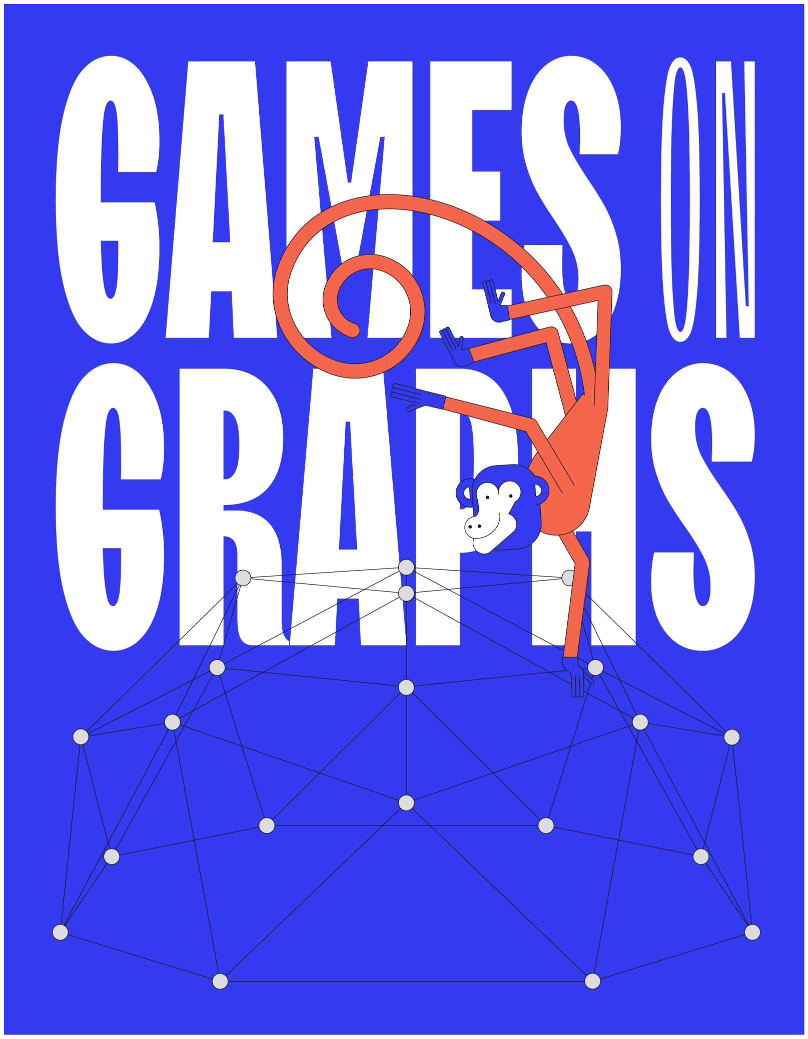
Games on graphs

Nathalie Bertrand, Romain Brenguier, Patricia Bouyer-Decitre, Arr

Feb 04, 2021

INTRODUCTION

That's the entry point to the book. Maybe a little logo?



CHAPTER

ONE

INTRODUCTION

Intro answers natural questions such as: what is the book about, at whom it is aimed, and how to read it. *Simple* defines a first model of games, which is the common denominator of (almost all) models studied in this book.

We introduce the computational models that we use in Computation and briefly define linear programming. In Conditions we list the main objectives appearing in all chapters. A few notions will be useful throughout the book: they are developed in this chapter. We start with the notion of automata, discussed in Automata, and then memory for strategies, in Memory. We then show how automata and memory structures can be used to construct reductions between games in Reductions. We introduce in Subgames the notions of subgames and traps.

The notion of fixed point algorithms is central to the study of games. We first recall the main two methods for proving the existence and computing fixed points in Fixed points. We then give an overview of two prominent families of fixed point algorithms for games: value iteration algorithms in Value iteration algorithms and strategy improvement algorithms in Strategy improvement algorithms.

1.1 Simple

The first model we define is the common denominator of most models studied in this book:

- 2-player,
- zero sum,
- turn based,
- perfect information game.

1.1.1 Players

The term 2-*player* means that there are two players, Eve and Adam. Many, many different names have been used: Player 0 and Player 1, Player I and Player II as in descriptive complexity, Éloïse and Abélard, Circle and Square, corresponding to the graphical representation, Even and Odd, mostly for parity objectives, Player and Opponent, Pathfinder and Verifier in the context of automata, Max and Min, which makes sense for quantitative objectives, and this is only a very partial list of names they have been given. In the names Eve and Adam, the first letters refer to $\exists$ and $\forall$ suggesting a duality between them. We will make use of their gender to distinguish between them, so we speak of her or his strategy.

We speak of 1-player games when there is only one player. In the context of stochastic games, we refer to random as a third player, and more precisely as half a player. Hence a 2 1/2-player game is a stochastic game with two players, and a 1 1/2-player game is a stochastic game with one player.

The situation where there are more than two players is called multiplayer games.

1.1.2 Graphs

A (directed) graph is given by a set V of vertices and a set $E \subseteq V \times V$ of edges. For an edge $e = (v, v')$ we write (e) for the incoming vertex v and (e) for the outgoing vertex v': we say that e is an outgoing edge of v and an incoming edge to v'.

A path $\pi = v_0 v_1 \cdots$ is a non empty finite or infinite sequence of consecutive vertices: for all i we have $(v_i, v_{i+1}) \in E$. We let (π) denote the first vertex occurring in π and (π) the last one if π is finite. We say that π starts from (π) and if π is finite π ends in (π). We sometimes talk of a path and let the context determine whether it is finite or infinite.

We let $(G) \subseteq V^+$ denote the set of finite paths in the graph G, sometimes omitting G when clear from the context. To restrict to paths starting from v we write (G, v). The set of infinite paths is $_\omega(G) \subseteq V^\omega$, and $_\omega(G, v)$ for those starting from v.

We use the standard terminology of graphs: for instance a vertex v' is a successor of v if $(v, v') \in E$, and then v is a predecessor of v', a vertex v' is reachable from v if there exists a path starting from v and ending in v', the outdegree of a vertex is its number of outgoing edges, the indegree is its number of incoming edges, a simple path is a path with no repetitions of vertices, a cycle is a path whose first and last vertex coincide, it is a simple cycle if it does not strictly contain another cycle, a self loop is an edge from a vertex to itself, and a sink is a vertex with only a self loop as outgoing edge.

1.1.3 Arenas

The arena is the place where the game is played, they have also been called game structures or game graphs.

In the turn based setting we define here, the set of vertices is divided into vertices controlled by each player. Since we are for now interested in 2-*player* games, we have $V = \uplus$, where is the set of vertices controlled by Eve and the set of vertices controlled by Adam. We represent vertices in by circles, and vertices in by squares, and also say that $v \in$ belongs to Eve, or that Eve owns or controls v, and similarly for Adam. An arena is given by a graph and the sets and . In the context of games, vertices are often referred to as positions.

The adjective "*finite*" means that the arena is finite, *i.e.* there are finitely many vertices (hence finitely many edges). We oppose "*deterministic*" to "*stochastic*": in the first definition we are giving here, there is no stochastic aspect in the game. An important assumption, called "*perfect information*", says that the players see everything about how the game is played out, in particular they see the other player's moves.

Our definition of an arena does not include the initial vertex.

We assume that all vertices have an outgoing edge. This is for technical convenience, as it implies that we do not need to explain what happens when a play cannot be prolonged.

1.1.4 Playing

The interaction between the two players consists in moving a token on the vertices of the arena. The token is initially on some vertex. When the token is in some vertex v, the player who controls the vertex chooses an outgoing edge e of v and pushes the token along this edge to the next vertex $(e) = v'$. The outcome of this interaction is the sequence of vertices traversed by the token: it is a path. In the context of games a path is also called a play and as for paths usually written . We note that plays can be finite (but non empty) or infinite.

1.1.5 Strategies

The most important notion in this book is that of *strategies* (sometimes called policies). A strategy for a player is a full description of his or her moves in all situations. Formally, a strategy is a function mapping finite plays to edges:

$$\sigma :\to E.$$

We use σ for strategies of Eve and τ for strategies of Adam so when considering a strategy σ it is implicitly for Eve, and similarly τ is implicitly a strategy for Adam.

We say that a play $= v_0 v_1 \dots$ is consistent with a strategy σ of Eve if for all i such that $v_i \in$ we have $\sigma(_{\leq i}) = (v_i, v_{i+1})$. The definition is easily adapted for strategies of Adam.

Once an initial vertex v and two strategies σ and τ have been fixed, there exists a unique infinite play starting from v and consistent with both strategies written $\pi^v_{\sigma,\tau}$. Note that the fact that it is infinite follows from our assumption that all vertices have an outgoing edge.

1.1.6 Conditions

The last ingredient to wrap up the definitions is (winning) conditions, which is what Eve wants to achieve. There are two types of conditions: the *qualitative*, or Boolean ones, and the *quantitative* ones.

A qualitative condition is $W \subseteq_\omega$: it separates winning from losing plays, in other words a play which belongs to W is winning and otherwise it is losing. We also say that the play satisfies W. In the zero sum context a play which is losing for Eve is winning for Adam, so Adam's condition is $_\omega \setminus W$.

A quantitative condition is $f :_\omega \to$: it assigns a real value (or plus or minus infinity) to a play, which can be thought of as a payoff or a score. In the zero sum context Eve wants to maximise while Adam wants to minimise the outcome.

Often we define W as a subset of V^ω and f as $f : V^\omega \to$, since $_\omega$ is included in V^ω.

1.1.7 Objectives

To reason about classes of games with the same conditions, we introduce the notions of objectives and colouring functions. An objective and a colouring function together induce a condition. The main point is that *objectives are independent of the arenas*, so we can speak of the class of conditions induced by a given objective, and by extension a class of games induced by a given objective, for instance parity games.

We fix a set C of colours. A qualitative objective is $\Omega \subseteq C^\omega$, and a quantitative objective is a function $\Phi : C^\omega \to$.

The link between an arena and an objective is given by a colouring function $: V \to C$ labelling vertices of the graph by colours. We extend componentwise to induce $:_\omega \to C^\omega$ mapping plays to sequences of colours: $(v_0 v_1 \dots) = (v_0)\,(v_1) \dots$

A qualitative objective Ω and a colouring function induce a qualitative condition $\Omega[]$ defined by:

$$\Omega[] = \in_\omega : () \in \Omega.$$

When is clear from the context we sometimes say that a play satisfies Ω but the intended meaning is that satisfies $\Omega[]$, equivalently that $() \in \Omega$.

Similarly, a quantitative objective $\Phi : C^\omega \to$ and a colouring function induce a quantitative condition $\Phi[] :_\omega \to$ defined by:

$$\Phi[]() = \Phi(()).$$

In our definition the colouring function labels vertices. Another more general definition would label edges, and yet another relaxation would be to allow partial functions, meaning that some vertices (or edges) are not labelled by a colour. In most cases the variants are all (in some sense) equivalent; whenever we use a different definition we will make it explicit by referring for instance to edge colouring functions or partial colouring functions.

1.1.8 Games

We can now give the following definitions.

- A graph is a tuple $G = (V, E)$ where V is a set of vertices and E is a set of edges.
- An arena is a tuple $= (G, ,)$ where G is a graph over the set of vertices V and $V = \uplus$.
- A colouring function is a function $: V \to C$ where C is a set of colours.
- A qualitative condition is $W \subseteq_\omega$.
- A qualitative objective is a subset $\Omega \subseteq C^\omega$. A colouring function and a qualitative objective Ω induce a qualitative condition $\Omega[]$.
- A qualitative game is a tuple $(, W)$ where is an arena and W a qualitative condition.
- A quantitative condition is $f :_\omega \to$.
- A quantitative objective Φ is a function $\Phi : C^\omega \to$. A colouring function and a quantitative objective Φ induce a quantitative condition $\Phi[]$.
- A quantitative game is a tuple $(, f)$ where is an arena and f a quantitative condition.

To be specific, the definition above is for 2-player zero sum turn based perfect information games. As a convention we use the condition to qualify games, so for instance "parity games" are games equipped with a parity condition. This extends to graphs: we speak of a "graph with condition W" for a graph equipped with a condition W, and for instance a "mean payoff graph" if W is a mean payoff condition.

We often introduce notations implicitly: for instance when we introduce a qualitative game $\Game$ without specifying the arena and the condition, it is understood that the arena is and the condition W.

We always implicitly take the point of view of Eve. Since we consider zero sum games we can easily reverse the point of view by considering the qualitative game $(,_\omega \setminus W)$ and the qualitative game $(, -f)$. Indeed for the latter Adam wants to minimise f, which is equivalent to maximising $-f$. The term zero sum comes from this: the total outcome for the two players is $f + (-f)$, meaning zero.

Unless otherwise stated we assume that graphs are finite, meaning that there are finitely many vertices (hence finitely many edges). We equivalently say that the arena or the game is finite. will study games over infinite graphs.

1.1.9 Winning in qualitative games

Now that we have the definitions of a game we can ask the main question: given a game and a vertex v, who wins from v?

Let be a qualitative game and v a vertex. A strategy σ for Eve is called winning from v if every play starting from v consistent with σ is winning, *i.e.* satisfies W. Another common terminology is that σ ensures W. In that case we say that Eve has a winning strategy from v in , or equivalently that Eve wins from v. This vocabulary also applies to Adam: for instance a strategy τ for Adam is called winning from v if every play starting from v consistent with τ is losing, *i.e.* does not satisfy W.

We let $()$ denote the set of vertices v such that Eve wins from v, it is called winning region, or sometimes winning set. A vertex in $()$ is said winning for Eve. The analogous notation for Adam is $()$.

We say that a strategy is optimal if it is winning from all vertices in $()$.

Lemma

For all qualitative games we have $() \cap () = \emptyset$.

Proof

Assume for the sake of contradiction that both players have a winning strategy from v, then $\pi^v_{\sigma,\tau}$ would both satisfy W and not satisfy W, a contradiction.

It is however not clear that for every vertex v, *some* player has a winning strategy from v, which symbolically reads $() \cup () = V$. One might imagine that if Eve picks a strategy, then Adam can produce a counter strategy beating Eve's strategy, and vice versa, if Adam picks a strategy, then Eve can come up with a strategy winning against Adam's strategy. A typical example would be rock-paper-scissors (note that this is a concurrent game, meaning the two players play simultanously, hence it does not fit in the definitions given so far), where neither player has a winning strategy.

Whenever $() \cup () = V$, we say that the game is determined. Being determined can be understood as follows: the outcome can be determined before playing assuming both players play optimally since one of them can ensure to win whatever is the strategy of the opponent.

Theorem

Qualitative games with Borel conditions are determined.

The definition of Borel sets goes beyond the scope of this book. Suffice to say that all conditions studied in this book are (very simple) examples of Borel sets, implying that our qualitative games are all determined (as long as we consider perfect infomation and turn based games, the situation will change with more general models of games).

1.1.10 Computational problems for qualitative games

We identify three computational problems. The first is that of solving a game, which is the simplest one and since it induces a decision problem, allows us to make complexity theoretic statements.

For qualitative games, "solving the game" means solving the following decision problem:

Solving a qualitative game

A "qualitative game" and a vertex v

Does Eve win from v?

The second problem extends the previous one: most algorithms solve games for all vertices at once instead of only for the given initial vertex. This is called computing the winning regions.

For qualitative games, "computing the winning regions" means solving the following computational task:

The third problem is constructing a winning strategy.

For qualitative games, "constructing a winning strategy" means solving the following computational task:

We did not specify how the winning regions or the winning strategies are represented, this will depend on the types of games we consider.

1.1.11 Values in quantitative games

Let be a quantitative game and v a vertex. Given $x \in$ called a threshold, we say that a strategy σ for Eve ensures x from v if every play π starting from v consistent with σ has value at least x under f, *i.e.* $f() \geq x$. In that case we say that Eve has a strategy ensuring x in from v.

Note that by doing so we are actually considering a qualitative game in disguise, where the qualitative condition is the set of plays having value at least x under f. Formally, a quantitative condition f and a threshold x induce a qualitative condition

$$f_{\geq x} = \in_\omega \mid f() \geq x.$$

Analogously, we say that a strategy τ for Adam ensures x from v if every play starting from v consistent with τ has value at most x under f, *i.e.* $f() \leq x$.

We let (v) denote the quantity

$$\sup_\sigma \inf_\tau f(^v_{\sigma,\tau}),$$

where σ ranges over all strategies of Eve and τ over all strategies of Adam. We also write $^\sigma(v) = \inf_\tau f(^v_{\sigma,\tau})$ so that $(v) = \sup^\sigma_\sigma(v)$. This is called the value of Eve in the game from v, and represents the best outcome that she can ensure against any strategy of Adam. Note that (v) is either a real number, ∞, or $-\infty$.

A strategy σ such that $^\sigma(v) = (v)$ is called optimal from v, and it is simply optimal if the equality holds for all vertices. Equivalently, σ is optimal from v if for every play consistent with σ starting from v we have $f() \geq (v)$.

There may not exist optimal strategies which is why we introduce the following notion. For $\varepsilon > 0$, a strategy σ such that $^\sigma(v) \geq (v) - \varepsilon$ is called ε-optimal. If (v) is finite there exist ε-optimal strategies for any $\varepsilon > 0$.

Symmetrically, we let (v) denote

$$\inf_\tau \sup_\sigma f(^v_{\sigma,\tau}).$$

Lemma

For all quantitative games and vertex v we have $(v) \leq (v)$.

Proof

For any function $F : X \times Y \to$, we have

$$\sup_{x \in X} \inf_{y \in Y} F(x,y) \leq \inf_{y \in Y} \sup_{x \in X} F(x,y).$$

If this inequality is an equality, we say that the game is *determined* in v, and let (v) denote the value in the game from v and $^\sigma(v)$ for $\inf_\tau f(^v_{\sigma,\tau})$. Similarly as for the qualitative case, being determined can be understood as follows: the outcome can be determined before playing assuming both players play optimally, and in that case the outcome is the value.

We say that a quantitative objective $f : C^\omega \to$ is Borel if for all $x \in$, the qualitative objective $f_{\geq x} \subseteq C^\omega$ is a Borel set.

Theorem

Quantitative games with Borel conditions are determined, meaning that for all quantitative games we have $=$.

Proof

If $(v) = \infty$ then thanks to the inequality above $(v) = \infty$ and the equality holds. Assume $(v) = -\infty$ and let r be a real number. (The argument is actually the same as for the finite case but for the sake of clarity we treat them independently.) We consider $f_{\geq r}$. By definition, this a qualitative Borel condition, so implies that it is determined. Since Eve cannot have a winning strategy for $f_{\geq r}$, as this would contradict the definition of (v), this implies that Adam has a winning strategy for $f_{\geq r}$, meaning a strategy τ such that every play starting from v consistent with τ satisfy $f()r$. In other words, ${}^{\tau}(v) = \sup_\sigma f({}^{v}_{\sigma,\tau}) \leq r$, which implies that $(v) \leq r$. Since this is true for any real number r, this implies $(v) = -\infty$.

Let us now assume that $x = (v)$ is finite and let $\varepsilon > 0$. We consider $f_{\geq x+\varepsilon}$. By definition, this a qualitative Borel condition, so implies that it is determined. Since Eve cannot have a winning strategy for $f_{\geq x+\varepsilon}$, as this would contradict the definition of (v), this implies that Adam has a winning strategy for $f_{\geq x+\varepsilon}$, meaning a strategy τ such that every play starting from v consistent with τ satisfy $f() < x + \varepsilon$. In other words, ${}^{\tau}(v) = \sup_\sigma f({}^{v}_{\sigma,\tau}) \leq x + \varepsilon$, which implies that $(v) \leq x + \varepsilon$. Since this is true for any $\varepsilon > 0$, this implies $(v) \leq (v)$. As we have seen the converse inequality holds, implying the equality.

Note that this determinacy result does not imply the existence of optimal strategies.

1.1.12 Computational problems for quantitative games

As for qualitative games, we identify different computational problems. The first is solving the game.

For quantitative games, “solving the game” means solving the following decision problem:

Decision problem

A “qualitative game” and a vertex v

Does Eve win from v?

A very close problem is the value problem.

Decision problem

A “qualitative game” and a vertex v

Does Eve win from v?

For quantitative games, “solving the value problem” means solving the following decision problem:

The two problems of solving a game and the value problem are not quite equivalent: they become equivalent if we assume the existence of optimal strategies.

The value problem is directly related to computing the value.

For quantitative games, “computing the value” means solving the following computational task:

What computing the value means may become unclear if the value is not a rational number, making its representation complicated. Especially in this case, it may be enough to approximate the value, which is indeed what the value problem gives us: by repeatedly applying an algorithm solving the value problem one can approximate the value to any given precision, using a binary search.

Lemma

If there exists an algorithm A for solving the value problem of a class of games, then there exists an algorithm for approximating the value of games in this class within precision ε using $\log(\frac{1}{\varepsilon})$ calls to the algorithm A.

The following problem is global, in the same way as computing the winning regions.

For quantitative games, "computing the value function" means solving the following computational task:

Decision problem

A "qualitative game" and a vertex v

Does Eve win from v?

Finally, we are sometimes interested in constructing optimal strategies provided they exist.

For quantitative games, "constructing an optimal strategy" means solving the following computational task:

A close variant is to construct ε-optimal strategies, usually with ε given as input.

(1-subsec:prefix-independent-objectives)

1.1.13 Prefix independent objectives

A qualitative objective Ω is:

- closed under adding prefixes if for every finite sequence ρ and for every infinite sequence ρ', if $\rho' \in \Omega$ then $\rho\rho' \in \Omega$;
- closed under removing prefixes if for every finite sequence ρ and for every infinite sequence ρ', if $\rho\rho' \in \Omega$ then $\rho' \in \Omega$;
- prefix independent if it is closed under both adding and removing prefixes; in other words whether a sequence satisfies Ω does not depend upon finite prefixes.

Let be a finite play consistent with σ, we write $\sigma_{|}$ for the strategy defined by

$$\sigma_{|}(') = \sigma(').$$

Fact

Let ∂ be a qualitative game with objective Ω closed under removing prefixes, σ a winning strategy from v, and a finite play consistent with σ starting from v. Then $\sigma_{|}$ is winning from $v' = ()$.

Proof

Let $'$ be an infinite play consistent with $\sigma_{|}$ from v', then $'$ is an infinite play consistent with σ starting from v, implying that it is winning, and since Ω is closed under removing prefixes the play $'$ is winning. Thus $\sigma_{|}$ is winning from v'.

Let ∂ be a qualitative game with objective Ω closed under removing prefixes and σ a winning strategy from v. Then all vertices reachable from v by a play consistent with σ are winning.

In other words, when playing a winning strategy the play does not leave the winning region.

Similarly, a quantitative objective Φ is:

- monotonic under adding prefixes if for every finite sequence ρ and for every infinite sequence ρ' we have $\Phi(\rho') \leq \Phi(\rho\rho')$;
- monotonic under removing prefixes if for every finite sequence ρ and for every infinite sequence ρ' we have $\Phi(\rho') \geq \Phi(\rho\rho')$;
- prefix independent if it is monotonic under both adding and removing prefixes.

The fact above extends to quantitative objectives with the same proof.

Fact

Let $\game$ be a quantitative game with objective Φ monotonic under removing prefixes, σ a strategy ensuring x from v, and a finite play consistent with σ starting from v. Then $\sigma_{|}$ ensures x from $v' = ()$.

Proof

Let $'$ be an infinite play consistent with $\sigma_{|}$ from v', then $'$ is an infinite play consistent with σ starting from v, implying that $\Phi(') \geq x$, and since Φ is monotonic under removing prefixes this implies that $\Phi(') \geq x$. Thus $\sigma_{|}$ ensures x from v'.

Let $\game$ be a quantitative game with objective Φ monotonic under removing prefixes and σ an optimal strategy from v. Then for all vertices v' reachable from v by a play consistent with σ we have $(v) \leq (v')$.

In other words, when playing an optimal strategy the value is non-decreasing along the play.

1.2 References

The study of games, usually called game theory, has a very long history rooted in mathematics, logic, and economics, among other fields. Foundational ideas and notions emerged from set theory with for instance backward induction by Zermelo [**?**], and topology with determinacy results by Martin [**?**] (stated as \cref{1-thm:borel_determinacy} in this chapter), and Banach-Mazur and Gale-Stewart games [**?**].

The topic of this book is a small part of game theory: we focus on infinite duration games played on graphs. In this chapter we defined deterministic games, meaning games with no source of randomness, which will be the focus of \cref{part:classic}. \Cref{part:stochastic} introduces stochastic games, which were initially studied in mathematics. We refer to~\cref{6-sec:references} for more bibliographic references on stochastic games, and focus in this chapter on references for deterministic games.

The model presented in this chapter emerged from the study of automata theory and logic, where it is used as a tool for various purposes. Let us first discuss the role of games in two contexts: for solving the synthesis problem of reactive systems and for automata and logic over infinite trees.

The synthesis problem for non-terminating reactive systems, sometimes called Church's problem, was formulated in general terms by Church [**?**][**?**]: from a specification of a step-by step transformation of an input stream given in some logical formalism, construct a system satisfying the specification. The first published paper solving Church's problem for monadic second-order logic was written by Büchi and Landweber [**?**], following a paper by Landweber [**?**] (then Büchi's PhD student) focussing on solving games. However, the idea of casting the synthesis problem as a game between a system and its environment is due to McNaughton: in the technical report [**?**] McNaughton attempted to give a solution to the synthesis problem using games, initiating many of the most important ideas for analysing games. Unfortunately the proof contained an error which Landweber detected and communicated to McNaughton, who then decided to let Landweber publish his complete solution. One of the most difficult step in the solution of Church's problem for monadic second-order logic by Büchi and Landweber [**?**] is the determinisation procedure from Büchi to Muller automata due to McNaughton [?]. We refer to Thomas' survey [?] for more details on some historical and technical aspects of the early papers on Church's synthesis problem.

Games emerged in another aspect of automata theory: for understanding the difficult result of Rabin [?] saying that automata over infinite trees can be effectively complemented. This is the key step for proving Rabin's seminal result that the monadic second-order theory of the infinite binary tree is decidable. The celebrated paper of Gurevich and Harrington [?] revisits Rabin's result by reducing the complementation question to a determinacy result for games. Interestingly, they credit McNaughton for ``airing the idea" of using games in this context and then for exploiting it to Landweber [?], Büchi and Landweber [?], and Büchi [?].

Both lines of work have been highly influential in automata theory and logic; we refer to the reference section in~\cref{2-chap:regular} for more bibliographic references on this connection. They bind automata theory and logic to the study of games on graphs and provide motivations and questions many of which are still open today.

Beyond these two examples there are many applications of games in theoretical computer science and logic in particular. The following quote is due to Hodges [?]:

``An extraordinary number of basic ideas in model theory can be expressed in terms of games."

Let us mention model checking games, which are used for checking whether a model satisfies a formula. They often form both a theoretical tool for understanding the model checking problem and proving its properties, as well as an algorithmic backend for effectively deciding properties of a logical formalism (we refer to [?] for a survey on model checking games). Another important construction of a game for understanding logical properties is the Ehrenfeucht-Fraïssé games [?][?][?] whose goal is to determine whether two models are equivalent against a logical formalism.

CHAPTER

TWO

REGULAR GAMES

CHAPTER

THREE

PARITY GAMES

CHAPTER

FOUR

GAMES WITH PAYOFFS

BIBLIOGRAPHY

[Zer13] Ernst Zermelo. Uber eine anwendung der mengenlehre auf die theorie des schachspiels. In *Proceedings of the International Congress of Mathematicians, ICM'13*, volume 2, 501–504. Cambridge University Press, 1913.

[Mar75] Donald A. Martin. Borel determinacy. *Annals of Mathematics*, 102(2):363–371, 1975.

[GS53] David Gale and F.M. Stewart. Infinite games with perfect information. *Annals of Mathematical Studies*, 28:245–266, 1953.

[Chu57] Alonzo Church. Applications of recursive arithmetic to the problem of cricuit synthesis. In *Summaries of the Summer Institute of Symbolic Logic*, volume I, pages 3–50. Cornell University, 1957.

[Chu62] Alonzo Church. Logic, arithmetic, and automata. In *Proceedings of the International Congress of Mathematicians*, 23–35. 1962.

[BuchiL69] J. Richard Büchi and Lawrence H. Landweber. Solving sequential conditions by finite-state strategies. *Transactions of the American Mathematical Society*, 138:295–311, 1969.

[Lan67] Lawrence H. Landweber. Finite state games–a solvability algorithm for restricted second-order arithmetic. *Notices of the American Mathematical Society*, 14:129–130, 1967.

[McN65] Robert McNaughton. Finite-state infinite games. Technical Report, Massachusetts Institute of Technology, 1965.

[McN66] Robert McNaughton. Testing and generating infinite sequences by a finite automaton. *Information and Computation*, 9(5):521–530, 1966.

[Tho09] Wolfgang Thomas. Facets of synthesis: revisiting Church's problem. In *Proceedings of the International Conference on the Foundations of Software Science and Computational Structures, FoSSaCS'09*, 1–14. 2009.

[Rab69] Michael O. Rabin. Decidability of second-order theories and automata on infinite trees. *Transactions of the American Mathematical Society*, 141:1–35, 1969.

[GH82] Yuri Gurevich and Leo Harrington. Trees, automata, and games. In *Proceedings of the Annual ACM Symposium on Theory of Computing, STOC'82*, 60–65. ACM Press, 1982.

[Buchi77] J. Richard Büchi. Using determinancy of games to eliminate quantifiers. In *Fundamentals of Computation Theory77*, 367–378. 1977.

[Hod93] Wilfrid Hodges. *Model theory*. Volume 42 of Encyclopedia of mathematics and its applications. Cambridge University Press, 1993.

[Gradel02] Erich Grädel. Model checking games. *Electronic Notes in Theoretical Computer Science*, 67:15–34, 2002.

[Ehr61] Andrzej Ehrenfeucht. An application of games to the completeness problem for formalized theories. *Comptes-rendus de l'Académie des Sciences*, 49:129–141, 1961.

[Fraisse50] Roland Fraïssé. Sur une nouvelle classification des systèmes de relations. In *Comptes Rendus de l'Académie des Sciences*, volume 230, 1022–1024. 1950.

[Fraisse53] Roland Fraïssé. Sur quelques classifications des systèmes de relations. In *Publications Scientifiques de l'Université d'Alger*, 35–182. 1953.