Logic and Automata II: Infinite Games

Wolfgang Thomas

RWTH AACHEN

Computability in Europe, Budapest, June 2014

Church's Problem

Alonzo Church (1903-1995)

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 $i_1, i_2, \cdots, i_\mu$, 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.)

$$\chi_1(x_1 + M + 1, 0,\dots,0, 0) \equiv \text{falsehood}$$

. . . .

. . . .

$$\chi_N(x_1 + M + 1, M,\dots,M, g) \equiv \text{falsehood}$$

$$\chi_1(x_1 + M + 1, x_2 + M + 1,\dots,0, 0) \equiv \text{falsehood}$$

. . . .

. . . .

$$\chi_1(x_1 + M + 1, x_2 + M + 1,\dots,x_m + M + 1, 0) \equiv \text{falsehood}$$

$$\chi_2(x_1 + M + 1, x_2 + M + 1,\dots,x_m + M + 1, 0) \equiv \text{falsehood}$$

. . . .

$$\chi_N(x_1 + M + 1, x_2 + M + 1,\dots,x_m + M + 1, 0) \equiv \text{falsehood}$$

$$\chi_1(x_1 + M + 1, x_2 + M + 1,\dots,x_m + M + 1, 1) \equiv \text{falsehood}$$

. . . .

. . . .

$$\chi_N(x_1 + M + 1, x_2 + M + 1,\dots,x_m + M + 1, g) \equiv \text{falsehood}$$

$$\chi_1(0, 0,\dots,0, t + g + 1) \equiv \chi_1(0, 0,\dots,0, t + g) \vee Q_{100\dots0}[\chi_1(0, 0,\dots,0, t),\dots, \chi_N(0, 0,\dots, 0, t), \chi_1(0, 0,\dots,1, t),\dots,\dots, \chi_N(M+1, M+1,\dots,M+1,t)$$

$$\chi_2(0, 0,\dots,0, t + g + 1) \equiv \chi_2(0, 0,\dots,0, t + g) \vee \overline{\chi}_1(0, 0,\dots,0, t + g)Q_{200\dots0}[\chi_1(0, 0,\dots,0, t),\dots, \chi_N(0, 0,\dots,0, t), \chi_1(0, 0,\dots,1, t),\dots,\dots, \chi_N(M + 1, M + 1,\dots,M + 1, t)]$$

. . . .

. . . .

$$\chi_N(M, M,\dots,M, t + g + 1) \equiv \chi_N(M, M,\dots,M, t + g) \vee \overline{\chi}_1(M, M,\dots,M, t + g)\, \overline{\chi}_2(M, M,\dots,M, t + g)\cdots \overline{\chi}_{N-1}(M, M,\dots,M, t + g)Q_{NMM\dots M}[\chi_1(0, 0,\dots, 0,t),\dots, \chi_N(0, 0,\dots,0, t), \chi_1(0, 0,\dots,1, t),\dots,\dots, \chi_N(2M + 1, 2M + 1,\dots,2M + 1,t)]$$

$$\chi_1(x_1 + M + 1, 0,\dots,0, t + g + 1) \equiv \chi_1(x_1 + M + 1, 0,\dots,0, t + g) \vee Q_{10\dots0}[\chi_1(x_1, 0,\dots,0, t),\dots, \chi_N(x_1,0,\dots,0, t), \chi_1(x_1, 0,\dots,1, t),\dots,\dots, \chi_N(x_1 + 2M + 2, M + 1,\dots,M + 1, t)]$$

$$\chi_2(x_1 + M + 1, 0,\dots,0, t + g + 1) \equiv \chi_2(x_1 + M + 1, 0,\dots,0, t + g) \vee \overline{\chi}_1(x_1 + M + 1, 0,\dots,0, t + g)Q_{20\dots0}[\chi_1(x_1, 0,\dots,0, t),\dots, \chi_N(x_1, 0,\dots,0, t), \chi_1(x_1, 0,\dots, 1, t),\dots,\dots, \chi_N(x_1 + 2M + 2, M + 1,\dots,M + 1, t)]$$

. . . .

. . . .

$$\chi_1(x_1 + M + 1, x_2 + M + 1,\dots,x_m + M + 1, t + g + 1) \equiv \chi_1(x_1 + M + 1, x_2 + M + 1,\dots,x_m + M + 1, t + g) \vee Q_1[\chi_1(x_1, x_2,\dots,x_m, t),\dots, \chi_N(x_1, x_2,\dots,x_m, t), \chi_1(x_1, x_2,\dots,x_m + 1, t),\dots,\dots, \chi_N(x_1 + 2M + 2, x_2 + 2M + 2,\dots, x_m + 2M + 2, t)]$$

$$\chi_2(x_1 + M + 1, x_2 + M + 1,\dots,x_m + M + 1, t + g + 1) \equiv \chi_2(x_1 + M + 1, x_2 + M + 1,\dots,x_m + M + 1, t + g) \vee \overline{\chi}_1(x_1 + M + 1, x_2 + M + 1,\dots,x_m + M + 1, t + g)Q_2[\chi_1(x_1, x_2,\dots,x_m, t),\dots, \chi_N(x_1, x_2,\dots,x_m, t), \chi_1(x_1, x_2,\dots,x_m + 1, t),\dots,\dots, \chi_N(x_1 + 2M + 2, x_2 + 2M + 2,\dots,x_m + 2M + 2, t)]$$

. . . .

$$\chi_N(x_1 + M + 1, x_2 + M + 1,\dots,x_m + M + 1, t + g + 1) \equiv \chi_N(x_1 + M + 1, x_2 + M + 1,\dots,x_m + M + 1, t + g) \vee \overline{\chi}_1(x_1 + M + 1, x_2 + M + 1,\dots,x_m + M+1, t+g)\, \overline{\chi}_2(x_1+M+1, x_2 + M + 1,\dots,x_m + M + 1, t+g)\cdots \overline{\chi}_{N-1}(x_1+M+1, x_2+M+1,\dots,x_m + M + 1, t + g)Q_N[\chi_1(x_1, x_2,\dots,x_m, t),\dots, \chi_N(x_1, x_2,\dots,x_m, t), \chi_1(x_1, x_2,\dots,x_m + 1, t),\dots,$$

A Pioneering Paper

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

Project MAC | MAC-M-125

Machine Structures Group Memo No. 14 — September, 1965

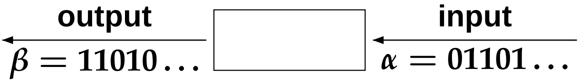
Finite-State Infinite Games*

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.

Requirements as Winning Conditions



Player 1:	a_0	a_1	a_2	$a_3 \cdots$	$=$	α
Player 2:	b_0	b_1	b_2	$b_3 \cdots$	$=$	β

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 $a_i = \alpha(i), b_i = \beta(i)$ ($i = 0, 1, \ldots$) in alternation.

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

Gale-Stewart Games

In a Gale-Stewart game, the winning condition (for Player 2) is just an abstract set L of plays $\alpha(0)\beta(0)\alpha(1)\beta(1)\alpha(2)\beta(2) \quad \ldots$

Descriptive set theory focusses on the question:

Which games are determined?

Martin's Theorem guarantees this for Borel sets L.

Church's Problem gives a shift towards computation of computable strategies.

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 $\varphi(X, Y)$:

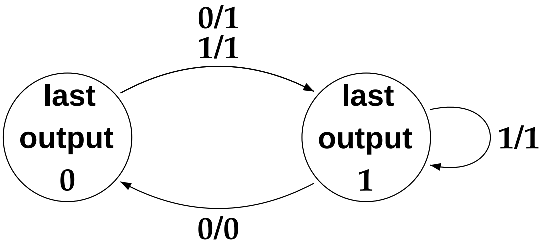
$$\forall t (X(t) \rightarrow Y(t))$$

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

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

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.

Büchi-Landweber Theorem

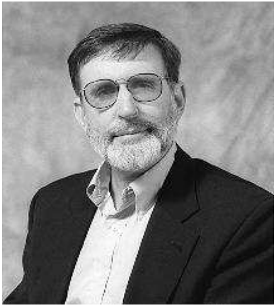
Solution of Church's Problem

Büchi-Landweber Theorem (1969)

For each MSO-requirement $\varphi(X, Y)$

- either Player 1 or Player 2 has a winning strategy (i.e., the game is "determined"),
- it is decidable who wins,
- and a finite-state winning strategy for the respective winner can be computed.

J.R. Büchi



L.H. Landweber

Applications

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

Uses of determinacy:

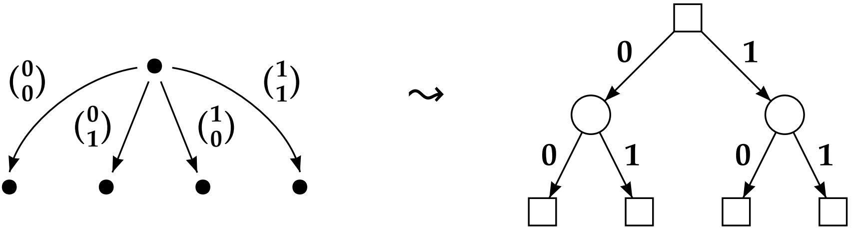
- Completeness of strategy constructions
- Complementation and model-checking

Applying McNaughton's Theorem

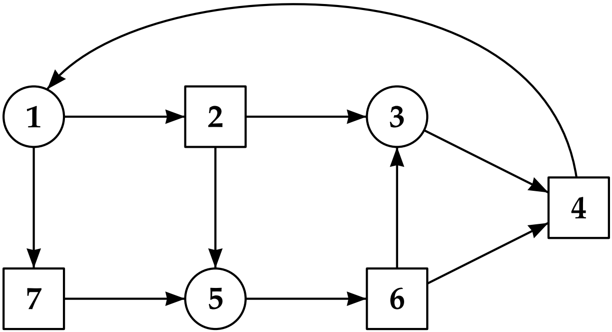
Transform $\varphi(X, Y)$ into a deterministic Muller automaton $\mathcal{A}$ over the alphabet $\{0, 1\}^2$.

$\mathcal{A}$ reads a play and accepts it iff Player 2 is the winner.

Reconfiguration of automaton to "game graph":



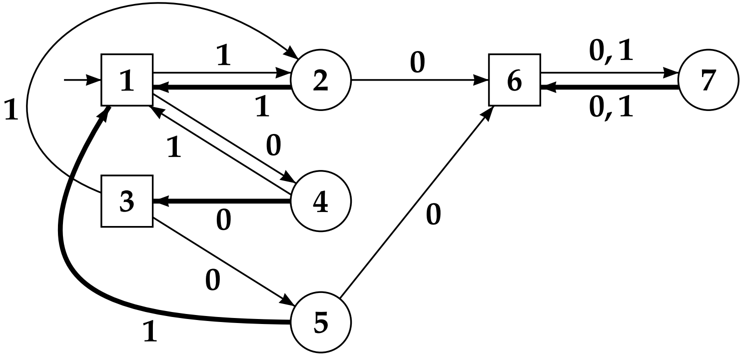
Game Graphs (Arenas)



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

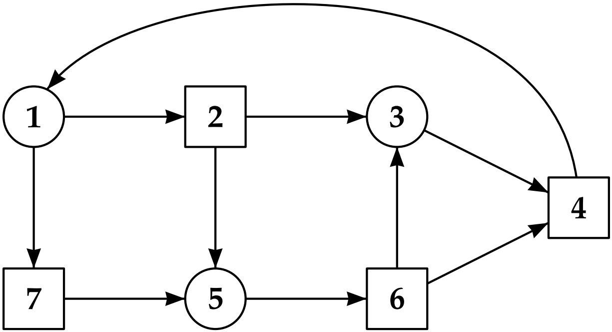
We use only very simple conditions (about infinitely many visits of states during a play).

A Muller Game



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

Further Example



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

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

Condition 2: Visit 2 again and again.

Here Player 2 has a positional (memoryless) winning strategy!

Parity Condition

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

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

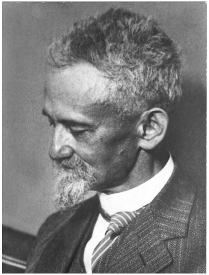
Formally:

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

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

History

The parity condition goes back to F. Hausdorff 1914

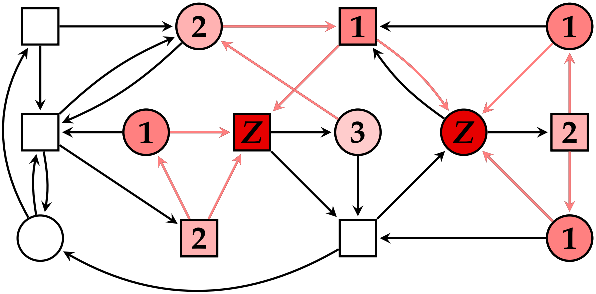


Felix Hausdorff (1868-1942)

and was re-introduced as “Rabin chain condition“ by A.W. Mostowski 1985,
rediscovered as “parity condition” by Emerson and Jutla 1991

Solving Muller Games

Simple Case: Reachability



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

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

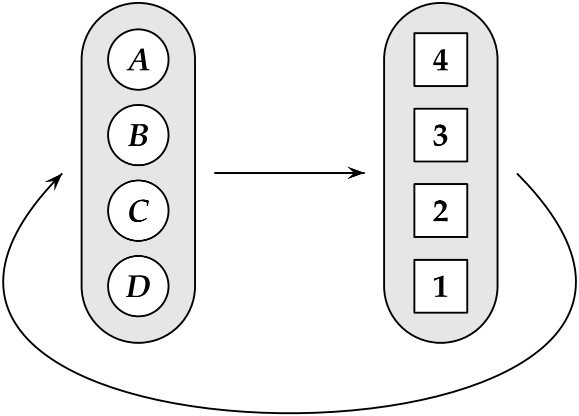
$$W_2 = \bigcup \mathbf{Attr}_2^i(F)$$

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

Over infinite graphs, a corresponding transfinite induction works.

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	B	B	D	C	B	A	B	A	C	B	A	B	A	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

In General: Memory Extensions

We have extended the game arena $G = (V, V_1, V_2, E)$ by a finite memory structure S

to a larger game arena $G' = (V \times S, V_1 \times S, V_2 \times S, E')$

at the same time changing the Muller winning condition to a parity winning condition.

Consequence: A positional winning strategy over G' can be converted to a finite-memory winning strategy over G.

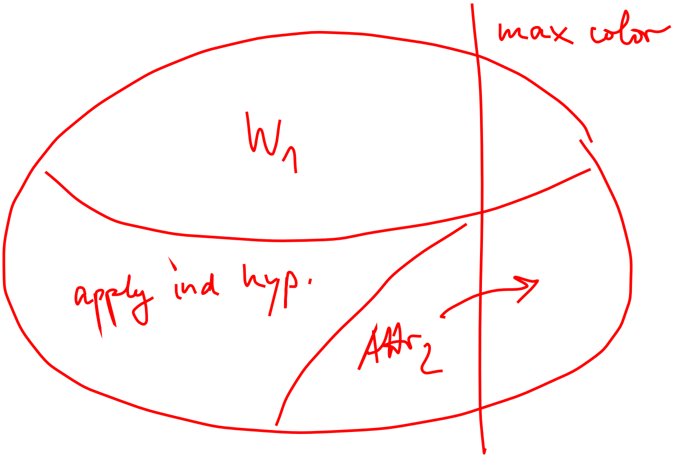
Positional Determinacy of Parity Games

Theorem (Emerson-Jutla 1991)

- Parity games are determined, 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.

Positional Determinacy of Parity Games

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



Finite Arenas: Computable Solution

For Player 2:

1. **Guess the winning region and a positional strategy over this region.**
2. **Check that this strategy is winning for Player 2.**

Step 2 can be done in polynomial time: Analyze the loops that Player 1 can realize.

The problem to determine the winner from a given vertex of a parity game is in NP $\cap$ co-NP.