Constructions and Algorithms for ω-Automata

Christof Löding

RWTH Aachen University, Germany

Workshop „Automaten und Logik“ beim Theorietag „Automaten und Formale Sprachen“ 25.27. September 2013, Ilmenau

Outline

Outline

1. The classical automaton logic connection
2. Complementation of Büchi automata
3. Linear arithmetic and weak automata
4. Summary and outlook

Origin – decidability of theories

First-order theory of $(\mathbb{N}, +, \cdot)$ is undecidable.

Goal: Identify decidable theories

Examples:

- $FO(\mathbb{N}, +)$ (Presburger arithmetic)
- $FO(\mathbb{N}, \cdot)$ (Skolem arithmetic)

Weak monadic second-order logic

Theorem (Büchi'60, Elgot'61, Trakhtenbrot'62). The weak monadic second-order theory WMSO$(\mathbb{N}, +1)$ is decidable.

WMSO: First-order logic plus quantification over finite sets.

Weak monadic second-order logic

Theorem (Büchi'60, Elgot'61, Trakhtenbrot'62). The weak monadic second-order theory $\mathrm{WMSO}(\mathbb{N}, +1)$ is decidable.

WMSO: First-order logic plus quantification over finite sets.

Proof strategy: Encode finite subsets of $\mathbb{N}$ by finite words over $\{0, 1\}$ and translate formula φ into finite automaton $\mathcal{A}_\varphi$

- disjunction $\leftrightarrow$ union
- negation $\leftrightarrow$ complement
- existential quantification $\leftrightarrow$ projection

Then: $\varphi(X_1, \ldots, X_n)$ satisfiable iff $L(\mathcal{A}_\varphi) \subseteq (\{0, 1\}^n)^*$ is non-empty

Complexity

Alternation of projection and complementation require an exponential step in the automaton construction.

Complexity

Alternation of projection and complementation require an exponential step in the automaton construction.

Theorem (Meyer, Stockmeyer'71) There is no translation of MSO formulas into automata such that the size of the resulting automaton can be bounded by a function of the form

$2^{2^{\cdot^{\cdot^{\cdot^{2^{|\varphi|}}}}}}\Big\}k$ for a fixed k.

Complexity

Alternation of projection and complementation require an exponential step in the automaton construction.

Theorem (Meyer, Stockmeyer'71) There is no translation of MSO formulas into automata such that the size of the resulting automaton can be bounded by a function of the form

$2^{2^{\cdot^{\cdot^{\cdot^{2^{|\varphi|}}}}}}\Big\}k$ for a fixed k.

Empirical observations (Tool MONA by Basin, Klarlund):

By minimizing the intermediate DFAs obtained during the translation, it is possible to translate long formulas into automata.

Complex MONA example

Hyman's mutual exclusion algorithm (two processes $i = 0, 1$):

while true do begin

0. noncritical section
1. b_i := true
2. while ($k \neq i$) do begin
3. while (b_{1-i}) do skip
4. $k := i$
5. critical section
6. b_i := false

end

Idea for MONA model

The word models represent executions of the protocol.

We use sets to model the variables b_i, k, and a binary encoding for the program counter.

```
var2 PC1', PC1'', PC1''', PC2', PC2'', PC2''', b1, b2, K, max;

pred p1_at_line_1(var1 t)
   = t notin PC1' & t notin PC1'' & t notin PC1''';
pred p1_at_line_2(var1 t)
   = t notin PC1' & t notin PC1'' & t in PC1''';
...
pred p1_proc_step(var1 t)
   = (p1_at_line_1(t) => p1_at_line_2(t+1) & unchanged_vars(t))
     & (p1_at_line_2(t) => ...
...
#Mutual exclusion
Valid => all1 p: (p<=max => ~(p1_at_line_6(p) & p2_at_line_6(p)));
```

Counter example computed by MONA

```
ANALYSIS
A counter-example of least length (10) is:


PC1'            X 0000011101
PC1''           X 0001100010
PC1'''          X 0010100001
PC2'            X 0000000111
PC2''           X 0000001000
PC2'''          X 0111110111
b1              X 0001111111
b2              X 0000001111
K               X 0000000011
```

Full monadic second-order logic

Theorem. (Büchi'62) The monadic second-order theory $\mathrm{MSO}(\mathbb{N}, +1)$ is decidable.

Proof strategy:

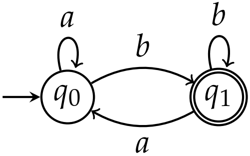
- Inductive translation into automata as for WMSO.
- Use automata over infinite words.

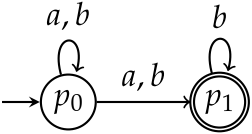
Büchi automata

Büchi automaton:

- same syntax as nondeterministic finite automata (NFAs)
- accepts all infinite words that admit a run visiting infinitely often a final state

Examples:



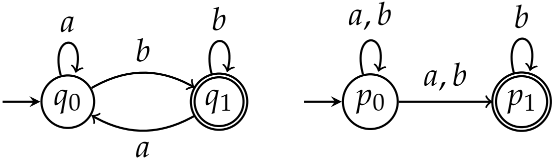


Büchi automata

Büchi automaton:

- same syntax as nondeterministic finite automata (NFAs)
- accepts all infinite words that admit a run visiting infinitely often a final state

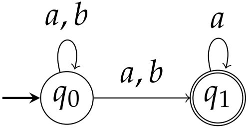
Examples:



Handling negation in the logic requires complementation of Büchi automata (union and projection are easy).

Outline

Classical subset construction fails



$aaaaaa\cdots$ and $abababab\cdots$ induce the same sequence of sets:

$$\{q_0\} \overset{a}{\underset{a}{\rightarrow}} \{q_0, q_1\} \overset{a}{\underset{b}{\rightarrow}} \{q_0, q_1\} \overset{a}{\underset{a}{\rightarrow}} \{q_0, q_1\} \overset{a}{\underset{b}{\rightarrow}} \{q_0, q_1\} \cdots$$

$\rightsquigarrow$ sequence of reachable state sets does not contain enough information.

Büchi's proof

Starting point: NBA $\mathcal{A}$ to be complemented

1. Assign a type (or color) to each finite word.
2. The type of a word u carries enough information about the behavior of $\mathcal{A}$ on u.

Büchi's proof

Starting point: NBA $\mathcal{A}$ to be complemented

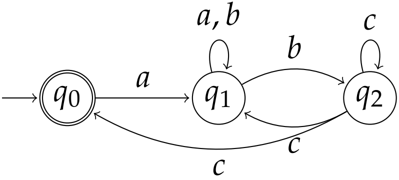
1. Assign a type (or color) to each finite word.
2. The type of a word u carries enough information about the behavior of $\mathcal{A}$ on u.
3. The sequence of types is enough to decide whether the word is in $L(\mathcal{A})$ or not for an arbitrary factorization of the word (because of 2.).

Büchi's proof

Starting point: NBA $\mathcal{A}$ to be complemented

1. Assign a type (or color) to each finite word.
2. The type of a word u carries enough information about the behavior of $\mathcal{A}$ on u.
3. The sequence of types is enough to decide whether the word is in $L(\mathcal{A})$ or not for an arbitrary factorization of the word (because of 2.).
4. Every infinite word can be factorized such that the resulting type sequence is very simple: type$_1$(type$_2$)$^\omega$.
5. For each type the set of words with this type is a regular language $\rightsquigarrow$ representation of the complement as $\bigcup U_i \cdot V_i^\omega$

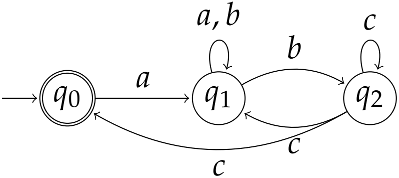
Transition profiles as types



For $u \in \Sigma^*$, the transition profile $\tau(u)$ contains for each state q

- which states are reachable from q by reading u,
- which states are reachable from q via a final state by reading u.

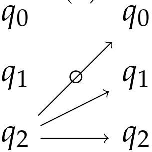
Transition profiles as types



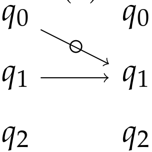
For $u \in \Sigma^*$, the transition profile $\tau(u)$ contains for each state q

- which states are reachable from q by reading u,
- which states are reachable from q via a final state by reading u.

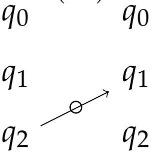
$\tau(c)$:



$\tau(a)$:

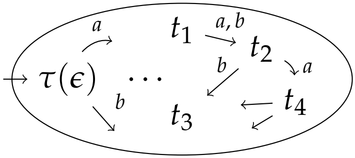


$\tau(ca)$:



Transition monoid automaton

Deterministic finite automaton with transition profiles as states



- Size: there are 3^{n^2} transition profiles.
- equipped with singleton acceptance set $\{t\}$,
  it recognizes all words u with $\tau(u) = t$

Büchi's proof

Starting point: NBA $\mathcal{A}$ to be complemented

1. Assign a type (or color) to each finite word.
2. The type of a word u carries enough information about the behavior of $\mathcal{A}$ on u.

Büchi's proof

Starting point: NBA $\mathcal{A}$ to be complemented

1. Assign a type (or color) to each finite word.
2. The type of a word u carries enough information about the behavior of $\mathcal{A}$ on u.
3. The sequence of types is enough to decide whether the word is in $L(\mathcal{A})$ or not for an arbitrary factorization of the word (because of 2.).

Büchi's proof

Starting point: NBA $\mathcal{A}$ to be complemented

1. Assign a type (or color) to each finite word.
2. The type of a word u carries enough information about the behavior of $\mathcal{A}$ on u.
3. The sequence of types is enough to decide whether the word is in $L(\mathcal{A})$ or not for an arbitrary factorization of the word (because of 2.).
4. Every infinite word can be factorized such that the resulting type sequence is very simple: type$_1$(type$_2$)$^\omega$.
5. For each type the set of words with this type is a regular language $\rightsquigarrow$ representation of the complement as $\bigcup U_i \cdot V_i^\omega$

Sequences of transition profiles

Consider an infinite word α.

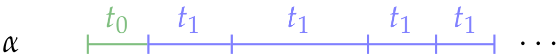
- For any factorization of α, the corresponding sequence of transition profiles contains enough information to decide whether $\alpha \in L$.

Sequences of transition profiles

Consider an infinite word α.

- For any factorization of α, the corresponding sequence of transition profiles contains enough information to decide whether $\alpha \in L$.

- Application of Ramsey's theorem yields a simple (periodic) factorization

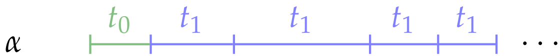


Sequences of transition profiles

Consider an infinite word α.

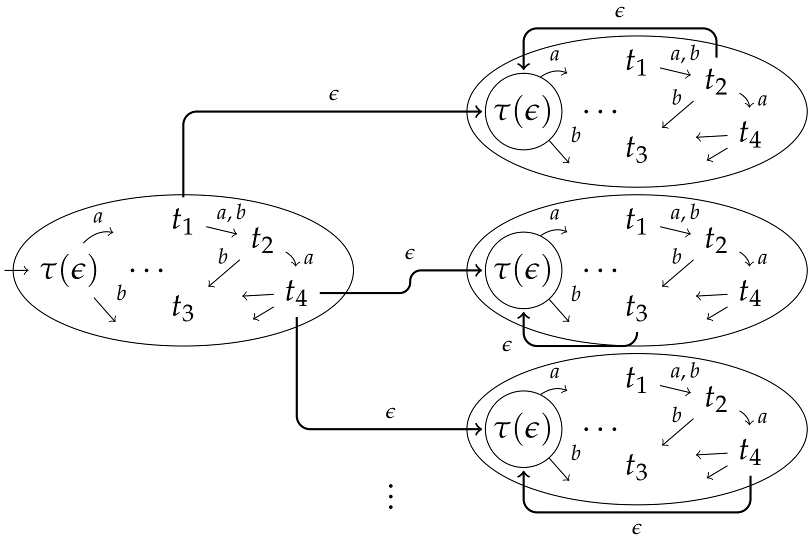
- For any factorization of α, the corresponding sequence of transition profiles contains enough information to decide whether $\alpha \in L$.

- Application of Ramsey's theorem yields a simple (periodic) factorization



- $\Sigma^\omega \setminus L(\mathcal{A})$ is of the form $\bigcup_{t_0,t_1} U_{t_0} U_{t_1}^\omega$ for those transition profiles t_0, t_1 for which $t_0 t_1^\omega$ contains no accepting run.

Structure of the complement automaton



Number of states at most $3^{n^2} \cdot 3^{n^2} \in 2^{\mathcal{O}(n^2)}$

Improvements, variations and lower bounds

Lower bounds:

- Michel'88: $n!$ states are required
- Yan'06: $(0.76n)^n$

Improvements, variations and lower bounds

Lower bounds:

- Michel'88: $n!$ states are required
- Yan'06: $(0.76n)^n$

Improved constructions:

- Safra'88: determinization in $2^{\mathcal{O}(n \log n)}$
- Klarlund'91 / Kupferman,Friedgut,Vardi'06 / Schewe'09: best known worst case upper bound (progress measures/ranks)
- Kähler, Wilke'08: Unified data structure for complementation, disambiguation, and determinization in $2^{\mathcal{O}(n \log n)}$
- Breuers, L., Olschewski'12: improvement of Büchi's original construction to $2^{\mathcal{O}(n \log n)}$

Experimental comparison (Tsai, Fogarty, Vardi, Tsay'10):

determinization approach best

Problems in practice

- Complementation constructions are still too complex
- Minimization of deterministic ω-automata is not yet well-understood and more difficult:

Theorem (Schewe'10). The problem

"Given a deterministic Büchi automaton $\mathcal{A}$ and a number k, is there a k-state deterministic Büchi automaton equivalent to $\mathcal{A}$?"

is NP-complete.

Outline

Presburger arithmetic and automata

Presburger arithmetic: $FO(\mathbb{N}, +, <)$

Translation to automata:

- Encode numbers in binary as words.
- There is an automaton over the alphabet $\{0, 1\}^3$ checking for three numbers x, y, z whether $x + y = z$ (similarly for $<$)

$$\begin{array}{lllllll} x & 1 & 0 & 0 & 1 & 1 & 0 \\ y & 0 & 0 & 1 & 0 & 1 & 1 \\ z & 1 & 1 & 0 & 0 & 0 & 1 \end{array}$$

- as before: disjunction $\rightsquigarrow$ union, negation $\rightsquigarrow$ complement, existential quantification $\rightsquigarrow$ projection

Complexity

As for WMSO, each alternation of projection and complementation leads to an exponential step in the inductive translation

$\rightsquigarrow$ unbounded tower of exponentials

Complexity

As for WMSO, each alternation of projection and complementation leads to an exponential step in the inductive translation

$\rightsquigarrow$ unbounded tower of exponentials

But:

Theorem (Klaedtke'04).: There is a triply exponential upper bound on the size of the DFAs produced from formulas of Presburger arithmetic.

Can be achieved by systematically minimizing the intermediate DFAs.

Real numbers

Now consider $FO(\mathbb{R}, <, +, \mathbb{Z})$ (linear real arithmetic)

- Choose the binary representation of real numbers $\leadsto$ ω-word.
- Code the dot by $\star$.
- Codings of 3.5:

$$(0^{+}11 \star 10^{\omega}) \text{ and } (0^{+}11 \star 01^{\omega})$$

- Vectors of real numbers are coded over the alphabet $\{0, 1, \star\}^{n}$ such that $\star$ is at the same position in all components.

Real numbers

Now consider $FO(\mathbb{R}, <, +, \mathbb{Z})$ (linear real arithmetic)

- Choose the binary representation of real numbers $\rightsquigarrow$ ω-word.
- Code the dot by $\star$.
- Codings of 3.5:

$$(0^{+}11 \star 10^{\omega}) \text{ and } (0^{+}11 \star 01^{\omega})$$

- Vectors of real numbers are coded over the alphabet $\{0, 1, \star\}^{n}$ such that $\star$ is at the same position in all components.
- For a formula $\varphi(x_1, \ldots, x_n)$ of FO$(\mathbb{R}, +, <, \mathbb{Z})$ let $L(\varphi)$ be the set of all codings of vectors $(r_1, \ldots, r_n)$ that make φ true.

Translation to automata

Theorem (Büchi'62). Every formula of $FO(\mathbb{R}, <, +, \mathbb{Z})$ can be translated into an equivalent MSO formula and thus also into an equivalent Büchi automaton.

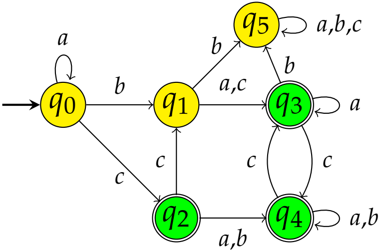
Problem: Although only a fragment of MSO is used, the translation into Büchi automata has to deal with the same difficulties as for full MSO.

In particular, minimization of intermediate automata is difficult.

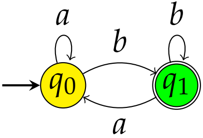
Deterministic weak automata

Deterministic Büchi automata with the following property:

Each SCC is either completely accepting or completely rejecting



weak



not weak

Minimization

Theorem (Boigelot,Jodogne,Wolper'01). Let φ be a formula of FO$(\mathbb{R}, +, <, \mathbb{Z})$. Then $L(\varphi)$ is recognizable by a deterministic weak Büchi automaton.

The proof uses topological arguments.

Reduction to Minimization of DFAs

Theorem (Staiger'83). Weak deterministic automata have canonical minimal automata, which can be defined in terms of the adaption of the Myhill/Nerode equivalence to infinite words.

Reduction to Minimization of DFAs

Theorem (Staiger'83). Weak deterministic automata have canonical minimal automata, which can be defined in terms of the adaption of the Myhill/Nerode equivalence to infinite words.

Theorem (L.'01). The minimization of deterministic weak Büchi automata can be reduced in linear time to the minimization of DFAs.

Reduction to Minimization of DFAs

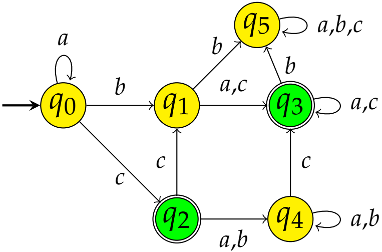
Theorem (Staiger'83). Weak deterministic automata have canonical minimal automata, which can be defined in terms of the adaption of the Myhill/Nerode equivalence to infinite words.

Theorem (L.'01). The minimization of deterministic weak Büchi automata can be reduced in linear time to the minimization of DFAs.

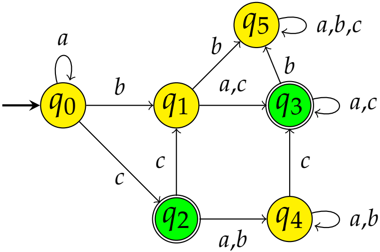
Some remarks on the proof:

- The algorithm computes from a weak Büchi automaton $\mathcal{A}$ a weak Büchi automaton $\mathcal{A}'$ that can be minimized as DFA and results in a minimal weak Büchi automaton.
- $\mathcal{A}'$ only differs from $\mathcal{A}$ on the set of accepting states. This set is recomputed on states that are not on a loop.

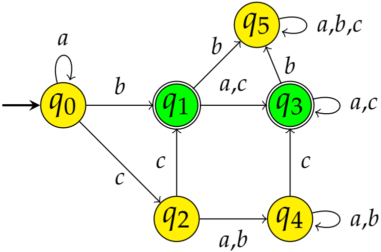
Illustration



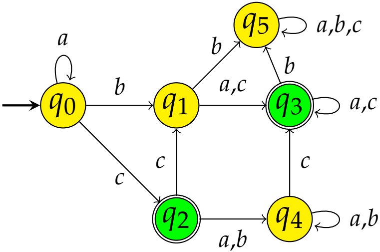
Illustration



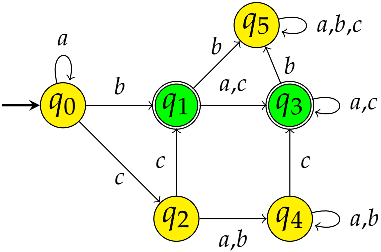
1. Adapt acceptance status of non-looping states:



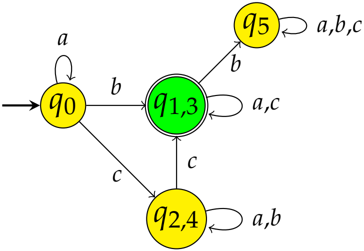
Illustration



1. Adapt acceptance status of non-looping states:



2. Minimize as DFA:



Implementation

- There is a similar triply exponential upper bound on the automata size as for finite words (Eisinger'08)
- Has been implemented in the LASH Toolset (Boigelot, Latour, Legay) and in LIRA (Becker, Dax, Eisinger, Klaedtke'07).
- Used, for example, to represent reachability sets of hybrid automata.

Outline

Summary

ω-automata as a useful tool for decision procedures:

- Monadic second-order logic
- Linear arithmetic over the reals

Problems:

- Constructions are more complex
- Minimization is difficult

Deterministic weak automata form a robust fragment with good algorithmic properties.

Some current topics

Towards practical algorithms

- Evaluate and improve existing constructions (complementation, determinization)
- Find more efficient algorithms for subclasses of logics/automata (for example unambiguous automata or fragments of temporal logics)

Stronger decidability results

- Extensions of logic/automata to express boundedness properties.