sigma^0_1 = \mathcal{O}$;
- $\Pi^0_\alpha = \mathrm{co}\Sigma^0_\alpha := \{S \setminus X : X \in \Sigma^0_\alpha\}$ for every α;
- $\Sigma^0_\alpha = \{\bigcup_{n\in\omega} X_n : X_n \in \Pi^0_\beta \text{ for } \beta < \alpha\}$ for $\alpha > 0$.

We are especially interested in the first levels of the Borel hierarchy:

- Σ^0_1: Open sets
- Π^0_1: Closed sets
- Σ^0_2: Countable unions of closed sets
- Π^0_2: Countable intersections of open sets
- Σ^0_3: Countable unions of Π^0_2-sets
- Π^0_3: Countable intersections of Σ^0_2-sets

Example 3.10. Let $d \in B$.

$$L_d = \{x \in B^\omega : x \text{ contains } d \text{ infinitely often}\} = \bigcap_{n\in\omega} \underbrace{(B^* \cdot d)^n \cdot B^\omega}_{\in \Sigma^0_1}.$$

Hence, $L_d \in \Pi^0_2$.

To determine the membership of an ω-language in a class Σ^0_α or Π^0_α of the Borel hierarchy and to relate the classes, we need a notion of reducibility between ω-languages.

Definition 3.11. A function $f : B^\omega \to C^\omega$ is called *continuous* if $f^{-1}(Y)$ is open for every open set $Y \subseteq C^\omega$.

Let $X \subseteq B^\omega$, $Y \subseteq C^\omega$. We say that X is *Wadge reducible* to Y, $X \leq Y$, if there exists a continuous function $f : B^\omega \to C^\omega$ such that $f^{-1}(Y) = X$, i.e. $x \in X$ iff $f(x) \in Y$ for all $x \in B^\omega$. For any such function f, we write $f : X \leq Y$.

Exercise 3.1. Prove that the relation $\leq$ satisfies the following properties:

- $X \leq Y$ and $Y \leq Z$ imply $X \leq Z$;
- $X \leq Y$ implies $B^\omega \setminus X \leq C^\omega \setminus Y$.

Theorem 3.12. Let $X \leq Y$ for $Y \in \Sigma^0_\alpha$ (or $Y \in \Pi^0_\alpha$). Then $X \in \Sigma^0_\alpha$ (respectively $X \in \Pi^0_\alpha$).

Proof. The claim is true by definition for Σ^0_1 (the open sets) and thus also for Π^0_1.

Let $\alpha > 1$, $f : X \leq Y$ and $Y \in \Sigma^0_\alpha$. $Y = \bigcup_{n\in\omega} Y_n$ where $Y_n \in \bigcup_{\beta<\alpha} \Pi^0_\beta$. Define $X_n := f^{-1}(Y_n)$. Then $X_n \leq Y_n$ for all $n \in \omega$, and thus by induction hypothesis $X_n \in \bigcup_{\beta<\alpha} \Pi^0_\beta$. We have:

$$\begin{aligned} x \in X &\Leftrightarrow f(x) \in Y \\ &\Leftrightarrow f(x) \in Y_n \text{ for some } n \in \omega \\ &\Leftrightarrow x \in X_n \text{ for some } n \in \omega. \end{aligned}$$

Hence, $X = \bigcup_{n\in\omega} X_n \in \Sigma^0_\alpha$. Q.E.D.

In the following we will present a game-theoretic characterisation of the relation $\leq$ in terms of the so-called *Wadge game*.

Definition 3.13. Let $X \subseteq B^\omega$, $Y \subseteq C^\omega$. The *Wadge game* $W(X,Y)$ is an infinite game between two players 0 and 1 who move in alternation. In the i-th round, Player 0 chooses a symbol $x_i \in B$, and afterwards Player 1 chooses a (possibly empty) word $y_i \in C^*$. After ω rounds, Player 0 has produced an ω-word $x = x_0x_1x_2 \cdots \in B^\omega$, and Player 1 has produced a finite or infinite word $y = y_0y_1y_2 \cdots \in C^{\leq\omega}$. Player 1 wins the play (x,y) if and only if $y \in C^\omega$ and $x \in X \Leftrightarrow y \in Y$.

Example 3.14. Let $B = C = \{0,1\}$.

- Player 1 wins $W(0^*1\{0,1\}^\omega, (0^*1)^\omega)$.
  Winning strategy for Player 1: Choose 0 until Player 0 chooses 1 for the first time. Afterwards, always choose 1.
- Player 0 wins $W((0^*1)^\omega, 0^*1\{0,1\}^\omega)$.
  Winning strategy for Player 0: Choose 1 until Player 1 chooses a word containing 1 for the first time. Afterwards, always choose 0.

Theorem 3.15 (Wadge)**.** Let $X \subseteq B^\omega$, $Y \subseteq C^\omega$. Then $X \leq Y$ if and only if Player 1 has a winning strategy for $W(X,Y)$.

Proof.
($\Leftarrow$) A winning strategy of Player 1 for $W(X,Y)$ induces a mapping $f : B^\omega \to C^\omega$ such that $x \in X$ iff $y \in Y$. It remains to show that f is continuous. Let $Z = U \cdot C^\omega$ be open. For every $u \in U$ denote by V_u the set of all words $v = x_0x_1 \ldots x_n \in B^*$ such that u is the answer of Player 1 to v, i.e. $u = f(x_0)f(x_1)\ldots f(x_n)$. Then $f^{-1}(U \cdot C^\omega) = V \cdot B^\omega$ where $V := \bigcup_{u\in U} V_u$.

($\Rightarrow$) Let $f : X \leq Y$. We construct a strategy for Player 1 as follows. Player 1 has to answer Player 0's moves $x_0x_1x_2\ldots$ by an ω-word $y_0y_1y_2\ldots$, but Player 1 can delay choosing y_i until he knows $x_0x_1\ldots x_n$ for some appropriate $n \geq i$.

Choice of y_0: Consider the partition $B^\omega = \dot\bigcup_{c\in C} f^{-1}(c \cdot C^\omega)$. Since $c \cdot C^\omega$ is clopen, $f^{-1}(c \cdot C^\omega)$ is also clopen. For every $x \in B^\omega$ there exists $c \in C$ such that $x \in f^{-1}(c \cdot C^\omega)$, and since $f^{-1}(c \cdot C^\omega)$ is clopen, there is a prefix $w_x \leq x$ such that $w_x \cdot B^\omega \subseteq f^{-1}(c \cdot C^\omega)$. So Player 1 can wait until Player 0 has chosen a prefix $w \in B^*$ that determines the set $f^{-1}(c \cdot C^\omega)$ the word x will belong to and choose $y_0 = c$.

The subsequent choices are done analogously. Let $y_0 \ldots y_i \in C^*$ be Player 1's answer to $x_0 \ldots x_n \in B^*$. For the choice of y_{i+1} we consider the partition

$$x_0 \cdots x_n \cdot B^\omega = \dot\bigcup_{c\in C} f^{-1}(y_0 \cdots y_i \cdot c \cdot C^\omega).$$

Since the sets $f^{-1}(y_0 \cdots y_i \cdot c \cdot C^\omega)$ are clopen, after finitely many moves, by choosing a prolongation $x_0 \cdots x_n x_{n+1} \cdots x_k$, Player 0 has determined in which set $f^{-1}(y_0 \cdots y_i \cdot c \cdot C^\omega)$ the word x will be. Player 1 then chooses $y_{i+1} = c$.

By using this strategy, Player 1 constructs the answer $y = f(x)$ for the sequence x chosen by Player 0. Otherwise, there would be a smallest i such that $y_i \neq f(x_i)$. This is impossible since $x \in f^{-1}(y_0 \cdots y_i \cdot C^\omega)$. Since $f : X \leq Y$, we have $x \in X$ iff $y \in Y$. Q.E.D.

Definition 3.16. A set $Y \subseteq C^\omega$ is Σ^0_α-*complete* if:

- $Y \in \Sigma^0_\alpha$;
- $X \leq Y$ for all $X \in \Sigma^0_\alpha$.

Π^0_α-completeness is defined analogously.

Note that Y is Σ^0_α-complete if and only if $C^\omega \setminus Y$ is Π^0_α-complete.

Proposition 3.17. Let $B = \{0,1\}$. Then:

- $0^*1\{0,1\}^\omega$ is Σ^0_1-complete;
- $\{0^\omega\}$ is Π^0_1-complete;
- $\{0,1\}^*0^\omega$ is Σ^0_2-complete;
- $(0^*1)^\omega$ is Π^0_2-complete.

Proof. By the above remark, it suffices to show that $0^*1\{0,1\}^\omega$ and $(0^*1)^\omega$ are Σ^0_1-complete and Π^0_2-complete, respectively.

- We know that $0^*1\{0,1\}^\omega \in \Sigma^0_1$. Let $X = W \cdot B^\omega$ be open. We describe a winning strategy for Player 1 in $W(X, 0^*1\{0,1\}^\omega)$: Pick 0 until Player 0 has completed a word contained in W; from this point onwards, pick 1. Hence, $X \leq 0^*1\{0,1\}^\omega$.
- We know that $(0^*1)^\omega \in \Pi^0_2$. Let $X = \bigcap_{n \in \omega} W_n \cdot B^\omega \in \Pi^0_2$. We describe a winning strategy for Player 1 in $W(X, \{0,1\}^*0^\omega)$: Start with $i := 0$; for arbitrary i, answer with 1 and set $i := i+1$ if the sequence $x_0 \ldots x_k$ of symbols chosen by Player 0 so far has a prefix in W_i, otherwise answer with 0 and leave i unaffected. Q.E.D.

3.2 Gale-Stewart Games

In this chapter we will show that, using the Axiom of Choice, one can construct a non-determined game. Later, we will mention which topological properties guarantee determinacy and how this is related to logic. Before we proceed to discuss the games, we shortly introduce the basic notions of ordinals, as these will be used in the proofs extensively.

3.2.1 Ordinals

The standard basic notion used in mathematics is the notion of a set, and all mathematical theorems follow from *the axioms of set theory*. The standard set of axioms, which (among others) guarantee the existence of an empty set, an infinite set, and the powerset of any set, and that no set is a member of itself (i.e. $\forall x \, \neg x \in x$) is called the *Zermelo-Fränkel Set Theory ZF*. It is standard in mathematics to use ZF extended by *the axiom of choice AC*, which together are called ZFC.

Since everything is a set in mathematics, there is a need to represent numbers as sets. The standard way to do this is to start with the empty set, let $0 = \emptyset$, and proceed by induction, defining $n + 1 = n \cup \{n\}$. Here are the first few numbers in this coding:

- $0 = \emptyset$,
- $1 = \{\emptyset\}$,
- $2 = \{\emptyset, \{\emptyset\}\}$,
- $3 = \{\emptyset, \{\emptyset\}, \{\emptyset, \{\emptyset\}\}\}$.

Observe that for each number n (as a set) it holds that

$$m \in n \implies m \subseteq n.$$

In particular, the relation $\in$ is *transitive* in such sets, i.e. if $k \in m$ and $m \in n$ then $k \in n$. We use this property of sets to define a more general class of numbers.

Definition 3.18. A set α is *an ordinal number* if $\in$ is transitive in α.

Except for natural numbers, what other ordinal numbers are there? The first example is $\omega = \bigcup_n n$, the union of all natural numbers. Indeed, it is easy to check that the union of ordinals is always an ordinal as well (as long as it is a set at all).

What is the next ordinal number after ω? We can again apply the $+1$ operation in the same way as for natural numbers, so

$$\omega + 1 = \omega \cup \{\omega\} = \{0, 1, 2, \ldots, \{0, 1, \ldots\}\}.$$

But does it make sense to say that $\omega + 1$ is the *next* ordinal, is there an order on ordinals? In fact both, each ordinal as a set and all ordinals as a class, are well-ordered, i.e. the following holds:

- for any two ordinal numbers α and β either $\alpha \subseteq \beta$ or $\beta \subseteq \alpha$;
- there exists no infinite sequence of ordinals

  $$\alpha_0 \supsetneq \alpha_1 \supsetneq \alpha_3 \supsetneq \cdots;$$

- each ordinal α is well-ordered by $\in$.

The well-ordering of ordinals follows from the mentioned axiom that no set is a member of itself, $\forall x\ \neg x \in x$.

Ordinals are intimately connected to well-orders, in fact any structure $(A, <)$ where $<$ is a well-ordering is isomorphic to some ordinal α. To get an intuition on how ordinals look like, consider the following examples of countable ordinals: $\omega + 1, \omega + \omega$, $\omega^2, \omega^3, \omega^\omega$.

The well-ordering of ordinals allows to define and prove the principle of *transfinite induction*. This principle states that the class of *all ordinals* is generated from $\emptyset$ by taking the successor ($+1$) and the union on limit steps, as shown on the examples before. Specifically, for each ordinal α it holds that either

- there exists an ordinal $\beta < \alpha$ such that $\alpha = \beta + 1 = \beta \cup \{\beta\}$, or
- there exist ordinals $\beta_\gamma < \alpha$ such that $\alpha = \bigcup_\gamma \beta_\gamma$.

Besides ordinals, we sometimes need cardinal numbers. A *cardinal number* κ is *the smallest ordinal* α for which a bijection to κ exists.

3.2.2 Non-determined Games

Let B be an alphabet (especially: $B = \{0,1\}$ or $B = \omega$). In a Gale-Stewart game the players alternately choose symbols from B and create an infinite sequence $\pi \in B^\omega$. Gale-Stewart games can be described as graph games in different ways. For $B = \{0,1\}$, for example, as a game on the infinite binary tree

$$\mathcal{T}^2 = (\{0,1\}^*, V_0, V_1, E, \Omega),$$

where

$$V_0 = \bigcup_{n\in\omega} \{0,1\}^{2n},$$
$$V_1 = \bigcup_{n\in\omega} \{0,1\}^{2n+1},$$
$$E = \{(x, xi) : x \in \{0,1\}^*, i \in \{0,1\}\},$$

and $\Omega : \{0,1\}^* \to \{0,1,\varepsilon\} : \varepsilon \mapsto \varepsilon, xi \mapsto i$. Alternatively, it can be described as a game on the graph depicted in Figure 3.2. Similar game graphs can be defined for arbitrary B.

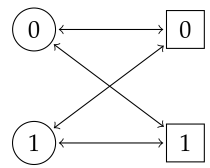


Figure 3.2. Game graph for Gale-Stewart game over $B = \{0,1\}$

Theorem 3.19 (Gale-Stewart)**.** There exists a non-determined game.

We will present two proofs: The first one uses ordinal numbers to enumerate the set of all strategies. The second one uses ultrafilters. Both rely on the Axiom of Choice (AC).

Proof. Let $T_0 = \{x \in B^* : |x| \text{ even}\}$ and $T_1 = \{x \in B^* : |x| \text{ odd}\}$. Then

$$F = \{f : T_0 \to B\} \text{ and } G = \{g : T_1 \to B\}$$

are the sets of strategies for Player 0 and for Player 1. Since B is countable, we have $|F| = |G| = |\mathcal{P}(\omega)| = 2^\omega$. Thus, using the well-ordering principle (which is equivalent to AC) we can enumerate the strategies by ordinals less than 2^ω:

$$F = \{f_\alpha : \alpha < 2^\omega\} \text{ and } G = \{g_\alpha : \alpha < 2^\omega\}.$$

For strategies f and g let $f\hat{}g \in B^\omega$ be the play uniquely defined by f and g. We will construct two increasing sequences of sets $X_\alpha, Y_\alpha \subseteq B^\omega$ for $\alpha < 2^\omega$ such that:

(1) $X_\alpha \cap Y_\alpha = \emptyset$,
(2) $|X_\alpha|, |Y_\alpha| < 2^\omega$,
(3) for all $\beta < \alpha$ there exists $f \in F$ such that $f\hat{}g_\beta \in X_\alpha$,
(4) for all $\beta < \alpha$ there exists $g \in G$ such that $f_\beta\hat{}g \in Y_\alpha$.

The construction proceeds as follows. For $\alpha = 0$ let $X_\alpha := Y_\alpha := \emptyset$. For limit ordinals λ let $X_\lambda := \bigcup_{\beta<\lambda} X_\beta$ and $Y_\lambda := \bigcup_{\beta<\lambda} Y_\beta$. Observe that the properties above are indeed satisfied.

For a successor ordinal $\alpha = \beta + 1$ consider the strategy f_β. The cardinality of X_β and Y_β is smaller than 2^ω by Property (2). But there are 2^ω different plays consistent with f_β, so there is one of them which is not yet in X_β. Choose such a play (AC again) and add it to Y_β to construct Y_α. Analogously, find such a play for g_β (which additionally is not in Y_α) and add it to X_β to construct X_α. Finally, we define $\text{Win} = \bigcup_{\alpha<2^\omega} X_\alpha$.

Assume that $f = f_\alpha$ for some $\alpha < 2^\omega$ is a winning strategy for Player 0. By the construction of Win, there is a strategy $g \in G$ such that $f_\alpha\hat{}g \in Y_\alpha$ and thus $f_\alpha\hat{}g \notin \text{Win}$, a contradiction.

Now assume that $g = g_\alpha$ for some $\alpha < 2^\omega$ is a winning strategy for Player 1. Analogously, there is a strategy $f \in F$ such that $f\hat{}g_\alpha \in X_\alpha \subseteq$ Win, a contradiction as well. Q.E.D.

The second proof we will present uses the concept of an ultrafilter. The intuition behind a filter is that it is a family of large sets.

Definition 3.20. Let I be a non-empty set. A non-empty set $F \subseteq \mathcal{P}(I)$ is a *filter* if

(1) $\emptyset \notin F$,
(2) $x \in F,\ y \in F \ \Rightarrow\ x \cap y \in F$, and
(3) $x \in F,\ y \supseteq x \ \Rightarrow\ y \in F$.

Example 3.21. The set $\{x \subseteq \omega : \omega \setminus x \text{ is finite}\}$ is a filter. We call it the *Fréchet filter.*

Definition 3.22. An *ultrafilter* is a filter that satisfies the additional requirement:

(4) for all $x \subseteq I$ either $x \in F$ or $I \setminus x \in F$.

Example 3.23. Fix $n_0 \in \omega$. Then $U = \{a \subseteq \omega : n_0 \in a\}$ is an ultrafilter.

Note that the Fréchet filter is not an ultrafilter. Observe as well, that any ultrafilter that contains a finite set must contain a singleton set as well, so it is of the form presented in the example above. Does there exist an ultrafilter which contains no finite set, i.e. one that contains the Fréchet filter? Indeed, we can show it does.

Theorem 3.24. The Fréchet filter F can be expanded to an ultrafilter $U \supset F$.

The proof uses AC or Zorn's Lemma or the compactness theorem for propositional logic and holds for every filter $F \subseteq 2^\omega$ such that $a_1 \cap \cdots \cap a_m \neq \emptyset$ for all $m \in \mathbb{N}$, $a_1, \ldots, a_m \in F$.

Proof. Let F be the Fréchet filter. We use propositional variables X_a for every $a \in \mathcal{P}(\omega)$. Let $\Phi = \Phi_U \cup \Phi_F$ where

$$\begin{aligned}\Phi_U = \{&\neg X_\emptyset\}\\ &\cup\ \{X_a \wedge X_b \rightarrow X_{a\cap b} : a, b \subseteq \omega\}\\ &\cup\ \{X_a \rightarrow X_b : a \subseteq b, a, b \subseteq \omega\}\\ &\cup\ \{X_a \leftrightarrow \neg X_{\omega\setminus a} : a \subseteq \omega\}\end{aligned}$$

and

$$\Phi_F = \{X_a : a \in F\}.$$

Every model $\mathcal{I}$ of Φ defines an ultrafilter U which expands F, namely $U = \{a \subseteq \omega : \mathcal{I}(X_a) = 1\}$. It remains to show that Φ is satisfiable.

By the compactness theorem, it suffices to show that every finite subset of Φ is satisfiable. Hence, let Φ_0 be a finite subset of Φ. Then the set $F_0 = \{a \in F : X_a \in \Phi_0 \cap \Phi_F\}$ is also finite. Now consider the following two cases:

- $F_0 = \emptyset$. Define the interpretation $\mathcal{I}$ by

$$\mathcal{I}(X_a) = \begin{cases} 1 & \text{if } 0 \in a, \\ 0 & \text{otherwise.} \end{cases}$$

Then $\mathcal{I} \models \Phi_0$.

- $F_0 = \{a_1, \ldots, a_m\}$. Since F is a filter, there exists $n_0 \in a_1 \cap \cdots \cap a_m$. Define the interpretation $\mathcal{I}$ by

$$\mathcal{I}(X_a) = \begin{cases} 1 & \text{if } n_0 \in a \\ 0 & \text{otherwise} \end{cases}$$

Again, we have $\mathcal{I} \models \Phi_0$.

Hence, Φ_0 is satisfiable. Q.E.D.

We are now able to give an alternative proof of the fact that there exists a non-determined game.

Proof (of Theorem 3.19). Let U be an ultrafilter that expands the Fréchet filter. We construct a non-determined Gale-Stewart game over $B = \omega$ with winning condition Win_U as follows. Player 0 wins a play $x = x_0 x_1 \ldots \in \omega^\omega$ if

- Player 1 has played a number that is not higher than the previously played one, i.e. $\min\{j : x_{j+1} \leq x_j\}$ exists and is odd, or
- $x_0 < x_1 < x_2 < \ldots$ and

$$A(x) := [0, x_0) \cup \bigcup_{i \in \omega} [x_{2i+1}, x_{2i+2}) \in U$$

(see Figure 3.3).

We claim that the Gale-Stewart game with winning condition Win_U is not determined. Towards a contradiction, assume that Player 0 has a

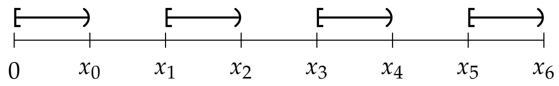


Figure 3.3. The winning condition of the ultrafilter game

winning strategy f. Then we can construct the following play, which is consistent with f:

- To $x_0 = f(\varepsilon)$, Player 1 answers with an arbitrary number $x_1 > x_0$.
- To x_{2i} for $i > 0$, Player 1 chooses the number chosen by f for the play prefix $x_0 x_2 x_3 \ldots x_{2i}$.

Consequently, Player 1 plays with strategy f against strategy f.

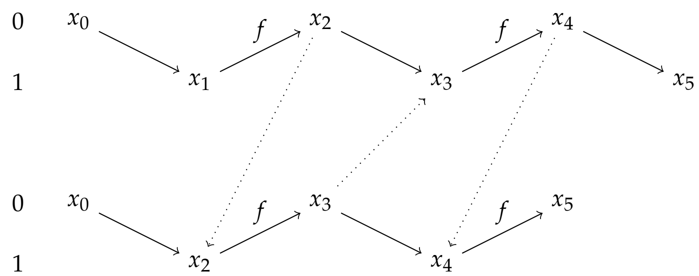


Figure 3.4. Playing the Ultrafilter game

This results in two plays $x = x_0 x_1 x_2 \ldots$ and $x' = x_0 x_2 x_3 x_4 \ldots$, where

$$x_{2i+2} = f(x_0 x_1 \ldots x_{2i+1}),$$

but also

$$x_{2i+1} = f(x_0 x_1 \ldots x_{2i}).$$

Both plays are consistent with the winning strategy f for Player 0. Thus we have $A(x) \in U$ and $A(x') \in U$. But

$$A(x) = [0, x_0) \cup \bigcup_{i \in \omega} [x_{2i+1}, x_{2i+2})$$

and

$$A(x') = [0, x_0) \cup \bigcup_{i \in \omega} [x_{2i+2}, x_{2i+3}).$$

Thus $A(x) \cap A(x') = [0, x_0) \in U$. However, since U expands the Fréchet filter, the co-finite set $\omega \setminus [0, x_0)$ is in U and thus $[0, x_0) \notin U$, a contradiction.

Analogously, one derives a contradiction from the assumption that Player 1 has a winning strategy. Q.E.D.

3.2.3 Determined Games

We call a game $\mathcal{G} = (V, V_0, V_1, E, \text{Win})$ clopen, open, closed, etc., or simply a *Borel game*, if the winning condition $\text{Win} \subseteq V^\omega$ has the respective property.

Clopen games are basically finite games: If $A \subseteq B^\omega$ is clopen, then for every $x \in B^\omega$ there exists a finite prefix $w_x \leq x$ such that:

- If $x \in A$ then $w_x\!\uparrow \subseteq A$;
- If $x \notin A$ then $w_x\!\uparrow \subseteq B^\omega \setminus A$.

Therefore, the game is equivalent to a finite game, in which a play is decided after a prefix w has been seen such that $w\!\uparrow \subseteq A$ or $w\!\uparrow \subseteq B^\omega \setminus A$. To be more precise: Given a game $\mathcal{G}$ and a starting position v_0, consider the tree $\mathcal{T}_G(v_0)$, i.e. the unfolding of $\mathcal{G}$ to the tree of all possible paths starting in v_0. If $A = W \cdot B^\omega$ and $B^\omega \setminus A = W' \cdot B^\omega$, then the tree can be truncated at the positions in $W \cup W'$. The resulting game is equivalent to the original game but allows only finite plays.

Corollary 3.25. Clopen games are determined.

A stronger result is the following:

Theorem 3.26. Every open game, and thus every closed game, is determined.

Proof. Let $\mathcal{G} = (V, V_0, V_1, E, \text{Win})$ where $\text{Win} = U \cdot V^\omega$ is open. First, we consider finite plays: Let $T_\sigma = \{v \in V_{1-\sigma} : vE = \emptyset\}$ and $A_\sigma =$

$\text{Attr}_\sigma(T_\sigma)$. From every position $v \in A_\sigma$ Player σ wins after finitely many moves with the attractor strategy.

For the infinite plays consider

$$\mathcal{G}' := \mathcal{G} \restriction V \setminus (A_0 \cup A_1)$$

with positions $V' := V \setminus (A_0 \cup A_1)$. In $\mathcal{G}'$ every play is infinite, and Player 0 wins $\pi = v_0 v_1 v_2 \dots$ if and only if $\pi \in U \cdot V^\omega$. Obviously, Player 0 wins in $\mathcal{G}'$ starting from v_0 if she can enforce a sequence $v_0 v_1 \dots v_n \in U$. Then every infinite prolongation of this sequence is a play in $U \cdot V^\omega$.

Instead of $\mathcal{G}'$ we consider again the equivalent game on the trees $\mathcal{T}(v) = \mathcal{T}_\mathcal{G}(v)$, the unfolding of $\mathcal{G}$ from $v \in V$. Positions in $\mathcal{T}(v)$ are words over V: $\mathcal{T}(v) \subseteq V^*$. Now consider the set

$$B_0 = \{v \in V' : v \in \text{Attr}_0^{\mathcal{T}(v)}(U \cdot V^*)\}$$

of positions from where player 0 can enforce a play prefix in $U \cdot V^*$. From every position in $V' \setminus A_0$, Player 1 has a strategy to guarantee that the play never reaches $U \cdot V^*$ since $V' \setminus A_0$ is a trap for Player 0. But a play that never reaches $U \cdot V^*$ is won by Player 1. It follows that $W_0 = A_0 \cup B_0$ and $W_1 = A_1 \cup (V' \setminus B_0)$, and thus $V = W_0 \cup W_1$. Q.E.D.

A much more subtle result was proven by Tony Martin in 1975.

Theorem 3.27 (Martin)**.** All Borel games are determined.

Here are some winning conditions for frequently used games in Computer Science:

- *Muller conditions*: Let B be finite, $\mathcal{F}_0 \subseteq \mathcal{P}(B)$, $\mathcal{F}_1 = \mathcal{P}(B) \setminus \mathcal{F}_0$. Player σ wins $\pi \in B^\omega$ if and only if

  $$\text{Inf}(\pi) := \{b \in B : b \text{ appears infinitely often in } \pi\} \in \mathcal{F}_\sigma.$$

  Hence, the winning condition is the set

  $$\{x \in B^\omega : \text{Inf}(\pi) \in \mathcal{F}_\sigma\} = \bigcup_{X \in \mathcal{F}_0} \big(\bigcap_{d \in X} L_d \cap \bigcup_{d \notin X} (B^\omega \setminus L_d)\big),$$

a finite Boolean combination of Π^0_2-sets.

- *Parity conditions* (see previous chapter) are special cases of Muller conditions and thus also finite Boolean combinations of Π^0_2-sets.
- Every ω-regular language is a Boolean combination of Π^0_2-sets. This follows from the recognisability of ω-regular languages by Muller automata and the fact that Muller conditions are Boolean combinations of Π^0_2-sets.

In practice, winning conditions are often specified in a suitable logic: ω-words $x \in B^\omega$ are interpreted as structures $\mathfrak{A}_x = (\omega, <, (P_b)_{b\in B})$ with unary predicates $P_b = \{i \in \omega : x_i = b\}$. A sentence ψ (for example in FO, MSO, etc.) over the signature $\{<\} \cup \{P_b : b \in B\}$ defines the ω-language (winning condition) $L(\psi) = \{x \in B^\omega : \mathfrak{A}_x \models \psi\}$.

Example 3.28. Let $B = \{0, \dots, m\}$. The parity condition is specified by the FO sentence

$$\psi := \bigwedge_{\substack{b\le m\\ b \text{ odd}}} \left(\exists y \forall z\, (y < z \rightarrow \neg P_b z) \vee \bigwedge_{c<b} \forall y \exists z\, (y < z \wedge P_c z) \right).$$

We have:

- FO and LTL define the same ω-languages (winning conditions);
- MSO defines exactly the ω-regular languages;
- There are ω-languages that are definable in MSO but not in FO;
- ω-regular languages are Boolean combinations of Π^0_2-sets.

In particular, graph games with winning conditions specified in LTL, FO, MSO, etc. are Borel games and therefore determined.

3.3 Muller Games and Game Reductions

Muller games are infinite games played over an arena $G = (V_0, V_1, E, \Omega : V \to C)$ with a winning condition depending only on the set of priorities seen infinitely often in a play. It is specified by a partition $\mathcal{P}(C) = \mathcal{F}_0 \uplus \mathcal{F}_1$, and a play $\pi = v_0 v_1 v_2 \dots$ is won by Player σ if

$$\mathrm{Inf}(\pi) = \{c : \Omega(v_i) = c \text{ for infinitely many } i \in \omega\} \in \mathcal{F}_\sigma.$$

We will only consider the case that the set C of priorities is finite. Then Muller games are Borel games specified by the FO sentence

$$\bigvee_{X\in\mathcal{F}_\sigma} \left(\bigwedge_{c\in X} \forall x \exists y (x < y \wedge P_c y) \wedge \bigwedge_{c\notin X} \exists x \forall y (x < y \rightarrow \neg P_c y) \right).$$

So Muller games are determined. Parity conditions are special Muller conditions, and we have seen that games with parity winning conditions are even positionally determined. The question arises what kind of strategies are needed to win Muller games. Unfortunately, there are simple Muller games that are not positionally determined, even solitaire games.

Example 3.29. Consider the game arena depicted in Figure 3.5 with the winning condition $\mathcal{F}_0 = \{\{1, 2, 3\}\}$, i.e. all positions have to be visited infinitely often. Obviously, player 0 has winning a winning strategy, but no positional one: Any positional strategy of player 0 will either visit only positions 1 and 2 or positions 2 and 3.

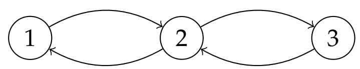


Figure 3.5. A solitaire Muller game

Although Muller games are, in general, not positionally determined, we will see that Muller games are determined via winning strategies that can be implemented using finite memory. To this end, we introduce the notions of a memory structure and of a memory strategy. Although we will not require that the memory is finite, we will use finite memory in most cases.

Definition 3.30. A *memory structure* for a game $\mathcal{G}$ with positions in V is a triple $\mathfrak{M} = (M, \mathrm{update}, \mathrm{init})$, where M is a set of *memory states*, $\mathrm{update} : M \times V \to M$ is a *memory update function* and $\mathrm{init} : V \to M$ is a *memory initialisation function*. The *size* of the memory is the cardinality of the set M.

A *strategy with memory* $\mathfrak{M}$ for Player σ is given by a next-move function $F : V_\sigma \times M \to V$ such that $F(v, m) \in vE$ for all $v \in V_\sigma, m \in$

M. If a play, from starting position v_0, has gone through positions $v_0 v_1 \dots v_n$, the memory state is $m(v_0 \dots v_n)$, defined inductively by $m(v_0) = \text{init}(v_0)$, and $m(v_0 \dots v_i v_{i+1}) = \text{update}(m(v_0 \dots v_i), v_{i+1})$, and in case $v_n \in V_\sigma$ the strategy leads to position $F(v_n, m(v_0 \dots, v_n))$.

Remark 3.31. In case $|M| = 1$, the strategy is positional, and it can be described by a function $F : V_\sigma \to V$.

Definition 3.32. A game $\mathcal{G}$ is determined via memory $\mathfrak{M}$ if it is determined and both players have winning strategies with memory $\mathfrak{M}$ on their winning regions.

Example 3.33. In the game from Example 3.29, Player 0 wins with a strategy with memory $\mathfrak{M} = (\{1,3\}, \text{update}, \text{init})$ where

- $\text{init}(1) = \text{init}(2) = 1$, $\text{init}(3) = 3$ and
- $\text{update}(m, v) = \begin{cases} v & \text{if } v \in \{1,3\}, \\ m & \text{if } v = 2. \end{cases}$

The corresponding strategy is defined by

$$F(v, m) = \begin{cases} 2 & \text{if } v \in \{1,3\}, \\ 3 & \text{if } v = 2, m = 1, \\ 1 & \text{if } v = 2, m = 3. \end{cases}$$

Let us consider a more interesting example now.

Example 3.34. Consider the game DJW_2 with its arena depicted in Figure 3.6. Player 0 wins a play π if the maximal number in $\text{Inf}(\pi)$ is equal to the number of letters in $\text{Inf}(\pi)$. Formally:

$$\mathcal{F}_0 = \{X \subseteq \{1, 2, a, b\} : |X \cap \{a, b\}| = \max(X \cap \{1, 2\})\}.$$

Player 0 has a winning strategy from every position, but no positional one. Assume that $f : \{a, b\} \to \{1, 2\}$ is a positional winning strategy for Player 0. If $f(a) = 2$ (or $f(b) = 2$), then Player 1 always picks a (respectively b) and wins, since this generates a play π with $\text{Inf}(\pi) = \{a, 2\}$ (respectively $\text{Inf}(\pi) = \{b, 2\}$). If $f(a) = f(b) = 1$, then Player 1 alternates between a and b and wins, since this generates a play

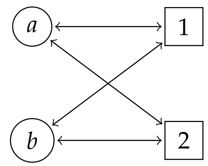


Figure 3.6. Muller game $\mathcal{G} = DJW_2$

π with $\text{Inf}(\pi) = \{a, b, 1\}$. However, Player 0 has a winning strategy that uses the memory depicted in Figure 3.7. The corresponding strategy is defined as follows:

$$F(c, m) = \begin{cases} 1 & \text{if } m = c\#d, \\ 2 & \text{if } m = \#cd. \end{cases}$$

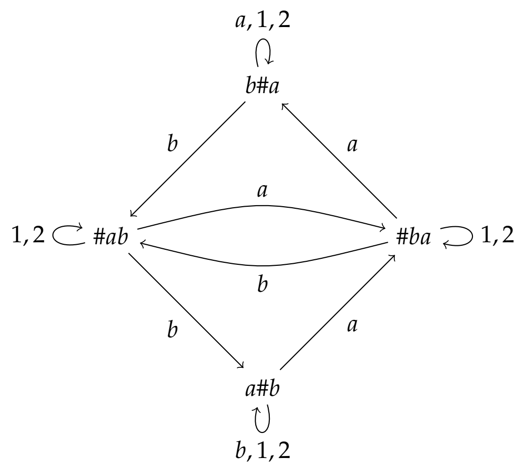


Figure 3.7. Memory for Player 0

Why is this strategy winning? If from some point onwards Player 1 picks only a or only b, then, from this point onwards, the memory state is always $b\#a$ or $a\#b$, respectively, and according to F Player 0 always picks 1 and wins. In the other case, Player 1 picks a and b again and again and the memory state is $\#ab$ or $\#ba$ infinitely often. Thus Player 0 picks 2 infinitely often and wins as well.

The memory structure used in this example is a special case of the LAR memory structure, which we will use for arbitrary Muller games. But first, let us look at a Muller game with infinitely many priorities that allows no winning strategy with finite memory but one with a simple infinite memory structure:

Example 3.35. Consider the game with its arena depicted in Figure 3.8 and with winning condition $\mathcal{F}_0 = \{\{0\}\}$. It is easy to see that every finite-memory strategy of Player 0 (the player who moves at position 0) is losing. A winning strategy with infinite memory is given by the memory structure $\mathfrak{M} = (\omega, \text{init}, \text{update})$ where $\text{init}(v) = v$ and $\text{update}(m, v) = \max(m, v)$ together with the strategy F defined by $F(0, m) = m + 1$.

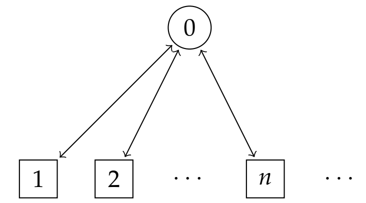


Figure 3.8. A game where finite-memory strategies do not suffice

Given a game graph $G = (V, V_0, V_1, E)$ and a memory structure $\mathfrak{M} = (M, \text{update}, \text{init})$, we obtain a new game graph

$$G \times \mathfrak{M} = (V \times M, V_0 \times M, V_1 \times M, E_{\text{update}})$$

where

$$E_{\text{update}} = \{((v, m), (v', m')) : (v, v') \in E \text{ and } m' = \text{update}(m, v')\}.$$

Obviously, every play $(v_0, m_0)(v_1, m_1) \dots$ in $G \times \mathfrak{M}$ has a unique projection to the play $v_0 v_1 \dots$ in G. Conversely, every play $v_0, v_1, \dots$ in

G has a unique extension to a play $(v_0, m_0)(v_1, m_1) \dots$ in $G \times \mathfrak{M}$ with $m_0 = \text{init}(v_0)$.

Definition 3.36. For games $\mathcal{G} = (G, \Omega, \text{Win})$ and $\mathcal{G}' = (G', \Omega', \text{Win}')$, we say that $\mathcal{G}$ *reduces to* $\mathcal{G}'$ *via memory* $\mathfrak{M}$, $\mathcal{G} \leq_{\mathfrak{M}} \mathcal{G}'$, if $G' = G \times \mathfrak{M}$ and every play in $\mathcal{G}'$ is won by the same player as the projected play in $\mathcal{G}$.

Given a memory structure $\mathfrak{M}$ for G and a memory structure $\mathfrak{M}'$ for $G \times \mathfrak{M}$, we obtain a memory structure $\mathfrak{M}^* = \mathfrak{M} \times \mathfrak{M}'$ for G. The set of memory locations is $M \times M'$, and we have memory initialisation

$$\text{init}^*(v) = (\text{init}(v), \text{init}'(v, \text{init}(v)))$$

with the update function

$$\text{update}^*((m, m'), v) = (\text{update}(m, v), \text{update}'(m', (v, \text{update}(m, v)))).$$

Theorem 3.37. Suppose that $\mathcal{G}$ reduces to $\mathcal{G}'$ via memory $\mathfrak{M}$ and that Player σ has a winning strategy for $\mathcal{G}'$ with memory $\mathfrak{M}'$ from position $(v_0, \text{init}(v_0)))$. Then Player σ has a winning strategy for $\mathcal{G}$ with memory $\mathfrak{M} \times \mathfrak{M}'$ from position v_0.

Proof. Given a strategy $F' : (V_\sigma \times M) \times M' \to (V \times M)$ for Player σ in $\mathcal{G}'$, we have to construct a strategy $F : (V_\sigma \times (M \times M')) \to V$ for Player σ in $\mathcal{G}$. For any $v \in V_\sigma$ and any pair $(m, m') \in M \times M'$ we have that $F'((v, m), m') = (w, \text{update}(m, w))$ for some $w \in vE$. We put $F(v, (m, m')) = w$. If a play in $\mathcal{G}$ that is consistent with F proceeds from position v with current memory location (m, m') to a new position w, then the memory is updated to (n, n') with $n = \text{update}(m, w)$ and $n' = \text{update}'(m', (w, n))$. In the extended play in $\mathcal{G}'$, we have an associated move from (v, m) to (w, n) with memory update from m' to n'. Thus, every play in $\mathcal{G}$ from initial position v_0 that is consistent with F is the projection of a play in $\mathcal{G}'$ from $(v_0, \text{init}(v_0))$ that is consistent with F'. Therefore, if F' is a winning strategy from $(v_0, \text{init}(v_0))$, then F is a winning strategy from v_0. Q.E.D.

Corollary 3.38. Every game that reduces via memory $\mathfrak{M}$ to a positionally determined game is determined via memory $\mathfrak{M}$.

Obviously, memory reductions between games can be composed. If $\mathcal{G}$ reduces to $\mathcal{G}'$ with memory $\mathfrak{M} = (M, \text{update}, \text{init})$ and $\mathcal{G}'$ reduces to $\mathcal{G}''$ with memory $\mathfrak{M}' = (M', \text{update}', \text{init}')$ then $\mathcal{G}$ reduces to $\mathcal{G}''$ with memory $(M \times M', \text{update}'', \text{init}'')$ where

$$\text{init}''(v) = (\text{init}(v), \text{init}'(v, \text{init}(v)))$$

and

$$\text{update}''((m, m'), v) = (\text{update}(m, v), \text{update}'(m', (v, \text{update}(m, v)))).$$

The classical example of a game reduction with finite memory is the reduction of Muller games to parity games via latest appearance records. Intuitively, a *latest appearance record* (LAR) is a list of priorities ordered by their latest occurrence. More formally, for a finite set C of priorities, $\text{LAR}(C)$ is the set of sequences $c_1 \dots c_k \# c_{k+1} \dots c_l$ of elements from $C \cup \{\#\}$ in which each priority $c \in C$ occurs at most once and # occurs precisely once. At a position v, the LAR $c_1 \dots c_k \# c_{k+1} \dots c_l$ is updated by moving the priority $\Omega(v)$ to the end, and moving # to the previous position of $\Omega(v)$ in the sequence. For instance, at a position with priority c_2, the LAR $c_1 c_2 c_3 \# c_4 c_5$ is updated to $c_1 \# c_3 c_4 c_5 c_2$. (If $\Omega(v)$ did not occur in the LAR, we simply append $\Omega(v)$ at the end). Thus, the LAR memory for an arena with priority labelling $\Omega : V \to C$ is the triple $(\text{LAR}(C), \text{update}, \text{init})$ with $\text{init}(v) = \#\Omega(v)$ and

$$\text{update}(c_1 \dots c_k \# c_{k+1} \dots c_l, v) = \begin{cases} c_1 \dots c_k \# c_{k+1} \dots c_l \Omega(v) & \text{if } \Omega(v) \notin \{c_1, \dots c_l\}, \\ c_1 \dots c_{m-1} \# c_{m+1} \dots c_l c_m & \text{if } \Omega(v) = c_m. \end{cases}$$

The *hit set* of an LAR $c_1 \dots c_k \# c_{k+1} \dots c_l$ is the set $\{c_{k+1} \dots c_l\}$ of priorities occurring after the symbol #. Note that if in a play $\pi = v_0 v_1 \dots$ the LAR at position v_n is $c_1 \dots c_k \# c_{k+1} \dots c_l$, then $\Omega(v_n) = c_l$ and the hit set $\{c_{k+1} \dots c_l\}$ is the set of priorities that have been visited since the latest previous occurrence of c_l in the play.

Lemma 3.39. Let π be a play of a Muller game $\mathcal{G}$ with finitely many priorities, and let $\text{Inf}(\pi)$ be the set of priorities occurring infinitely often in π. Then the hit set of the latest appearance record is, from some point onwards, always a subset of $\text{Inf}(\pi)$ and infinitely often coincides with $\text{Inf}(\pi)$.

Proof. For each play $\pi = v_0 v_1 v_2 \dots$ there is a position v_m such that $\Omega(v_n) \in \text{Inf}(\pi)$ for all $n \geq m$. Since no priority outside $\text{Inf}(\pi)$ is seen after position v_m, the hit set will, from that position onwards, always be contained in $\text{Inf}(\pi)$, and the LAR will always have the form $c_1 \dots c_{j-1} c_j \dots c_k \# c_{k+1} \dots c_l$ where $c_1, \dots c_{j-1}$ remains fixed and $\text{Inf}(\pi) = \{c_j, \dots, c_l\}$. Since all priorities in $\text{Inf}(\pi)$ are seen again and again, it happens infinitely often that, among these, the one occurring leftmost in the LAR is hit. At such positions, the LAR is updated to $c_1, \dots, c_{j-1} \# c_{j+1} \dots c_l c_j$, and the hit set coincides with $\text{Inf}(\pi)$. Q.E.D.

Theorem 3.40. Every Muller game with finitely many priorities reduces via LAR memory to a parity game.

Proof. Let $\mathcal{G}$ be a Muller game with game graph G, priority labelling $\Omega : V \to C$ and winning condition $(\mathcal{F}_0, \mathcal{F}_1)$. We have to prove that $\mathcal{G} \leq_{\text{LAR}} \mathcal{G}'$ for a parity game $\mathcal{G}'$ with game graph $G \times \text{LAR}(C)$ and an appropriate priority labelling Ω' on $V \times \text{LAR}(C)$, which is defined as follows:

$$\Omega'(v, c_1 c_2 \dots c_k \# c_{k+1} \dots c_l) = \begin{cases} 2k & \text{if } \{c_{k+1}, \dots, c_l\} \in \mathcal{F}_0, \\ 2k+1 & \text{if } \{c_{k+1}, \dots, c_l\} \in \mathcal{F}_1. \end{cases}$$

Let $\pi = v_0 v_1 v_2 \dots$ be a play on $\mathcal{G}$ and fix a number m such that, for all $n \geq m$, $\Omega(v_n) \in \text{Inf}(\pi)$ and the LAR at position v_n has the form $c_1 \dots c_j c_{j+1} \dots c_k \# c_{k+1} \dots c_l$ where $\text{Inf}(\pi) = \{c_{j+1}, \dots c_l\}$ and the prefix $c_1 \dots c_j$ remains fixed. In the corresponding play $\pi' = (v_0, r_0)(v_1, r_1) \dots$ in $\mathcal{G}'$, all nodes (v_n, r_n) for $n \geq m$ have a priority $2k + \rho$ with $k \geq j$ and $\rho \in \{0, 1\}$. Assume that the play π is won by Player σ, i.e., $\text{Inf}(\pi) \in \mathcal{F}_\sigma$.

Since the hit set of the LAR coincides with $\mathrm{Inf}(\pi)$ infinitely often, the minimal priority seen infinitely often on the extended play is $2j + \sigma$. Thus the extended play in the parity game $\mathcal{G}'$ is won by the same player as the original play in $\mathcal{G}$. Q.E.D.

Corollary 3.41. Muller games are determined via finite memory strategies. The size of the memory is bounded by $(|C| + 1)!$.

The question arises which Muller conditions $(\mathcal{F}_0, \mathcal{F}_1)$ guarantee positional winning strategies for arbitrary game graphs? One obvious answer are parity conditions. But there are others:

Example 3.42. Let $C = \{0, 1\}$, $\mathcal{F}_0 = \{C\}$ and $\mathcal{F}_1 = \mathcal{P}(C) \setminus \{C\} = \{\{0\}, \{1\}, \emptyset\}$. $(\mathcal{F}_0, \mathcal{F}_1)$ is not a parity condition, but every Muller game with winning condition $(\mathcal{F}_0, \mathcal{F}_1)$ is positionally determined.

Definition 3.43. The *Zielonka tree* for a Muller condition $(\mathcal{F}_0, \mathcal{F}_1)$ over C is a tree $Z(\mathcal{F}_0, \mathcal{F}_1)$ whose nodes are labelled with pairs (X, σ) such that $X \in \mathcal{F}_\sigma$. We define $Z(\mathcal{F}_0, \mathcal{F}_1)$ inductively as follows. Let $C \in \mathcal{F}_\sigma$ and $C_0, \ldots, C_{k-1}$ be the maximal sets in $\{X \subseteq C : X \in \mathcal{F}_{1-\sigma}\}$. Then $Z(\mathcal{F}_0, \mathcal{F}_1)$ consists of a root, labelled with (C, σ), to which we attach as subtrees the Zielonka trees $Z(\mathcal{F}_0 \cap \mathcal{P}(C_i), \mathcal{F}_1 \cap \mathcal{P}(C_i))$, $i = 0, \ldots, k - 1$.

Example 3.44. Let $C = \{0, 1, 2, 3, 4\}$ and $\mathcal{F}_0 = \{\{0, 1\}, \{2, 3, 4\}, \{2, 3\}, \{2, 4\}, \{3\}, \{4\}\}$, $\mathcal{F}_1 = \mathcal{P}(C) \setminus \mathcal{F}_0$. The Zielonka tree $Z(\mathcal{F}_0, \mathcal{F}_1)$ is depicted in Figure 3.9.

A set $Y \subseteq C$ belongs to $\mathcal{F}_\sigma$ if there is a node of $Z(\mathcal{F}_0, \mathcal{F}_1)$ that is labelled with (X, σ) for some $X \supseteq Y$ and for all children $(Z, 1 - \sigma)$ of (X, σ) we have $Y \not\subseteq Z$.

Example 3.45. Consider again the tree $Z(\mathcal{F}_0, \mathcal{F}_1)$ from Example 3.44. It is the case that $\{2, 3\} \in \mathcal{F}_0$, since $(\{2, 3, 4\}, 0)$ is a node of $Z(\mathcal{F}_0, \mathcal{F}_1)$ and

- $\{2, 3\} \subseteq \{2, 3, 4\}$;
- $\{2, 3\} \not\subseteq \{2\}$;
- $\{2, 3\} \not\subseteq \{3, 4\}$.

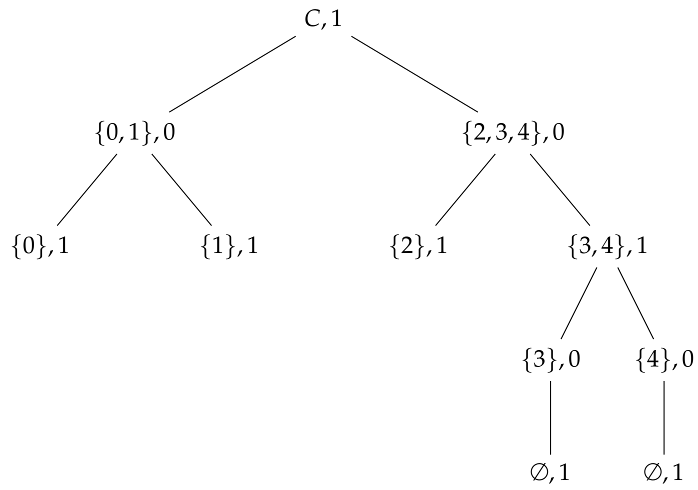


Figure 3.9. A Zielonka tree

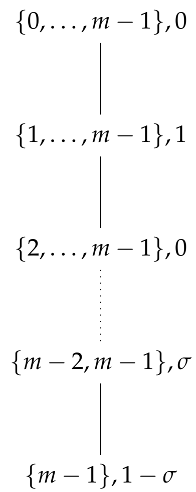


Figure 3.10. The Zielonka tree of a parity-condition with m priorities

The Zielonka tree of a parity-condition is especially simple, as Figure 3.10 shows.

Besides parity games there are other important special cases of Muller games. Of special relevance are games with Rabin and Streett conditions because these admit positional winning strategies for one player.

Definition 3.46. A *Streett-Rabin condition* is a Muller condition $(\mathcal{F}_0, \mathcal{F}_1)$ such that $\mathcal{F}_0$ is closed under union.

In the Zielonka tree for a Streett-Rabin condition, the nodes labelled with $(X, 1)$ have only one successor. It follows that if both $\mathcal{F}_0$ and $\mathcal{F}_1$ are closed under union, then the Zielonka tree $Z(\mathcal{F}_0, \mathcal{F}_1)$ is a path, which implies that $(\mathcal{F}_0, \mathcal{F}_1)$ is equivalent to a parity condition.

In a Streett-Rabin game, Player 1 has a positional winning strategy on his winning region. On the other hand, Player 0 can win on his winning region via a finite-memory strategy, and the size of the memory can be directly read off from the Zielonka tree. We present an elementary proof of this result.

Theorem 3.47. Let $\mathcal{G} = (V, V_0, V_1, E, \Omega)$ be a game with a Streett-Rabin winning condition $(\mathcal{F}_0, \mathcal{F}_1)$. Then $\mathcal{G}$ is determined, i.e. $V = W_0 \cup W_1$, with a finite memory winning strategy for Player 0 on W_0, and a positional winning strategy for Player 1 on W_1. The size of the memory required by the winning strategy for Player 0 is bounded by the number of leaves of the Zielonka tree $Z(\mathcal{F}_0, \mathcal{F}_1)$.

Proof. We proceed by induction on the number of priorities in C or, equivalently, the depth of the Zielonka tree $Z(\mathcal{F}_0, \mathcal{F}_1)$. Let l be the number of leaves of $Z(\mathcal{F}_0, \mathcal{F}_1)$. We distinguish two cases.

Case 1: $C \in \mathcal{F}_1$. Let

$$X_0 := \left\{ v : \begin{array}{l} \text{Player 0 has a winning strategy with memory} \\ \text{of size} \leq l \text{ from } v \end{array} \right\},$$

and $X_1 = V \setminus X_0$. It suffices to prove that Player 1 has a positional winning strategy on X_1. To construct this strategy, we combine three

positional strategies of Player 1: A trap strategy, an attractor strategy, and a winning strategy on a subgame with fewer priorities.

At first, we observe that X_1 is a trap for Player 0. This means that Player 1 has a positional trap strategy t on X_1 to enforce that the play stays within X_1.

Since $\mathcal{F}_0$ is closed under union, there is a unique maximal subset $C' \subseteq C$ with $C' \in \mathcal{F}_0$. Let $Y := X_1 \cap \Omega^{-1}(C \setminus C')$, and let $Z = \text{Attr}_1(Y) \setminus Y$. Observe that Player 1 has a positional attractor strategy a, by which he can force, from any position $z \in Z$, that the play reaches Y.

Finally, let $V' = X_1 \setminus (Y \cup Z)$ and let $\mathcal{G}'$ be the subgame of $\mathcal{G}$ induced by V', with winning condition $(\mathcal{F}_0 \cap \mathcal{P}(C'), \mathcal{F}_1 \cap \mathcal{P}(C'))$ (see Figure 3.11). Since this game has fewer priorities, the induction hypothesis applies, i.e. we have $V' = W_0' \cup W_1'$, and Player 0 has a winning strategy with memory $\leq l$ on W_0', whereas Player 1 has a positional winning strategy g' on W_1'. However, $W_0' = \emptyset$: Otherwise we could combine the strategies of Player 0 to obtain a winning strategy with memory $\leq l$ on $X_0 \cup W_0' \supsetneq X_0$, a contradiction to the definition of X_0. Hence $W_1' = V'$.

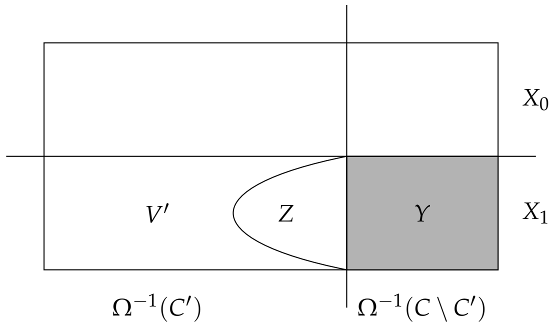


Figure 3.11. Constructing a winning strategy for Player 1

We can now define a positional strategy g for Player 1 on X_1 by

$$g(x) = \begin{cases} g'(x) & \text{if } x \in V', \\ a(x) & \text{if } x \in Z, \\ t(x) & \text{if } x \in Y. \end{cases}$$

Consider any play π that starts at a position $v \in X_1$ and is consistent with g. We have to show that π is won by Player 1. Obviously, π stays within X_1. If it hits $Y \cup Z$ only finitely often, then from some point onwards it stays within V' and coincides with a play consistent with g'. It is therefore won by Player 1. Otherwise, π hits $Y \cup Z$, and hence also Y, infinitely often. Thus, $\mathrm{Inf}(\pi) \cap (C \setminus C') \neq \emptyset$ and $\mathrm{Inf}(\pi) \in \mathcal{F}_1$. So Player 1 has a positional winning strategy on X_1.

Case 2: $C \in \mathcal{F}_0$. There exist maximal subsets $C_0, \dots, C_{k-1} \subseteq C$ with $C_i \in \mathcal{F}_1$ (see Figure 3.12). Observe that if $D \cap (C \setminus C_i) \neq \emptyset$ for all $i < k$ then $D \in \mathcal{F}_0$. Now let

$$X_1 := \{v \in V : \text{Player 1 has a positional winning strategy from } v\},$$

and $X_0 = V \setminus X_1$. We claim that Player 0 has a finite memory winning strategy of size $\leq l$ on X_0. To construct this strategy, we proceed in a similar way as above, for each of the sets $C \setminus C_i$. We will obtain strategies $f_0, \dots, f_{k-1}$ for Player 0 such that each f_i has finite memory M_i, and we will use these strategies to build a winning strategy f on X_0 with memory $M_0 \cup \dots \cup M_{k-1}$.

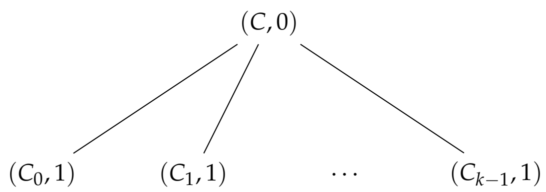


Figure 3.12. The top of the Zielonka tree $Z(\mathcal{F}_0, \mathcal{F}_1)$

For $i = 0, \dots, k-1$, let $Y_i = X_0 \cap \Omega^{-1}(C \setminus C_i)$, and $Z_i = \mathrm{Attr}_0(Y_i) \setminus Y_i$, and let a_i be a positional attractor strategy by which Player 0 can force a play from any position in Z_i to reach Y_i. Furthermore, let $U_i = X_0 \setminus (Y_i \cup Z_i)$, and let $\mathcal{G}_i$ be the subgame of $\mathcal{G}$ induced by U_i with winning condition $(\mathcal{F}_0 \cap \mathcal{P}(C_i), \mathcal{F}_1 \cap \mathcal{P}(C_i))$. The winning region of Player 1 in $\mathcal{G}_i$ is empty: Indeed, if Player 1 could win $\mathcal{G}_i$ from v, then, by the induction hypothesis, he could win with a positional winning strategy. By combining this strategy with the positional winning strategy of Player 1 on X_1, this would imply that $v \in X_1$, but $v \in U_i \subseteq V \setminus X_1$.

Hence, by the induction hypothesis, Player 0 has a winning strategy f_i with finite memory M_i on U_i. Let $(f_i + a_i)$ be the combination of f_i with the attractor strategy a_i, defined by

$$(f_i + a_i)(v) := \begin{cases} f_i(v) & \text{if } v \in U_i, \\ a_i(v) & \text{if } v \in Z_i. \end{cases}.$$

From any position $v \in U_i \cup Z_i$ this strategy ensures that the play either remains inside U_i and is winning for Player 1, or that it eventually reaches a position in Y_i.

We now combine the strategies $(f_0 + a_0), \dots, (f_{k-1} + a_{k-1})$ to a winning strategy f on X_0, which ensures that either the play ultimately remains within one of the regions U_i and coincides with a play according to f_i, or that it cycles infinitely often through all the regions $Y_0, \dots, Y_{k-1}$.

At positions in $\widetilde{Y} := \bigcap_{i<k} Y_i$, Player 0 just plays with a (positional) trap strategy t ensuring that the play remains in X_0. At the first position $v \notin \widetilde{Y}$, Player 0 takes the minimal i such that $v \notin Y_i$, i.e. $v \in U_i \cup Z_i$, and uses the strategy $(f_i + a_i)$ until a position $w \in Y_i$ is reached. At this point, Player 0 switches from i to $j = i + l \pmod{k}$ for the minimal l such that $w \notin Y_j$. Hence $w \in U_j \cup Z_j$; Player 0 now plays with strategy $(f_j + a_j)$ until a position in Y_j is reached. There Player 0 again switches to the appropriate next strategy, as he does every time he reaches $\widetilde{Y}$.

Assuming that $M_i \cap M_j = \emptyset$ for $i \neq j$, it is not difficult to see that f can be implemented with memory $M = M_0 \cup \dots \cup M_{k-1}$. We leave the formal definition of f as an exercise.

Note that, by the induction hypothesis, the size of the memory M_i is bounded by the number of leaves of the Zielonka subtrees $Z(\mathcal{F}_0 \cap$

$\mathcal{P}(C_i), \mathcal{F}_1 \cap \mathcal{P}(C_i))$. Consequently, the size of M is bounded by the number of leaves of $Z(\mathcal{F}_0, \mathcal{F}_1)$.

It remains to prove that f is winning on X_0. Let π be a play that starts in X_0 and is consistent with f. If π eventually remains inside some U_i, then from some point onwards it coincides with a play that is consistent with f_i and is therefore won by Player 0. Otherwise, it is easy to see that π hits each of the sets $Y_0, \ldots, Y_{k-1}$ infinitely often. But this means that $\mathrm{Inf}(\pi) \cap (C \setminus C_i) \neq \emptyset$ for all $i \leq k$; as observed above this implies that $\mathrm{Inf}(\pi) \in \mathcal{F}_0$. Q.E.D.

An immediate consequence of Theorem 3.47 is that parity games are positionally determined.

3.4 Complexity

We will now determine the complexity of computing the winning regions for games over finite game graphs. The associated decision problem is

> *Given*: Game graph $\mathcal{G}$, winning condition $(\mathcal{F}_0, \mathcal{F}_1)$, $v \in V$, $\sigma \in \{0, 1\}$.
> *Question*: $v \in W_\sigma$?

For parity games, we already know that this problem is in NP ∩ coNP, and it is conjectured to be in P. Moreover, for many special cases, we know that it is indeed in P. Now we will examine the complexity of Streett-Rabin games and games with arbitrary Muller conditions.

Theorem 3.48. Deciding whether Player σ wins from a given position in a Streett-Rabin game is

- coNP-hard for $\sigma = 0$,
- NP-hard for $\sigma = 1$.

Proof. It is sufficient to prove the claim for $\sigma = 1$ since Streett-Rabin games are determined. We will reduce the satisfiability problem for

Boolean formulae in CNF to the given problem. For every formula

$$\Psi = \bigwedge_i C_i, \quad C_i = \bigvee_j Y_{ij}$$

in CNF, we define the game $\mathcal{G}_\Psi$ as follows: Positions for Player 0 are the literals $X_1, \ldots, X_k, \neg X_1, \ldots, \neg X_k$ occurring in Ψ; positions for Player 1 are the clauses $C_1, \ldots, C_n$. Player 1 can move from a clause C to a literal $Y \in C$; Player 0 can move from Y to any clause. The winning condition is given by

$$\mathcal{F}_0 = \{Z : \{X, \neg X\} \subseteq Z \text{ for at least one variable X}\}.$$

Obviously, $(\mathcal{F}_0, \mathcal{F}_1)$ is a Streett-Rabin condition.

We claim that Ψ is satisfiable if and only if Player 1 wins $\mathcal{G}_\Psi$ (from any initial position).

(⇒) Assume that Ψ is satisfiable. There exists a satisfying interpretation $\mathcal{I} : \{X_1, \ldots, X_k\} \to \{0, 1\}$. Player 1 moves from a clause C to a literal $Y \in C$ such that $[\![Y]\!]^\mathcal{I} = 1$. In the resulting play only literals with $[\![Y]\!]^\mathcal{I} = 1$ are seen, and thus Player 1 wins.

(⇐) Assume that Ψ is unsatisfiable. It is sufficient to show that Player 1 has no positional winning strategy. Every positional strategy f for Player 1 chooses a literal $Y = f(C) \in C$ for every clause C. Since Ψ is unsatisfiable, there exists clauses C, C' and a variable X such that $f(C) = X$, $f(C') = \neg X$. Otherwise, f would define a satisfying interpretation for Ψ. Player 0's winning strategy is to move from $\neg X$ to C and from any other literal to C'. Then X and $\neg X$ are seen infinitely often, and Player 0 wins. Thus, f is not a winning strategy for Player 1. If Player 1 has no positional winning strategy, he has no winning strategy at all.

Is $\Psi \mapsto \mathcal{G}_\Psi$ a polynomial reduction? The problem that arises is the winning condition: Both $\mathcal{F}_0$ and $\mathcal{F}_1$ contain exponentially many sets. Moreover, the Zielonka tree $Z(\mathcal{F}_0, \mathcal{F}_1)$ has exponential size. On the other hand, $\mathcal{F}_0$ and $\mathcal{F}_1$ can be represented in a very compact way using a Boolean formula in the following sense: Let $(\mathcal{F}_0, \mathcal{F}_1)$ be a Muller condition over C. A Boolean formula Ψ with variables in C defines the set $\mathcal{F}_\Psi = \{Y \subseteq C : \mathcal{I}_Y \models \Psi\}$ where

$$\mathcal{I}_Y(c) = \begin{cases} 1 & \text{if } c \in Y \\ 0 & \text{if } c \notin Y. \end{cases}$$

Ψ defines $(\mathcal{F}_0, \mathcal{F}_1)$ if $\mathcal{F}_\Psi = \mathcal{F}_0$ (and thus $\mathcal{F}_{\neg\Psi} = \mathcal{F}_1$). Representing the winning condition by a Boolean formula makes the reduction a polynomial reduction. Q.E.D.

Another way of defining Streett-Rabin games is by a collection of pairs (L, R) with $L, R \subseteq C$. The collection $\{(L_1, R_1), \ldots, (L_k, R_k)\}$ defines the Muller condition $(\mathcal{F}_0, \mathcal{F}_1)$ given by:

$$\mathcal{F}_0 = \{X \subseteq C : X \cap L_i \neq \emptyset \Rightarrow X \cap R_i \neq \emptyset \text{ for all } i \leq k\}.$$

We have:

- Every Muller condition defined by a collection of pairs is a Streett-Rabin condition.
- Every Streett-Rabin condition is definable by a collection of pairs.
- Representing a Streett-Rabin condition by a collection of pairs can be exponentially more succinct than a representation by its Zielonka tree or an explicit enumeration of $\mathcal{F}_0$ or $\mathcal{F}_1$: There are Streett-Rabin conditions definable with k pairs such that the corresponding Zielonka tree has $k!$ leaves.

The reduction $\Psi \mapsto \mathcal{G}_\Psi$ can be modified such that the winning condition is given by $2m$ pairs, where m is the number of variables in Ψ:

$$L_{2i} = \{X_i\}, \quad R_{2i} = \{\neg X_i\}, \quad L_{2i-1} = \{\neg X_i\}, \quad R_{2i-1} = \{X_i\}.$$

For the Streett-Rabin condition defined by $\{(L_1, R_1), \ldots, (L_{2m}, R_{2m})\}$ we have that

$$\mathcal{F}_1 = \left\{ Z : \begin{array}{l} Z \text{ contains a Literal } X_i \text{ (or } \neg X_i\text{) such that the} \\ \text{complementary literal } \neg X_i \text{ (respectively } X_i\text{) is} \\ \text{not contained in } Z \end{array} \right\}.$$

The winning strategies used in the proof remain winning for the modified winning condition.

To prove the upper bounds for the complexity of Streett-Rabin games we will consider solitaire games first.

Theorem 3.49. Let $\mathcal{G}$ be a Streett-Rabin game such that only Player 0 can do non-trivial moves. Then the winning regions W_0 and W_1 can be computed in polynomial time.

Proof. Let us assume that the winning condition is given by the collection $\mathcal{P} = \{(L_1, R_1), \ldots, (L_k, R_k)\}$ of pairs. It is sufficient to prove the claim for W_0 since Streett-Rabin games are determined. Every play π will ultimately stay in a strongly connected set $U \subseteq V$ such that all positions in U are seen infinitely often. Therefore, we call a strongly connected set U *good for Player* 0 if for all $i \leq k$

$$\Omega(U) \cap L_i \neq \emptyset \Rightarrow \Omega(U) \cap R_i \neq \emptyset.$$

For every such U, $\text{Attr}_0(U) \subseteq W_0$. If U is not good for Player 0 then there is a node in U which violates a pair (L_i, R_i). In this case Player 0 wants to find a (strongly connected) subset of U where she can win nevertheless. We can eliminate the pairs (L_i, R_i) where $\Omega(U) \cap L_i = \emptyset$ since they never violate the winning condition. On the other hand, Player 0 loses if a node of

$$\widetilde{U} = \{u \in U \mid \Omega(u) \in L_i \text{ for some i such that } \Omega(U) \cap R_i = \emptyset\}$$

is visited again and again. Thus we will reduce the game from U to $U \setminus \widetilde{U}$ with the modified winning condition $\mathcal{P}' = \{(L_i, R_i) \in \mathcal{P} : \Omega(U) \cap L_i \neq \emptyset\}$. This yields Algorithm 3.1.

The SCC decomposition can be computed in linear time. The decomposition algorithm will be called less than $|V|$ times, the rest are elementary steps. Therefore, the algorithm runs in polynomial time.

It remains to show that $W_0 = \text{WinReg}(G, \mathcal{P})$:

($\subseteq$) Let $v \in W_0$. Player 0 can reach from v a strongly connected set S that satisfies the winning condition. S is a subset of an SCC U of G. If U satisfies the winning condition, then $v \in \text{WinReg}(G, \mathcal{P})$. Otherwise,

Algorithm 3.1. A polynomial time algorithm solving solitaire Streett-Rabin games

Algorithm WinReg$(G, \mathcal{P})$
Input: Streett-Rabin game with game graph G and pairs condition $\mathcal{P}$.
Output: W_0, the winning region for Player 0.

```
W_0 := ∅;
Decompose G into its SCCs;
For every SCC U do
    P' := {(L_i, R_i) : Ω(U) ∩ L_i ≠ ∅};
    Ũ := {u ∈ U : Ω(u) ∈ L_i for some i such that Ω(U) ∩ R_i = ∅};
    if Ũ = ∅ then W := W ∪ U;
    else W := W ∪ WinReg(G ↾_{U\(Ũ)}, P');
enddo;
W_0 := Attr_0(W);
Output W_0;
```

$S \subseteq U \setminus \widetilde{U}$, and S is contained in an SCC of $G \restriction_{U \setminus \widetilde{U}}$. The repetition of the argument leads to $S \subseteq W$ and therefore $v \in \text{WinReg}(G, \mathcal{P})$

($\supseteq$) Let $v \in \text{WinReg}(G, \mathcal{P})$. The algorithm finds a strongly connected set U (an SCC of a subgraph) that is reachable from v and that satisfies the winning condition. By moving from v into U, staying there, and visiting all positions in U infinitely often, Player 0 wins. Thus $v \in W_0$. Q.E.D.

Theorem 3.50. Deciding whether Player σ wins from a given position in a Streett-Rabin game is

- coNP-complete for $\sigma = 0$,
- NP-complete for $\sigma = 1$.

Proof. It suffices to prove the claim for Player 1 since W_0 is the complement of W_1. Hardness follows from Theorem 3.48. To decide whether $v \in W_1$, guess a positional strategy for Player 1 and construct the induced solitaire game, in which only Player 0 has non-trivial moves. Then decide in polynomial time whether v is in the winning region of Player 1 in the solitaire game (according to Theorem 3.49), i.e. whether

the strategy is winning from v. If this is the case, accept; otherwise reject. Q.E.D.

Remark 3.51. The complexity of computing the winning regions in arbitrary Muller games depends to a great amount on the representation of the winning condition. For any reasonable representation, the problem is in PSPACE, and many representations are so succinct as to render the problem PSPACE-hard. Only recently, it was shown that, given an explicit representation of the winning condition, the problem of deciding the winner is in P.

4 Basic Concepts of Mathematical Game Theory

Up to now we considered finite or infinite games

- with two players,
- played on finite or infinite graphs,
- with perfect information (the players know the whole game, the history of the play and the actual position),
- with qualitative (win or loss) winning conditions (zero-sum games),
- with ω-regular winning conditions (or Borel winning conditions) specified in a suitable logic or by automata, and
- with asynchronous interaction (turn-based games).

Those games are used for verification or to evaluate logic formulae.

In this section we move to concurrent multi-player games in which players get real-valued *payoffs*. The games will still have perfect information and additionally throughout this chapter we assume that the set of possible plays is *finite*, so there exist only finitely many strategies for each of the players.

4.1 Games in Strategic Form

Definition 4.1. A *game in strategic form* is described by a tuple $\Gamma = (N, (S_i)_{i \in N}, (p_i)_{i \in N})$ where

- $N = \{1, \dots, n\}$ is a finite set of players
- S_i is a set of *strategies* for Player i
- $p_i : S \to \mathbb{R}$ is a *payoff function* for Player i

and $S := S_1 \times \cdots \times S_n$ is the set of *strategy profiles*. Γ is called a *zero-sum game* if $\sum_{i \in N} p_i(s) = 0$ for all $s \in S$.

The number $p_i(s_1, \dots, s_n)$ is called the *value* or *utility* of the strategy profile $(s_1, \dots, s_n)$ for Player i. The intuition for zero-sum games is that the game is a closed system.

Many important notions can best be explained by two-player games, but are defined for arbitrary multi-player games.

In the sequel, we will use the following notation: Let Γ be a game. Then $S_{-i} := S_1 \times \cdots \times S_{i-1} \times S_{i+1} \times \cdots \times S_n$ is the set of all strategy profiles for the players except i. For $s \in S_i$ and $s_{-i} \in S_{-i}$, (s, s_{-i}) is the strategy profile where Player i chooses the strategy s and the other players choose s_{-i}.

Definition 4.2. Let $s, s' \in S_i$. Then s *dominates* s' if

- for all $s_{-i} \in S_{-i}$ we have $p_i(s, s_{-i}) \geq p_i(s', s_{-i})$, and
- there exists $s_{-i} \in S_{-i}$ such that $p_i(s, s_{-i}) > p_i(s', s_{-i})$.

A strategy s is *dominant* if it dominates some other strategy of the player.

Definition 4.3. An *equilibrium in dominant strategies* is a strategy profile $(s_1, \dots, s_n) \in S$ such that all s_i are dominant strategies.

Definition 4.4. A strategy $s \in S_i$ is a *best response* to $s_{-i} \in S_{-i}$ if $p_i(s, s_{-i}) \geq p_i(s', s_{-i})$ for all $s' \in S_i$.

Remark 4.5. A dominant strategy is a best response for all strategy profiles of the other players.

Example 4.6.

- Prisoner's Dilemma:

  Two suspects are arrested, but there is insufficient evidence for a conviction. Both prisoners are questioned separately, and are offered the same deal: if one testifies for the prosecution against the other and the other remains silent, the betrayer goes free and the silent accomplice receives the full 10-year sentence. If both stay silent, both prisoners are sentenced to only one year in jail for a minor charge. If both betray each other, each receives a five-year sentence. So this dilemma poses the question: How should the prisoners act?

	stay silent	betray
stay silent	$(-1, -1)$	$(-10, 0)$
betray	$(0, -10)$	$(-5, -5)$

  An entry (a, b) at position i, j of the matrix means that if profile (i, j) is chosen, Player 1 (who chooses the rows) receives payoff a and Player 2 (who chooses the columns) receives payoff b.

  Betraying is a dominant strategy for every player, call this strategy b. Therefore, (b, b) is an equilibrium in dominant strategies. Problem: The payoff $(-5, -5)$ of the dominant equilibrium is not optimal.

  The Prisoner's Dilemma is an important metaphor for many decision situations, and there exists extensive literature concerned with the problem. Especially interesting is the situation, where the Prisoner's Dilemma is played infinitely often.

- Battle of the sexes:

	meat	fish
red wine	$(2, 1)$	$(0, 0)$
white wine	$(0, 0)$	$(1, 2)$

  There are no dominant strategies, and thus there is no dominant equilibrium. The pairs (red wine, meat) and (white wine, fish) are distinguished since every player plays with a best response against the strategy of the other player: No player would change his strategy unilaterally.

Definition 4.7. A strategy profile $s = (s_1, \dots, s_n) \in S$ is a *Nash equilibrium* in Γ if

$$p_i(\underbrace{s_i, s_{-i}}_{s}) \geq p_i(s', s_{-i})$$

holds for all $i \in N$ and all strategies $s'_i \in S_i$, i.e., for every Player i, s_i is a best response for s_{-i}.

Is there a Nash equilibrium in every game? Yes, but not necessarily in pure strategies!

Example 4.8. Rock, paper, scissors:

	rock	scissors	paper
rock	$(0,0)$	$(1,-1)$	$(-1,1)$
scissors	$(-1,1)$	$(0,0)$	$(1,-1)$
paper	$(1,-1)$	$(-1,1)$	$(0,0)$

There are no dominant strategies and no Nash equilibria: For every pair (f,g) of strategies one of the players can change to a better strategy. Note that this game is a zero-sum game. But there is a reasonable strategy to win this game: Randomly pick one of the three actions with equal probability.

This observation leads us to the notion of mixed strategies, where the players are allowed to randomise over strategies.

Definition 4.9. A *mixed strategy* of Player i in Γ is a probability distribution $\mu_i : S_i \to [0,1]$ on S_i where $\sum_{s\in S_i} \mu(s) = 1$.
$\Delta(S_i)$ denotes the set of probability distributions on S_i. $\Delta(S) := \Delta(S_1) \times \cdots \times \Delta(S_n)$ is the set of all strategy profiles in mixed strategies. The expected payoff is $\widehat{p}_i : \Delta(S) \to \mathbb{R}$,

$$\widehat{p}_i(\mu_1,\ldots,\mu_n) = \sum_{(s_1,\ldots,s_n)\in S} \left(\prod_{j\in N} \mu_j(s_j) \right) \cdot p_i(s_1,\ldots,s_n)$$

For every game $\Gamma = (N,(S_i)_{i\in N},(p_i)_{i\in N})$ we define the *mixed expansion* $\widehat{\Gamma} = (N,(\Delta(S_i))_{i\in N},(\widehat{p}_i)_{i\in N})$.

Definition 4.10. A *Nash equilibrium of* Γ *in mixed strategies* is a Nash equilibrium in $\widehat{\Gamma}$, i.e. a Nash equilibrium in Γ in mixed strategies is a mixed strategy profile $\overline{\mu} = (\mu_1,\ldots,\mu_n) \in \Delta(S)$ such that, for every player i and every $\mu_i' \in \Delta(S)$, $\widehat{p}_i(\mu_i,\mu_{-i}) \geq \widehat{p}_i(\mu_i',\mu_{-i})$.

Theorem 4.11 (Nash)**.** Every finite game Γ in strategic form has at least one Nash equilibrium in mixed strategies.

To prove this theorem, we will use a well-known fixed-point theorem.

Theorem 4.12 (Brouwer's fixed-point theorem)**.** Let $X \subseteq \mathbb{R}^n$ be compact (i.e., closed and bounded) and convex. Then every continuous function $f : X \to X$ has a fixed point.

Proof (of Theorem 4.11). Let $\Gamma = (N,(S_i)_{i\in N},(p_i)_{i\in N})$. Every mixed strategy of Player i is a tuple $\mu_i = (\mu_{i,s})_{s\in S_i} \in [0,1]^{|S_i|}$ such that $\sum_{s\in S_i} \mu_{i,s} = 1$. Thus, $\Delta(S_i) \subseteq [0,1]^{|S_i|}$ is a compact and convex set, and the same applies to $\Delta(S) = \Delta(S_1) \times \cdots \times \Delta(S_n)$ for $N = \{1,\ldots,n\}$. For every $i \in N$, every pure strategy $s \in S_i$ and every mixed strategy profile $\overline{\mu} \in \Delta(S)$ let

$$g_{i,s}(\overline{\mu}) := \max\left(\widehat{p}_i(s,\overline{\mu}_{-i}) - \widehat{p}_i(\overline{\mu}), 0\right)$$

be the gain of Player i if he unilaterally changes from the mixed profile $\overline{\mu}$ to the pure strategy s (only if this is reasonable).

Note that if $g_{i,s}(\overline{\mu}) = 0$ for all i and all $s \in S_i$, then $\overline{\mu}$ is a Nash equilibrium. We define the function

$$\begin{aligned} f : \Delta(S) &\to \Delta(S) \\ \overline{\mu} &\mapsto f(\overline{\mu}) = (\nu_1,\ldots,\nu_n) \end{aligned}$$

where $\nu_i : S_i \to [0,1]$ is a mixed strategy defined by

$$\nu_{i,s} = \frac{\mu_{i,s} + g_{i,s}(\overline{\mu})}{1 + \sum_{s\in S_i} g_{i,s}(\overline{\mu})}.$$

For every Player i and all $s \in S_i$, $\overline{\mu} \mapsto \nu_{i,s}$ is continuous since $\widehat{p}_i$ is continuous and thus $g_{i,s}$, too. $f(\overline{\mu}) = (\nu_1,\ldots,\nu_n)$ is in $\Delta(S)$: Every $\nu_i = (\nu_{i,s})_{s\in S_i}$ is in $\Delta(S_i)$ since

$$\sum_{s\in S_i} \nu_{i,s} = \frac{\sum_{s\in S_i} \mu_{i,s} + \sum_{s\in S_i} g_{i,s}(\overline{\mu})}{1 + \sum_{s\in S_i} g_{i,s}(\overline{\mu})} = \frac{1 + \sum_{s\in S_i} g_{i,s}(\overline{\mu})}{1 + \sum_{s\in S_i} g_{i,s}(\overline{\mu})} = 1.$$

By the Brouwer fixed point theorem f has a fixed point. Thus, there is a $\overline{\mu} \in \Delta(S)$ such that

$$\mu_{i,s} = \frac{\mu_{i,s} + g_{i,s}(\overline{\mu})}{1 + \sum_{s \in S_i} g_{i,s}(\overline{\mu})}$$

for all i and all s.
Case 1: There is a Player i such that $\sum_{s \in S_i} g_{i,s}(\overline{\mu}) > 0$.
Multiplying both sides of the fraction above by the denominator, we get $\mu_{i,s} \cdot \sum_{s \in S_i} g_{i,s}(\mu) = g_{i,s}(\overline{\mu})$. This implies $\mu_{i,s} = 0 \;\Leftrightarrow\; g_{i,s}(\overline{\mu}) = 0$, and thus $g_{i,s}(\overline{\mu}) > 0$ for all $s \in S_i$ where $\mu_{i,s} > 0$.

But this leads to a contradiction: $g_{i,s}(\overline{\mu}) > 0$ means that it is profitable for Player i to switch from (μ_i, μ_{-i}) to (s, μ_{-i}). This cannot be true for all s where $\mu_{i,s} > 0$ since the payoff for (μ_i, μ_{-i}) is the mean of the payoffs (s, μ_{-i}) with arbitrary $\mu_{i,s}$. However, the mean cannot be smaller than all components:

$$\begin{aligned}
\widehat{p}_i(\mu_i, \mu_{-i}) &= \sum_{s \in S_i} \mu_{i,s} \cdot \widehat{p}_i(s, \mu_{-i}) \\
&= \sum_{\substack{s \in S_i \\ \mu_{i,s} > 0}} \mu_{i,s} \cdot \widehat{p}_i(s, \mu_{-i}) \\
&> \sum_{\substack{s \in S_i \\ \mu_{i,s} > 0}} \mu_{i,s} \cdot \widehat{p}_i(\mu_i, \mu_{-i}) \\
&= \widehat{p}_i(\mu_i, \mu_{-i})
\end{aligned}$$

which is a contradiction.
Case 2: $g_{i,s}(\overline{\mu}) = 0$ for all i and all $s \in S_i$, but this already means that $\overline{\mu}$ is a Nash equilibrium as stated before. Q.E.D.

The *support* of a mixed strategy $\mu_i \in \Delta(S_i)$ is $\operatorname{supp}(\mu_i) = \{s \in S_i : \mu_i(s) > 0\}$.

Theorem 4.13. Let $\mu^* = (\mu_1, \dots, \mu_n)$ be a Nash equilibrium in mixed strategies of a game Γ. Then for every Player i and every pure strategy $s, s' \in \operatorname{supp}(\mu_i)$

$$\widehat{p}_i(s, \mu_{-i}) = \widehat{p}_i(s', \mu_{-i}).$$

Proof. Assume $\widehat{p}_i(s, \mu_{-i}) > \widehat{p}_i(s', \mu_{-i})$. Then Player i could achieve a

higher payoff against μ_{-i} if she played s instead of s': Define $\tilde{\mu}_i \in \Delta(S_i)$ as follows:

- $\tilde{\mu}_i(s) = \mu_i(s) + \mu_i(s')$,
- $\tilde{\mu}_i(s') = 0$,
- $\tilde{\mu}_i(t) = \mu_i(t)$ for all $t \in S_i - \{s, s'\}$.

Then

$$\begin{aligned}
\widehat{p}_i(\tilde{\mu}_i, \mu_{-i}) &= \widehat{p}_i(\mu_i, \mu_{-i}) + \underbrace{\mu_i(s')}_{>0} \cdot \underbrace{(\widehat{p}_i(s, \mu_{-i}) - \widehat{p}_i(s', \mu_{-i}))}_{>0} \\
&> \widehat{p}_i(\mu_i, \mu_{-i})
\end{aligned}$$

which contradicts the fact that μ is a Nash equilibrium. Q.E.D.

We want to apply Nash's theorem to two-person games. First, we note that in every game $\Gamma = (\{0,1\}, (S_0, S_1), (p_0, p_1))$

$$\max_{f \in \Delta(S_0)} \min_{g \in \Delta(S_1)} p_0(f, g) \le \min_{g \in \Delta(S_1)} \max_{f \in \Delta(S_0)} p_0(f, g).$$

The maximal payoff which one player can enforce cannot exceed the minimal payoff the other player has to cede. This is a special case of the general observation that for every function $f : X \times Y \to \mathbb{R}$

$$\sup_x \inf_y h(x, y) \le \inf_y \sup_x h(x, y).$$

(For all x', y: $h(x', y) \le \sup_x h(x, y)$. Thus $\inf_y h(x', y) \le \inf_y \sup_x h(x, y)$ and $\sup_x \inf_y h(x, y) \le \inf_y \sup_x h(x, y)$.)

Remark 4.14. Another well-known special case is

$$\exists x \, \forall y \, Rxy \models \forall y \, \exists x \, Rxy.$$

Example 4.15. Consider the following two-player "traveller" game $\Gamma = (\{1,2\}, (S_1, S_2), (p_1, p_2))$ with $S_1 = S_2 = \{2, \dots, 100\}$ and

$$p_1(x,y) = \begin{cases} x+2 & \text{if } x < y, \\ y-2 & \text{if } y < x, \\ x & \text{if } x = y, \end{cases}$$

$$p_2(x,y) = \begin{cases} x-2 & \text{if } x < y, \\ y+2 & \text{if } y < x, \\ y & \text{if } x = y. \end{cases}$$

Let's play this game! These are the results from the lecture in 2009:

$$2, 49, 49, 50, 51, 92, 97, 98, 99, 99, 100.$$

But what are the Nash equilibria? Observe that the only pure-strategy Nash equilibrium is $(2,2)$ since for each (i,j) with $i \neq j$ the player that has chosen the greater number, say i, can do better by switching to $j-1$, and also, for every (i,i) with $i > 2$ each player can do better by playing $i-1$ (and getting the payoff $i+1$ then). But would you really expect such a good payoff playing 2? Look at how others played: 97 seems to be much better against what people do in most cases!

Theorem 4.16 (v. Neumann, Morgenstern)**.**
Let $\Gamma = (\{0,1\}, (S_0, S_1), (p, -p))$ be a two-person zero-sum game. For every Nash equilibrium (f^*, g^*) in mixed strategies

$$\max_{f \in \Delta(S_0)} \min_{g \in \Delta(S_1)} p(f,g) = p(f^*, g^*) = \min_{g \in \Delta(S_1)} \max_{f \in \Delta(S_0)} p(f,g).$$

In particular, all Nash equilibria have the same payoff which is called the *value* of the game. Furthermore, both players have optimal strategies to realise this value.

Proof. Since (f^*, g^*) is a Nash equilibrium, for all $f \in \Delta(S_0)$, $g \in \Delta(S_1)$

$$p(f^*, g) \geq p(f^*, g^*) \geq p(f, g^*).$$

Thus

$$\min_{g \in \Delta(S_1)} p(f^*, g) = p(f^*, g^*) = \max_{f \in \Delta(S_1)} p(f, g^*).$$

So

$$\max_{f \in \Delta(S_0)} \min_{g \in \Delta(S_1)} p(f,g) \geq p(f^*, g^*) \geq \min_{g \in \Delta(S_1)} \max_{f \in \Delta(S_0)} p(f,g)$$

and

$$\max_{f \in \Delta(S_0)} \min_{g \in \Delta(S_1)} p(f,g) \leq \min_{g \in \Delta(S_1)} \max_{f \in \Delta(S_0)} p(f,g)$$

imply the claim. Q.E.D.

4.2 Iterated Elimination of Dominated Strategies

Besides Nash equilibria, the iterated elimination of dominated strategies is a promising solution concept for strategic games which is inspired by the following ideas. Assuming that each player behaves rational in the sense that he will not play a strategy that is dominated by another one, dominated strategies may be eliminated. Assuming further that it is common knowledge among the players that each player behaves rational, and thus discards some of her strategies, such elimination steps may be iterated as it is possible that some other strategies become dominated due to the elimination of previously dominated strategies. Iterating these elimination steps eventually yields a fixed point where no strategies are dominated.

Example 4.17.

	X: L	X: R	Y: L	Y: R
T	$(1,0,1)$	$\boxed{(1,1,0)}$	$(1,0,1)$	$(0,1,0)$
B	$(1,1,1)$	$(0,0,1)$	$(1,1,1)$	$(1,0,0)$

Player 1 picks rows, Player 2 picks columns, and Player 3 picks matrices.

- No row dominates the other (for Player 1);
- no column dominates the other (for Player 2);
- matrix X dominates matrix Y (for Player 3).

Thus, matrix Y is eliminated.

- In the remaining game, the upper row dominates the lower one (for Player 1).

Thus, the lower row is eliminated.

- Of the remaining two possibilities, Player 2 picks the better one.

The only remaining profile is (T, R, X).

There are different variants of strategy elimination that have to be considered:

- dominance by *pure* or *mixed* strategies;
- (*weak*) dominance or *strict* dominance;
- dominance by strategies in the *local* subgame or by strategies in the *global* game.

The possible combinations of these parameters give rise to eight different operators for strategy elimination that will be defined more formally in the following.

Let $\Gamma = (N, (S_i)_{i \in N}, (p_i)_{i \in N})$ such that S_i is finite for every Player i. A subgame is defined by $T = (T_1, \dots, T_n)$ with $T_i \subseteq S_i$ for all i. Let $\mu_i \in \Delta(S_i)$, and $s_i \in S_i$. We define two notions of dominance:

(1) Dominance with respect to T:
$\mu_i >_T s_i$ if and only if
- $p_i(\mu_i, t_{-i}) \geq p_i(s_i, t_{-i})$ for all $t_{-i} \in T_{-i}$
- $p_i(\mu_i, t_{-i}) > p_i(s_i, t_{-i})$ for some $t_{-i} \in T_{-i}$.

(2) Strict dominance with respect to T:
$\mu_i \gg_T s_i$ if and only if $p_i(\mu_i, t_{-i}) > p_i(s_i, t_{-i})$ for all $t_{-i} \in T_{-i}$.

We obtain the following operators on $T = (T_1, \dots, T_n)$, $T_i \subseteq S_i$, that are defined component-wise:

$$\mathrm{ML}(T)_i := \{t_i \in T_i : \neg\exists \mu_i \in \Delta(T_i)\ \mu_i >_T t_i\},$$
$$\mathrm{MG}(T)_i := \{t_i \in T_i : \neg\exists \mu_i \in \Delta(S_i)\ \mu_i >_T t_i\},$$
$$\mathrm{PL}(T)_i := \{t_i \in T_i : \neg\exists t'_i \in T_i\ t'_i >_T t_i\}, \text{and}$$
$$\mathrm{PG}(T)_i := \{t_i \in T_i : \neg\exists s_i \in S_i\ s_i >_T t_i\}.$$

MLS, MGS, PLS, PGS are defined analogously with $\gg_T$ instead of $>_T$. For all T we have the following obvious inclusions:

- Every M-operator eliminates more strategies than the corresponding P-operator.
- Every operator considering (weak) dominance eliminates more strategies than the corresponding operator considering strict dominance.
- With dominance in global games more strategies are eliminated than with dominance in local games.

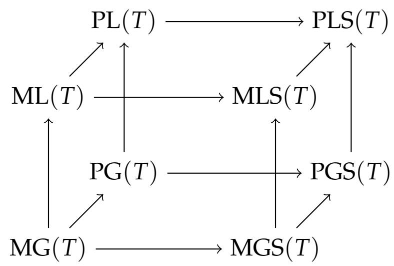


Figure 4.1. Inclusions between the eight strategy elimination operators

Each of these operators is deflationary, i.e. $F(T) \subseteq T$ for every T and every operator F. We iterate an operator beginning with $T = S$, i.e. $F^0 := S$ and $F^{\alpha+1} := F(F^\alpha)$. Obviously, $F^0 \supseteq F^1 \supseteq \cdots \supseteq F^\alpha \supseteq F^{\alpha+1}$. Since S is finite, we will reach a fixed point F^α such that $F^\alpha = F^{\alpha+1} =: F^\infty$. We expect that for the eight fixed points MG^∞, ML^∞, etc. the same inclusions hold as for the operators $\mathrm{MG}(T)$, $\mathrm{ML}(T)$, etc. But this is not the case: For the following game $\Gamma = (\{0,1\}, (S_0, S_1), (p_0, p_1))$ we have $\mathrm{ML}^\infty \not\subseteq \mathrm{PL}^\infty$.

	X	Y	Z
A	$(2,1)$	$(0,1)$	$(1,0)$
B	$(0,1)$	$(2,1)$	$(1,0)$
C	$(1,1)$	$(1,0)$	$(0,0)$
D	$(1,0)$	$(0,1)$	$(0,0)$

We have:

- Z is dominated by X and Y.
- D is dominated by A.
- C is dominated by $\frac{1}{2}A + \frac{1}{2}B$.

Thus:

$$\begin{aligned} \mathrm{ML}(S) = \mathrm{ML}^1 = (\{A,B\},\{X,Y\}) \subset \mathrm{PL}(S) = \mathrm{PL}^1 \\ = (\{A,B,C\},\{X,Y\}). \end{aligned}$$

$\mathrm{ML}(\mathrm{ML}^1) = \mathrm{ML}^1$ since in the following game there are no dominated strategies:

	X	Y
A	$(2,1)$	$(0,1)$
B	$(0,1)$	$(2,1)$

$\mathrm{PL}(\mathrm{PL}^1) = (\{A,B,C\},\{X\}) = \mathrm{PL}^2 \subsetneq \mathrm{PL}^1$ since Y is dominated by X (here we need the presence of C). Since B and C are now dominated by A, we have $\mathrm{PL}^3 = (\{A\},\{X\}) = \mathrm{PL}^\infty$. Thus, $\mathrm{PL}^\infty \subsetneq \mathrm{ML}^\infty$ although ML is the stronger operator.

We are interested in the inclusions of the fixed points of the different operators. But we only know the inclusions for the operators. So the question arises under which assumptions can we prove, for two deflationary operators F and G on S, the following claim:

If $F(T) \subseteq G(T)$ for all T, then $F^\infty \subseteq G^\infty$?

The obvious proof strategy is induction over α: We have $F^0 = G^0 = S$, and if $F^\alpha \subseteq G^\alpha$, then

$$\begin{aligned} F^{\alpha+1} = F(F^\alpha) \subseteq G(F^\alpha) \\ F(G^\alpha) \subseteq G(G^\alpha) = G^{\alpha+1} \end{aligned}$$

If we can show one of the inclusions $F(F^\alpha) \subseteq F(G^\alpha)$ or $G(F^\alpha) \subseteq G(G^\alpha)$, then we have proven the claim. These inclusions hold if the operators are monotone: $H : S \to S$ is monotone if $T \subseteq T'$ implies $H(T) \subseteq H(T')$. Thus, we have shown:

Lemma 4.18. Let $F, G : \mathcal{P}(S) \to \mathcal{P}(S)$ be two deflationary operators such that $F(T) \subseteq G(T)$ for all $T \subseteq S$. If either F or G is monotone, then $F^\infty \subseteq G^\infty$.

Corollary 4.19. PL and ML are not monotone on every game.

Which operators are monotone? Obviously, MGS and PGS are monotone: If $\mu_i \gg_T s_i$ and $T' \subseteq T$, then also $\mu_i \gg_{T'} s_i$. Let $T' \subseteq T$ and $s_i \in \mathrm{PGS}(T')_i$. Thus, there is no $\mu_i \in S_i$ such that $\mu_i \gg_{T'} s_i$, and there is also no $\mu_i \in S_i$ such that $\mu_i \gg_T s_i$ and we have $s_i \in \mathrm{PGS}(T)_i$. The reasoning for MGS is analogous if we replace S_i by $\Delta(S_i)$.

MLS and PLS are not monotone. Consider the following simple game:

	X
A	$(1,0)$
B	$(0,0)$

$$\begin{aligned} \mathrm{MLS}(\{A,B\},\{X\}) = \mathrm{PLS}(\{A,B\},\{X\}) = (\{A\},\{X\}) \text{ and} \\ \mathrm{MLS}(\{B\},\{X\}) = \mathrm{PLS}(\{B\},\{X\}) = (\{B\},\{X\}), \end{aligned}$$

but $(\{B\},\{X\}) \not\subseteq (\{A\},\{X\})$.

Thus, none of the local operators (those which only consider dominant strategies in the current subgame) is monotone. We will see that also MG and PG are not monotone in general. The monotonicity of the global operators MGS and PGS will allow us to prove the expected inclusions $\mathrm{ML}^\infty \subseteq \mathrm{MLS}^\infty \subseteq \mathrm{PLS}^\infty$ and $\mathrm{PL}^\infty \subseteq \mathrm{PLS}^\infty$ between the local operators. To this end, we will show that the fixed points of the local and corresponding global operators coincide (although the operators are different).

Lemma 4.20. $\mathrm{MGS}^\infty = \mathrm{MLS}^\infty$ and $\mathrm{PGS}^\infty = \mathrm{PLS}^\infty$.

Proof. We will only prove $\mathrm{PGS}^\infty = \mathrm{PLS}^\infty$. Since $\mathrm{PGS}(T) \subseteq \mathrm{PLS}(T)$ for all T and PGS is monotone, we have $\mathrm{PGS}^\infty \subseteq \mathrm{PLS}^\infty$. Now we will prove by induction that $\mathrm{PLS}^\alpha \subseteq \mathrm{PGS}^\alpha$ for all α. Only the induction step $\alpha \mapsto \alpha+1$ has to be considered: Let $s_i \in \mathrm{PLS}_i^{\alpha+1}$. Therefore, $s_i \in \mathrm{PLS}_i^\alpha$ and there is no $s_i' \in \mathrm{PLS}_i^\alpha$ such that $s_i' \gg_{\mathrm{PLS}^\alpha} s_i$. Assume $s_i \notin \mathrm{PGS}_i^{\alpha+1}$, i.e.

$$A = \{s_i' \in S_i : s_i' \gg_{\mathrm{PGS}^\alpha} s_i\} \neq \emptyset$$

(note: By induction hypothesis $\mathrm{PGS}^\alpha = \mathrm{PLS}^\alpha$). Pick an $s_i^* \in A$ which is maximal with respect to $\gg_{\mathrm{PLS}^\alpha}$. Claim: $s_i^* \in \mathrm{PLS}^\alpha$. Otherwise, there exists a $\beta \leq \alpha$ and an $s_{i'} \in S_i$ with $s_i' \gg_{\mathrm{PLS}^\beta} s_{i^*}$. Since $\mathrm{PLS}^\beta \supseteq \mathrm{PLS}^\alpha$, it follows that $s_i' \gg_{\mathrm{PLS}^\alpha} s_i^* \gg_{\mathrm{PLS}^\alpha} s_i$. Therefore, $s_i' \in A$ and s_i^* is not maximal with respect to $\gg_{\mathrm{PLS}^\alpha}$ in A. Contradiction.

But if $s_i^* \in \mathrm{PLS}^\alpha$ and $s_i^* \gg_{\mathrm{PLS}^\alpha} s_i$, then $s_i \notin \mathrm{PLS}^{\alpha+1}$ which again constitutes a contradiction.

The reasoning for MGS^∞ and MLS^∞ is analogous. Q.E.D.

Corollary 4.21. $\mathrm{MLS}^\infty \subseteq \mathrm{PLS}^\infty$.

Lemma 4.22. $\mathrm{MG}^\infty = \mathrm{ML}^\infty$ and $\mathrm{PG}^\infty = \mathrm{PL}^\infty$.

Proof. We will only prove $\mathrm{PG}^\infty = \mathrm{PL}^\infty$ by proving $\mathrm{PG}^\alpha = \mathrm{PL}^\alpha$ for all α by induction. Let $\mathrm{PG}^\alpha = \mathrm{PL}^\alpha$ and $s_i \in \mathrm{PG}_i^{\alpha+1}$. Then $s_i \in \mathrm{PG}_i^\alpha = \mathrm{PL}_i^\alpha$ and hence there is no $s_i' \in S_i$ such that $s_i' >_{\mathrm{PG}^\alpha} s_i$. Thus, there is no $s_i' \in \mathrm{PL}_i^\alpha$ such that $s_i' >_{\mathrm{PL}^\alpha} s_i$ and $s_i \in \mathrm{PL}^{\alpha+1}$. So, $\mathrm{PG}^{\alpha+1} \subseteq \mathrm{PL}^{\alpha+1}$.

Now, let $s_i \in \mathrm{PL}_i^{\alpha+1}$. Again we have $s_i \in \mathrm{PL}_i^\alpha = \mathrm{PG}_i^\alpha$. Assume $s_i \notin \mathrm{PG}_i^{\alpha+1}$. Then

$$A = \{s_i' \in S_i : s_i' >_{\mathrm{PL}^\alpha} s_i\} \neq \emptyset.$$

For every $\beta \leq \alpha$ let $A^\beta = A \cap \mathrm{PL}_i^\beta$. Pick the maximal β such that $A^\beta \neq \emptyset$ and a $s_i^* \in A^\beta$ which is maximal with respect to $>_{\mathrm{PL}^\beta}$.

Claim: $\beta = \alpha$. Otherwise, $s_i \notin \mathrm{PL}_i^{\beta+1}$. Then there exists an $s_i' \in \mathrm{PL}_i^\beta$ with $s_i' >_{\mathrm{PL}^\beta} s_i^*$. Since $\mathrm{PL}^\beta \supseteq \mathrm{PL}^\alpha$ and $s_i^* >_{\mathrm{PL}^\alpha} s_i$, we have $s_i' >_{\mathrm{PL}^\alpha} s_i$, i.e. $s_i' \in A^\beta$ which contradicts the choice of s_i^*. Therefore, $s_i^* \in \mathrm{PL}_i^\alpha$. Since $s_i^* >_{\mathrm{PL}^\alpha} s_i$, we have $s_i \notin \mathrm{PL}_i^{\alpha+1}$. Contradiction, hence the assumption is wrong, and we have $s_i \in \mathrm{PG}^{\alpha+1}$. Altogether $\mathrm{PG}^\alpha = \mathrm{PL}^\alpha$. Again, the reasoning for $\mathrm{MG}^\infty = \mathrm{ML}^\infty$ is analogous. Q.E.D.

Corollary 4.23. $\mathrm{PL}^\infty \subseteq \mathrm{PLS}^\infty$ and $\mathrm{ML}^\infty \subseteq \mathrm{MLS}^\infty$.

Proof. We have $\mathrm{PL}^\infty = \mathrm{PG}^\infty \subseteq \mathrm{PGS}^\infty = \mathrm{PLS}^\infty$ where the inclusion $\mathrm{PG}^\infty \subseteq \mathrm{PGS}^\infty$ holds because $\mathrm{PG}(T) \subseteq \mathrm{PGS}(T)$ for any T and PGS is monotone. Analogously, we have $\mathrm{ML}^\infty = \mathrm{MG}^\infty \subseteq \mathrm{MGS}^\infty = \mathrm{MLS}^\infty$. Q.E.D.

This implies that MG and PG cannot be monotone. Otherwise, we would have $\mathrm{ML}^\infty = \mathrm{PL}^\infty$. But we know that this is wrong.

4.3 Beliefs and Rationalisability

Let $\Gamma = (N, (S_i)_{i\in N}, (p_i)_{i\in N})$ be a game. A *belief* of Player i is a probability distribution over S_{-i}.

Remark 4.24. A belief is not necessarily a product of independent probability distributions over the individual S_j ($j \neq i$). A player may believe that the other players play correlated.

A strategy $s_i \in S_i$ is called a *best response to a belief* $\gamma \in \Delta(S_{-i})$ if $\widehat{p}_i(s_i, \gamma) \geq \widehat{p}_i(s_i', \gamma)$ for all $s_i' \in S_i$. Conversely, $s_i \in S_i$ is *never a best response* if s_i is not a best response for any $\gamma \in \Delta(S_{-i})$.

Lemma 4.25. For every game $\Gamma = (N, (S_i)_{i\in N}, (p_i)_{i\in N})$ and every $s_i \in S_i$, s_i is never a best response if and only if there exists a mixed strategy $\mu_i \in \Delta(S_i)$ such that $\mu_i \gg_S s_i$.

Proof. If $\mu_i \gg_S s_i$, then $\widehat{p}_i(\mu_i, s_{-i}) > \widehat{p}_i(s_i, s_{-i})$ for all $s_{-i} \in S_{-i}$. Thus, $\widehat{p}_i(\mu_i, \gamma) > \widehat{p}_i(s_i, \gamma)$ for all $\gamma \in \Delta(S_{-i})$. Then, for every belief $\gamma \in \Delta(S_{-i})$, there exists an $s_i' \in \mathrm{supp}(\mu_i)$ such that $\widehat{p}_i(s_i', \gamma) > \widehat{p}_i(s_i, \gamma)$. Therefore, s_i is never a best response.

Conversely, let $s_i^* \in S_i$ be never a best response in Γ. We define a two-person zero-sum game $\Gamma' = (\{0,1\}, (T_0, T_1), (p, -p))$ where $T_0 = S_i - \{s_i^*\}$, $T_1 = S_{-i}$ and $p(s_i, s_{-i}) = p_i(s_i, s_{-i}) - p_i(s_i^*, s_{-i})$. Since s_i^* is never a best response, for every mixed strategy $\mu_1 \in \Delta(T_1) = \Delta(S_{-i})$ there is a strategy $s_0 \in T_0 = S_i - \{s_i^*\}$ such that $\widehat{p}_i(s_0, \mu_1) > \widehat{p}_i(s_i^*, \mu_1)$ (in Γ), i.e. $p(s_0, \mu_1) > 0$ (in Γ'). So, in Γ'

$$\min_{\mu_1 \in \Delta(T_1)} \max_{s_0 \in T_0} p(s_0, \mu_1) > 0,$$

and therefore

$$\min_{\mu_1 \in \Delta(T_1)} \max_{\mu_0 \in \Delta(T_0)} p(\mu_0, \mu_1) > 0.$$

By Nash's Theorem, there is a Nash equilibrium (μ_0^*, μ_1^*) in Γ'. By von Neumann and Morgenstern we have

$$\begin{aligned} \min_{\mu_1 \in \Delta(T_1)} \max_{s_0 \in \Delta(T_0)} p(\mu_0, \mu_1) &= p(\mu_0^*, \mu_1^*) \\ &= \max_{s_0 \in \Delta(T_0)} \min_{\mu_1 \in \Delta(T_1)} p(\mu_0, \mu_1) > 0. \end{aligned}$$

Thus, $0 < p(\mu_0^*, \mu_1^*) \leq p(\mu_0^*, \mu_1)$ for all $\mu_1 \in \Delta(T_1) = \Delta(S_{-i})$. So, we have in Γ $\widehat{p}_i(\mu_0^*, s_{-i}) > p_i(s_i^*, s_{-i})$ for all $s_{-i} \in S_{-i}$ which means $\mu_0^* \gg_S s_i^*$. Q.E.D.

Definition 4.26. Let $\Gamma = (N, (S_i)_{i \in N}, (p_i)_{i \in N})$ be a game. A strategy $s_i \in S_i$ is *rationalisable* in Γ if for any Player j there exists a set $T_j \subseteq S_j$ such that

- $s_i \in T_i$, and
- every $s_j \in T_j$ (for all j) is a best response to a belief $\gamma_j \in \Delta(S_{-j})$ where $\text{supp}(\gamma_j) \subseteq T_{-j}$.

Theorem 4.27. For every finite game Γ we have: s_i is rationalisable if and only if $s_i \in \text{MLS}_i^\infty$. This means, the rationalisable strategies are exactly those surviving iterated elimination of strategies that are strictly dominated by mixed strategies.

Proof. Let $s_i \in S_i$ be rationalisable by $T = (T_1, \ldots, T_n)$. We show $T \subseteq \text{MLS}^\infty$. We will use the monotonicity of MGS and the fact that $\text{MLS}^\infty = \text{MGS}^\infty$. This implies $\text{MGS}^\infty = \text{gfp}(\text{MGS})$ and hence, MGS^∞ contains all other fixed points. It remains to show that $\text{MGS}(T) = T$. Every $s_j \in T_j$ is a best response (among the strategies in S_j) to a belief γ with $\text{supp}(\gamma) \subseteq T_{-j}$. This means that there exists no mixed strategy $\mu_j \in \Delta(S_j)$ such that $\mu_j \gg_T s_j$. Therefore, s_j is not eliminated by MGS: $\text{MGS}(T) = T$.

Conversely, we have to show that every strategy $s_i \in \text{MLS}_i^\infty$ is rationalisable by MLS^∞. Since $\text{MLS}^\infty = \text{MGS}^\infty$, we have $\text{MGS}(\text{MLS}^\infty) = \text{MLS}^\infty$. Thus, for every $s_i \in \text{MLS}_i^\infty$ there is no mixed strategy $\mu_i \in \Delta(S_i)$ such that $\mu_i \gg_{\text{MLS}^\infty} s_i$. So, s_i is a best response to a belief in MLS_i^∞. Q.E.D.

Intuitively, the concept of rationalisability is based on the idea that every player keeps those strategies that are a best response to a possible combined rational action of his opponents. As the following example shows, it is essential to also consider correlated actions of the players.

Example 4.28. Consider the following cooperative game in which every player receives the same payoff:

	L	R	L	R	L	R	L	R
T	8	0	4	0	0	0	3	3
B	0	0	0	4	0	8	3	3
	1		2		3		4	

Matrix 2 is not strictly dominated. Otherwise there were $p, q \in [0, 1]$ with $p + q \leq 1$ and

$$8 \cdot p + 3 \cdot (1 - p - q) > 4 \text{ and}$$
$$8 \cdot q + 3 \cdot (1 - p - q) > 4.$$

This implies $2 \cdot (p + q) + 6 > 8$, i.e. $2 \cdot (p + q) > 2$, which is impossible.

So, matrix 2 must be a best response to a belief $\gamma \in \Delta(\{T, B\} \times \{L, R\})$. Indeed, the best responses to $\gamma = \frac{1}{2} \cdot ((T, L) + (B, R))$ are matrices 1, 2 or 3.

On the other hand, matrix 2 is not a best response to a belief of independent actions $\gamma \in \Delta(\{T, B\}) \times \Delta(\{L, R\})$. Otherwise, if matrix 2 was be a best response to $\gamma = (p \cdot T + (1 - p) \cdot B, q \cdot L + (1 - q) \cdot R)$, we would have that

$$4pq + 4 \cdot (1 - p) \cdot (1 - q) \geq \max\{8pq, 8 \cdot (1 - p) \cdot (1 - q), 3\}.$$

We can simplify the left side: $4pq + 4 \cdot (1 - p) \cdot (1 - q) = 8pq - 4p - 4q + 4$. Obviously, this term has to be greater than each of the terms from which we chose the maximum:

$$8pq - 4p - 4q + 4 \geq 8pq \Rightarrow p + q \geq 1$$

and

$$8pq - 4p - 4q + 4 \geq 8 \cdot (1-p) \cdot (1-q) \Rightarrow p + q \leq 1.$$

So we have $p + q = 1$, or $q = 1 - p$. But this allows us to substitute q by $p - 1$, and we get

$$8pq - 4p - 4q + 4 = 8p \cdot (1-p).$$

However, this term must still be greater or equal than 3, so we get

$$\begin{aligned} & 8p \cdot (1-p) \geq 3 \\ \Leftrightarrow\ & p \cdot (1-p) \geq \frac{3}{8}, \end{aligned}$$

which is impossible since $\max(p \cdot (1-p)) = \frac{1}{4}$ (see Figure 4.2).

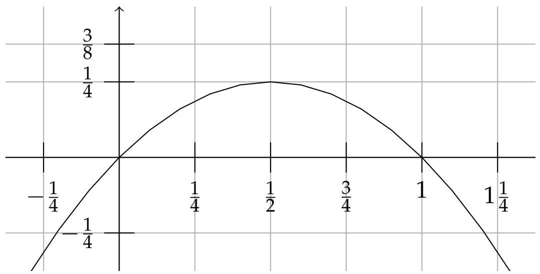


Figure 4.2. Graph of the function $p \mapsto p \cdot (1-p)$

4.4 Games in Extensive Form

A *game in extensive form* (with perfect information) is described by a game tree. For two-person games this is a special case of the games on graphs which we considered in the earlier chapters. The generalisation to n-person games is obvious: $\mathcal{G} = (V, V_1, \ldots, V_n, E, p_1, \ldots, p_n)$ where

(V, E) is a directed tree (with root node w), $V = V_1 \uplus \cdots \uplus V_n$, and the payoff function $p_i : \text{Plays}(\mathcal{G}) \to \mathbb{R}$ for Player i, where $\text{Plays}(\mathcal{G})$ is the set of paths through (V, E) beginning in the root node, which are either infinite or end in a terminal node.

A strategy for Player i in $\mathcal{G}$ is a function $f : \{v \in V_i : vE \neq \emptyset\} \to V$ such that $f(v) \in vE$. S_i is the set of all strategies for Player i. If all players $1, \ldots, n$ each fix a strategy $f_i \in S_i$, then this defines a unique play $f_1\hat{}\cdots\hat{}f_n \in \text{Plays}(\mathcal{G})$.

We say that $\mathcal{G}$ has *finite horizon* if the depth of the game tree (the length of the plays) is finite.

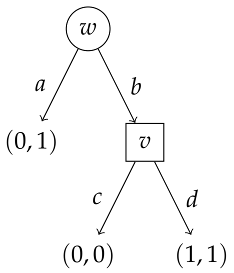
For every game $\mathcal{G}$ in extensive form, we can construct a game $S(\mathcal{G}) = (N, (S_i)_{i \in N}, (p_i)_{i \in N})$ with $N = \{1, \ldots, n\}$ and $p_i(f_1, \ldots, f_n) = p_i(f_1\hat{}\cdots\hat{}f_n)$. Hence, we can apply all solution concepts for strategic games (Nash equilibria, iterated elimination of dominated strategies, etc.) to games in extensive form. First, we will discuss Nash equilibria in extensive games.

Example 4.29. Consider the game $\mathcal{G}$ (of finite horizon) depicted in Figure 4.3 presented as (a) an extensive-form game and as (b) a strategic-form game. The game has two Nash equilibria:

- The natural solution (b, d) where both players win.
- The second solution (a, c) which seems to be irrational since both players pick an action with which they lose.

What seems irrational about the second solution is the following observation. If Player 0 picks a, it does not matter which strategy her opponent chooses since the position v is never reached. Certainly, if Player 0 switches from a to b, and Player 1 still responds with c, the payoff of Player 0 does not increase. But this threat is not credible since if v is reached after action a, then action d is better for Player 1 than c. Hence, Player 0 has an incentive to switch from a to b.

This example shows that the solution concept of Nash equilibria is not sufficient for games in extensive form since they do not take the sequential structure into account. Before we introduce a stronger notion of equilibrium, we will need some more notation: Let $\mathcal{G}$ be a game in extensive form and v a position of $\mathcal{G}$. $\mathcal{G}\restriction_v$ denotes the *subgame* of $\mathcal{G}$



(a) extensive form

	c	d
a	$(0,1)$	$(0,1)$
b	$(0,0)$	$(1,1)$

(b) strategic form

Figure 4.3. A game of finite horizon

beginning in v (defined by the subtree of $\mathcal{G}$ rooted at v). Payoffs: Let h_v be the unique path from w to v in $\mathcal{G}$. Then $p_i^{\mathcal{G}\restriction_v}(\pi) = p_i^{\mathcal{G}}(h_v \cdot \pi)$. For every strategy f of Player i in $\mathcal{G}$ let $f\restriction_v$ be the restriction of f to $\mathcal{G}\restriction_v$.

Definition 4.30. A *subgame perfect equilibrium* of $\mathcal{G}$ is a strategy profile $(f_1, \dots, f_n)$ such that, for every position v, $(f_1\restriction_v, \dots, f_n\restriction_v)$ is a Nash equilibrium of $\mathcal{G}\restriction_v$. In particular, $(f_1, \dots, f_n)$ itself is a Nash equilibrium.

In the example above, only the natural solution (b, d) is a subgame perfect equilibrium. The second Nash equilibrium (a, c) is not a subgame perfect equilibrium since $(a\restriction_v, c\restriction_v)$ is not a Nash equilibrium in $\mathcal{G}\restriction_v$.

Let $\mathcal{G}$ be a game in extensive form, $f = (f_1, \dots, f_n)$ be a strategy profile, and v a position in $\mathcal{G}$. We denote by $\widetilde{f}(v)$ the play in $\mathcal{G}\restriction_v$ that is uniquely determined by $f_1 \dots, f_n$.

Lemma 4.31. Let $\mathcal{G}$ be a game in extensive form with finite horizon. A strategy profile $f = (f_1, \dots, f_n)$ is a subgame perfect equilibrium of $\mathcal{G}$ if and only if for every Player i, every $v \in V_i$, and every $w \in vE$: $p_i(\widetilde{f}(v)) \geq p_i(\widetilde{f}(w))$.

Proof. Let f be a subgame perfect equilibrium. If $p_i(\widetilde{f}(w)) > p_i(\widetilde{f}(v))$ for some $v \in V_i$, $w \in vE$, then it would be better for Player i in $\mathcal{G}\restriction_v$ to

change her strategy in v from f_i to f_i' with

$$f_i'(u) = \begin{cases} f_i(u) & \text{if } u \neq v \\ w & \text{if } u = w\,. \end{cases}$$

This is a contradiction.

Conversely, if f is not a subgame perfect equilibrium, then there is a Player i, a position $v_0 \in V_i$ and a strategy $f_i' \neq f_i$ such that it is better for Player i in $\mathcal{G}\restriction_{v_0}$ to switch from f_i to f_i' against f_{-i}. Let $g := (f_i', f_{-i})$. We have $q := p_i(\widetilde{g}(v_0)) > p_i(\widetilde{f}(v_0))$. We consider the path $\widetilde{g}(v_0) = v_0 \dots v_t$ and pick a maximal $m < t$ with $p_i(\widetilde{g}(v_0)) > p_i(\widetilde{f}(v_m))$. Choose $v = v_m$ and $w = v_{m+1} \in vE$. Claim: $p_i(\widetilde{f}(v)) < p_i(\widetilde{f}(w))$ (see Figure 4.4):

$$p_i(\widetilde{f}(v)) = p_i(\widetilde{f}(v_m)) < p_i(\widetilde{g}(v_m)) = q$$

$$p_i(\widetilde{f}(w)) = p_i(\widetilde{f}(v_{m+1})) \geq p_i(\widetilde{g}(v_{m+1})) = q$$

Q.E.D.

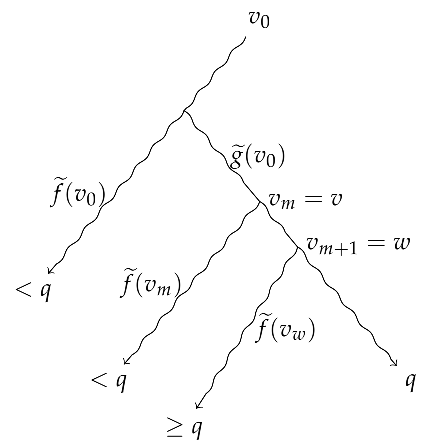


Figure 4.4. $p_i(\widetilde{f}(v)) < p_i(\widetilde{f}(w))$

If f is not a subgame perfect equilibrium, then we find a subgame $\mathcal{G}\restriction_v$ such that there is a profitable deviation from f_i in $\mathcal{G}\restriction_v$, which only differs from f_i in the first move.

In extensive games with finite horizon we can directly define the payoff at the terminal nodes (the leaves of the game tree). We obtain a payoff function $p_i : T \to \mathbb{R}$ for $i = 1, \ldots, n$ where $T = \{v \in V : vE = \emptyset\}$.

Backwards induction: For finite games in extensive form we define a strategy profile $f = (f_1, \ldots, f_n)$ and values $u_i(v)$ for all positions v and every Player i by backwards induction:

- For terminal nodes $t \in T$ we do not need to define f, and $u_i(t) := p_i(t)$.
- Let $v \in V \setminus T$ such that all $u_i(w)$ for all i and all $w \in vE$ are already defined. For i with $v \in V_i$ define $f_i(v) = w$ for some w with $u_i(w) = \max\{u_i(w') : w' \in vE\}$ and $u_j(v) := u_j(f_i(v))$ for all j.

We have $p_i(\widetilde{f}(v)) = u_i(v)$ for every i and every v.

Theorem 4.32. The strategy profile defined by backwards induction is a subgame perfect equilibrium.

Proof. Let $f_i' \neq f_i$. Then there is a node $v_0 \in V_i$ with minimal height in the game tree such that $f_i'(v) \neq f_i(v)$. Especially, for every $w \in vE$, $\widetilde{(f_i', f_{-i})}(w) = \widetilde{f}(w)$. For $w = f_i'(v)$ we have

$$\begin{aligned} p_i(\widetilde{(f_i', f_{-i})}(v)) &= p_i(\widetilde{(f_i', f_{-i})}(w)) \\ &= p_i(\widetilde{f}(w)) \\ &= u_i(w) \leq \max_{w' \in vE} \{u_i(w')\} \\ &= u_i(v) \\ &= p_i(\widetilde{f}(v)). \end{aligned}$$

Therefore, $f \restriction_v$ is a Nash equilibrium in $\mathcal{G} \restriction_v$. Q.E.D.

Corollary 4.33. Every finite game in extensive form has a subgame perfect equilibrium (and thus a Nash equilibrium) in pure strategies.