Constructing Winning Strategies in Infinite Games

Tbilisi, October 2007

Wolfgang Thomas

thomas@informatik.rwth-aachen.de

RWTH AACHEN

Church's Problem

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)."

APPLICATION OF RECURSIVE ARITHMETIC TO THE PROBLEM OF CIRCUIT SYNTHESIS

Alonzo Church

RESTRICTED RECURSIVE ARITHMETIC

Primitive symbols are individual (i.e., numerical) variables $x, y, z, t, \cdots$, singulary functional constants $i_1, i_2, \cdots, i_\mu$, the individual constant 0, the accent ' as a notation for <u>successor</u> (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.

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

. . . .

. . . .

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

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

. . . .

. . . .

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

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

. . . .

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

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

. . . .

. . . .

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

$$\chi_1(0,\ 0,\ldots,0,\ t + g + 1) \equiv \chi_1(0,\ 0,\ldots,0,\ t + g)\ v\ Q_{100\ldots0}[\chi_1(0,\ 0,\ldots,0,\ t),\ldots,\chi_N(0,\ 0,\ldots,0,\ t),\ \chi_1(0,\ 0,\ldots,1,\ t),\ldots,\ldots,\chi_N(M{+}1,\ M{+}1,\ldots,M{+}1,t)$$

$$\chi_2(0,\ 0,\ldots,0,\ t + g + 1) \equiv \chi_2(0,\ 0,\ldots,0,\ t + g)\ v\ \overline{\chi}_1(0,\ 0,\ldots,0,\ t + g)Q_{200\ldots0}[\chi_1(0,\ 0,\ldots,0,\ t),\ldots,\chi_N(0,\ 0,\ldots,0,\ t),\ \chi_1(0,\ 0,\ldots,1,\ t),\ldots,\ldots,\chi_N(M + 1,\ M + 1,\ldots,M + 1,\ t)]$$

. . . .

. . . .

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

$$\chi_1(x_1 + M + 1,\ 0,\ldots,0,\ t + g + 1) \equiv \chi_1(x_1 + M + 1,\ 0,\ldots,0,\ t + g)\ v\ Q_{10\ldots0}[\chi_1(x_1,\ 0,\ldots,0,\ t),\ldots,\chi_N(x_1,0,\ldots,0,\ t),\ \chi_1(x_1,\ 0,\ldots,1,\ t),\ldots,\ldots,\chi_N(x_1 + 2M + 2,\ M + 1,\ldots,M + 1,\ t)]$$

$$\chi_2(x_1 + M + 1,\ 0,\ldots,0,\ t + g + 1) \equiv \chi_2(x_1 + M + 1,\ 0,\ldots,0,\ t + g)\ v\ \overline{\chi}_1(x_1 + M + 1,\ 0,\ldots,0,\ t + g)Q_{20\ldots0}[\chi_1(x_1,\ 0,\ldots,0,\ t),\ldots,\chi_N(x_1,\ 0,\ldots,0,\ t),\ \chi_1(x_1,\ 0,\ldots,1,\ t),\ldots,\ldots,\chi_N(x_1 + 2M + 2,\ M + 1,\ldots,M + 1,\ t)]$$

. . . .

. . . .

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

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

. . . .

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

Requirements as Winning Conditions

$$\xleftarrow{\text{output}} \beta = 010101\ldots \quad \boxed{} \quad \xleftarrow{\text{input}} \alpha = 001101\ldots$$

Requirement $\varphi(\alpha, \beta)$ is considered as winning condition in an infinite two-person game.

Player 1 for input bits, Player 2 for output bits

Players 1 and 2 choose their bits $\alpha(t)$ and $\beta(t)$ in alternation.

Play $\binom{\alpha(0)}{\beta(0)} \binom{\alpha(1)}{\beta(1)} \binom{\alpha(2)}{\beta(2)} \ldots$ is won by Player 2

if $\varphi(\alpha, \beta)$ is satisfied

Strategies

A strategy for Player 1 is a map

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

A strategy for Player 2 is a map

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

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: If β should be "double α",
then a finite-state strategy does not suffice.

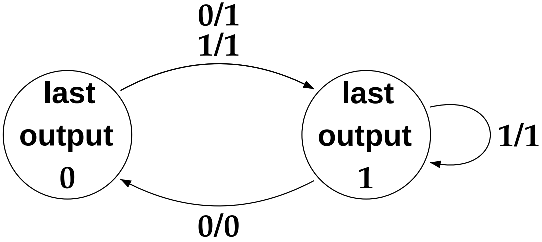
Example

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. $\exists^{\omega} t\ \alpha(t) = 0 \;\rightarrow\; \exists^{\omega} t\ \beta(t) = 0$

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": a solution for the specification

Infinite Games in Computer Science

- Area: Nonterminating reactive systems (operating systems, control systems, business software, etc.)
- Strategy construction is program synthesis
- Asymmetric view in applications (controller against environment)
  Symmetric view is helpful in game analysis
- Cantor space rather than Baire space
- Winning conditions define special $B(\Sigma_2^0)$ sets rather than open (Σ_1^0) sets or Borel sets

Overview

1. **Church's Problem and the Büchi-Landweber Theorem**
2. **From logic to Muller games**
3. **An interesting Muller game**
4. **Solving Muller games**
5. **Refinement of Büchi-Landweber Theorem**

Part 1

Church's Problem and the Büchi-Landweber Theorem

Specification Language

Underlying structure: $(\mathbb{N}, +1, <)$

$t, s, \ldots$ **as number variables (for time instances)**
$\alpha, \beta, \gamma, \ldots$ **as sequence variables**

Use Boolean connectives and quantifiers (over both kinds of variables)

Write $\exists^\omega t \ldots$ **for** $\forall s \exists t (s < t \wedge \ldots$**))**

The logic is called S1S (second-order theory of one successor)

or MSO-logic (monadic second-order logic)

Church’s Problem and its Solution

Church’s Problem asks to decide, for an S1S-specification $\varphi(\alpha, \beta)$, whether Player 2 wins the corresponding game, and in this case to construct a finite-state winning strategy.

Büchi-Landweber Theorem (1969)

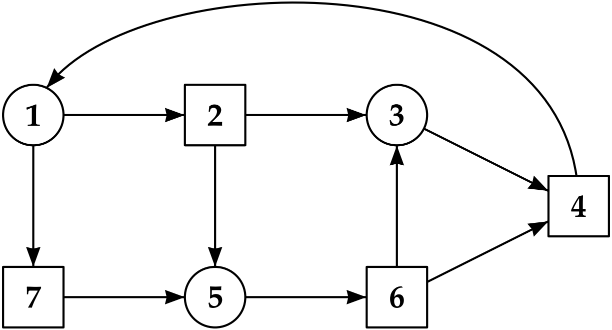
For each S1S-specification $\varphi(\alpha, \beta)$ one can decide whether Player 2 can win the corresponding game, in this case synthesize a finite automaton that executes a winning strategy.

Present approach is from W.T., LNCS 900 (1995).

Part 2

From Logic to Muller Games

Muller Games: Intuition



Winning condition for Player 2 for play ρ depends on the set $\mathrm{Inf}(\rho)$ of vertices visited infinitely often.

Example: “Visit 2 and 6 again and again”

Strategy: From 1 go to 2 and 7 in alternation

Two Winning Conditions

- **Muller condition**, given by a family $\mathcal{F} = \{F_1, \ldots, F_k\}$

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

  Example: Take for $\mathcal{F}$ all sets which contain 2 and 6

  We speak of a **Muller game**

- **Reachability condition**, given by a set F of vertices
  Play ρ is won by Player 2 iff $\exists t\ \rho(t) \in F$

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

Easy:

Solution of reachability games by memoryless strategies

Method: Computation of “2-attractor of F”

From Logic to Games

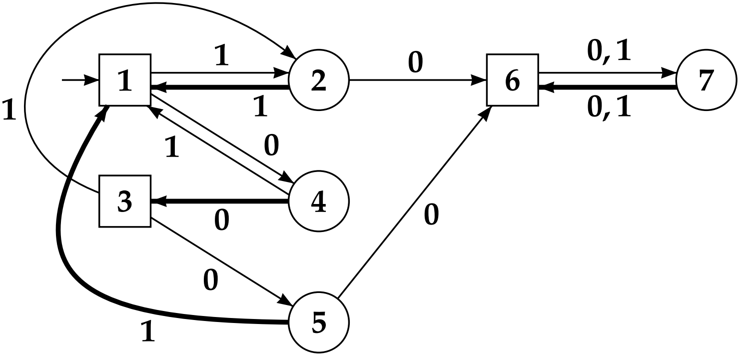
Büchi (1960), McNaughton (1966):

Each S1S-formula $\varphi(\alpha, \beta)$ can be transformed into a Muller game with designated vertex v_0 such that

- **Player 2 has a winning strategy to satisfy the condition $\varphi(\alpha, \beta)$ 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 $\varphi(\alpha, \beta)$**

So it remains to solve Muller games.

Example



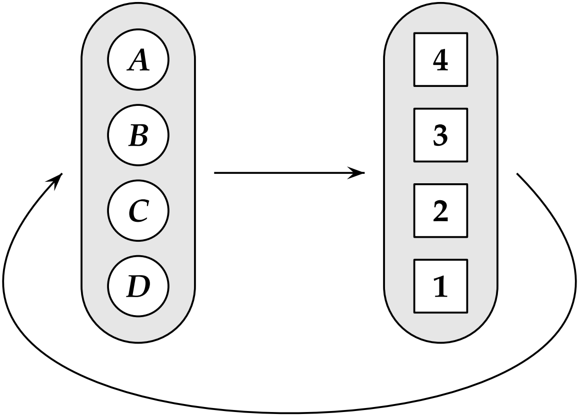
$\mathcal{F}$ **contains** $\{1,2,3,4\}, \{1,2,3,4,5\}, \{1,3,4,5\}, \{1,4\}$

Part 3

An Interesting Muller Game

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

Visited letter	LAR
A	$\underline{A}BCD$
C	$CA\underline{B}D$
C	$\underline{C}ABD$
D	$DCA\underline{B}$
B	$BDC\underline{A}$
D	$D\underline{B}CA$
C	$CD\underline{B}A$
D	$D\underline{C}BA$
D	$\underline{D}CBA$

Solution of the DJW-Game

Player 2 wins the DJW game with the LAR strategy.

This is a finite-state strategy,
although the number of memory states is large:
$n! \cdot n$ **states for** n **letter-vertices**

- **Use letter-vertices as input**
- **Use update of LAR for the transition function**
- **Use hit position for the output (choice of next step)**

An Essential Observation

Call the letters up to hit position the "hit set".

For the maximal hit occurring infinitely often in the LAR-sequence,

call the corresponding hit set the permanence set.

The set of letters chosen infinitely often coincides with the permanence set of the LAR-sequence.

Part 4

Solving Muller Games

General Idea

Step 1

Over a game graph G with states $1, \ldots, n$ we will use the finite automaton with

- ***all* LAR's $(i_1 \ldots \underline{i_h} \ldots i_n)$ as memory states**
- **the vertices of G as "input letters"**
- **the LAR update rule as transition function**

Step 2

We have to determine the outputs of the LAR-automaton

Build a new game graph $G' = G \times \mathrm{LAR}(G)$

A play ρ over G is mapped to a play ρ' over G'

Analyzing the Muller Condition over G

Over G' we can reformulate the Muller winning condition.

- **The set $\mathrm{Inf}(\rho)$ is the permanence set of the LAR sequence**
- **The permanence set is the hit set for the highest hit occurring infinitely often**
- **So the Muller winning condition says:**
  The hit set for the highest hit occurring infinitely often belongs to $\{F_1, \ldots, F_m\}$

Merge hit value h and status of hit set into a color:
color $2h$ if hit set belongs to $\{F_1, \ldots, F_m\}$, otherwise $2h - 1$

So the Muller winning condition says:

The highest LAR-color occurring infinitely often is even
("parity condition")

Intermediate Summary

- We have transformed a game graph G into an expanded game graph $G' = G \times \mathrm{LAR}(G)$.
- A play ρ over G induces a play ρ' over G'.
- The play ρ' records ρ plus the state sequence which the LAR-automaton assumes during ρ.
- The Muller winning condition on ρ becomes the parity condition for ρ'.
- Conclusion:
  Suppose we have a memoryless winning strategy for Player 2 in the parity game over G'.
  This gives the output function of the LAR automaton, and we have a finite-state winning strategy for the Muller game over G.

Solving Parity Games

Memoryless Determinacy of Parity Games:

Given a parity game
(by a finite game graph G and a coloring c),

one can compute the winning regions of the two players and corresponding memoryless winning strategies.

Moreover, the two winning regions cover the whole game graph.

Proof by induction over the number of vertices of G.

Part 5

Refinement of Büchi-Landweber Theorem

Definability of Strategies

A strategy $f : \binom{\alpha(0)}{\beta(0)} \binom{\alpha(1)}{\beta(1)} \dots \binom{P(k-1)}{Q(k-1)} \binom{\alpha(k)}{*} \mapsto$ **0/1**

is **MSO-definable** **iff there is an MSO-formula** $\psi(X, Y, x)$ **which says when the output bit is 1:**

$$([0,k], <, \alpha[0,k], \ \beta[0,k-1], \ k) \models \psi$$

iff

$$f(\binom{\alpha(0)}{\beta(0)} \dots \binom{\alpha(k-1)}{\beta(k-1)} \binom{\alpha(k)}{*}) = 1$$

Büchi, Elgot, Trakhtenbrot (1957-1960):
Finite-state strategies are MSO-definable.

$\mathcal{L}$-Definable Games and Strategies

An $\mathcal{L}$-defined game is **determined with $\mathcal{L}'$-definable strategies** if

for each $\mathcal{L}$-formula $\varphi(\alpha, \beta)$, there is either an $\mathcal{L}'$-definable winning strategy of Player 1 or an $\mathcal{L}'$-definable winning strategy for Player 2.

Büchi-Landweber:

MSO-defined games are determined with MSO-definable strategies.

What about other logics?

Results

Theorem

For $\mathcal{L} =$ MSO, FO($<$), FO($+1$):

Each $\mathcal{L}$-definable game is determined with $\mathcal{L}$-definable winning strategies (which are computable from the specification).

Theorem

If $\mathcal{L} =$ Presburger arithmetic, this fails.

(A. Rabinovich, W.T., CSL 2007)

Part 6

Perspectives

Research Areas

- Games over infinite graphs
- Concurrent games
- Games with quantitative winning conditions
- Timed games
- Stoachstic games

A fundamental problem: Is there a methodology to solve games “compositionally”, i.e. following the structure of the formula that defines the winning condition?