Strategies in a Logical Setting

Wolfgang Thomas

RWTH AACHEN

Francqui Lecture, Mons, April 2013

General Perspective: Uniformization

Games and Uniformization

A game specification $\varphi(X, Y)$ defines a relation

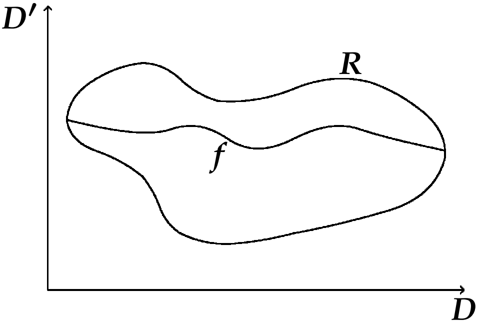
$$R_\varphi = \{(\alpha, \beta) \mid (\mathbb{N}, +1) \models \varphi[P_\alpha, P_\beta]\}$$

The question: Is there a function $F : \alpha \mapsto \beta$ computable by a finite-state strategy such that

$$\forall \alpha \; R_\varphi(\alpha, F(\alpha))$$

This is a "uniformization problem".

Illustration



Two Examples over Finite Words

- **Recursively enumerable relations**
  (whose elements (u, v) are enumerated by a procedure)
- **Rational relations**
  (whose elements (u, v) are accepted by a nondeterministic transducer)

A function f is recursive iff its graph $\{(u, f(u)) \mid u \in \mathrm{dom}(f)\}$ is r.e.

A function is rational iff its graph $\{(u, f(u)) \mid u \in \mathrm{dom}(f)\}$ is rational.

R.E. Relations

Assume R is enumerated as

$(u_0, v_0), (u_1, v_1), (u_2, v_2), \ldots$

Define a function f as follows

$f(u)$ is v_i for the first i such that $u = u_i$,

otherwise undefined.

So:

R.e. relations are uniformizable by recursive functions.

A more complicated construction shows:

Rational relations are uniformizable by rational functions.

S1S and S2S

Theorem (Siefkes 1975): **An S1S-definable relation is uniformizable by an S1S-definable function.**

Example:

$\varphi(X,Y)$: "Y starts with 1 iff there are infinitely many 1 in X".

$\psi(X,Y) : (\exists^{\omega} z X(z) \wedge \forall t Y(t)) \vee (\neg \exists^{\omega} z X(z) \wedge \forall t \neg Y(t))$

Theorem (Gurevich-Shelah 1983, Carayol-Löding 2007):

In general, an S2S-definable relation over trees is not unifomizable by an S2S-definable function.

For games, we are dealing with special functions (realized by strategies); they are "online-computable".

Logical Definability of Strategies

Strategies

A strategy for Player 1 is a map

$$\binom{P(0)}{Q(0)} \binom{P(1)}{Q(1)} \ldots \binom{P(k)}{Q(k)} \mapsto \mathbf{0/1}$$

A strategy for Player 2 is a map

$$\binom{P(0)}{Q(0)} \binom{P(1)}{Q(1)} \ldots \binom{P(k)}{*} \mapsto \mathbf{0/1}$$

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.

Definability of Strategies

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

is MSO-definable iff there is an MSO-formula $\psi(X, Y, z)$ **such that**

$$([0,k],<) \models \psi(P \cap [0,k]),\ (Q \cap [0,k-1]),\ k)$$

iff

$$f(\binom{P(0)}{Q(0)} \ldots \binom{P(k-1)}{Q(k-1)} \binom{P(k)}{*}) = 1$$

Büchi, Elgot, Trakhtenbrot:
Finite-state strategies are MSO-definable.

Büchi-Landweber Theorem

For each MSO-requirement $\varphi(X, Y)$ either Player 1 or Player 2 has a finite-state winning strategy.

It is decidable who wins, and a finite-state winning strategy for the respective winner is computable.

A comment:

More generally, the following type of game problem is naturally suggested by automata theory. Given a class of games G: (1) can one effectively decide, for any $\mathfrak{C} \in G$, which player has a winning strategy? (2) Just how simple winning strategies do exist for games in G? For example, is there a recursive or even a finite automata winning strategy for $\mathfrak{C} \in G$?

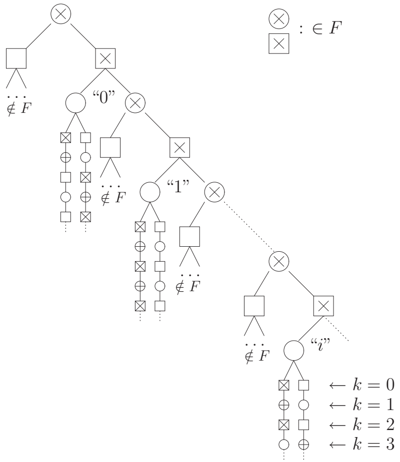
Recursive Winning Strategies?

It seems unlikely that there is a presentation for recursive-sup-conditions which admits a method for deciding which of the players has a winning strategy. Note that our Theorem 6 states the existence for sequential conditions.

PROBLEM. Is it true that for every recursive-sup-game either of the players has a winning strategy which is arithmetical? If yes, how high does it occur?

PROBLEM. For any $\forall_3$-game is there a winning strategy in the arithmetic hierarchy of operators? If yes, how high do they occur in the hierarchy?

A Recursive Infinite Graph



Below node i the membership in F switches at level k iff the Turing machine M_i stops on the empty tape at step k

Answer to Büchi-Landweber's Question

Andreas Blass (Discr. Math. 3 (1979)):

- There is a recursive game with no arithmetical winning strategy.
- There is a recursively enumerable game with no hyperarithmetical winning strategy.

A recursive (r.e.) game is presented by a recursive (r.e.) relation R between ω-sequences.

Refining the Büchi-Landweber Theorem

Some Logics

1. **MSO, monadic second-order logic over $(\mathbb{N}, <)$ (with free set variables, similarly for the FO-logics below),**
2. **FO($<$), first-order logic over $(\mathbb{N}, <)$**
3. **FO($<$)+MOD, the extension of FO($<$) by modular counting quantifers,**
4. **FO(S), first-order logic over $(\mathbb{N}, S)$ with successor relation S,**
5. **Presburger arithmetic, first-order logic over $(\mathbb{N}, +)$**

$\mathcal{L}, \mathcal{L}', \ldots$ **will stand for any of these logics.**

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

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

for each $\mathcal{L}$-formula $\varphi(X, Y)$, 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.

Linking Strategies to Requirements

Let $\mathcal{L}$ be any of the logics MSO, FO($<$), FO($<$)+MOD, FO(S). Then each $\mathcal{L}$-definable game is determined with $\mathcal{L}$-definable winning strategies.

If $\mathcal{L}$ is FO+$\exists^\omega(S)$ or FO(S)+MOD or Presburger arithmetic, then there are $\mathcal{L}$-definable games that are are not determined with $\mathcal{L}$-definable winning strategies.

(Rabinovich, Th., CSL 2007)

Proof Strategy for FO($<$)

Essential steps:

1. **Recall k-types**
2. **Recall Composition Theorem**
3. **Transform given $\mathcal{L}$-formula $\varphi(X, Y)$ into a "bounded normal form", say of quantifier depth k**
4. **Use k-types as vertices of finite game graph, with Muller winning condition**
5. **Transform into parity game over k'-types (for some $k' > k$)**
6. **Use $\mathcal{L}$-definability of k'-types and positional determinacy of parity games to obtain $\mathcal{L}$-definable winning strategies**

k-Types

M, M' **are models** $(\mathbb{N}, \ldots, P, Q)$ **or** $([m,n], \ldots, P, Q)$

A k-type is an equivalence class of $\equiv_k^{\mathcal{L}}$**:**

- $M \equiv_k^{\mathcal{L}} M'$
  iff $M \models \varphi \Leftrightarrow M' \models \varphi$ **for every $\mathcal{L}$-formula** $\varphi(X,Y)$ **of quantifier depth** k**.**

H_k **:= set of k-types (finite!)**

k-type t is $\mathcal{L}$-definable by a formula φ_t of quantifier-depth k.

For each $\mathcal{L}$-formula φ of quantifier-depth k and any model M:

$M \models (\varphi \leftrightarrow \bigvee_{\varphi_t \models \varphi} \varphi_t)$

Composition Theorem

Let $\mathcal{L}$ be the logic FO$(<)$.

(a) **The k-types of M_0, M_1 for $\mathcal{L}$ determine the k-type of the ordered sum $M_0 + M_1$ for $\mathcal{L}$, which moreover can be computed from the k-types of M_0, M_1.**

(b) **If $M_0, M_1, \ldots$ all have the same k-type for $\mathcal{L}$, then this k-type determines the k-type of the ordered sum $\Sigma_{i \in \mathbb{N}} M_i$, which moreover can be computed from the k-type of M_0.**

“Bounded Normal Form”

We use a first-order version of McNaughton’s Theorem (W. Th., Inf. Contr. 1981)

An $\mathcal{L}$-formula $\varphi(X,Y)$ is equivalent to a formula in bounded normal form:

$$\bigvee_{i=1}^{n} (\exists^{\omega} z\ \psi_i(X,Y,z)\ \wedge \neg\exists^{\omega} z\ \psi_i'(X,Y,z))$$

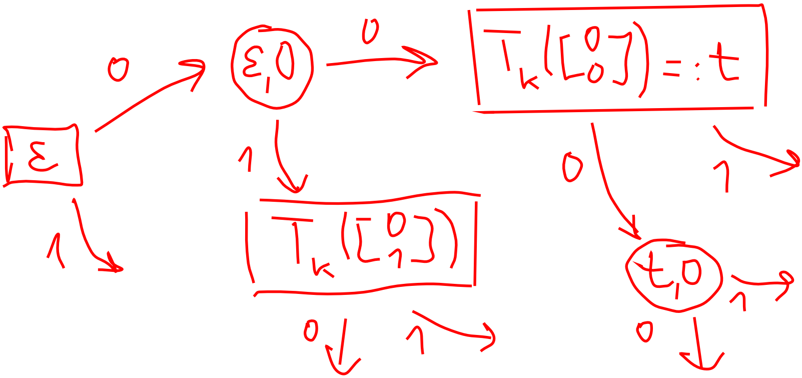
where the ψ_i, ψ_i' are bounded in z.

Let k be the quantifier depth of the ψ_i, ψ_i'.

Construct a game graph over the set H_k of k-types.

Using Types as Vertices of Game Graph

After a play prefix $\binom{P(0)}{Q(0)} \ldots \binom{P(n)}{Q(n)}$, the vertex $T_k([0,n], \ldots P \cap [0,n], Q \cap [0,n])$ is reached.



Definition of Muller Game

Take as game graph $G_\varphi = (V, V_1, V_2, E)$ with

- $V_1 = H_k \cup \{\varepsilon\}$, $V_2 = V_1 \times \{0,1\}$, $V = V_1 \cup V_2$
- **an edge from $t \in V_1$ to (t, a) for each $t \in V_1$, $a \in \{0,1\}$,**
- **an edge from (t, a) to "$t + (a, b)$" for each $b \in \{0,1\}$.**

Winning condition:

$$\bigvee_{i, t \models \psi_i, t' \models \psi'_i} (t \text{ is visited infinitely often} \wedge t' \text{ only finitely often})$$

Call these pairs (t, t') "good" (for φ)

LAR (Latest Appearance Record)

Let $\varrho = t_0 t_1 \dots t_j \dots$ be a play over V.

Consider the associated play ϱ' of LAR's.

LAR at time point j: $(t_j, t_{i_1}, \dots, t_{i_m})$ where $(t_{i_1}, t_{i_2}, \dots, t_{i_m})$ is the list of types visited before j in the order of last visits (most recent noted first).

Assume t_j occurs in $t_{i_1}, \dots t_{i_m}$ at place h.
Color$(t_j, t_{i_1}, \dots, t_{i_m}) := 2h$
if $\exists$ good pair (t, t') s.t. t but not t' occurs in $\{t_{i_1}, \dots, t_{i_h}\}$
otherwise take color $2h - 1$.

Fact: ϱ satisfies the Muller condition iff ϱ' satisfies the parity condition.

Expanding the Game Graph

Extend the k-types t by LAR-information:

Example of LAR-information on a play prefix:

t_1, t_2, t_3 **occur at** $x_1 < x_2 < x_3 \ \wedge \ \bigwedge_i \neg\exists y > x_i$ **:** t_i **occurs at** y

Proceed from k-types to $(k + |H_k| + 1)$-types of *same* logic.
Let $k' = k + |H_k| + 1$.

Apply memoryless determinacy of parity games.

Fix a winning strategy for Player 2 by choosing, for each (t, a) in $H_{k'} \times \{0, 1\}$, a bit $b(t, a)$.

Defining the Winning Strategy

Define the winning strategy by $\psi(X, Y, x) :=$

$$\bigvee_{(t,a);b(t,a)=1} (T_{k'}([0, x-1], X \cap [0, x], Y \cap [0, x]) = t \wedge X(x) = a)$$

Mojzesz Presburger (1904-1943)

Presburger Arithmetic

A winning condition $\varphi(X,(Y,Z))$ can fix that $Z =$ Squares:

$$0,1 \in Z \wedge \forall x_1, x_2, x_3 (x_1 < x_2 < x_3 \text{ successive in } Z \\ \rightarrow\ x_3 - x_2 = (x_2 - x_1) + 2)$$

Putnam 1957: In FO$(+, \mathrm{Squ})$ multiplication is definable.

Proof: $2xy = (x+y)^2 - x^2 - y^2$

$$x^2 = y \Leftrightarrow$$
$$y \in \mathrm{Squ}\ \wedge\ y - (2x-1) \text{ is the greatest square } < y$$

Consequence: Each winning condition $\exists^\omega x R(X,(Y,\mathrm{Squ}),x)$ with recursive R can be expressed.

Even hyperarithmetical winning strategies do not suffice (but the winning conditions are all arithmetical).

Conclusion

Problems

- **How essential is the Composition Theorem?**
  Are there serious extensions of MSO where the main result still holds?

- **General perspective:**
  Develop a precise understanding of the relation between requirements and winning strategies.

- **Language theoretical view:**
  Relate classes of ω-languages (specifications) to classes of $*$-languages (winning strategies).