Logic, Automata, and Games I: Fundamentals on Regular ω-Languages

Wolfgang Thomas

RWTH AACHEN

Nordic Spring School, Nordfjordeid, May 2013

Plan

1. Fundamentals on regular ω-languages
2. Solving regular games
3. Rabin's Tree Theorem
4. Further (un-) decidability results

Prologue: Problems of Tarski and Church

Tarski's Problem

Gödel's and Turing's results implied:

The first-order theory of $(\mathbb{N}, +, \cdot, 0, 1)$ is undecidable.

Alfred Tarski asked:

Is the monadic second-order theory of $(\mathbb{N}, +1, 0)$ decidable?

Today we call this a model-checking problem:

Is the model-checking problem

$(\mathbb{N}, +1, 0) \models \varphi$?

w.r.t. MSO -logic decidable?

Other names: S1S, SC, Büchi's arithmetic

Alfred Tarski

MSO Logic over $(\mathbb{N}, +1, 0)$

We have

- **first-order variables $x, y, z, \ldots$ ranging over natural numbers**
- **set variables $X, Y, Z, \ldots$ ranging over sets of natural numbers**
- **terms formed from first-order variables and 0 by application of "$+1$"**
- **atomic formulas $s = t$ and $X(t)$ for terms s, t and set variables X**
- **connectives $\neg, \vee, \wedge, \rightarrow, \leftrightarrow$ and quantifiers $\exists$, $\forall$**

Example Formulas

- **Over graphs (V, E), we can express 3-colorability:**

$$\exists X_1 \exists X_2 \exists X_3 (\mathbf{Partition}(X_1, X_2, X_3) \\ \wedge \forall x \forall y (E(x,y) \rightarrow \bigvee_{i \neq j} (X_i(x) \wedge X_j(y))))$$

- **Over $(\mathbb{N}, +1, 0)$ the induction axiom:**

$$\forall X (X(0) \wedge \forall y (X(y) \rightarrow X(y+1)) \rightarrow \forall z X(z))$$

- **Over $(\mathbb{N}, +1, 0)$ the existence of automaton runs (e.g., for three states):**

$$\exists X_1 \exists X_2 \exists X_3 \\ (\mathbf{Partition}(X_1, X_2, X_3) \\ \wedge \text{ transition and acceptance condition})$$

Transitive Closure

The relation $\leq$ is the transitive closure of successor.

We have $x \leq y$ iff for all sets X containing x and closed under successor, $X(y)$ holds

Notation: $x < y$, $\exists^{\omega} y \ldots ...$ for $\forall x \exists y (x < y \wedge \ldots$, etc.

Taking closure under predecessor starting from y, a quantifier of finite sets suffices (weak MSO logic).

For any MSO-formula $\varphi(z, z')$, we write

$$\varphi^*(x, y) \ :=$$

$$\forall X (X(x) \wedge \forall z, z' (X(z) \wedge \varphi(z, z') \rightarrow X(z')) \ \rightarrow \ X(y))$$

Example

“Each set with two successive elements contains an even number”

First define “y is even”:

Set $\varphi_2(z, z') \;:=\; (z+1)+1 = z'$

$\mathbf{Even}(y) := \varphi_2^*(0, y)$

Then we take the following formula:

$\forall X(\exists x(X(x) \wedge X(x+1)) \rightarrow \exists y(X(y) \wedge \mathbf{Even}(y)))$

Büchi's Theorem

The MSO theory of $(\mathbb{N}, +1, 0)$ is decidable.

MTh$(\mathbb{N}, +1, 0)$ is decidable.

Method: A version of quantifier elimination:

Reduction of arbitrary formulas $\varphi(X_1, \ldots, X_n)$ to the form

"automaton $\mathcal{A}_\varphi$ accepts $(X_1, \ldots, X_n)$"

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,···,i_μ, 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,

from $P \supset S^a_{a'}P|$ and $S^a_0P|$ to infer P;

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

A Citation

Alonzo Church

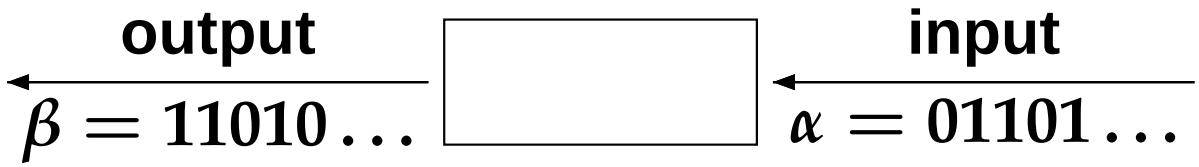
at the “Summer Institute of Symbolic Logic”

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

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

Strategies

A strategy for Player 1 is a map

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

A strategy for Player 2 is a map

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

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

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

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 $\varphi(X, Y)$:

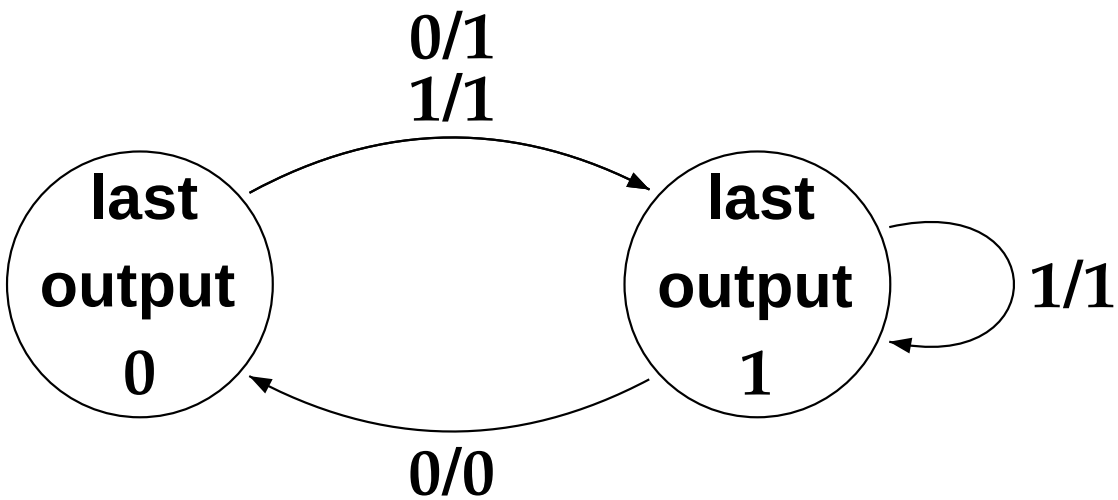
$$\forall t(X(t) \rightarrow Y(t))$$

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

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

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 of Church's Problem

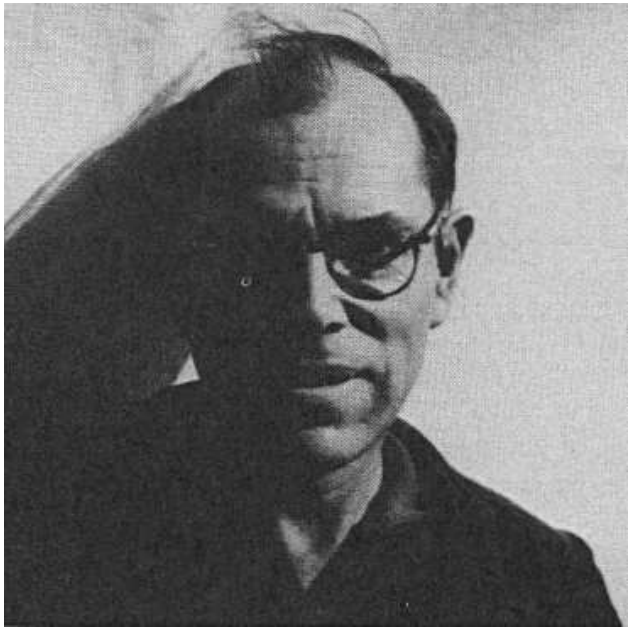
Specification language:
MSO = monadic second-order logic over $(\mathbb{N}, +1)$

Büchi-Landweber Theorem (1969)

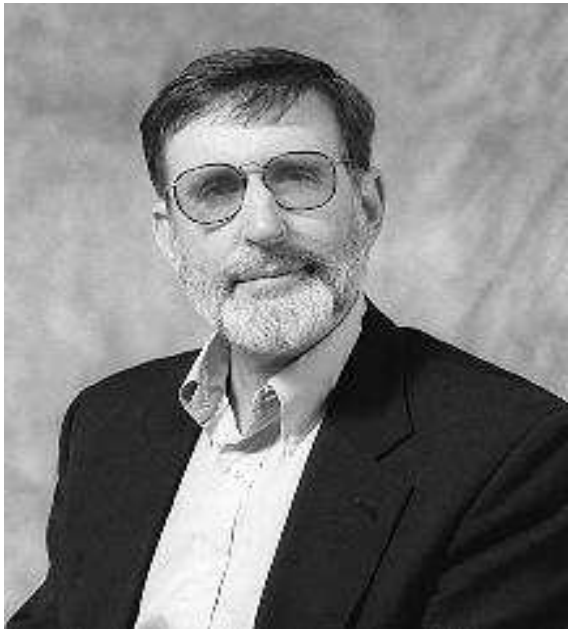
For each MSO-requirement $\varphi(X, Y)$ **either Player 1 or Player 2 has a finite-state winning strategy**
(i.e., the game is "determined"),

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

MSO and Regular ω-Languages

SECTION

I

MATHEMATICAL LOGIC

Symposium on Decision Problems

ON A DECISION METHOD IN RESTRICTED SECOND ORDER ARITHMETIC

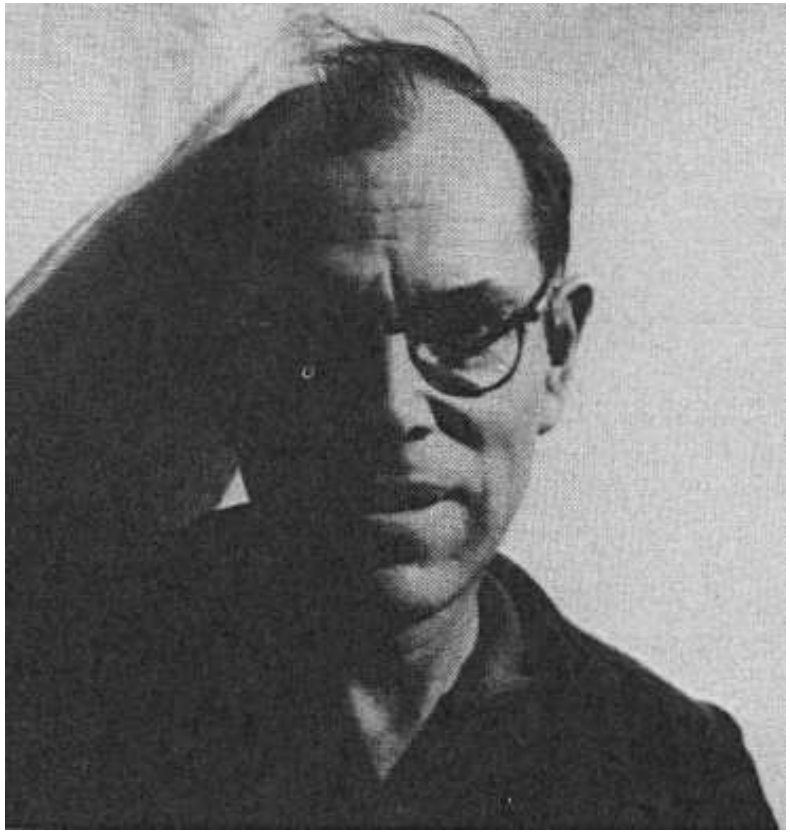
J. RICHARD BÜCHI

University of Michigan, Ann Arbor, Michigan, U.S.A.

Let SC be the interpreted formalism which makes use of individual variables t, x, y, z, . . . ranging over natural numbers, monadic predicate variables q( ), r( ), s( ), i( ), . . . ranging over arbitrary sets of natural numbers, the individual symbol 0 standing for zero, the function symbol ′ denoting the successor function, propositional connectives, and quantifiers for both types of variables. Thus SC is a fraction of the restricted second order theory of natural numbers, or of the first order theory of real numbers. In fact, if predicates on natural numbers are interpreted as binary expansions of real numbers, it is easy to see that SC is equivalent to the first order theory of [Re, +, Pw, Nn], whereby Re, Pw, Nn are, respectively, the sets of non-negative reals, integral powers of 2, and natural numbers.

The purpose of this paper is to obtain a rather complete understanding of definability in SC, and to outline an effective method for deciding truth

This work was done under a grant from the National Science Foundation to the Logic of Computers Group, and with additional assistance through contracts with the Office of Naval Research, Office of Ordnance Research, and the Army Signal Corps.



J. Richard Büchi

Sets versus Words

A set $K \subseteq \mathbb{N}$ can be identified with the infinite 0-1-word α_K where $\alpha_K(i) = 1$ iff $i \in K$.

$\alpha_{\mathbb{P}} = 0\,0\,1\,1\,0\,1\,0\,1\,\ldots$

A tuple $(K_1, \ldots K_n)$ corresponds to an ω-word over $\{0,1\}^n$

$\alpha_{\mathrm{Even},\mathbb{P}} = \binom{1}{0}\binom{0}{0}\binom{1}{1}\binom{0}{1}\binom{1}{0}\binom{0}{1}\ \ldots$

An MSO-formula $\varphi(X_1, \ldots, X_n)$ defines an ω-language:

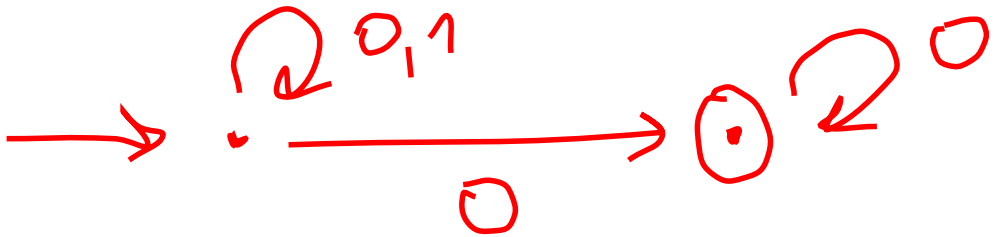
$L(\varphi) = \{\alpha_{(K_1,\ldots,K_n)} \mid (\mathbb{N}, +1, 0) \models \varphi[K_1, \ldots, K_n]\}$

L is MSO definable (over $(\mathbb{N}, +1, 0)$) iff $L = L(\varphi)$ for some MSO-formula φ.

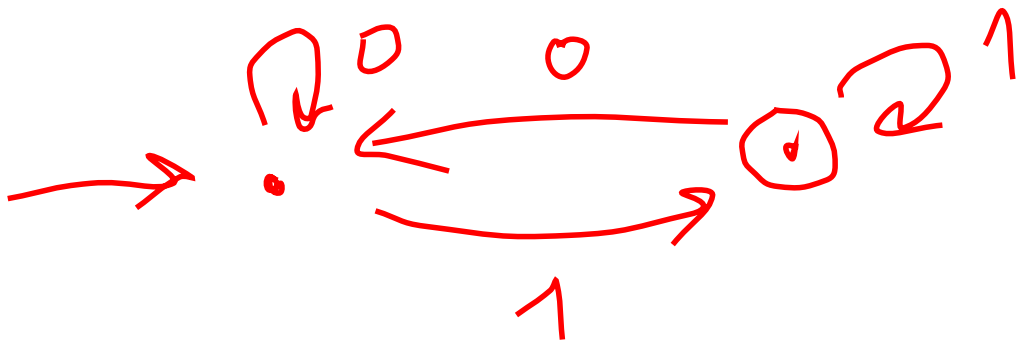
Consider alphabets $\Sigma = \{0,1\}^n$ for notational simplicity.

Example

$\varphi(X) \; : \; \exists x \forall y (x < y \rightarrow \neg X(y))$



$\psi(X) \; : \; \forall x \exists y (x < y \wedge X(y))$



Büchi's Version of "Büchi Automaton"

$$\Sigma_1^\omega : (\exists r) \cdot A[r(0)] \wedge \forall t\ B[i(t), r(t), r(t')] \wedge (\exists^\omega t)\ C[r(t)]$$

Büchi showed closure properties of this formula class and derived that this is a normal form of formulas of S1S.

Consequence: Each formula of S1S can be transformed into a Büchi automaton. MTh$(\mathbb{N}, +1, 0)$ is decidable.

This was new kind of "quantifier elimination".

Büchi-Automata

Format: $\mathcal{A} = (Q, \Sigma, q_0, \Delta, F)$ **with**

- **finite state-set** Q**, initial state** q_0**, set** $F \subseteq Q$ **of final states,**
- **transition relation** $\Delta \subseteq Q \times \Sigma \times Q$

$\mathcal{A}$ **accepts the input word** $\alpha \in \Sigma^\omega$ **if there is a run** ϱ **of** $\mathcal{A}$ **on** α **such that** $\exists^\omega i \; \varrho(i) \in F$

$L(\mathcal{A}) := \{\alpha \in \Sigma^\omega \mid \mathcal{A} \text{ accepts } \alpha\}$

is the ω**-language recognized by** $\mathcal{A}$**.**

L **is called Büchi recognizable if** $L = L(\mathcal{A})$ **for some Büchi automaton** $\mathcal{A}$**.**

Periodicity

Given $\mathcal{A} = (Q, \Sigma, q_0, \Delta, F)$ define

$$W_{pq} = \{w \in \Sigma^* \mid \mathcal{A} : p \xrightarrow{w} q\}$$

Then

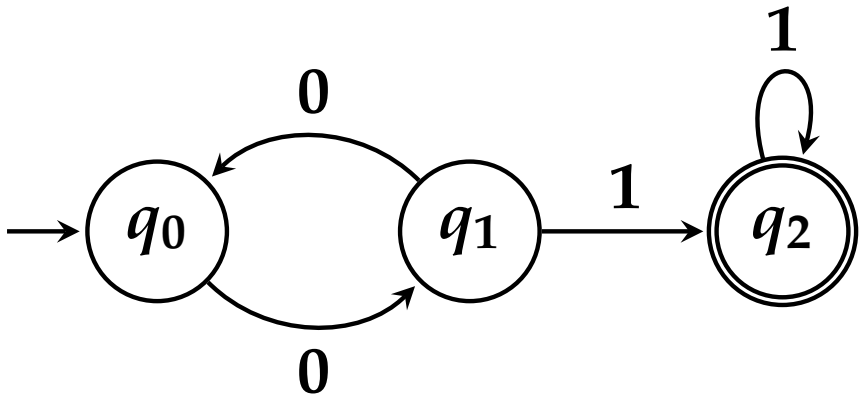
$$L(\mathcal{A}) \;=\; \bigcup_{q \in F} W_{q_0 q} \cdot W_{q,q}^{\omega}$$

An ω-language is Büchi recognizable iff it is a finite union of ω-languages $U \cdot V^{\omega}$ with regular $U, V \subseteq \Sigma^*$

Büchi's Theorem:

An ω-language is MSO-definable iff it is Büchi recognizable

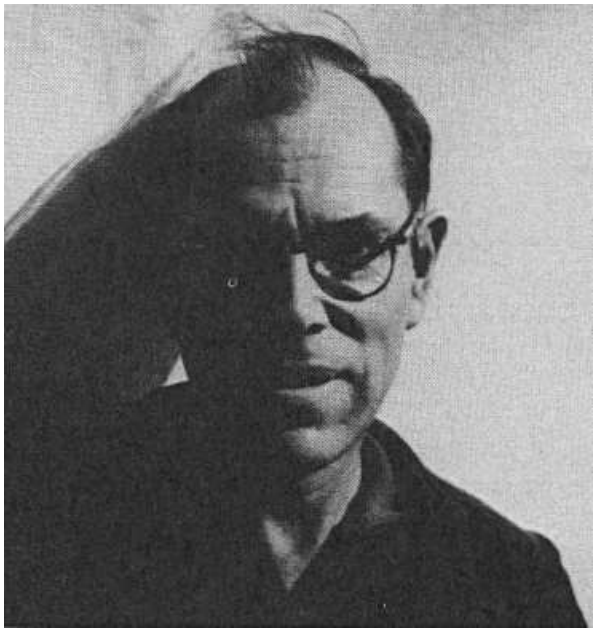
From Automata to MSO-Logic



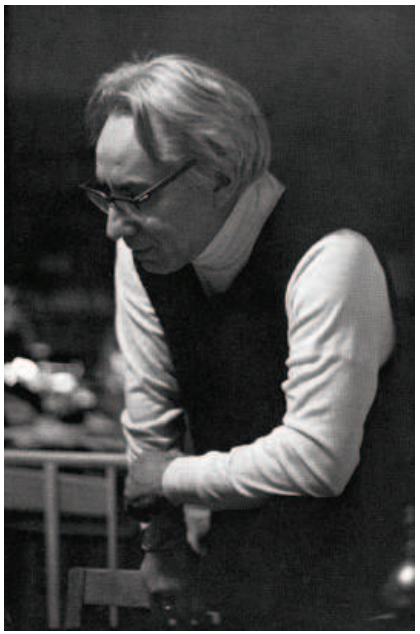
$$\varphi_{\mathcal{A}}(X) :=$$

$$\begin{aligned}
&\exists Y_0 \exists Y_1 \exists Y_2 (\mathbf{Partition}(Y_0, Y_1, Y_2) \wedge Y_0(0) \\
&\quad \wedge \forall x ((Y_0(x) \wedge \neg X(x) \wedge Y_1(x+1)) \\
&\qquad \vee (Y_1(x) \wedge \neg X(x) \wedge Y_0(x+1)) \\
&\qquad \vee (Y_1(x) \wedge X(x) \wedge Y_2(x+1)) \\
&\qquad \vee (Y_2(x) \wedge X(x) \wedge Y_2(x+1))) \\
&\quad \wedge \forall x \exists y (x < y \wedge Y_2(y)))
\end{aligned}$$

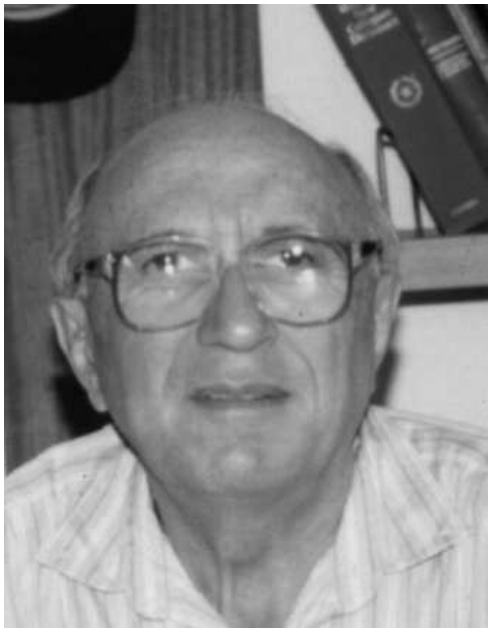
MSO over Finite Words



J.R. Büchi



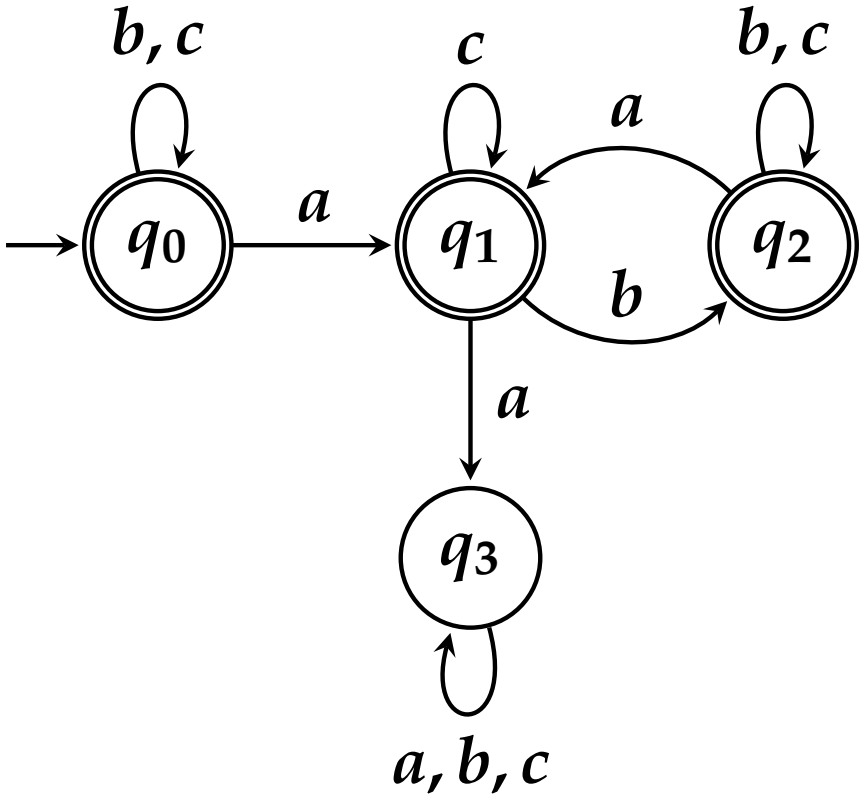
C.C. Elgot



B.A. Trakhtenbrot

Theorem of Büchi-Elgot-Trakhtenbrot (1960):

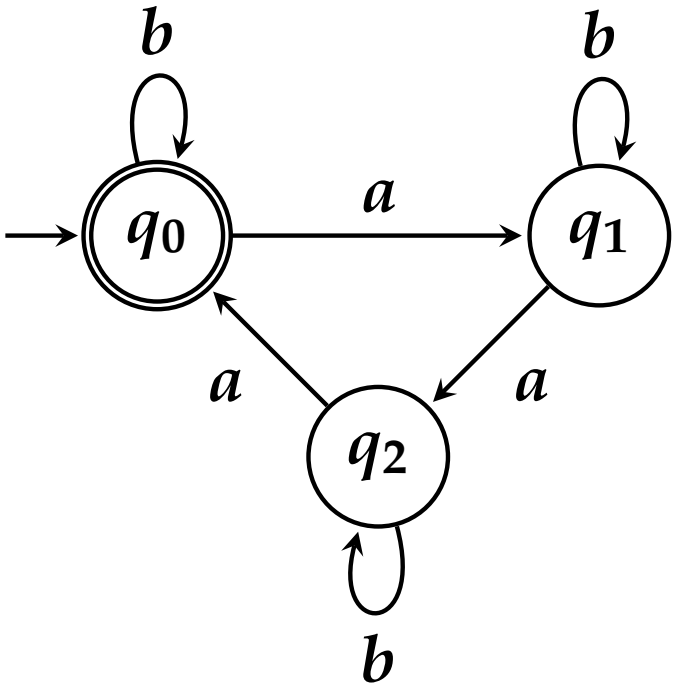
Finite automata and monadic second-order fomulas can express the same properties of finite words.



“between any two letters a there is somewhere a b”

$$\forall x \forall y (x < y \wedge P_a(x) \wedge P_a(y) \rightarrow \exists z (x < z < y \wedge P_b(z)))$$

First-order formula



b	a	a	b	a	a	a	a	b	a	a	b	b	a	b
1	2	3	4	5	6	7	8	9	10	11	12	13	15	16

$\neg\exists x\, P_a(x)$ $\vee$ "$\exists$ set X of positions, containing each third position with a and also the last position with a)"

Formula of monadic second-order logic

Preparing MSO for Easy Induction

Work with a dialect of MSO in which the first-order variables are cancelled.

Simulate x by a singleton variable $\{x\}$.

Atomic formulas: $X \subseteq Y$, Sing(X), Succ(X, Y).

Formulas $\varphi(X_1, \ldots, X_n)$

$\varphi(X_1, \ldots, X_n)$ defines a set of ω-words:

$$L(\varphi) = \{\alpha_{\overline{P}} \mid (\mathbf{Pow}(\mathbb{N}), \mathbf{Succ}, \mathbf{Sing}) \models \varphi[\overline{P}]$$

From MSO to Automata (Finite Words)

Proceed by induction on formulas.

Use nondeterministic automata:

Then atomic formulas, $\vee, \exists$ are easy.

For complementation use the subsetset construction to obtain a deterministic automaton which is easily complementable.

The transformation of MSO to automata is of non-elementary complexity.

From MSO to Automata: Infinite Words

Proceed again by induction. Only complementation is difficult.

Idea: Represent the complement-ω-language as a finite union of sets $U \cdot V^\omega$ with regular U, V.

As U, V use equivalence classes of an equivalence relation:

$$u \sim_{\mathcal{A}} v \;\; :\Leftrightarrow \mathcal{A} : p \xrightarrow{u} q \Leftrightarrow \mathcal{A} : p \xrightarrow{v} q$$

and $\mathcal{A} : p \xrightarrow{u} q$ via F iff $\mathcal{A} : p \xrightarrow{v} q$ via F

- **$\sim_{\mathcal{A}}$ is a finite congruence, and each $\sim_{\mathcal{A}}$-class is a regular.**
- **For $\sim_{\mathcal{A}}$-classes U, V either $UV^\omega \subseteq L(\mathcal{A})$ or $UV^\omega \cap L(\mathcal{A}) = \emptyset$**
- **Then: $\overline{L(\mathcal{A})} = \bigcup \{UV^\omega \mid UV^\omega \cap L(\mathcal{A}) = \emptyset\}$**

Consequences

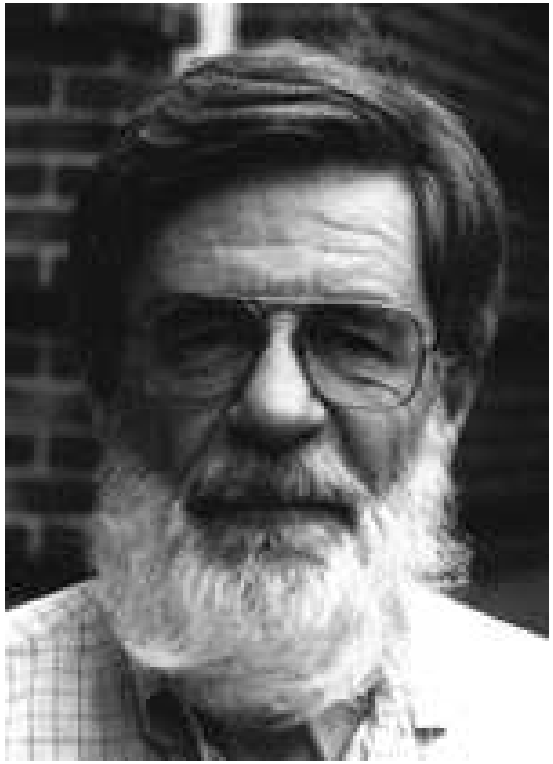
1. **The MSO-theory of $(\mathbb{N}, +1, 0)$ is decidable.**

2. **MSO-formulas can be rewritten as EMSO-formulas.**

The equivalence between MSO and Büchi automata is enough for verification, but not for solving games.

Determinization

McNaughton's Theorem



R. McNaughton

Each Büchi automaton an be transformed into a (deterministic) Muller automaton.

Muller Automata

Format: $\mathcal{A} = (Q, \Sigma, q_0, \delta, \mathcal{F})$
with $\delta : Q \times \Sigma \to Q$, $\mathcal{F} = \{F_1, \ldots, F_k\}$ **where** $F_i \subseteq Q$

Acceptance: $\mathcal{A}$ **accepts** α **iff for the unique run** ϱ **we have**

$$\bigvee_{F \in \mathcal{F}} \Big(\bigwedge_{q \in F} \exists^\omega i \; \varrho(i) = q \wedge \bigwedge_{q \in Q \setminus F} \neg \exists^\omega i \; \varrho(i) = q \Big)$$

Write $\mathcal{A}_q$ **for the det. Büchi automaton** $(Q, \Sigma, q_0, \delta, \{q\})$.

$$L(\mathcal{A}) = \bigcup_{F \in \mathcal{F}} \Big(\bigcap_{q \in F} L(\mathcal{A}_q) \cap \bigcap_{q \in Q \setminus F} \overline{L(\mathcal{A}_q)} \Big)$$

L is Muller recognizable iff L is a Boolean combination of deterministic-Büchi recognizable ω-languages.

Deterministic Büchi Automata in Logic

Given a finite automaton $\mathcal{A}$.

There is a monadic second-order formula $\varphi(y)$ which expresses over an ω-word α:

"the initial segment up to position y is accepted by $\mathcal{A}$"

In $\varphi(y)$ one uses quantifiers "bounded by y":
$\exists x(x \leq y \wedge \ldots)$, $\exists X(\forall z(X(z) \rightarrow z \leq y) \wedge \ldots)$,
similarly for $\forall$.

L is det. Büchi recognizable iff it is definable in the form $\forall x \exists y(x < y \wedge \varphi(y))$ where $\varphi(y)$ is bounded in y.

There are only two unbounded quantifiers (x and y), all other quantifiers are bounded (by y) to a finite domain.

McNaughton's Theorem Logically

Given a Büchi recognizable ω-language

defined by an EMSO formula

McNaughton's Theorem allows to express this

by a Boolean combination of formulas

$\forall x \exists y (x < y \wedge \varphi(y))$ **where** $\varphi(y)$ **is bounded in** y

In logical terminology we are reducing a Σ^1_1 statement to a Boolean combination of Π^0_2-statements.

This amounts to a drastic reduction of quantifier complexity.