The Fundamental Results on Infinite Games

Wolfgang Thomas

RWTH AACHEN

Francqui Lecture, Mons, April 2013

Winning and Losing

Church's Problem

APPLICATION OF RECURSIVE ARITHMETIC TO THE PROBLEM OF CIRCUIT SYNTHESIS

Alonzo Church

RESTRICTED RECURSIVE ARITHMETIC

Primitive symbols are individual (i.e., numerical) variables x, y, z, t,•••, singulary functional constants i_1, i_2,•••,i_μ, the individual constant 0, the accent ' as a notation for successor (of a number), the notation ( ) for application of a singulary function to its argument, connectives of the propositional calculus, and brackets [ ].

Axioms are all tautologous wffs. Rules are modus ponens; substitution for individual variables; mathematical induction,

$$\text{from } P \supset S^{a}_{a'}P| \quad \text{and} \quad S^{a}_{0}P| \quad \text{to infer } P;$$

and any one of several alternative recursion schemata or sets of recursion schemata.

A Citation

Alonzo Church

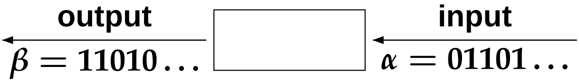
at the "Summer Institute of Symbolic Logic"

Cornell University, 1957:

"Given a requirement which a circuit is to satisfy, we may suppose the requirement expressed in some suitable logistic system which is an extension of restricted recursive arithmetic. The *synthesis problem* is then to find recursion equivalences representing a circuit that satisfies the given requirement (or alternatively, to determine that there is no such circuit)."

(By "circuits", Church means finite automata with output.)

Requirements as Winning Conditions



Player 1: $a_0 \quad a_1 \quad a_2 \quad a_3 \cdots = \alpha$
Player 2: $b_0 \quad b_1 \quad b_2 \quad b_3 \cdots = \beta$

Bitstreams α, β are identified with subsets of $\mathbb{N}$.

Use variables X, Y for subsets of $\mathbb{N}$.

Requirement $\varphi(X, Y)$ is considered as winning condition in an infinite two-person game:

Players 1 and 2 choose bits $a_i = \alpha(i), b_i = \beta(i)$ ($i = 0, 1, \dots$) in alternation.

Play $\binom{\alpha(0)}{\beta(0)} \binom{\alpha(1)}{\beta(1)} \binom{\alpha(2)}{\beta(2)} \dots$ is won by 2 if $(\mathbb{N}, \dots) \models \varphi(\alpha, \beta)$

Strategies

A strategy for Player 1 is a map

$$\binom{\alpha(0)}{\beta(0)}\binom{\alpha(1)}{\beta(1)}\binom{\alpha(2)}{\beta(2)}\ \dots \binom{\alpha(k)}{\beta(k)} \ \mapsto \ 0/1$$

A strategy for Player 2 is a map

$$\binom{\alpha(0)}{\beta(0)}\binom{\alpha(1)}{\beta(1)}\ \dots\ \binom{\alpha(k)}{*} \ \mapsto \ 0/1$$

A strategy is called winning strategy for Player i if every play compatible with it satisfies the winning condition for Player i.

Finite-state strategy: computable by a finite automaton over

$$\Sigma = \{\binom{0}{0}, \binom{0}{1}, \binom{1}{0}, \binom{1}{1}, \binom{0}{*}, \binom{1}{*}\}$$

with output function.

Example

Consider the conjunction of three conditions on the input-output stream (α, β):

1. $\forall t : \alpha(t) = 1 \;\rightarrow\; \beta(t) = 1$
2. $\neg \exists t : \beta(t) = \beta(t+1) = 0$
3. $\exists^\omega t\ \alpha(t) = 0 \;\rightarrow\; \exists^\omega t\ \beta(t) = 0$

MSO-formula $\varphi(X, Y)$:

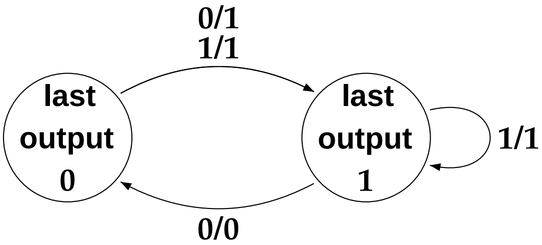
$$\forall t (X(t) \rightarrow Y(t))$$

$$\wedge \neg \exists t (\neg Y(t) \wedge \neg Y(t+1))$$

$$\wedge (\forall s \exists t (s < t \wedge \neg X(t)) \;\rightarrow\; \forall u \exists v (u < v \wedge \neg Y(v)))$$

Common-Sense Solution

- for input 1 produce output 1
- for input 0 produce
  - output 1 if last output was 0
  - output 0 if last output was 1



This is a finite-state strategy.

Solution – Solvability – Synthesis

The basic problems:

- **Solution: Does a given finite automaton compute a winning strategy?**
- **Solvability: Does there exist such a finite automaton?**
- **Synthesis: Construct such a finite automaton.**

Solution of Church's Problem

Specification language:
MSO = monadic second-order logic over $(\mathbb{N}, +1)$

"Solution" is settled by decidability of S1S (model-checking).

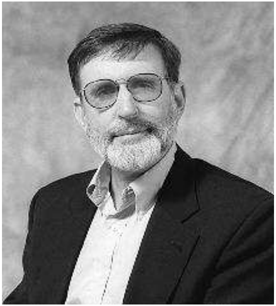
Büchi-Landweber Theorem (1969)

For each MSO-requirement $\varphi(X, Y)$ either Player 1 or Player 2 has a finite-state winning strategy (i.e., the game is "determined"),

it is decidable who wins,

and a finite-state winning strategy for the respective winner is computable.

J.R. Büchi



L.H. Landweber

Applications

- Controller synthesis
- Complementation results from determinacy
- Model Checking (μ-calculus)

Uses of determinacy:

- Completeness of strategy constructions
- Complementation and model-checking

Proof Strategy

1. **Transform the logical specification into a a graph-based game (finite-state game with Muller winning condition).**

2. **Transform the Muller game into a parity game**

3. **Solve the parity game**

A Pioneering Paper

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

Project MAC

MAC-M-125

Machine Structures Group Memo No. 14

September, 1965

Finite-State Infinite Games*

by

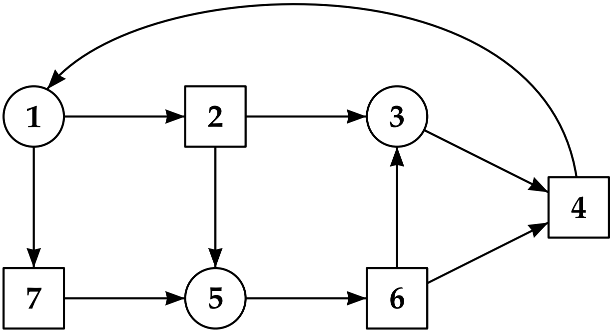
Robert McNaughton

This paper will begin with a discussion of infinite games in general, followed by a definition of finite-state infinite games. The main contribution is a proof that every such game has an effectively determinable finite-state winning strategy. The closing remarks establish the Corollary that Büchi's Sequential Calculus has a solvability algorithm, which was an open problem in the Theory of Automata that stimulated interest in the problem of finite-state infinite games.

M.O. Rabin, J. Hartmanis, R. McNaughton

Automata Theoretic Framework

Game Graphs (Arenas)



This view introduces the aspect of memory which does not appear in a game tree.

A game is now given as a pair (G, φ) of a graph $G = (V, V_1, V_2, E)$ and a winning condition for Player 2.

Solving a Game

Given a game graph and a winning condition for Player 2,

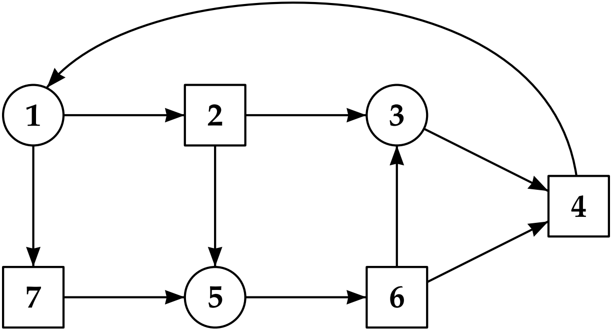
- **decide for each vertex v whether Player 2 has a winning strategy for plays starting from v**
  ("v belongs to the winning region W_2 of Player 2")
- **for $v \in W_2$ provide a winning strategy for Player 2 from v**

For a play $\rho = v_0 v_1 v_2 \ldots$ let $\mathrm{Inf}(\rho) := \{v \in V \mid \exists^\omega i\, v = v_i\}$

A Muller condition is given by a set $\mathcal{F} = \{F_1, \ldots, F_k\}$ with $F_i \subseteq V$ with the interpretation

"Play ρ is won by Player 2 iff $\mathrm{Inf}(\rho)$ is one of the sets F_i"

Example



Example 1: "Visit 2 and 6 again and again"

A winning strategy: From 1 go to 2 and 7 in alternation.

Example 2: Visit 2 again and again.

Player 2 has a positional (memoryless) winning strategy.

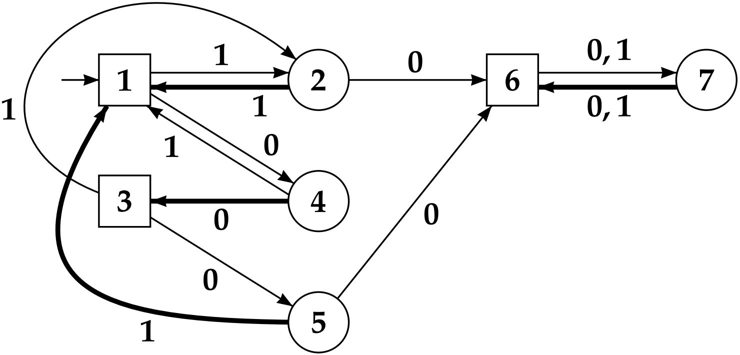
From Logic to Muller Games

Büchi (1960), McNaughton (1966):

Each MSO-formula $\varphi(X, Y)$ can be transformed into a Muller game with designated vertex v_0 such that

- **Player 2 has a winning strategy to satisfy the condition φ iff Player 2 wins the Muller game from v_0,**
- **a finite-state winning strategy for Player 2 in the Muller game from v_0 allows to construct a finite-state strategy for Player 2 to satisfy φ**

Example



$\mathcal{F}$ **contains** $\{1,2,3,4\}, \{1,2,3,4,5\}, \{1,3,4,5\}, \{1,4\}$

Parity Condition

goes back to F. Hausdorff 1914,
introduced as “Rabin chain condition“ by A.W. Mostowski 1985
rediscovered as “parity condition” by Emerson and Jutla 1991

We assume a coloring $c : V \to \{1, \ldots, k\}$ of the game graph.

A play $\rho \in V^\omega$ satisfies the parity condition iff the maximal color occurring infinitely often in ρ is even.

The parity condition says for play ρ:

$$\bigvee_{j \text{ even}} (\exists^\omega i : c(\rho(i)) = j \ \wedge \neg \exists^\omega i : c(\rho(i)) > j)$$

A parity game is given by a game graph with finite coloring and the parity condition as winning condition for player 2.

Proof of Büchi-Landweber Theorem

Prelude: Reachability Games

Given a finite game graph $G = (V, V_1, V_2, E)$ and $F \subseteq Q$

Player 2 wins $\rho :\Leftrightarrow \exists i \; \rho(i) \in F$

Inductive construction of $\mathbf{Attr}_2^i(F)$:

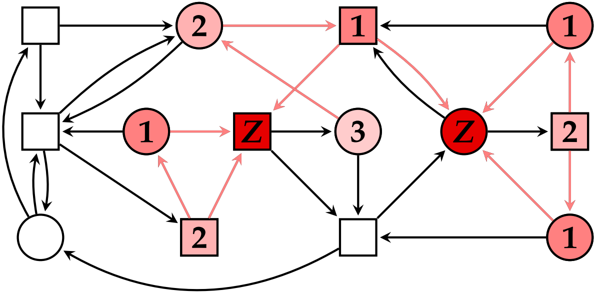
$$\begin{aligned}
\mathbf{Attr}_2^0(F) &= F, \\
\mathbf{Attr}_2^{i+1}(F) &= \mathbf{Attr}_2^i(F) \\
&\cup \{u \in V_2 \mid \exists (u,v) \in E : v \in \mathbf{Attr}_2^i(F)\} \\
&\cup \{u \in V_1 \mid \forall (u,v) \in E : v \in \mathbf{Attr}_2^i(F)\}
\end{aligned}$$

$$W_2 = \bigcup \mathbf{Attr}_2^i(F)$$

$$W_1 = V \setminus \mathbf{Attr}_2(F)$$

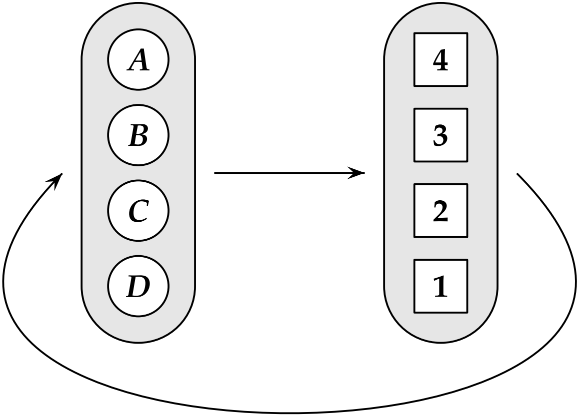
Over infinite graphs, a corresponding transfinite induction works.

Example



Towards Parity Games: DJW-Game

invented by Dziembowski, Jurdzinski and Walukiewicz (1997)



Winning condition:

$$|\mathrm{Inf}(\rho) \cap \{A, B, C, D\}| = \max(\mathrm{Inf}(\rho) \cap \{1, 2, 3, 4\})$$

Latest Appearance Record

$D \quad B \quad B \quad D \quad C \quad B \quad A \quad B \quad A \quad C \quad B \quad A \quad B \quad A \quad C$

A	D	B	$\underline{B}$	D	C	B	A	B	A	C	B	A	B	A	C
B	A	D	D	$\underline{B}$	D	C	B	$\underline{A}$	$\underline{B}$	A	C	B	$\underline{A}$	$\underline{B}$	A
C	B	$\underline{A}$	A	A	B	$\underline{D}$	C	C	C	$\underline{B}$	$\underline{A}$	$\underline{C}$	C	C	$\underline{B}$
$\underline{D}$	$\underline{C}$	C	C	C	$\underline{A}$	A	$\underline{D}$	D	D	D	D	D	D	D	D

Solution of the DJW-Game

LAR-strategy for Player 2:

During play, update and use the LAR as follows:

- **shift the current letter vertex to the front**
  underline the position from where the current letter was taken
- **move to the number vertex given by underlined position**

These are the two items performed by the strategy:

- **update of memory**
- **choice of next step ("output")**

Result: Finite-state winning strategy with $n! \cdot n$ states for a game graph with $2n$ vertices

Analyzing the Winning Strategy

Call the underlined position the hit

The states of a LAR up to the hit are called the recent states.

The Muller winning condition says:

For the highest hit occurring infinitely often, the corresponding recent states form a set in $\mathcal{F}$.

We merge the hit value h and the status of the recent states into a LAR-color:

- **color $2h$ if the recent states form set in $\mathcal{F}$**
- **color $2h-1$ otherwise**

So the Muller winning condition says:

The highest LAR-color occurring infinitely often is even

McNaughton 1965

$$\langle N_{i_1}, N_{i_2}, \ldots, N_{i_n} \rangle \quad (1 \leqq i_x \leqq n \text{ and } i_x \neq i_y \text{ if } x \neq y)$$

where n is the number of nodes of the graph, which are labeled $N_1, \ldots, N_n$. For time 0, the order vector is $\langle N_{in}, \ldots \rangle$, where N_{in} is the initial node and where the other nodes occur arbitrarily. Then at any time t, if the order vector is $\langle N_{i_1}, \ldots, N_{i_n} \rangle$ and at time t+1 the marker goes to node N_{i_j}, the order vector for time t+1 is $\langle N_{i_j}, N_{i_1}, \ldots, N_{i_{j-1}}, N_{i_{j+1}}, \ldots N_{i_n} \rangle$.

Memory Extensions

Use (finite) memory space S, initialized with s_0

Transform game graph $G = (V, V_1, V_2, E)$ into
$G' = (S \times V, S \times V_1, S \times V_2, E')$

with memory update function $\delta : S \times V \to S$

For $(u, v) \in E$ put $((s, u), (\delta(s, u), v))$ into E'

Each play ρ over G induces a play ρ' over G'.

Write $(G, \varphi) \leq (G', \varphi')$ if for each ρ:

Player 2 wins ρ w.r.t. φ iff Player 2 wins ρ' w.r.t. φ'

Application of Game Reduction

Assume $(G, \varphi) \leq (G', \varphi')$ via memory extension with finite S.

If (G', φ') is determined with memoryless winning strategies, then (G, φ) is determined with finite-state strategies.

Proof

Use state set S of memory expansion.

Use memory update for transition function.

Use memoryless winning strategy for definition of output function $\eta : S \times V_2 \to V_1$

If $(s, u) \to (s', v)$ is an edge of the winning strategy then define $\eta(s, u) = v$.

Positional Determinacy of Parity Games

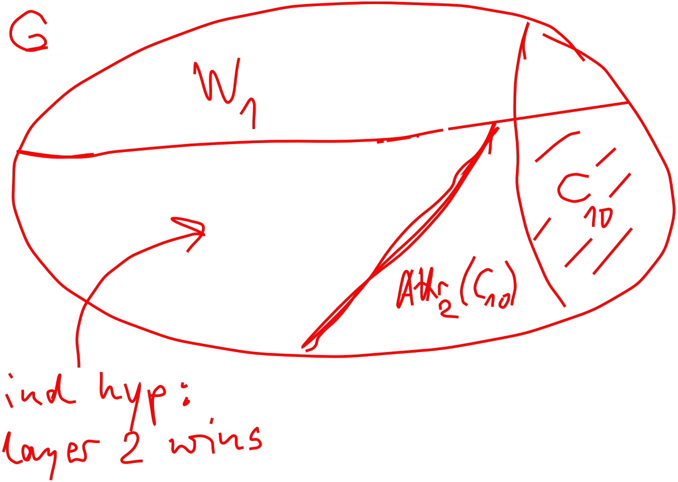
Theorem (Emerson-Jutla 1991)

- **Parity games are determined (i.e., each vertex belongs to W_A or W_B), and the winner from a given vertex has a positional winning strategy.**
- **Over finite graphs, the winning regions and winning strategies of the two players can be computed in (at most) exponential time in the number of vertices of the game graph.**

Whether polynomial time suffices in the second item, is open.

Positional Determinacy of Parity Games

by induction on number of colors, say with even highest color



Finite Arenas: Find Winning Regions

1. **Guess W_1 and W_2 and positional strategies given by edge sets: E_1 and E_2**
2. **Check that E_1 defines a winning strategy from each $q \in W_1$ and that E_2 a winning strategy from each $q \in W_2$**

Pursue Step 2: Check whether a given positional strategy is a winning strategy for Player 2 from q.

Checking Strategies

Remark: For a fixed positional strategy f of Player 2 one can decide in polynomial time for any $q \in Q$, whether f is a winning strategy from q

Proof:

Consider the graph G_2 induced by a given by a positional strategy of Player 2. We have a one-player game (of Player 1).

Check whether in G_2 there is a path from q to a loop whose highest color is odd.

For this, check for reachable SCC's in the restriction of G_2 to colors $\leq m$ for odd m.

Decision Problem "Parity Game"

Given: A finite game graph G with coloring, $q \in Q$

Question: Does player 2 win the parity game from q?
(Short: "$q \in W_2$ in the corresponding parity game?")

Theorem: The Problem "Parity Game" is in the complexity class NP $\cap$ co-NP

Proof: The above nondeterministic procedure shows that the problem is in NP.

It remains to show that the complementary problem "$q \notin W_2$?" is also in NP.

This problem means "$q \in W_1$?". It is solvable in the same way as "$q \in W_2$?", hence in NP.

Intriguing: Open problem: Is "parity game" in P?

Rabin's Solution

The Tree Setting

Consider moves of Players 1 and 2 in $\{0, 1\}$.

The players construct a labelled path in T_2.

A dummy value is associated to the root (say 0).

Player 1 chooses directions.

Player 2 chooses a label at a node reached by Player 1.

When Player 1 has chosen the bit sequence w, Player 2 puts his choice as label at position w.

So a strategy of Player 2 is a $\{0, 1\}$-labelling of T_2.

Defining Winning Strategies in S2S

A play is a sequence "direction-label-direction-label etc."

The winning condition is a condition on labelled paths through T_2.

If the winning condition definable by an S1S-formula $\varphi(X, Y)$, one can reformulate it as an S2S condition on paths through the tree.

Consequence: There is an S2S-formula $\psi(Z)$ expressing:

The tree labelling given by Z defines a winning strategy for Player 2.

("For all paths, the sequence Y of directions and the sequence X of Z-labels along Y satisfies $\varphi(X, Y)$")

Application of Tree Automata

- **By Rabin's Tree Theorem,**
  one can decide truth of $\exists Z \psi(Z)$,
  so can decides whether Player 2 wins the game
  defined by φ.

- **In this case, by Rabin's Basis Theorem**
  a regular tree exists satisfying $\psi(Z)$, which gives a
  finite-state strategy for Player 2.

The problem of solving sequential games is a satisfiability problem over trees.