Logic and Automata I

Wolfgang Thomas

RWTH AACHEN

EATCS School, Telc, July 2014

The Plan

We present automata theory as a tool to make logic effective.

Four parts:

1. **Some history**
2. **Automata on infinite words**
    - **First step: MSO-logic over finite words**
    - **Büchi automata**
    - **Determinization**
3. **Tree automata and infinite games**
4. **Decidability of monadic theories**

Some History

Why MSO-Logic? Tarski's Problem

Gödel, Turing, Tarski in the 1930's:

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

If we use second-order logic, we get undecidability even for $(\mathbb{N}, +1, 0)$. We define addition (and similarly multiplication):

$x + y = z$ iff each binary relation that contains $(0, x)$ and is closed under $(m, n) \mapsto (m + 1, n + 1)$ contains also (y, z)

Alfred Tarski asked:

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

Alfred Tarski (1901 - 1983)

"Model-Checking"

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

In computer science, the emphasis has shifted:

Does a system model G satisfy a specification φ?

G is often an infinite (edge-labelled and vertex-labelled) graph.

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

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

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

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

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

  $$\exists X_1 \exists X_2 \exists X_3$$
  $$(\mathrm{Partition}(X_1, X_2, X_3)$$
  $\wedge$ **initial-, transition-, and acceptance-condition**$)$

Transitive Closure

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

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

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

Then we take the following formula:

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

Comparing with Presburger Arithmetic

R.M. Robinson 1958:

The MSO-theory of $(\mathbb{N}, +1, 2x, 0)$ is undecidable.

We follow a proof idea of Elgot and Rabin (1966).

Idea: Code binary relations by sets and simulate relation quantifiers by set quantifiers.

$$R = \{(m_0, n_0), (m_1, n_1), \ldots\} \quad \mapsto$$
$$M_R = \{m_0' < n_0' < m_1' < n_1' \ldots\}.$$

$(m, n) \in R$ **iff**
in the M_R-list there are successive codes m', n' of m, n where m' sits on an odd position.

Defining the Code

For each x we need an infinite set C_x of code numbers.

such that from $z \in C_x$ the number x can be retrieved (by an MSO-formula).

Set $C_x := \{2^i \cdot (2x+1) \mid i \geq 0\}$.

From a number $z = 2^i \cdot (2x+1)$ obtain x:

- iteratively divide by 2
- if this is no more possible subtract 1 and divide by 2

Obtain an MSO-formula for "z is a code of x"

Details

Let $\varphi_2(y, y') :=$ y' is half of y)

Then

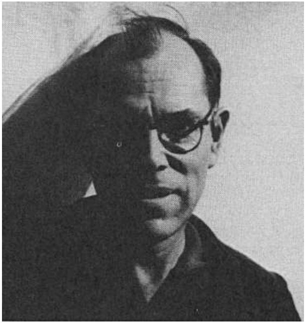
"z is a code of x": $\exists s(\varphi_2^*(z, s) \wedge s = 2x + 1)$

Translation of $\exists R(R(x, y) \ldots)$:

$$\exists X(\exists z \exists z'(z \text{ is code of } x \wedge z' \text{ is code of } y$$

$$\wedge \mathrm{OddPos}(X, z) \wedge \mathrm{Next}(X, z, z'))$$

Büchi's Theorem



Richard J. Büchi (1924 - 1984)

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

MSO-formulas can be translated into "Büchi automata".

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.

First Step: MSO over Finite Words

Words coded by Sets

Consider $w = \binom{0}{1}\binom{0}{0}\binom{0}{0}\binom{1}{1}\binom{1}{0}$

Domain of letter positions: $D = \{0, 1, 2, 3, 4\}$

For an infinite word this domain is $\mathbb{N}$.

w is identified with a pair of sets: $K_1 = \{3, 4\}$, $K_2 = \{0, 3\}$

A word over $\{0, 1\}^n$ with domain D can be identified with an n-tuple $(K_1, \ldots, K_n)$ of subsets of D.

A Definable Language

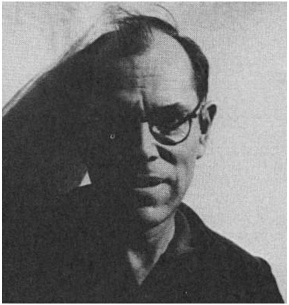
Consider the regular language L_0 over $\{0,1\}^2$ containing the words where

between any two letters $\binom{*}{1}$ there is a letter $\binom{0}{*}$

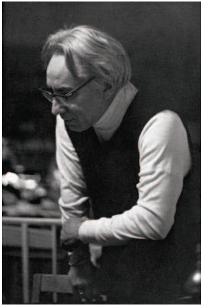
$w = \binom{0}{1}\binom{0}{0}\binom{0}{0}\binom{1}{1}\binom{1}{0}$ **satsfies this.**

$$\begin{aligned} \varphi(X_1, X_2) \ = \ & \forall x \forall y (x < y \wedge X_2(x) \wedge X_2(y) \ \rightarrow \\ & \quad \exists z \ (x < z \wedge z < y \wedge \neg X_1(z))) \end{aligned}$$

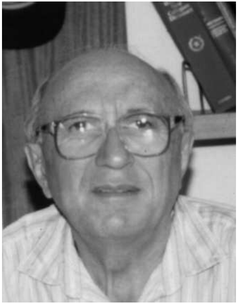
BET Theorem



J.R. Büchi



C.C. Elgot



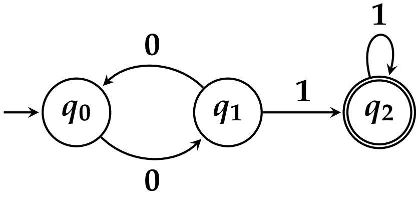
B.A. Trakhtenbrot

Theorem of Büchi-Elgot-Trakhtenbrot (1960):

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

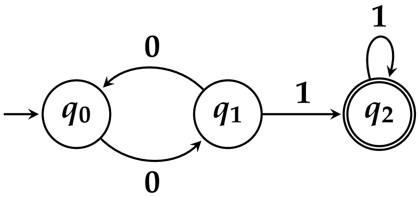
From Automata to MSO-Logic

$\mathcal{A}_0$:



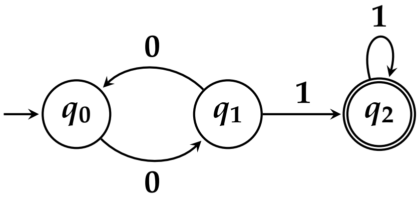
We look for an MSO-formula φ "equivalent" to $\mathcal{A}_0$

It should express over w that $\mathcal{A}_0$ accepts w.



An input word and an accepting run:

$$
\begin{array}{cccccc}
0 & 0 & 0 & 1 & 1 & \\
\begin{pmatrix}1\\0\\0\end{pmatrix} & \begin{pmatrix}0\\1\\0\end{pmatrix} & \begin{pmatrix}1\\0\\0\end{pmatrix} & \begin{pmatrix}0\\1\\0\end{pmatrix} & \begin{pmatrix}0\\0\\1\end{pmatrix} & \begin{pmatrix}0\\0\\1\end{pmatrix}
\end{array}
$$



$$
\begin{aligned}
&\varphi(X) := \\
&\exists Y_0 \exists Y_1 \exists Y_2 (\text{Partition}(Y_0, Y_1, Y_2) \wedge Y_0(\min) \\
&\qquad \wedge \forall x ((Y_0(x) \wedge \neg X(x) \wedge Y_1(x+1)) \\
&\qquad\qquad \vee (Y_1(x) \wedge \neg X(x) \wedge Y_0(x+1)) \\
&\qquad\qquad \vee (Y_1(x) \wedge X(x) \wedge Y_2(x+1)) \\
&\qquad\qquad \vee (Y_2(x) \wedge X(x) \wedge Y_2(x+1))) \\
&\qquad \wedge (Y_1(\max) \vee Y_2(\max)) \wedge X(\max))
\end{aligned}
$$

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), $X < Y$.

Example

$$\varphi(X_1, X_2) \;=\; \forall x \forall y (x < y \wedge X_2(x) \wedge X_2(y) \;\rightarrow\; \exists z\, \neg X_1(z))$$

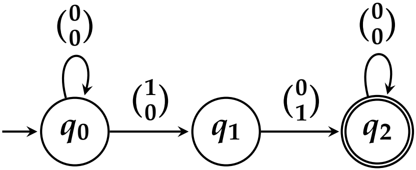
$$\varphi'(X_1, X_2) \;=\; \forall X \forall Y (X < Y \wedge X \subseteq X_2 \wedge Y \subseteq X_2(y) \;\rightarrow\; \exists Z(\mathrm{Sing}(Z) \wedge \neg Z \subseteq X_1))$$

Task: Find for any $\varphi(X_1, \ldots, X_n)$ a corresponding automaton over $\{0,1\}^n$.

From MSO to Automata (Finite Words)

Proceed by induction on formulas.

Example: $\mathrm{Succ}(X_1, X_2)$



Use nondeterministic automata:

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

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

Büchi Automata

Definition

A Büchi automaton (NBA) has the form $\mathcal{A} = (Q, \Sigma, q_0, \Delta, F)$ 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 $\mathcal{A}$.

Büchi's Version of "Büchi Automaton"

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

This formula type is motivated by a representation of Σ_1^1-sets in the Cantor space.

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

This is a kind of "quantifier elimination".

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 $U, V \subseteq \Sigma^*$

Consequence: A nonempty Büchi-recognizable ω-language contains an ultimately periodic ω-word.

Büchi's Theorem

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

- **From automata to MSO: as over finite words.**
- **From MSO to automata: Proceed again by induction.**
  Only complementation is difficult.

Complementation

Idea: Represent also 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:

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

and $\mathcal{A} : p \xrightarrow{u} q$ **via** $F \quad \Leftrightarrow \quad \mathcal{A} : p \xrightarrow{v} q$ **via** F

This is an equivalence relation.

The $\sim_{\mathcal{A}}$-class of u is captured by two lists of edges (p, q):

- **those (p, q) with $\mathcal{A} : p \xrightarrow{u} q$**
- **those (p, q) with $\mathcal{A} : p \xrightarrow{u} q$ via F**

$\sim_{\mathcal{A}}$ is of finite index, and each $\sim_{\mathcal{A}}$-class is regular.

A Crucial Property

Assume $U \cdot V^\omega \cap L(\mathcal{A}) \neq \emptyset$, where U, V are $\sim_{\mathcal{A}}$-classes.

Then $U \cdot V^\omega \subseteq L(\mathcal{A})$.

By assumption we have:

$\mathcal{A}: q_0 \xrightarrow{u} p_1 \xrightarrow{v_1} p_2 \xrightarrow{v_2} p_3 \xrightarrow{v_3} \ldots$ with $u \in U, v_i \in V$
where via infinitely many v_i a final state is passed.

Consider $\beta = u'v_1'v_2'v_3' \ldots$ in $U \cdot V^\omega$ with $u' \in U, v_i' \in V$.

Apply $u \sim_{\mathcal{A}} u'$ and $v_i \sim_{\mathcal{A}} v_i'$:

$$\mathcal{A}: q_0 \xrightarrow{u'} p_1 \xrightarrow{v_1'} p_2 \xrightarrow{v_2'} p_3 \xrightarrow{v_3'} \ldots$$

where via infinitely many v_i' a final state is passed.

Last Missing Step

We want to show (with $\sim_{\mathcal{A}}$-classes U, V):

$$\overline{L(\mathcal{A})} = \bigcup\{U \cdot V^\omega | U \cdot V^\omega \cap L(\mathcal{A}) = \emptyset\}$$

Missing step: Show that each α belongs to some $U \cdot V^\omega$ where U, V are $\sim_{\mathcal{A}}$-classes.

Ramsey's Theorem:

Given a coloring of all pairs $i < j$ of natural numbers, there is an infinite "homogeneous" set $H \subseteq \mathbb{N}$ and a fixed color c such that each pair $i < j$ with $i, j \in H$ is colored with c.

Take as color for (i, j) the $\sim_{\mathcal{A}}$-class of $\alpha[i, j)$

Consequences

1. **The MSO-theory of $(\mathbb{N}, +1, 0)$ is decidable.**
   Proof: Transform a sentence φ into a Büchi automaton over $\{0,1\}^0$ (i.e. with unlabelled transitions) and check whether there is a reachable $q \in F$ with a loop back to q.

2. **MSO-formulas over $(\mathbb{N}, +1, 0)$ can be rewritten as EMSO-formulas.**

3. **Model checking:**
   Check whether for each path of a transition system $\mathcal{S}$ a formula φ holds.
   Check whether the intersection automaton of $\mathcal{S}$ and $\mathcal{A}_{\neg\varphi}$ accepts some ω-word.

Determinization

Deterministic Büchi Automata (DBA)

are of the form $\mathcal{A} = (Q, \Sigma, q_0, \delta, F)$ with $\delta : Q \times \Sigma \to Q$

$\mathcal{A}$ accepts α if its unique run on α visits F infinitely often.

This means: The DFA $\mathcal{A}$ accepts infinitely many prefixes of α

Let $\varphi(X_1, \ldots, X_n)$ be a formula equivalent to the DFA $\mathcal{A}$

Modify $\varphi(X_1, \ldots, X_n)$ to a formula $\varphi'(X_1, \ldots, X_n, y)$ saying "the segment up to position y satisfies φ"

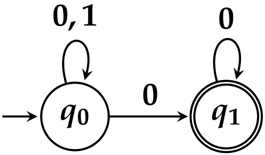
Then the DBA $\mathcal{A}$ is equivalent to the formula

$\forall x \exists y \; (x < y \wedge \varphi'(X_1, \ldots, X_n, y))$ where φ' is bounded in y.

So: DBA-recognizable ω-languages are Π^0_2-sets of the Borel hierarchy (as opposed to Σ^1_1 sets for NBA).

Weakness of DBA

Consider the ω-language $(0+1)^*0^\omega$, recognized by an NBA:



Assume the DBA $\mathcal{A}$ recognizes $(0+1)^*0^\omega$.

- **Reading 0^ω it will reach a final state after 0^{n_0}**
- **Reading $0^{n_0}10^\omega$ it will reach a final state after $0^{n_0}10^{n_1}$**
- **etc.**

So on $0^{n_0}10^{n_1}10^{n_2}1\ldots$ $\mathcal{A}$ will visit final states infinitely often, contradiction.

Muller Automata

are a form of ω-automata that recognize the Boolean combinations of DBA-recognizable sets.

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** $F_i \subseteq Q$

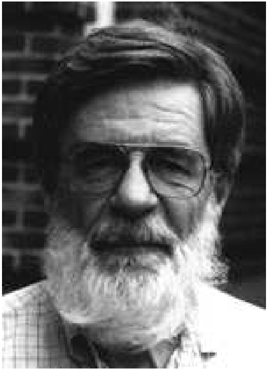
Acceptance: $\mathcal{A}$ **accepts** α **iff for the unique run** ϱ **we have**

$$\bigvee_{i=1}^{k} \Big(\bigwedge_{q \in F_i} \exists^\omega m \; \varrho(m) = q \wedge \bigwedge_{q \in Q \setminus F_i} \neg \exists^\omega m \; \varrho(m) = q \Big)$$

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

$$L(\mathcal{A}) = \bigcup_{i=1}^{k} \Big(\bigcap_{q \in F_i} L(\mathcal{A}_q) \cap \bigcap_{q \in Q \setminus F_i} \overline{L(\mathcal{A}_q)} \Big)$$

McNaughton's Theorem



R. McNaughton (1924 - 2014)

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

McNaughton's Theorem Logically

The ω-language defined by a Büchi automaton can be defined 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.

In other words: The Büchi recognizable ω-languages are all included in a low level of the Borel hierarchy (of the Cantor space).

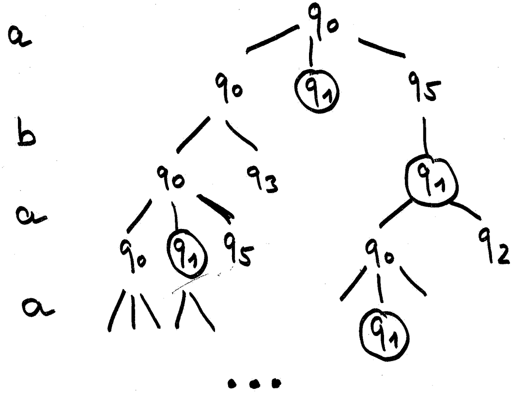
Constructions of Automata

- **Muller (1963), with a flaw**
- **McNaughton (1966)**
- **Choueka (1974)**
- **Th., using logic (1981)**
- **Safra (1988)**
  with optimal growth rate $2^{O(n \log n)}$ for number of states
- **Muller and Schupp (1995)**
- **Fogharty, Kähler, Vardi, Wilke (2013)**
- **and others . . .**

Büchi Determinization: The Task

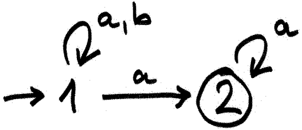
Given Büchi automaton $\mathcal{B} = (Q, \Sigma, q_0, \Delta, F)$ and input α

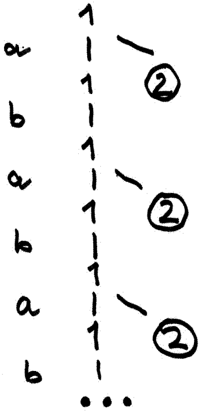
Computation tree of the $\mathcal{B}$-runs on α:



A First Attempt: Subset Construction

Collect for each input-prefix w the states reachable via w, thus obtaining “macrostates”.





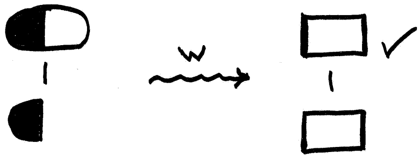
Branching Off and Ticking

Assume we have a nonterminating thread of macrostates.

Branch off a second thread starting from final states, redoing the subset construction from there.

When the second thread merges with the first thread,

- **cancel the macrostate of the second thread**
- **signal the past visit to a final state by ticking the first macrostate.**



Ticking Infinitely Often

Assume that infinitely many ticks occur on the first thread.

$$\begin{array}{ccccccccccccc}
P_1 & \rightsquigarrow & R_1\checkmark & \rightsquigarrow & P_2 & \rightsquigarrow & R_2\checkmark & \rightsquigarrow & & \rightsquigarrow & P_i & \rightsquigarrow & R_i\checkmark \\
\cup\!\!\mid & & \| & & \cup\!\!\mid & & \| & & \ldots & & \cup\!\!\mid & & \| \\
F_1 & \rightsquigarrow & R_1 & & F_2 & \rightsquigarrow & R_2 & & & & F_i & \rightsquigarrow & R_i
\end{array}$$

From each R_i state we can trace back a finite run with $\geq i$ visits to F.

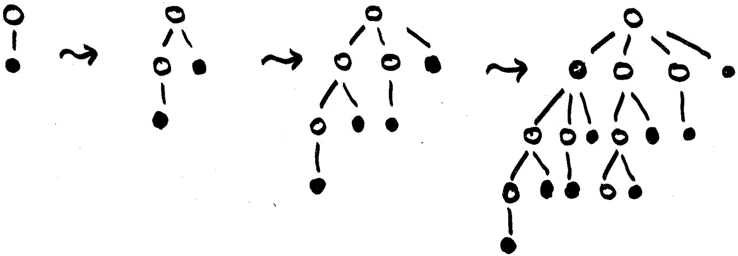
Build a tree from these run segments (each from some R_j to R_{j+1}) constructed backwards from the R_i-states for $i \geq 0$.

Apply König's Lemma to this infinite, finitely branching tree.

Obtain an infinite run infinitely many visits to F; so $\mathcal{B}$ accepts.

The Hydra

We have to open a new thread whenever final states are met!



Defeating the Hydra: Safra Trees

- **Observe that a parent set properly contains the union of the children.**
- **Avoid double occurrences of states among brother nodes: Delete a state when it already occurs in an older brother.**

We obtain a Safra tree:

- **a finite ordered tree of macrostates, possibly some of them ticked**
- **where the children of a node are disjoint and their union a proper subset of the parent.**

There are only finitely many such trees. These are the states of the desired Muller automaton $\mathcal{M}$.

Transitions of $\mathcal{M}$

From a Safra tree s via letter a the following Safra tree results:

1. **For each node whose label contains final states, branch off a new son containing these final states.**
2. **To each node label apply the subset construction via input letter a.**
3. **Cancel state q if it occurs also in an older brother node. Cancel a node if it has label $\emptyset$**
4. **Cancel all children (and their descendants) if the union of their labels equals the parent label, and in this case tick the parent.**

Acceptance

A run $s_0s_1s_2\ldots$ is accepting if from some s_i onwards some node stays uncancelled in $s_is_{i+1}s_{i+2}\ldots$ and is ticked infinitely often.

Then by construction we have $L(\mathcal{M}) \subseteq L(\mathcal{B})$.

This is a Boolean combination of Büchi conditions on the run $s_0s_1s_2\ldots$:

For some node name k:

- there are only finitely many s_i where k is missing.
- there are infinitely many s_i where the label of k is ticked.

Finishing the formal definition

We name the nodes of a Safra tree s by natural numbers $1, 2, \ldots$

In a Safra tree s, a given state q occurs along a path up to some point, the “characteristic node”.

The Safra tree is given by a function $\sigma : Q \to \mathbb{N}$ mapping q to its characteristic node ($\sigma(q) = 0$ indicates that q is not present.)

Consequence: Each Safra tree has $\leq |Q|$ nodes.

**We need a node name reservoir from 1 to $2|Q|$:
When starting a new branch we should use a name not used in the previous tree.**

Completeness of the Construction

Show $L(\mathcal{B}) \subseteq L(\mathcal{M})$. Let $\alpha \in L(\mathcal{B})$.

Consider a run of $\mathcal{B}$ on α, visiting say the final state q again and again. Analyze the $\mathcal{M}$-run of Safra trees on α.

If root is ticked infinitely often, $\mathcal{M}$ accepts.
Otherwise look at first occurrence of q afterwards:

Here q is put into a son of the root, and the accepting $\mathcal{B}$-run finally stays in a fixed son k_1 of the root.

If k_1 is ticked infinitely often, $\mathcal{M}$ accepts.
Otherwise continue analogously and get a son k_2 of k_1 where the $\mathcal{B}$-run finally stays.

At some stage the tick occurs infinitely often, otherwise the height of the Safra trees would be unbounded.

Number of Safra Trees

If q occurs in Safra tree, there is a unique node ("characteristic node") which contains q but such that no son contains q.

We need four functions to describe a Safra tree:

1. **function $Q \to \{0,\ldots,2n\}$ giving the characteristic node**
2. **$\checkmark$-label function: $\{1,\ldots,2n\} \to \{0,1\}$**
3. **parent function: $\{1,\ldots,2n\} \to \{0,\ldots,2n\}$**
4. **older-brother function: $\{1,\ldots,2n\} \to \{0,\ldots,2n\}$**

Number of Safra trees $\leq$ number of such functions

$$\leq (2n+1)^n \cdot 2^{2n} \cdot (2n+1)^{2n} \cdot (2n+1)^{2n}$$
$$\leq (2n+1)^{7n} \in 2^{O(n\log n)}$$

The Conclusion of Büchi's Paper

Problem 1. Let SC^2 be like SC, except that the functions $2x+1$ and $2x+2$ are taken as primitives in place of $x+1$. Is SC^2 decidable?

This is of some interest, because the functions $2x+1$ and $2x+2$ can be interpreted as the right-successor functions x1 and x2 on the set of all words on two generators 1 and 2.

Problem 2. Let $SC(\alpha)$ be like SC, except that the domain of individuals is the ordinal α, and the well ordering on α is added as a primitive. Is $SC(\omega^2)$ decidable?

As outlined in the introduction, Theorem 2 may be interpreted as a method for deciding whether or not a given finite automaton satisfies a given condition in SC.

Problem 3. Is there a solvability algorithm for SC, i.e., is there a method which applies to any formula $C(\mathbf{i}, \mathbf{u})$ of SC and decides whether or not there is a finite automata recursion $A(\mathbf{i}, \mathbf{r}, \mathbf{u})$ which satisfies the condition C (i.e., $A(\mathbf{i}, \mathbf{r}, \mathbf{u}) \supset C(\mathbf{i}, \mathbf{u})$)?