Infinite Games for Verification and Synthesis: Basic Theory and Quantitative Aspects

Wolfgang Thomas

QMC PhD School, Copenhagen, February 2012
Part 1

Basic Theory
Background: 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 $x, y, z, t, \cdots$, 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,

from $P \supset S^a_{\alpha} P |$ and $S^a_{\alpha} P |$ to infer $P$;

and any one of several alternative recursion schemata or sets of recursion schemata.
Alonzo Church
“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.)
Church’s Problem

Given a requirement on a bit stream transformation

\[ \beta = 11010 \ldots \]

fill the box by a machine with output, satisfying the requirement (or state that the requirement is not satisfiable).

An important concrete case:

- Requirements are formulated in MSO-logic.
- The machine should be a finite automaton.

Helpful perspective: Infinite two-person game.
Example

Requirement:

1. $\forall t : \alpha(t) = 1 \rightarrow \beta(t) = 1$
2. $\neg \exists t : \beta(t) = \beta(t + 1) = 0$
3. $\exists^\omega t \alpha(t) = 0 \rightarrow \exists^\omega t \beta(t) = 0$
This is a "finite-state strategy", implemented by a Mealy automaton.
The Framework
Arenas and Winning Conditions

We consider two-person games between players 0 and 1.

An infinite game is a pair $\Gamma = (G, \varphi)$ of two components:

- a transition graph $G$ (arena of the game), so that the infinite paths through $G$ are the plays of the game,
- a winning condition $\varphi$ on plays, describing those plays which are won by player 0.

For the second item $\varphi$ one can use

- abstract sets of plays (an $\omega$-language $L \subseteq \{0, 1\}^\omega$)
- conditions on plays specified in a logic,
- automata theoretic acceptance conditions.

For this, one often uses a finite coloring of $V$. 

Wolfgang Thomas
A game graph has the form $G = (Q, Q_0, E)$ where $Q_0 \subseteq Q$ and $E \subseteq Q \times Q$ is the transition relation satisfying

$$\forall q \in Q : qE \neq \emptyset \quad (\text{i.e. } \forall q \exists q' : (q, q') \in E)$$

We set $Q_1 := Q \setminus Q_0$
A play is a sequence $\rho = r_0 r_1 r_2 \ldots$ with $(r_i, r_{i+1}) \in E$

It is built up starting in $\rho(0)$ as follows: If $\rho(i) \in Q_0$ then player 0 chooses an edge $(\rho(i), q)$, otherwise player 1 picks such an edge, leading to $\rho(i + 1) = q$

Intuitively, a token is moved from vertex to vertex via edges:

From $Q_0$-vertices player 0 moves the token, from $Q_1$-vertices player 1 moves the token.
A **strategy** for player 0 from $q$ is a function $f : Q^+ \rightarrow Q$, specifying for any play prefix $q_0 \ldots q_k$ with $q_0 = q$ and $q_k \in Q_0$ some vertex $r \in Q$ with $(q_k, r) \in E$

A play $\rho = q_0q_1\ldots$ from $q_0 = q$ is **played according to strategy** $f$ if for each $q_i \in Q_0$ we have $q_{i+1} = f(q_0 \ldots q_i)$

A strategy $f$ for player 0 from $q$ is called **winning strategy** for player 0 from $q$ if any play from $q$ which is played according to $f$ is won by player 0 (according to the winning condition).

In the analogous way, one introduces strategies and winning strategies for player 1.

Player 0 (resp. 1) **wins from** $q$ if s/he has a winning strategy from $q$
Winning Regions

For a game $\Gamma = (G, \varphi)$ with $G = (Q, Q_0, E)$, the winning regions of players 0 and 1 are the sets

$$W_0 := \{ q \in Q \mid \text{player 0 wins from } q \}$$
$$W_1 := \{ q \in Q \mid \text{player 1 wins from } q \}$$

Remark: Each vertex $q$ belongs at most to $W_0$ or $W_1$.

Example: Consider game graph above. Winning condition requires that vertex 3 is reached.

$$W_0 = \{3\}, \quad W_1 = \{1, 2, 4, 5, 6, 7\}$$
Determinacy

In general, the winning regions $W_0$, $W_1$ of players 0 and 1 satisfy $W_0 \cap W_1 = \emptyset$

A game is called determined if from each vertex either of the two players has a winning strategy.

Remark:

1. There are (exotic) games which are not determined.
2. In descriptive set theory one investigates which abstract winning conditions define determined games.
3. All games in this tutorial will be determined. (They are “Borel games”.)
Three Fundamental Questions

Given a game \( \Gamma = (G, \varphi) \), \( G = (Q, Q_0, E) \)

1. Decide for each \( q \in Q \) whether \( q \in W_0 \) (i.e. whether player 0 wins from \( q \))

2. In this case:
   Construct a suitable winning strategy from \( q \) (in the form of a program)

3. Optimize the construction of the winning strategy (e.g., time complexity) or optimize parameters of the winning strategy (e.g., size of memory).

Solving a game means to provide algorithms for 1. and 2.
Special Strategies

If $Q$ is finite, then a strategy is a word function $f : Q^+ \rightarrow Q$

There are three basic types of strategies:

1. A natural requirement: $f$ is computable (computable or recursive strategy)

2. More restrictive requirement: $f$ is a finite-state strategy or computable by a finite automaton (automaton strategy, which allows to use “bounded information” from the past)

3. Simplest kind of strategy: Current vertex determines choice of next vertex (positional strategy)

Other types: pushdown strategy, counter strategy etc.
Positional Strategies

Given $G = (Q, Q_0, E)$

A strategy $f : Q^+ \to Q$ (for player 0) is called positional (sometimes also called local or memoryless) if the value $f(q_0 \ldots q_k)$ only depends on $q_k \in Q_0$.

A positional strategy for 0 is representable as

1. a function $F : Q_0 \to Q$ or
2. a set $E_0$ of edges of the game graph, containing for every $q \in Q_0$ precisely one edge with source vertex $q$ (and for $q \in Q_1$ all edges from $E$ with source vertex $q$).
Finite-State Strategies

A finite strategy automaton for player 0 over $G = (Q, E)$ is a Mealy-automaton $\mathcal{A} = (S, Q, s_0, \sigma, \tau)$ with

- finite memory $S$, input alphabet $Q$
- initial memory content $s_0 \in S$
- memory update function $\sigma : S \times Q \to S$
- transition choice function $\tau : S \times Q_0 \to Q$

Extend $\sigma$ to a function $\sigma : S \times Q^* \to S$ by

$$\sigma(s, \varepsilon) = s, \quad \sigma(s, wq) = \sigma(\sigma(s, w), q)$$

$\sigma(s, w)$ is the memory content reached from $s$ by input $w$.

The strategy computed by $\mathcal{A}$ is the function $f_\mathcal{A}$ with

$$f_\mathcal{A}(q_0 \ldots q_k) := \tau(\sigma(s_0, q_0q_1 \ldots q_{k-1}), q_k)$$
Application I: Reactive Systems

- Application areas: communication protocols, control systems
- Use game graph to model reactive system
- Use temporal logics or modal $\mu$-calculus as specification logic for the winning condition
- Include non-discrete aspects (time, probability)

However, the core theory originating from Church’s Problem is still fundamental.

Main contributions by

- McNaughton, Büchi, Landweber, Rabin in the 1960’s
- Gurevich, Harrington, Emerson, Jutla, Mostowski in the 1980’ and 1990’s
Application II: Model-Checking

Question:

Given Kripke structure $\mathcal{K}$ and formula $\varphi$ (e.g. of modal $\mu$-calculus), decide whether $\mathcal{K} \models \varphi$

Combine $\mathcal{K}$ and $\varphi$ into a “product game graph” $G_{\mathcal{K},\varphi}$ with the so-called “parity winning condition” such that

$\mathcal{K} \models \varphi$ iff

from a designated vertex in $G_{\mathcal{K},\varphi}$, player 0 has a (positional) winning strategy.
Application III: Automata on Infinite Trees

Nondeterministic tree automaton (e.g. Rabin tree automaton) generates run on input tree and accepts if all paths of the run satisfy the acceptance condition.

Existence of successful run corresponds to existence of winning strategy in a game.

Complementation of (e.g.) Rabin tree automata can be solved with a determinacy theorem on the game:

From non-existence of accepting run of given automaton conclude the existence of accepting run for the complement automaton.

Determinacy helps.
Simple Games
Simple Winning Conditions

For plays $\rho = \rho(0)\rho(1) \ldots$ we fix the winning condition using a subset $F \subseteq Q$ of the game graph.

Consider four basic winning conditions:

- **Reachability**: $\exists i : \rho(i) \in F$
- **Safety**: $\forall i : \rho(i) \in F$
- **Recurrence (Büchi condition)**: $\forall i \exists j > i : \rho(j) \in F$
- **Persistence (co-Büchi condition)**: $\exists i \forall j > i : \rho(j) \in F$

We speak of reachability, safety, Büchi, and co-Büchi games. All of these can be solved with positional strategies.

Wolfgang Thomas
Given a finite game graph $G = (Q, Q_0, E)$, $F \subseteq Q$ with the winning condition

player 0 wins $\rho :\iff \exists i \rho(i) \in F$

Then the winning regions $W_0, W_1$ of players 0 and 1 are computable, as well as corresponding positional winning strategies.

**Proof:** Define

$$Attr^i_0(F) := \{ q \in Q \mid \text{from } q \text{ player 0 can force a visit of } F \text{ in } \leq i \text{ moves} \}$$
Example
Computing the Attractor

Inductive construction of $Attr^i_0(F)$:

$$Attr^0_0(F) = F,$$
$$Attr^{i+1}_0(F) = Attr^i_0(F) \cup \{q \in Q_0 \mid \exists (q, r) \in E : r \in Attr^i_0(F)\} \cup \{q \in Q_1 \mid \forall (q, r) \in E : r \in Attr^i_0(F)\}$$

Then

$$Attr^0_0(F) \subseteq Attr^1_0(F) \subseteq Attr^2_0(F) \subseteq \ldots$$

Since $Q$ is finite, for some $l \leq |Q|$: $Attr^l_0(F) = Attr^{l+1}_0(F)$

so the sequence becomes stationary at

$Attr_0(F) := \bigcup_{i=0}^{\left|Q\right|} Attr^i_0(F)$

Easy: $W_0 = Attr_0(F)$ and $W_1 = Q \setminus Attr_0(F)$
Attractor Strategy

Positional winning strategies for player 0/1 on $W_0$ resp. $W_1$:

- On $\text{Attr}_0(F) \setminus F$ player 0 picks edges which decrease the distance to $F$
- On $Q \setminus \text{Attr}_0(F)$ player 1 picks edges which lead again into $Q \setminus \text{Attr}_0(F)$

Remarks:

1. Over infinite graphs with infinite branching, the number of iteration steps may be transfinite.
2. The attractor construction remains effective over certain infinite graphs, notably the pushdown graphs.
Parity Games
Parity Condition

is a boolean combination of Büchi conditions.

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

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

One can formulate this using a chain of subsets (as it occurs in the so-called ”difference hierarchy” introduced by Hausdorff):

Let $F_i = \{q \in Q \mid c(q) \geq 2i\}$
and $E_i = \{q \in Q \mid c(q) \geq 2i + 1\}$

Note $F_0 \supseteq E_0 \supseteq F_1 \supseteq E_1 \ldots \supseteq F_{2k} \supseteq E_{2k+1}$

Parity condition says:

$\forall_i (\text{Inf}(\rho) \cap F_i \neq \emptyset) \land (\text{Inf}(\rho) \cap E_i = \emptyset)$
Felix Hausdorff
Solution of Parity Games

**Theorem** (Emerson-Jutla 1991)

- Parity games are determined (i.e., each vertex belongs to $W_0$ or $W_1$), 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.

The proof of the first part (for finite game graphs) proceeds by induction on the size of the game graph (see e.g. W. Thomas, Proc. STACS 1995, LNCS 900).
Towards an Algorithmic Solution

We use a nondeterministic algorithm:

1. Guess $W_0$ and $W_1$ and positional strategies given by edge sets: $E_0$ with one out-edge from $Q_0$-vertices, and $E_1$ with one out-edge from $Q_1$-vertices.

2. Check that $E_0$ is a uniform winning strategy from each $q \in W_0$ and that $E_1$ is a uniform winning strategy from each $q \in W_1$

**Step 1:** is done in nondeterministic polynomial time (or by exhaustive search of all $2^{|Q|} \cdot 2^{|Q|^2}$ possibilities)

**Step 2:** Check whether a given positional strategy is a winning strategy for player 0 from $q$. 
Step 2: Checking Strategies

Remark: For a fixed positional strategy $f$ of player 0 one can decide in polynomial time for any $q \in Q$, whether $f$ is a winning strategy from $q$

Proof:
Consider the graph $G(f)$ given by the positional strategy $f$:
Each $Q_0$-vertex has precisely one out-edge given by $f$, each $Q_1$-vertex has all out-edges which are present in $G$.
Player 1 wins from $q$ against strategy $f$ iff in $G(f)$ there is a path from $q$ to a loop whose highest color is odd.
For odd $m \leq k$ define $G_m$ as the graph $G$ restricted to vertices with color $\leq m$

Player 1 wins from $q$ against strategy $f$ iff there is an odd $m \leq k$ and a path in $G(f)$ from $q$ to an SCC of $G_m$ with color $m$

For each $m$ this can be checked by the SCC detection algorithm.
Decision Problem “Parity Game”

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

**Question:** Does player 0 win the corresponding parity game from $q$?

(Short: “$q \in W_0$ 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 $NP$. The complementary problem “$q \not\in W_0?$” is also in $NP$ (by determinacy switch the players).

**Open Problem:** Is ”Parity Game” in $P$?

Wolfgang Thomas
From Logical Winning Conditions to Muller Games
LTL over $\omega$-Sequences

Consider $\omega$-words $\alpha \in \{0, 1\}^\omega$ for some $n$, e.g.,

$$\alpha = (1) \ (1) \ (0) \ (1) \ (0) \ (0) \ (0) \ldots$$

Plays are now coded this way.

Use propositional variables $p_1, \ldots, p_n$ to express the bit values in the components.
Linear-time Temporal Logic LTL

The LTL-formulas over the atomic propositions $p_1, \ldots, p_n$ are inductively defined as follows:

- $p_i$ is an LTL-formula
- if $\varphi, \psi$ are LTL-formulas, then so are $\neg \varphi, \varphi \lor \psi, \varphi \land \psi, \varphi \rightarrow \psi$
- if $\varphi, \psi$ are LTL-formulas, then so are
  - $X\varphi$ “nexttime $\varphi$”
  - $F\varphi$ “eventually $\varphi$”
  - $G\varphi$ “henceforth $\varphi$”
  - $\varphi U \psi$ “$\varphi$ until $\psi$”

Winning condition $\varphi$ is satisfied for $\rho$ :iff $\rho \models \varphi$. 
Examples

- $GFp_1$ “infinitely often 1 in first component”
- $XX(p_1 \rightarrow Fp_2)$
  “if after two steps 1 occurs in first component, then therfe or sometime after also in the second component”
- $F(p_1 \land X(\neg p_2 Up_1))$
  “sometime 1 in first component such that this is true later again and in between 0 in second component”
A Stronger Logic: S1S

Recall the LTL-formulas:

- $GFp_1$
- $XX(p_1 \rightarrow Fp_2)$
- $F(p_1 \land X(\neg p_2 U p_1))$

Corresponding S1S-formulas:

$$\varphi_1(X_1) : \forall s \exists t (s < t \land X_1(t))$$
$$\varphi_2(X_1, X_2) : X_1(0'') \rightarrow \exists t (0'' \leq t \land X_2(t))$$
$$\varphi_3(X_1, X_2) : \exists t_1 (X_1(t_1) \land \exists t_2 (t'_1 \leq t_2 \land X_1(t_2) \land \forall t ((t'_1 \leq t \land t < t_2) \rightarrow \neg X_2(t))))$$

Also set quantifiers are allowed ("monadic second-order logic").
LTL vs. S1S vs. Büchi Automata

LTL-formulas can be converted into S1S-formulas, but the converse fails in general.

For each S1S-formula $\varphi(X_1, \ldots, X_n)$ (defining a language $L \subseteq \{0, 1\}^n$) one can construct an equivalent Büchi automaton, and conversely für each Büchi automaton one can construct an equivalent S1S-formula.

(A Büchi automaton is a nondeterministic finite automaton used with the ”Büchi acceptance” condition for infinite words, namely that there is an infinite run over the given $\omega$-words that visits an accepting state infinitely often.)
McNaughton’s Theorem: Muller Automata

A nondeterministic Büchi automaton over $\Sigma$ can be transformed into a deterministic Muller automaton

$$
\mathcal{A} = (Q, \Sigma, q_0, \delta, \mathcal{F})
$$

where $Q, \Sigma, q_0, \delta$ are as usual for deterministic finite automata, and $\mathcal{F}$ is a collection of subsets $P \subset Q$.

The Muller automaton accepts an input $\omega$-word if in the unique run $\rho$ over this word the infinitely often occurring states form a set in $\mathcal{F}$.

One often writes $\text{Inf}(\rho) \in \mathcal{F}$. 
Muller Games

Applied to input words that are plays we may distinguish the states from which a letter of player 0 is processed from those where a letter of player 1 is processed.

Then from a Muller automaton we get a Muller game graph:
The transition graph is then a game graph, with ”Muller winning condition given by the item $\mathcal{F}$. 
Transformation of Muller Games into Parity Games
A Muller Game

Player 2 wins if the infinitely often visited states are:
{1, 2, 3, 4} or {1, 2, 3, 4, 5} or {1, 3, 4, 5} or {1, 4}

This is a Muller game (standard form of regular infinite games).

Wolfgang Thomas
From Muller to Parity Games

Theorem (Simulation Theorem):

For a game graph \( G = (Q, Q_0, E) \) and Muller winning condition (given by \( \mathcal{F} \subseteq 2^Q \)), there is a parity game over a graph \( G' = (Q', Q'_0, E') \) with a coloring \( c \) of \( Q' \) such that a play \( \rho \) over \( G \) induces a play \( \rho' \) over \( G' \) where \( \rho \) satisfies the Muller condition iff \( \rho' \) satisfies the parity condition.

Then a solution of the parity game over \( G' \) will yield a solution of the Muller game over \( G \).
The Idea

A latest appearance record over $Q = \{1, \ldots, n\}$ is a permutation

$$(i_1, \ldots, i_h, \ldots i_n)$$

of $(1, \ldots, n)$ with a pointer to position $h$.

We use the latest appearance records as vertices of the new game graph $G'$.

$Q' := \text{LAR}(Q)$

Definition of $E'$:

Introduce edges

from $(i_1 \ldots i_h \ldots i_n)$ to $(i_m i_1 \ldots i_{m-1} i_{m+1} \ldots i_n)$

if $(i_1, i_m) \in E$
Essential Observation

- Let \( i_0, i_1, i_2, \ldots \) be an infinite play over \( Q \) and \((i_0, \ldots), (i_1, \ldots)(i_2, \ldots)\) be the corresponding play over \( \text{LAR}(Q) \).

- Let \( h \) be the maximal position such that underlining at \( h \) occurs again and again in this play, say from time \( t \) onwards.

- Then after \( t \) in a \( \text{LAR} \left( i_1 \ldots \underline{i_h} \ldots i_n \right) \) the states \( i_{h+1}, \ldots, i_n \) stay fixed (and are never visited again)
  and precisely the states \( i_1, \ldots, i_h \) will be visited again and again

- \( \{i_1, \ldots, i_h\} \) is the set of states visited infinitely often in \( \rho \), and player 0 wins \( \rho \) if \( \{i_1, \ldots, i_h\} \in \mathcal{F} \)
Defining the colors

Coloring \( c : \text{LAR}(Q) \to \{1, \ldots, 2n\} \):

\[
c(i_1 \ldots i_h \ldots i_n) = \begin{cases} 
2h - 1 & \text{if } \{i_1, \ldots, i_h\} \not\in \mathcal{F} \\
2h & \text{if } \{i_1, \ldots, i_h\} \in \mathcal{F}
\end{cases}
\]

Then for a play \( \rho \) and the corresponding play \( \rho' \) over \( \text{LAR}(Q) \):

the set of infinitely often visited states in \( \rho \) belongs to \( \mathcal{F} \)

iff

the maximal color occurring infinitely often in \( \rho' \) is even
Application of Game Reduction

The parity game over $G'$ can be solved with positional strategies.

**Consequence:**

If player 0 wins over $G'$ from $(i_1 \ldots i_n)$ by a positional winning strategy, then player 0 wins the Muller game over $G$ by a finite-state strategy, where the memory is the set $\text{LAR}(Q)$. 
Summary

We arrive at the "Büchi-Landweber Theorem":

A game with S1S-definable winning condition is determined, one can compute who wins, and one can construct a finite-state winning strategy for the winner.

Recall the four steps (which can be merged to some extent for better efficiency):

1. Transform the S1S-formula into a Büchi automaton.
2. Determinize the Büchi automaton to obtain a Muller automaton and thus a game graph with Muller winning condition.
3. Transform the Muller game into a parity game.
4. Solve the parity game with positional strategies.

Wolfgang Thomas
Trends in the Theory of Infinite Games

- Generalizations of the game model:
  
  Infinite-state, concurrent, stochastic, timed, weighted, distributed, multi-player games

- Here we concentrate on certain quantitative aspects: Construction of “good” strategies
Part 2

Quantitative Aspects
A strategy may be called “good” if . . .

- it is memory-efficient,
  i.e., requires a small amount of memory,
- it is generous
  i.e., discloses moves in advance to the opponent,
- it is simple
  i.e., is definable with weak logical means,
- it is finite-time
  i.e., pursues just a safety objective and wins infinite plays in finite time.
- it is responsive
  i.e., serves requests quickly,
- it is permissive
  i.e., allows several choices (nondeterministically),
Strategies with Lookahead
The Idea

If a player discloses at time $i$ his moves for times $i + 1, \ldots, f(i)$ then the opponent has a corresponding lookahead.

Question:
If Player 0 wins the standard game, for which rates $f$ of generosity can he still win?
At which rates will the game be won by the opponent?

Recall: $F : \alpha \mapsto \beta$ is continuous (in the Cantor topology over the space of infinite sequences) if

$\beta(i)$ is determined by a finite prefix $\alpha(0) \ldots \alpha(j)$ of $\alpha$

Strategies with given lookahead rate are just uniformly continuous functions.
Two Results

A Collapse Result:

In a regular infinite game, a player wins with any lookahead $f$ iff he wins with a constant lookahead $g(i) = i + k$, and the minimal such $k$ can be computed.

So the possibility to win with lookahead can be decided.

A complementary result:

For context-free infinite games, the possibility to win with lookahead is undecidable, and in this case the lookahead cannot be bounded by any elementary function.

(Fridman, Holtmann, Kaiser, Löding, Th., Zimmermann 2009/10)
Definability of Strategies
Strategies and Definability

A strategy for Player 2 is a map
\[
(\alpha(0) \beta(0)) (\alpha(1) \beta(1)) \ldots (\alpha(k) \beta(k)) \mapsto 0/1
\]

Strategies can be defined by sentences interpreted over such finite play prefixes,
in the sense that the truth value is the bit to be chosen.

The Büchi-Landweber Theorem says:
MSO-definable games can be solved with MSO-definable strategies.

General Problem: Relate the logic for describing winning strategies to the logic used to define the game.

Wolfgang Thomas
Some Results

- Games definable in LTL can be solved with strategies definable again in LTL.
- This fails for fragments of LTL (e.g. by bounding the nesting of temporal operators).
- For games defined in Presburger arithmetic, even full first-order arithmetic is insufficient for defining winning strategies.
Winning Muller Games in Finite Time
Winning Games by Scores

Given a Muller game with the collection $\mathcal{F}$ of “winning loops” for Player 0
play this like a card game in the evening
... and of course go to sleep at some time.

Question: How can one terminate a play after finite time, declaring correctly the winner?

McNaughton’s approach: Count for each loop $F$ how often the loop $F$ (as a set) was completely traversed without interruptions.

At time $i$ denote this repetition number as $\text{score}_F(i)$. 
A player wins a Muller game iff he has a strategy which guarantees that score 3 is reached for one of his winning loops.

(McNaughton had shown this with the bound $n!$ for a game graph with $n$ vertices.)

This can be achieved by just ensuring that the opponent’s scores stay $\leq 2$.

This amounts to solve a “safety game”.

(Fearnley, Neider, R. Rabinovich, Zimmermann (2010/11))

We obtain a new approach to strategy construction — and to strategy optimization.
Responsive Strategies
Optimality in Request-Response Games

Game arena $G = (Q, Q_0, E)$

Subsets $Rqu_1, \ldots, Rqu_k \subseteq Q$: “Requests”

Subsets $Rsp_1, \ldots, Rsp_k \subseteq Q$: “Responses”

RR-condition:

$$\bigwedge_{i=1}^{k} \forall s (Rqu_i(s) \rightarrow \exists t (s < t \land Rsp_i(t)))$$

LTL:

$$\bigwedge_{i=1}^{k} G (Rqu_i \rightarrow XF Rsp_i)$$
Standard Solution of RR-Games

- It suffices to keep a memory for the set of "open requests"
  Memory size: $2^k$ for $k$ conditions

- Reduction to Büchi games

- Result: Winning strategy which ensures bounded waiting
time between request and response
  (Bound $B := k \cdot |Q|$).

How to optimize better than just by measuring the maximum of waiting times?
Measuring Quality of Solution

- Linear Penalty model:
  For each moment of waiting (for each RR-condition) pay 1 unit

- Quadratic Penalty model:
  For the $i$-th moment of waiting pay $i$ units

Activation of $i$-th condition in a play $\varrho$ is a visit to $Rqu_i$ such that all previous visits to $Rqu_i$ are already matched by an $Rsp_i$-visit.
Values of Plays and Strategies

For both linear and quadratic penalty define:

- $w_\varrho(n) = \text{sum of penalties in } \varrho(0) \ldots \varrho(n) \text{ divided by number of activations}$
  
  "average penalty sum per activation"

- $w(\varrho) = \lim_{n \to \infty} \sup w_\varrho(n)$

Given a strategy $\sigma$ for controller and a strategy $\tau$ for adversary

- $\varrho(\sigma, \tau) := \text{the play induced by } \sigma \text{ and } \tau$

- $w(\sigma) := \sup_{\tau} w(\varrho(\sigma, \tau))$

Call $\sigma$ optimal if there is no other strategy with smaller value.
On the Linear Penalty

For the linear penalty model, a finite-state optimal strategy does not exist in general:

$Rqu_1, Rqu_2$

$Rsp_1$

$Rsp_2$
Theorem

For the quadratic penalty function (in fact, for any strictly increasing divergent penalty) one can decide whether a RR-game is won by controller and in this case one can compute a finite-state optimal winning strategy.

(Horn, Ths. Wallmeier, ATVA 2008)

Proof ingredients:

- It suffices to consider strategies with value $\leq M$ (induced by bounded waiting time of standard solution).
- Conversely: For strategies with value $\leq M$ one can assume bounded waiting time.
- Reduction to mean-payoff games.
Building a Mean-Payoff Game

From a game graph $G = (Q, Q_0, E)$ with $k$ conditions proceed to a game graph over $Q \times \mathbb{N}^k$

State format: $(q, n_1, \ldots, n_k)$

$n_i =$ current waiting time for $i$-th condition since last activation

Derived mean-payoff game:

For each edge $e = (p, m) \rightarrow (q, n)$ introduce edge weight

$w(e) = n_1 + \ldots + n_k$ (sum of current penalties)
Boundedness Lemma

Let $\sigma$ be a winning strategy of value $\leq M$

Then one can construct a winning strategy $\sigma'$ with bounded waiting times such that $w(\sigma') \leq w(\sigma)$.

Consequence:

In the mean-payoff game, it suffices to consider waiting time vectors in a domain $[0, B]^k$ rather than $\mathbb{N}^k$.

So we obtain a finite MPG which can be solved.
Permissive Strategies
Definitions

Model: Game on finite graph with the parity winning condition. After a finite play prefix, a strategy just blocks certain edges.

Idea:

- A strategy should narrow the system’s behaviour as little as possible.
- This supports modular constructions: Adding requirements leads to a refinement of strategies.
New Penalties

For strategy $\sigma$ (of Player 0):

- After a play prefix $q(0) \ldots q(n)$ it is the total number of edges blocked by $\sigma$ up to time $n$. Call this $\pi_\sigma(q(0) \ldots q(n))$.
- Penalty of complete play $q$ is the lim sup over the average values $\frac{1}{n} \cdot \pi_\sigma(q(0) \ldots q(n))$.
- Penalty associated with strategy $\sigma$: supremum of penalties of plays consistent with $\sigma$.
- Permissiveness value of game = infimum of this value over all winning strategies of Player 2.
For parity games on finite graphs, the permissiveness value is computable.

There are games where this value is not realizable by a finite-state strategy.

(Bouyer, Markey, Olschewski, Ummels 2010)

Open: Finite presentation of more general strategies such that computability of most permissive strategies is possible.
Mean Payoff Games
The games studied in the first part of this tutorial were win-lose games.

In quantitative games a value is associated to each play.

Usually, one player tries to maximize and the other player tries to minimize the value.
For a finite play $v_0 \cdots v_n$ we are interested in the mean value

$$\frac{1}{n} \cdot \sum_{i=0}^{n-1} r(v_i, v_{i+1})$$

In the limit, Player 0 tries to maximize and Player 1 tries to minimize this value.
A mean payoff game is of the form $G = (Q, Q_0, E, r)$ where $(Q, Q_0, E)$ is a finite game graph as we know it, and

$$r : E \rightarrow \mathbb{Z}$$

is a function assigning a reward to each edge.

As usual the players built up a play $\pi = v_0 v_1 v_2 \cdots$ where

- Player 0 tries to maximize
  $$r_0(\pi) := \lim_{n \to \infty} \inf \left( \frac{1}{n} \cdot \sum_{i=0}^{n-1} r(v_i, v_{i+1}) \right)$$

- Player 1 tries to minimize
  $$r_1(\pi) := \lim_{n \to \infty} \sup \left( \frac{1}{n} \cdot \sum_{i=0}^{n-1} r(v_i, v_{i+1}) \right)$$
Remark. The values \( r_0(\pi) \) and \( r_1(\pi) \) can be different.

Remark. Both values are the same for ultimately periodic plays
\[ \pi = v_0 \cdots v_i (v_{i+1} \cdots v_n)^\omega \]
with \( v_i = v_n \):

\[
r_0(\pi) = r_1(\pi) = \frac{1}{n-i} \sum_{j=i}^{n-1} r(v_j, v_{j+1})
\]
Strategies

Strategies for Player $i$ are as before mappings $\sigma : V^* V_i \rightarrow V$.

For two strategies $\sigma$ and $\tau$ of Player 0 and Player 1, respectively, and a starting vertex $v$ we denote by $\pi_{\sigma,\tau,v}$ the unique play starting in $v$ and played according to $\sigma$ and $\tau$.

The Player 0 value of the game from $v$ is

$$\text{val}_0(v) := \sup_{\sigma} \inf_{\tau} r_0(\pi_{\sigma,\tau,v}),$$

the Player 1 value of the game from $v$ is

$$\text{val}_1(v) := \inf_{\tau} \sup_{\sigma} r_1(\pi_{\sigma,\tau,v}),$$

where $\sigma$ ranges over Player 0 strategies and $\tau$ over Player 1 strategies.

Remark. $\text{val}_0(v) \leq \text{val}_1(v)$
Theorem. For each finite mean payoff game there are positional strategies $\sigma^*$ and $\tau^*$ for Player 0 and Player 1, respectively, such that for each vertex $v$

\[
\text{val}_0(v) = \sup_{\sigma} \inf_{\tau} r_0(\pi_{\sigma,\tau,v}) \\
= \inf_{\tau} r_0(\pi_{\sigma^*,\tau,v}) \\
= \sup_{\sigma} r_1(\pi_{\sigma,\tau^*,v}) \\
= \inf_{\tau} \sup_{\sigma} r_1(\pi_{\sigma,\tau,v}) \\
= \text{val}_1(v)
\]

In particular, the values of the game for the two players are the same and we can simply write $\text{val}(v)$. 
From Parity Games to Mean Payoff Games

**Theorem.** For each parity game $G$ one can construct a mean payoff game $G'$ over the same game graph such that for each vertex $v$ Player 0 has a winning strategy in $G$ from $v$ iff $\text{val}(v) \geq 0$ in $G'$.

**Construction:**

- Let $n$ be the number of vertices of $G$.
- Let $(u, v)$ be an edge of $G$ and $p$ be the priority of $u$.
- Define $r(u, v) := \begin{cases} n^p & \text{if } p \text{ is even} \\ -n^p & \text{if } p \text{ is odd} \end{cases}$
Proof

We can restrict to positional strategies in both games.

Consider a play determined by two positional strategies:

\[ \pi = v_0 \ldots v_i (v_{i+1} \ldots v_k) \omega \]

In the mean payoff game \( G' \) the value is

\[ \frac{1}{k-i} \sum_{j=i}^{k-1} r(v_j, v_{j+1}) \]

This is positive iff \( \sum_{j=i}^{k-1} r(v_j, v_{j+1}) \) is positive

Note that \( k - i \leq n \). In case highest color in loop is even:

\[ \sum_{j=i}^{k-1} r(v_j, v_{j+1}) \geq n^p - (n - 1)n^{p-1} > 0 \]

Similar for odd highest color.
Determinacy of Mean Payoff Games

We recall:

**Theorem.** For each finite mean payoff game there are positional strategies $\sigma^*$ and $\tau^*$ for Player 0 and Player 1, respectively, such that for each vertex $v$

\[
\text{val}_0(v) = \sup_{\sigma} \inf_{\tau} r_0(\pi_{\sigma,\tau},v) \\
= \inf_{\tau} r_0(\pi_{\sigma^*,\tau},v) \\
= \sup_{\sigma} r_1(\pi_{\sigma,\tau^*},v) \\
= \inf_{\tau} \sup_{\sigma} r_1(\pi_{\sigma,\tau},v) \\
= \text{val}_1(v)
\]

In particular, the values of the game for the two players are the same and we can simply write $\text{val}(v)$. 
Proof Steps

- **We use a finitary variant of the game** that stops as soon as a cycle is closed. The value of such a finite play is the mean value on the cycle.

- **Finite duration games** (zero sum with perfect information) are determined / have a value.

- **First main step:** The value of the infinite duration game is the value of the finite duration game. This shows the determinacy of the game.

- **Second main step:** By induction on the number of edges for Player 0 we show that positional strategies are enough (symmetrically for Player 1).

Wolfgang Thomas
Finite Duration MPG

Let \( \mathcal{G} = (Q, Q_0, E, r) \) be a mean payoff game (MPG).

The game \( \mathcal{G}_{\text{fin}} \) is defined over the same graph. A play stops as soon as the first vertex repeats, i.e., it is of the form \( \beta = v_0 \cdots v_n \) with \( v_n = v_i \) for some \( i < n \), and \( v_0, \ldots, v_{n-1} \) are all pairwise different.

The value of such a finite play is the mean value of the loop:

\[
r_{\text{fin}}(\beta) := \frac{1}{n - i} \sum_{j=i}^{n-1} r(v_j, v_{j+1})
\]

The notion of strategy is the same as before. For two strategies \( \sigma \) and \( \tau \) of the two players and a vertex \( v \) we denote by \( \beta_{\sigma, \tau, v} \) the unique (finite) play of \( \mathcal{G}_{\text{fin}} \) from \( v \) using \( \sigma, \tau \).
Determinacy of Finite Duration MPGs

Below $\sigma$ ranges over strategies for Player 0 and $\tau$ over strategies of Player 1.

**Theorem.** Let $\mathcal{G} = (Q, Q_0, E, r)$ be an MPG and let $v$ be a vertex. Then

$$\max_\sigma \min_\tau r_{\text{fin}}(\beta_{\sigma,\tau,v}) = \min_\tau \max_\sigma r_{\text{fin}}(\beta_{\sigma,\tau,v}) \equiv: \text{val}(v, \mathcal{G}_{\text{fin}}).$$

**Proof idea:** From the finite tree of all possible plays one can derive the value and corresponding strategies from the leaves to the root.
Terminology

We say that a strategy $\sigma$ for Player 0 secures value $a$ from $v$ (in $G$ or $G_{\text{fin}}$) if every play that is played according to $\sigma$ from $v$ has value at least $a$.

The same terminology applies to Player 1 if he has a strategy that ensures that the value of the resulting plays is at most $a$. 
From Finite to Infinite Duration

Let $\sigma$ be a strategy for one of the players in $G_{\text{fin}}$. We define the strategy $\hat{\sigma}$ for $G$ as follows:

$$\hat{\sigma}(v_0 \cdots v_n) = \sigma(u_0 \cdots u_m)$$

where $u_0 \cdots u_m$ is obtained by removing the loops from $v_0 \cdots v_n$: find the smallest $j$ such there is $i < j$ with $v_i = v_j$, and remove $v_{i+1} \cdots v_j$. Continue until no more loops are left.

Lemma. If $\sigma$ secures value $a$ from $v$ in $G_{\text{fin}}$ for one of the players, then $\hat{\sigma}$ secures value $a$ from $v$ in $G$ (for the same player).

Theorem. Let $G$ be a mean payoff game and $v$ be a vertex of $G$. Then $\text{val}_0(v) = \text{val}_1(v) = \text{val}(v, G_{\text{fin}})$. 

Wolfgang Thomas
Closing Remarks
Parity and mean payoff games are basic classes of infinite games – for "regular objectives", respectively "quantitative objectives".

Even for the (classical) infinite two-person games, we have not yet understood completely how to construct strategies that are “good” — and it is even less clear how to handle multiple optimization criteria.

A fundamental problem: Is there a compositional framework of strategy construction which reflects the structure of the (logical) specifications and works without the detour through automata theory (algorithmic theory of labelled graphs)?