Monadic second order logic as a model companion

Sam van Gool

samvangool@me.com

University of Utrecht (until 31 Aug) / Paris (starting 1 Sept)

Logic Colloquium, Prague

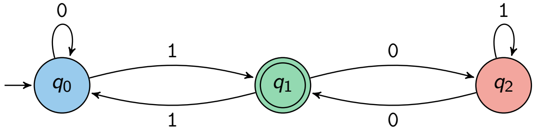
13 August 2019

Automata and logic: example

- A programming problem: given a natural number in binary, $w \in \{0, 1\}^*$, determine if w is congruent to 1 modulo 3.

Automata and logic: example

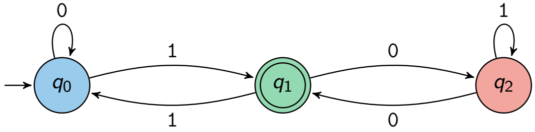
- A programming problem: given a natural number in binary, $w \in \{0,1\}^*$, determine if w is congruent to 1 modulo 3.
- Solution 1: a (deterministic) automaton A:



Answer yes iff A accepts w.

Automata and logic: example

- A programming problem: given a natural number in binary, $w \in \{0,1\}^*$, determine if w is congruent to 1 modulo 3.
- Solution 1: a (deterministic) automaton A:



Answer yes iff A accepts w.

- Solution 2: a monadic second order formula $\varphi(W_0, W_1)$:

$$\exists Q_0 \exists Q_1 \exists Q_2 (Q_0(\texttt{first}) \wedge Q_1(\texttt{last}) \wedge \text{Partition}(Q_0, Q_1, Q_2) \wedge$$

$$\forall x([W_0(x) \wedge Q_0(x) \to Q_0(\texttt{S}x)] \wedge [W_1(x) \wedge Q_0(x) \to Q_1(\texttt{S}x)] \wedge \ldots))$$

Answer yes iff $w = (W_0, W_1)$ makes φ true.

Regular languages

Regular languages over a finite alphabet Σ are subsets $L \subseteq \Sigma^\omega$ which are ...

- recognizable by a finite automaton;

  or, equivalently,

- definable by a formula of S1S,
  the monadic second order logic of one successor.

Büchi 1960

A model complete theory

A functional language $\mathcal{L}$: Boolean algebra operations $(\bot, \cup, -)$, two unary functions, $\mathbf{X}$ and $\mathbf{F}$, and a constant $\mathbf{I}$.

A model complete theory

A functional language $\mathcal{L}$: Boolean algebra operations $(\perp, \cup, -)$, two unary functions, $\mathbf{X}$ and $\mathbf{F}$, and a constant $\mathbf{I}$.

The Boolean algebra $\mathcal{P}(\omega)$ is an $\mathcal{L}$-structure with:

- $\mathbf{X}a := \{t \in \omega \mid t + 1 \in a\}$,
- $\mathbf{F}a := \{t \in \omega \mid \exists t' \geq t \colon t' \in a\}$,
- $\mathbf{I} := \{0\}$.

A model complete theory

A functional language $\mathcal{L}$: Boolean algebra operations $(\bot, \cup, -)$, two unary functions, **X** and **F**, and a constant **I**.

The Boolean algebra $\mathcal{P}(\omega)$ is an $\mathcal{L}$-structure with:

- $\mathbf{X}a := \{t \in \omega \mid t+1 \in a\}$,
- $\mathbf{F}a := \{t \in \omega \mid \exists t' \geq t \colon t' \in a\}$,
- $\mathbf{I} := \{0\}$.

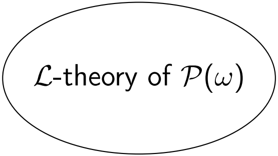
Theorem

The first order $\mathcal{L}$-theory of $\mathcal{P}(\omega)$ is model complete.

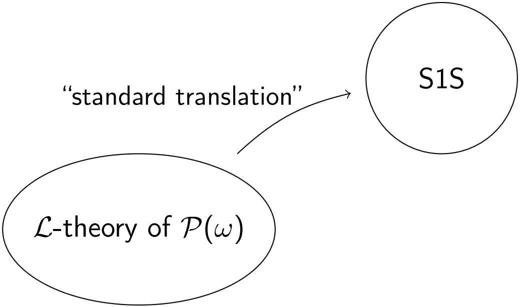
A theory T^* is model complete iff every formula is T^*-equivalent to an existential formula.

Ghilardi, G. JSL 2017

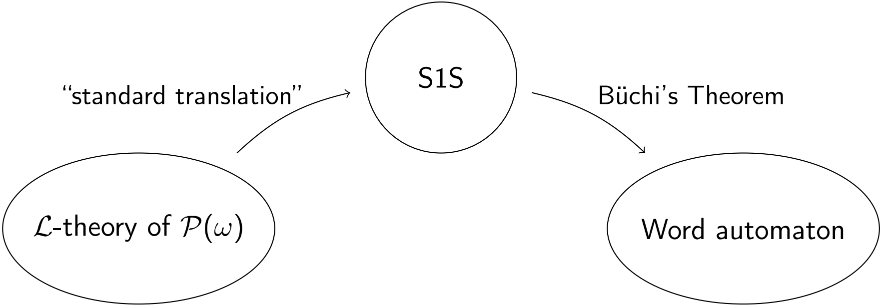
Proving model completeness with automata



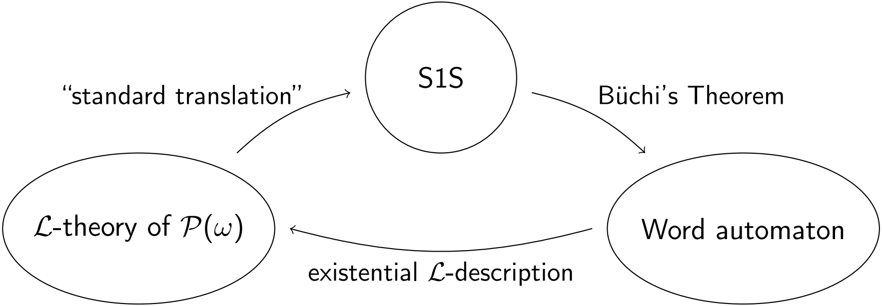
Proving model completeness with automata



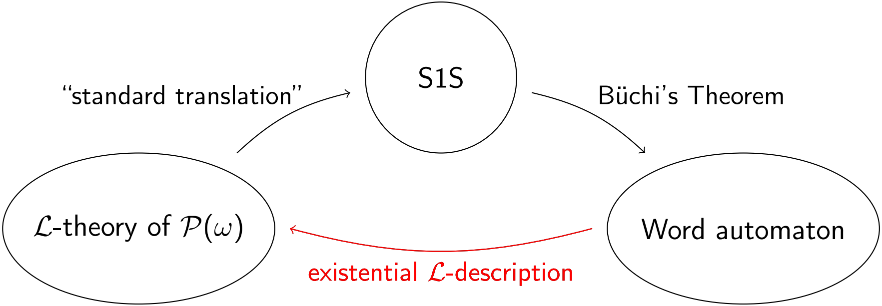
Proving model completeness with automata



Proving model completeness with automata



Proving model completeness with automata



An existential $\mathcal{L}$-description of a word automaton

- Let $A = (Q, \Sigma, \delta, q_0, F)$ be a **word automaton** over a finite alphabet Σ, i.e., a function $\delta\colon Q \times \Sigma \to \mathcal{P}(Q)$, an initial state $q_0 \in Q$ and a subset $F \subseteq Q$ of final states.

An existential $\mathcal{L}$-description of a word automaton

- Let $A = (Q, \Sigma, \delta, q_0, F)$ be a **word automaton** over a finite alphabet Σ, i.e., a function $\delta\colon Q \times \Sigma \to \mathcal{P}(Q)$, an initial state $q_0 \in Q$ and a subset $F \subseteq Q$ of final states.
- Write $\Sigma = \{0, \ldots, s\}$, $Q = \{0, \ldots, m\}$, $q_0 = 0$.
- A word $W\colon \omega \to \Sigma$ is a partition $(W_0, \ldots, W_s)$ of ω; $W_j = W^{-1}(j)$.

An existential $\mathcal{L}$-description of a word automaton

- Let $A = (Q, \Sigma, \delta, q_0, F)$ be a **word automaton** over a finite alphabet Σ, i.e., a function $\delta\colon Q \times \Sigma \to \mathcal{P}(Q)$, an initial state $q_0 \in Q$ and a subset $F \subseteq Q$ of final states.
- Write $\Sigma = \{0, \ldots, s\}$, $Q = \{0, \ldots, m\}$, $q_0 = 0$.
- A word $W\colon \omega \to \Sigma$ is a partition $(W_0, \ldots, W_s)$ of ω; $W_j = W^{-1}(j)$.

Key Observation. The automaton A accepts a word $W\colon \omega \to \Sigma$ iff $\mathcal{P}(\omega), [w_i \mapsto W_i] \models \alpha(w_0, \ldots, w_s)$, where α is the $\exists$ $\mathcal{L}$-formula:

An existential $\mathcal{L}$-description of a word automaton

- Let $A = (Q, \Sigma, \delta, q_0, F)$ be a **word automaton** over a finite alphabet Σ, i.e., a function $\delta \colon Q \times \Sigma \to \mathcal{P}(Q)$, an initial state $q_0 \in Q$ and a subset $F \subseteq Q$ of final states.
- Write $\Sigma = \{0, \ldots, s\}$, $Q = \{0, \ldots, m\}$, $q_0 = 0$.
- A word $W \colon \omega \to \Sigma$ is a partition $(W_0, \ldots, W_s)$ of ω; $W_j = W^{-1}(j)$.

Key Observation. The automaton A accepts a word $W \colon \omega \to \Sigma$ iff $\mathcal{P}(\omega), [w_i \mapsto W_i] \models \alpha(w_0, \ldots, w_s)$, where α is the $\exists$ $\mathcal{L}$-formula:

$$\exists q_0, \ldots, q_m(\text{“the } q_i \text{ partition } \omega\text{”} \wedge \bigwedge_{\substack{0 \le i \le m \\ 0 \le j \le s}} \left(q_i \cap w_j \subseteq \bigcup_{k \in \delta(i,j)} \mathbf{X} q_k \right)$$

An existential $\mathcal{L}$-description of a word automaton

- Let $A = (Q, \Sigma, \delta, q_0, F)$ be a **word automaton** over a finite alphabet Σ, i.e., a function $\delta \colon Q \times \Sigma \to \mathcal{P}(Q)$, an initial state $q_0 \in Q$ and a subset $F \subseteq Q$ of final states.
- Write $\Sigma = \{0, \ldots, s\}$, $Q = \{0, \ldots, m\}$, $q_0 = 0$.
- A word $W \colon \omega \to \Sigma$ is a partition $(W_0, \ldots, W_s)$ of ω; $W_j = W^{-1}(j)$.

Key Observation. The automaton A accepts a word $W \colon \omega \to \Sigma$ iff $\mathcal{P}(\omega), [w_i \mapsto W_i] \models \alpha(w_0, \ldots, w_s)$, where α is the $\exists$ $\mathcal{L}$-formula:

$$\exists q_0, \ldots, q_m (\text{“the } q_i \text{ partition } \omega\text{”} \wedge \bigwedge_{\substack{0 \le i \le m \\ 0 \le j \le s}} \left(q_i \cap w_j \subseteq \bigcup_{k \in \delta(i,j)} \mathbf{X} q_k \right)$$

$$\wedge\ I \subseteq q_0 \wedge \mathbf{F} \left(\textstyle\bigcup_{i \in F} q_i \right) = \top).$$

The theory is a model companion

A theory T^* is a model companion of a theory T iff T^* is model complete, and T and T^* have the same universal consequences.

Theorem

The $\mathcal{L}$-theory of $\mathcal{P}(\omega)$ is the model companion of the theory of $\mathcal{L}$-structures axiomatized by the following universal sentences:

The theory is a model companion

A theory T^* is a model companion of a theory T iff T^* is model complete, and T and T^* have the same universal consequences.

Theorem

The $\mathcal{L}$-theory of $\mathcal{P}(\omega)$ is the model companion of the theory of $\mathcal{L}$-structures axiomatized by the following universal sentences:

- *equations for Boolean algebras;*
- $\mathbf{X}$ *is a Boolean homomorphism;*
- $\mathbf{F}a$ *is the least fixed point of* $x \mapsto a \vee \mathbf{X}x$*;*
- $\mathbf{I}$ *is an atom which is below* $\mathbf{F}a$ *for any* $a \neq \bot$*, and* $\mathbf{X}\mathbf{I} = \bot$*.*

Ghilardi, G. JSL 2017

Binary trees

The full binary tree is 2^*, finite sequences of 0's and 1's.

Binary trees

The full binary tree is 2^*, finite sequences of 0's and 1's.

To obtain a model complete theory, more complex operators are needed to express acceptance by a tree automaton.

Binary trees

The full binary tree is 2^*, finite sequences of 0's and 1's.

To obtain a model complete theory, more complex operators are needed to express acceptance by a tree automaton.

A functional language $\mathcal{L}_2$: Boolean algebra operations $(\perp, \cup, -)$, constant $\mathbf{I}$, unary operations $\mathbf{X}_0$, $\mathbf{X}_1$, binary operations **EU** and **AF**.

Binary trees

The full binary tree is 2^*, finite sequences of 0's and 1's.

To obtain a model complete theory, more complex operators are needed to express acceptance by a tree automaton.

A functional language $\mathcal{L}_2$: Boolean algebra operations $(\bot, \cup, -)$, constant $\mathbf{I}$, unary operations $\mathbf{X}_0$, $\mathbf{X}_1$, binary operations **EU** and **AF**.

The Boolean algebra $\mathcal{P}(2^*)$ is an $\mathcal{L}_2$-structure with

- $\mathbf{I} := \{\epsilon\}$,
- $\mathbf{X}_i a := \{t \in \omega \mid t \cdot i \in a\}$ for $i = 0, 1$,
- $t \in \mathbf{EU}(a, b)$ iff there **E**xists a path $t = t_0, \ldots, t_n$ such that, for $i < n$, $t_i \in a$, and (**U**ntil) $t_n \in b$,

Binary trees

The full binary tree is 2^*, finite sequences of 0's and 1's.

To obtain a model complete theory, more complex operators are needed to express acceptance by a tree automaton.

A functional language $\mathcal{L}_2$: Boolean algebra operations $(\bot, \cup, -)$, constant $\mathbf{I}$, unary operations $\mathbf{X}_0$, $\mathbf{X}_1$, binary operations **EU** and **AF**.

The Boolean algebra $\mathcal{P}(2^*)$ is an $\mathcal{L}_2$-structure with

- $\mathbf{I} := \{\epsilon\}$,
- $\mathbf{X}_i a := \{t \in \omega \mid t \cdot i \in a\}$ for $i = 0, 1$,
- $t \in \mathbf{EU}(a, b)$ iff there **E**xists a path $t = t_0, \ldots, t_n$ such that, for $i < n$, $t_i \in a$, and (**U**ntil) $t_n \in b$,
- $t \in \mathbf{AF}(a, -b)$ iff for **A**ll infinite paths $t = t_0, t_1, \ldots$ there is a (**F**uture) $t_i \in a$, provided that $t_j \in b$ for infinitely many j.

Model companion for binary trees

Theorem

The $\mathcal{L}_2$-theory of $\mathcal{P}(2^*)$ is model complete, and is in fact the model companion of an $\mathcal{L}_2$-theory with a finite universal axiomatization.

Ghilardi, G. LICS 2016

Model companion for binary trees

Theorem

The $\mathcal{L}_2$-theory of $\mathcal{P}(2^*)$ is model complete, and is in fact the model companion of an $\mathcal{L}_2$-theory with a finite universal axiomatization.

Ghilardi, G. LICS 2016

- Proving model completeness crucially uses tree automata originally developed for deciding S2S (Rabin 1969).
- We obtain an analogous result for 'bisimulation-invariant' MSO, i.e., the modal μ-calculus (Janin-Walukiewicz 1996).

Ongoing work and questions

- Ongoing work: extending these results to general trees; this requires an infinite language that can count successors.
- Where do $\mathcal{L}$-structures and $\mathcal{L}_2$-structures fit in model theory?
  - Context: model companions also exist for Heyting algebras and certain modal algebras; but the methods are different.
- Can automata methods be useful for proving the model completeness of other theories (especially if they have a 'computation' flavor)?

Monadic second order logic as a model companion

Sam van Gool

samvangool@me.com

University of Utrecht (until 31 Aug) / Paris (starting 1 Sept)

Logic Colloquium, Prague

13 August 2019