What is an existentially closed Heyting algebra and what does it have to do with automata?

Sam van Gool

IRIF, Université de Paris

IITK Logic Webinar
2 April 2021

About this talk

- **Objective.** Show an instance of interaction between
  logical algebra, model theory and automata.

About this talk

- **Objective.** Show an instance of interaction between
  logical algebra, model theory and automata.
- **Format.** Part tutorial (∼20m), part research talk (∼40m).

Overview

Part I: A tutorial on algebra in logic

Regular languages and logic

Logical algebra

Part II: Model completeness in logical algebra

Model completeness and model companions

MSO on omega is the model companion of LTL

An excursion to trees

An excursion to Heyting

Overview

Overview

Part I: A tutorial on algebra in logic

Regular languages and logic

Logical algebra

Part II: Model completeness in logical algebra

Model completeness and model companions

MSO on omega is the model companion of LTL

An excursion to trees

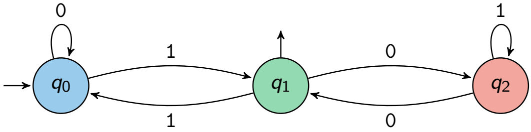
An excursion to Heyting

Three ways of describing a regular language

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

Three ways of describing a regular language

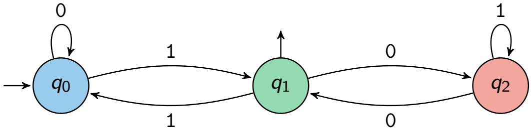
- A programming problem: given a natural number in binary, $w \in \{0,1\}^*$, determine if w is congruent 1 modulo 3.
- Solution 1: a (deterministic) automaton A:



Answer yes iff A accepts w.

Three ways of describing a regular language

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



Answer yes iff A accepts w.

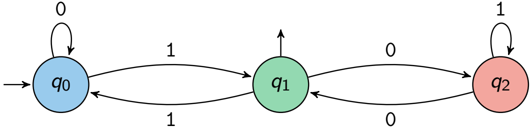
- Solution 2: a homomorphism $\varphi\colon \{0,1\}^* \to S_3$ defined by

$$0 \mapsto (1\,2), \quad 1 \mapsto (0\,1).$$

Answer yes iff the permutation $\varphi(w)$ sends 0 to 1.

Three ways of describing a regular language

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



Answer yes iff A accepts w.

- Solution 3: a predicate formula φ describing A:

$\exists Q_0 \exists Q_1 \exists Q_2 (Q_0(\texttt{first}) \land Q_1(\texttt{last}) \land$

$\forall x[\texttt{0}(x) \land Q_0(x) \to Q_0(\texttt{S}x)] \land [\texttt{1}(x) \land Q_0(x) \to Q_1(\texttt{S}x)] \land \dots).$

Answer yes iff w satisfies the formula φ.

Predicate logic on finite words

- **Syntax.** Monadic Second Order (MSO) logic over $<$, Σ.
  - Basic propositional connectives: $\wedge$, $\neg$.
  - Quantification over first-order variables x, y, ... and one-place (monadic) second-order variables P, Q, ...
  - Relational signature: $x < y$, $\mathtt{a}(x)$ for $a \in \Sigma$.

Predicate logic on finite words

- **Syntax.** Monadic Second Order (MSO) logic over $<$, Σ.
  - Basic propositional connectives: $\wedge$, $\neg$.
  - Quantification over first-order variables x, y, ... and one-place (monadic) second-order variables P, Q, ...
  - Relational signature: $x < y$, $\mathrm{a}(x)$ for $a \in \Sigma$.

- **Semantics.** A finite word $w = a_1 \ldots a_n$ gives a structure W.
  - The underlying set of W is $\{1, \ldots, n\}$.
  - The natural linear order $<^W$ interprets the binary predicate $<$.
  - For every letter $a \in \Sigma$, $\mathrm{a}^W := \{i \in \{1, \ldots, n\} \colon a_i = a\}$.

Predicate logic on finite words

- **Syntax.** Monadic Second Order (MSO) logic over $<$, Σ.
- **Semantics.** A finite word $w = a_1 \ldots a_n$ gives a structure W.

Predicate logic on finite words

- **Syntax.** Monadic Second Order (MSO) logic over $<$, Σ.
- **Semantics.** A finite word $w = a_1 \dots a_n$ gives a structure W.

- For a sentence φ, $L_\varphi := \{w \in \Sigma^* \mid w \models \varphi\}$.
- Shortcuts $\mathtt{S}x$, $\mathtt{first}$, $\mathtt{last}$, $\subseteq$ are MSO-definable.

Overview

Part I: A tutorial on algebra in logic

Regular languages and logic

Logical algebra

Part II: Model completeness in logical algebra

Model completeness and model companions

MSO on omega is the model companion of LTL

An excursion to trees

An excursion to Heyting

Algebra on two levels

- The collection of finite Σ-words has monoid structure, i.e., an associative operation with unit.

Algebra on two levels

- The collection of finite Σ-words has monoid structure, i.e., an associative operation with unit.
- The collection of sets of finite Σ-words (Σ-languages), has Boolean algebra structure, i.e., a ring in which all elements are idempotent, which is moreover equipped with 'modal' operators, such as, for any Σ-language L,

$$a^{-1}L := \{w \in \Sigma^* \ : \ aw \in L\}, \quad a \in \Sigma.$$

Algebra on two levels

- The collection of finite Σ-words has monoid structure, i.e., an associative operation with unit.
- The collection of sets of finite Σ-words (Σ-languages), has Boolean algebra structure, i.e., a ring in which all elements are idempotent, which is moreover equipped with 'modal' operators, such as, for any Σ-language L,

$$a^{-1}L := \{w \in \Sigma^* \ : \ aw \in L\}, \quad a \in \Sigma.$$

- This phenomenon of 'algebra on two levels' is a special instance of Stone duality.

Boolean algebras of languages

- The set Sent(Σ) of all MSO-sentences over a fixed finite alphabet Σ carries a preorder, $\vdash$:

$$\varphi \vdash \psi \iff \text{for every finite word } W, \text{ if } W \models \varphi, \text{ then } W \models \psi.$$

Boolean algebras of languages

- The set Sent(Σ) of all MSO-sentences over a fixed finite alphabet Σ carries a preorder, $\vdash$:

$$\varphi \vdash \psi \iff \text{for every finite word } W, \text{ if } W \models \varphi, \text{ then } W \models \psi.$$

- The quotient of Sent(Σ) under $\vdash$-equivalence is a *Boolean algebra*, $R(\Sigma)$:

$$[\varphi] \cdot [\psi] := [\varphi \wedge \psi], \quad [\varphi] + [\psi] := [\varphi \text{ xor } \psi], \quad 0 := [\bot].$$

Boolean algebras of languages

- The set $\mathrm{Sent}(\Sigma)$ of all MSO-sentences over a fixed finite alphabet Σ carries a preorder, $\vdash$:

$$\varphi \vdash \psi \iff \text{for every finite word } W, \text{ if } W \models \varphi, \text{ then } W \models \psi.$$

- The quotient of $\mathrm{Sent}(\Sigma)$ under $\vdash$-equivalence is a *Boolean algebra*, $R(\Sigma)$:

$$[\varphi] \cdot [\psi] := [\varphi \wedge \psi], \quad [\varphi] + [\psi] := [\varphi \text{ xor } \psi], \quad 0 := [\bot].$$

- **Theorem (Büchi 1960).** The image of the injection

$$R(\Sigma) \hookrightarrow \mathcal{P}(\Sigma^*), \quad \varphi \mapsto \{W \in \Sigma^* \mid W \models \varphi\}$$

consists of the regular Σ-languages.

Logical algebra

- The abstractions of arithmetic algebra allow us not to think about concrete numbers, but also treat other entities, such as polynomials and permutations, as if they are numbers.

Logical algebra

- The abstractions of arithmetic algebra allow us not to think about concrete numbers, but also treat other entities, such as polynomials and permutations, as if they are numbers.

- The abstractions of logical algebra allow us not to think about concrete formulas, but also treat other entities, such as languages and other sets, as if they are formulas.

Boolean algebras as lattices

- A bounded lattice is a tuple $(L, \leq, \vee, \wedge, \bot, \top)$, where $\leq$ is a partial order, and for any $a, b \in L$, $a \vee b = \sup\{a, b\}$, $a \wedge b = \inf\{a, b\}$, $\bot = \sup \emptyset$, and $\top = \inf \emptyset$.

Boolean algebras as lattices

- A bounded lattice is a tuple $(L, \leq, \vee, \wedge, \bot, \top)$, where $\leq$ is a partial order, and for any $a, b \in L$, $a \vee b = \sup\{a, b\}$, $a \wedge b = \inf\{a, b\}$, $\bot = \sup \emptyset$, and $\top = \inf \emptyset$.
  - Note that $\leq$ is definable from $\vee$ or $\wedge$.

Boolean algebras as lattices

- A bounded lattice is a tuple $(L, \leq, \vee, \wedge, \bot, \top)$, where $\leq$ is a partial order, and for any $a, b \in L$, $a \vee b = \sup\{a, b\}$, $a \wedge b = \inf\{a, b\}$, $\bot = \sup \emptyset$, and $\top = \inf \emptyset$.
  - Note that $\leq$ is definable from $\vee$ or $\wedge$.
  - Bounded lattices can be axiomatized without $\leq$.

Boolean algebras as lattices

- A bounded lattice is a tuple $(L, \leq, \vee, \wedge, \bot, \top)$, where $\leq$ is a partial order, and for any $a, b \in L$, $a \vee b = \sup\{a, b\}$, $a \wedge b = \inf\{a, b\}$, $\bot = \sup \emptyset$, and $\top = \inf \emptyset$.
  - Note that $\leq$ is definable from $\vee$ or $\wedge$.
  - Bounded lattices can be axiomatized without $\leq$.
  - A bounded lattice L is distributive if, for any $a, b, c \in L$,

$$a \wedge (b \vee c) = (a \wedge b) \vee (a \wedge c).$$

Boolean algebras as lattices

- A bounded lattice is a tuple $(L, \leq, \vee, \wedge, \bot, \top)$, where $\leq$ is a partial order, and for any $a, b \in L$, $a \vee b = \sup\{a, b\}$, $a \wedge b = \inf\{a, b\}$, $\bot = \sup \emptyset$, and $\top = \inf \emptyset$.
  - Note that $\leq$ is definable from $\vee$ or $\wedge$.
  - Bounded lattices can be axiomatized without $\leq$.
  - A bounded lattice L is distributive if, for any $a, b, c \in L$,

$$a \wedge (b \vee c) = (a \wedge b) \vee (a \wedge c).$$

- A Boolean algebra is a tuple $(B, \vee, \wedge, \neg, \bot, \top)$ where
  - $(B, \vee, \wedge, \bot, \top)$ is a bounded distributive lattice;
  - for any $a \in B$, $a \vee \neg a = \top$ and $a \wedge \neg a = \bot$.

Boolean algebras as lattices

- A bounded lattice is a tuple $(L, \leq, \vee, \wedge, \bot, \top)$, where $\leq$ is a partial order, and for any $a, b \in L$, $a \vee b = \sup\{a, b\}$, $a \wedge b = \inf\{a, b\}$, $\bot = \sup \emptyset$, and $\top = \inf \emptyset$.
  - Note that $\leq$ is definable from $\vee$ or $\wedge$.
  - Bounded lattices can be axiomatized without $\leq$.
  - A bounded lattice L is distributive if, for any $a, b, c \in L$,

$$a \wedge (b \vee c) = (a \wedge b) \vee (a \wedge c).$$

- A Boolean algebra is a tuple $(B, \vee, \wedge, \neg, \bot, \top)$ where
  - $(B, \vee, \wedge, \bot, \top)$ is a bounded distributive lattice;
  - for any $a \in B$, $a \vee \neg a = \top$ and $a \wedge \neg a = \bot$.
- **Exercise.** Boolean algebras are term-equivalent with idempotent commutative rings with unit.

Terms as formulas

- A polynomial, e.g., $xy - x^2$, gets a *value* whenever its variables are assigned values in a ring.

Terms as formulas

- A polynomial, e.g., $xy - x^2$, gets a *value* whenever its variables are assigned values in a ring.
- A propositional formula, e.g., $(x \wedge y) \vee \neg(x \vee x)$, gets a *value* whenever its variables are assigned values in a Boolean algebra.

Terms as formulas

- A polynomial, e.g., $xy - x^2$, gets a *value* whenever its variables are assigned values in a ring.

- A propositional formula, e.g., $(x \wedge y) \vee \neg(x \vee x)$, gets a *value* whenever its variables are assigned values in a Boolean algebra.

- The familiar *truth tables* compute these values under assignment to the two-element Boolean algebra $\{\bot, \top\}$.

Terms as formulas

- A polynomial, e.g., $xy - x^2$, gets a *value* whenever its variables are assigned values in a ring.
- A propositional formula, e.g., $(x \wedge y) \vee \neg(x \vee x)$, gets a *value* whenever its variables are assigned values in a Boolean algebra.
- The familiar *truth tables* compute these values under assignment to the two-element Boolean algebra $\{\bot, \top\}$.
- In other logics, a single finite algebra is *not* enough.

Logic from algebra

- *Propositional logic* in variables $x_1, \ldots, x_n$ is the set of Boolean algebra terms that evaluate to $\top$ under any assignment to a Boolean algebra.

Logic from algebra

- *Propositional logic* in variables $x_1, \ldots, x_n$ is the set of Boolean algebra terms that evaluate to $\top$ under any assignment to a Boolean algebra.
- *Modal logic* in variables $x_1, \ldots, x_n$ is the set of modal algebra terms that evaluate to $\top$ under any assignment to a modal algebra.

Logic from algebra

- *Propositional logic* in variables $x_1, \ldots, x_n$ is the set of Boolean algebra terms that evaluate to $\top$ under any assignment to a Boolean algebra.
- *Modal logic* in variables $x_1, \ldots, x_n$ is the set of modal algebra terms that evaluate to $\top$ under any assignment to a modal algebra.
  - A modal algebra is a pair $(B, \Box)$ where B is a Boolean algebra and $\Box \colon B \to B$ preserves $\wedge$ and $\top$.

Logic from algebra

- *Propositional logic* in variables $x_1, \ldots, x_n$ is the set of Boolean algebra terms that evaluate to $\top$ under any assignment to a Boolean algebra.
- *Modal logic* in variables $x_1, \ldots, x_n$ is the set of modal algebra terms that evaluate to $\top$ under any assignment to a modal algebra.
  - A modal algebra is a pair $(B, \Box)$ where B is a Boolean algebra and $\Box \colon B \to B$ preserves $\wedge$ and $\top$.
- *Linear temporal logic* in variables $x_1, \ldots, x_n$ is the set of LTL-algebra terms that evaluate to $\top$ under any assignment to an LTL-algebra.
  - We will define (a variant of) LTL-algebras in Part II.

Logic from algebra

- *Propositional logic* in variables $x_1, \ldots, x_n$ is the set of Boolean algebra terms that evaluate to $\top$ under any assignment to a Boolean algebra.
- *Modal logic* in variables $x_1, \ldots, x_n$ is the set of modal algebra terms that evaluate to $\top$ under any assignment to a modal algebra.
  - A modal algebra is a pair $(B, \Box)$ where B is a Boolean algebra and $\Box \colon B \to B$ preserves $\wedge$ and $\top$.
- *Linear temporal logic* in variables $x_1, \ldots, x_n$ is the set of LTL-algebra terms that evaluate to $\top$ under any assignment to an LTL-algebra.
  - We will define (a variant of) LTL-algebras in Part II.
- *Intuitionistic logic* ... Heyting algebras.

Heyting algebras

- A Heyting algebra is a tuple $(H, \wedge, \vee, \bot, \top, \to)$, where
  - $(H, \wedge, \vee, \bot, \top)$ is a bounded distributive lattice,
  - $\to$ is a *relative pseudocomplement*, that is, for any $a, b, c \in H$,

$$a \wedge b \leq c \text{ iff } a \leq b \to c.$$

Heyting algebras

- A Heyting algebra is a tuple $(H, \wedge, \vee, \bot, \top, \to)$, where
  - $(H, \wedge, \vee, \bot, \top)$ is a bounded distributive lattice,
  - $\to$ is a *relative pseudocomplement*, that is, for any $a, b, c \in H$,

$$a \wedge b \leq c \text{ iff } a \leq b \to c.$$

- *Intuitionistic logic* in variables $x_1, \ldots, x_n$ is the set of Heyting algebra terms that evaluate to $\top$ under any assignment to a Heyting algebra.

Heyting algebras

- A Heyting algebra is a tuple $(H, \wedge, \vee, \bot, \top, \to)$, where
  - $(H, \wedge, \vee, \bot, \top)$ is a bounded distributive lattice,
  - $\to$ is a *relative pseudocomplement*, that is, for any $a, b, c \in H$,

$$a \wedge b \leq c \text{ iff } a \leq b \to c.$$

- *Intuitionistic logic* in variables $x_1, \ldots, x_n$ is the set of Heyting algebra terms that evaluate to $\top$ under any assignment to a Heyting algebra.
- **Exercise.** Write $\neg a := a \to \bot$, $a \leftrightarrow b := (a \to b) \wedge (b \to a)$. The term $(x \wedge \neg x) \leftrightarrow \bot$ is in intuitionistic logic, but $(x \vee \neg x) \leftrightarrow \top$ is not.

Heying algebras

- A Heyting algebra is a tuple $(H, \wedge, \vee, \bot, \top, \to)$, where
  - $(H, \wedge, \vee, \bot, \top)$ is a bounded distributive lattice,
  - $\to$ is a *relative pseudocomplement*, that is, for any $a, b, c \in H$,

$$a \wedge b \leq c \text{ iff } a \leq b \to c.$$

- *Intuitionistic logic* in variables $x_1, \ldots, x_n$ is the set of Heyting algebra terms that evaluate to $\top$ under any assignment to a Heyting algebra.
- **Exercise.** Write $\neg a := a \to \bot$, $a \leftrightarrow b := (a \to b) \wedge (b \to a)$. The term $(x \wedge \neg x) \leftrightarrow \bot$ is in intuitionistic logic, but $(x \vee \neg x) \leftrightarrow \top$ is not.
- **Exercise (non-trivial).** There are infinitely many non-equivalent terms in a single variable x.

Summary of Part I: Tutorial

- Monadic second order logic has the same expressive power as finite automata.
- Boolean algebras are abstract algebraic models for propositional logic.
- Generalizing Boolean algebras in various directions (modal, temporal, and Heyting algebras) allows one to talk about different logics in one algebraic framework.

Overview

Part I: A tutorial on algebra in logic

- Regular languages and logic
- Logical algebra

Part II: Model completeness in logical algebra

- Model completeness and model companions
- MSO on omega is the model companion of LTL
- An excursion to trees
- An excursion to Heyting

Overview

Model theory

- Model theory studies classes of (algebraic) structures through the lens of first order logic.

Model theory

- Model theory studies classes of (algebraic) structures through the lens of first order logic.
- Usually, the structures studied are classical: fields, groups, ...

Model theory

- Model theory studies classes of (algebraic) structures through the lens of first order logic.
- Usually, the structures studied are classical: fields, groups, ...
- In this work, we apply model theory to structures *from logical algebra*, that is, to Boolean algebras, to Heyting algebras, to LTL algebras, and more.

Solving equations

- Solve for $x \in \mathbb{R}$: $x^2 + 1 = 0$.

Solving equations

- Solve for $x \in \mathbb{C}$: $x^2 + 1 = 0$.

Solving equations

- Solve for $x \in \mathbb{C}$: $x^2 + 1 = 0$.
- A field F is existentially closed if any existential sentence that becomes true in some field extension of F already holds in F.

Solving equations

- Solve for $x \in \mathbb{C}$: $x^2 + 1 = 0$.
- A field F is existentially closed if any existential sentence that becomes true in some field extension of F already holds in F.
- This is first order definable: F is existentially closed iff

  for every non-constant polynomial p, $F \models \exists \overline{x} p(\overline{x}) = 0$.

Solving equations

- Solve for $x \in \mathbb{C}$: $x^2 + 1 = 0$.
- A field F is existentially closed if any existential sentence that becomes true in some field extension of F already holds in F.
- This is first order definable: F is existentially closed iff

  for every non-constant polynomial p, $F \models \exists \overline{x} p(\overline{x}) = 0$.

- A T-structure A is existentially closed* if any existential sentence that becomes true in some T-structure extending A already holds in A.

* If the class of T-structures does not have amalgamation, a more complicated definition is needed.

Solving equations

- Solve for $x \in \mathbb{C}$: $x^2 + 1 = 0$.
- A field F is existentially closed if any existential sentence that becomes true in some field extension of F already holds in F.
- This is first order definable: F is existentially closed iff

  for every non-constant polynomial p, $F \models \exists \overline{x} p(\overline{x}) = 0$.

- A T-structure A is existentially closed* if any existential sentence that becomes true in some T-structure extending A already holds in A.
- This property is often first order definable:
  - *Linear orders without endpoints:* density;
  - *Boolean algebras:* atomless;
  - *Heyting algebras:* I will sketch this a few slides from now.

* If the class of T-structures does not have amalgamation, a more complicated definition is needed.

Model companion

A first order theory T^* which captures the existentially closed models for a universal theory T is called a model companion of T.

Theorem.
The theory T^*, if it exists, is the unique theory such that:

1. T and T^* believe the same universal sentences;

2. For any sentence φ, there is an existential sentence φ' such that T^* believes $\varphi \leftrightarrow \varphi'$.

Robinson, 1963

Model companion

A first order theory T^* which captures the existentially closed models for a universal theory T is called a model companion of T.

Theorem.
The theory T^*, if it exists, is the unique theory such that:

1. T and T^* believe the same universal sentences;

   T and T^* are co-theories

2. For any sentence φ, there is an existential sentence φ' such that T^* believes $\varphi \leftrightarrow \varphi'$.

   T^* is model complete

Robinson, 1963

Overview

MSO on omega is the model companion of LTL

Joint work with Silvio Ghilardi (Milan).

Model companions and languages

Theorem.

The first order theory T^* of an algebra for word languages, $\mathcal{P}(\omega)$,

is the model companion of

a theory T of algebras for a fragment of linear temporal logic.

"MSO on ω is the model companion of LTL"

Ghilardi & G. JSL 2017

For convenience, we switch from finite words to ω-words for this part.

The theory T^*: the generic LTL-algebra

- The Boolean algebra $\mathcal{P}(\omega)$ carries temporal operators:
  - $\mathbf{X}a := \{t \in \omega \mid t+1 \in a\}$,
  - $\mathbf{F}a := \{t \in \omega \mid \exists t' \geq t \colon t' \in a\}$,
  - $\mathbf{I} := \{0\}$.

The theory T^*: the generic LTL-algebra

- The Boolean algebra $\mathcal{P}(\omega)$ carries temporal operators:
  - $\mathbf{X}a := \{t \in \omega \mid t + 1 \in a\}$,
  - $\mathbf{F}a := \{t \in \omega \mid \exists t' \geq t \colon t' \in a\}$,
  - $\mathbf{I} := \{0\}$.

- The theory T^* is the theory, $\mathrm{Th}(\mathcal{P}(\omega))$, of this (single) structure, in the signature $\{\vee, \wedge, \bot, \top, \neg, \mathbf{X}, \mathbf{F}, \mathbf{I}\} \cup \{=\}$.

The theory T^*: the generic LTL-algebra

- The Boolean algebra $\mathcal{P}(\omega)$ carries temporal operators:
  - $\mathbf{X}a := \{t \in \omega \mid t+1 \in a\}$,
  - $\mathbf{F}a := \{t \in \omega \mid \exists t' \geq t \colon t' \in a\}$,
  - $\mathbf{I} := \{0\}$.

- The theory T^* is the theory, $\mathrm{Th}(\mathcal{P}(\omega))$, of this (single) structure, in the signature $\{\vee, \wedge, \bot, \top, \neg, \mathbf{X}, \mathbf{F}, \mathbf{I}\} \cup \{=\}$.

- **Exercise.** Let φ be an $(\mathbf{X}, \mathbf{F}, \mathbf{I})$-formula in variables $x_1, \ldots, x_n$. For each $1 \leq i \leq n$, let $X_i \subseteq \omega$. For any $t \in \omega$, we have

$$t \in \varphi^{\mathcal{P}(\omega)}(\overline{X}) \text{ iff } \varphi \text{ holds at } t \text{ in the Kripke model } (\omega, \overline{X}).$$

The theory T: general LTL-algebras

- A linear temporal algebra is a tuple $(\mathbf{B}, \mathbf{X}, \mathbf{F}, \mathbf{I})$, where
  - $\mathbf{B} = (B, \vee, \wedge, \neg, \bot, \top)$ is a Boolean algebra;
  - $\mathbf{X}$ is an endomorphism of B;
  - $\mathbf{I}$ is an atom, $\mathbf{XI} = \bot$, and $\mathbf{I} \leq \mathbf{X}a$ when $a \neq \bot$.
  - for any $a \in B$, $\mathbf{F}a$ is the least fixed point of $x \mapsto a \vee \mathbf{X}x$, i.e.,

The theory T: general LTL-algebras

- A linear temporal algebra is a tuple $(\mathbf{B}, \mathbf{X}, \mathbf{F}, \mathbf{I})$, where
  - $\mathbf{B} = (B, \vee, \wedge, \neg, \bot, \top)$ is a Boolean algebra;
  - $\mathbf{X}$ is an endomorphism of B;
  - $\mathbf{I}$ is an atom, $\mathbf{XI} = \bot$, and $\mathbf{I} \leq \mathbf{X}a$ when $a \neq \bot$.
  - for any $a \in B$, $\mathbf{F}a$ is the least fixed point of $x \mapsto a \vee \mathbf{X}x$, i.e.,
    - $a \vee \mathbf{XF}a \leq \mathbf{F}a$, and
    - for any $b \in B$, if $a \vee \mathbf{X}b \leq b$, then $\mathbf{F}a \leq b$.

The theory T: general LTL-algebras

- A linear temporal algebra is a tuple $(\mathbf{B}, \mathbf{X}, \mathbf{F}, \mathbf{I})$, where
  - $\mathbf{B} = (B, \vee, \wedge, \neg, \bot, \top)$ is a Boolean algebra;
  - $\mathbf{X}$ is an endomorphism of B;
  - $\mathbf{I}$ is an atom, $\mathbf{XI} = \bot$, and $\mathbf{I} \leq \mathbf{X}a$ when $a \neq \bot$.
  - for any $a \in B$, $\mathbf{F}a$ is the least fixed point of $x \mapsto a \vee \mathbf{X}x$, i.e.,
    - $a \vee \mathbf{XF}a \leq \mathbf{F}a$, and
    - for any $b \in B$, if $a \vee \mathbf{X}b \leq b$, then $\mathbf{F}a \leq b$.
- T is the theory of the linear temporal algebras.

The theory T: general LTL-algebras

- A linear temporal algebra is a tuple $(\mathbf{B}, \mathbf{X}, \mathbf{F}, \mathbf{I})$, where
  - $\mathbf{B} = (B, \vee, \wedge, \neg, \bot, \top)$ is a Boolean algebra;
  - $\mathbf{X}$ is an endomorphism of B;
  - $\mathbf{I}$ is an atom, $\mathbf{XI} = \bot$, and $\mathbf{I} \leq \mathbf{X}a$ when $a \neq \bot$.
  - for any $a \in B$, $\mathbf{F}a$ is the least fixed point of $x \mapsto a \vee \mathbf{X}x$, i.e.,
    - $a \vee \mathbf{XF}a \leq \mathbf{F}a$, and
    - for any $b \in B$, if $a \vee \mathbf{X}b \leq b$, then $\mathbf{F}a \leq b$.

- T is the theory of the linear temporal algebras.

- **Theorem.** $T^* = \mathrm{Th}(\mathcal{P}(\omega))$ is the model companion of T.

The theory T: general LTL-algebras

- A linear temporal algebra is a tuple $(\mathbf{B}, \mathbf{X}, \mathbf{F}, \mathbf{I})$, where
  - $\mathbf{B} = (B, \vee, \wedge, \neg, \bot, \top)$ is a Boolean algebra;
  - $\mathbf{X}$ is an endomorphism of B;
  - $\mathbf{I}$ is an atom, $\mathbf{XI} = \bot$, and $\mathbf{I} \leq \mathbf{X}a$ when $a \neq \bot$.
  - for any $a \in B$, $\mathbf{F}a$ is the least fixed point of $x \mapsto a \vee \mathbf{X}x$, i.e.,
    - $a \vee \mathbf{XF}a \leq \mathbf{F}a$, and
    - for any $b \in B$, if $a \vee \mathbf{X}b \leq b$, then $\mathbf{F}a \leq b$.

- T is the theory of the linear temporal algebras.

- **Theorem.** $T^* = \mathrm{Th}(\mathcal{P}(\omega))$ is the model companion of T.
  - Co-theories: a non-trivial **exercise**.

The theory T: general LTL-algebras

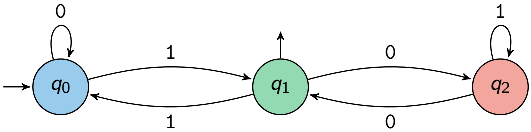
- A linear temporal algebra is a tuple $(\mathbf{B}, \mathbf{X}, \mathbf{F}, \mathbf{I})$, where
  - $\mathbf{B} = (B, \vee, \wedge, \neg, \bot, \top)$ is a Boolean algebra;
  - $\mathbf{X}$ is an endomorphism of B;
  - $\mathbf{I}$ is an atom, $\mathbf{XI} = \bot$, and $\mathbf{I} \leq \mathbf{X}a$ when $a \neq \bot$.
  - for any $a \in B$, $\mathbf{F}a$ is the least fixed point of $x \mapsto a \vee \mathbf{X}x$, i.e.,
    - $a \vee \mathbf{XF}a \leq \mathbf{F}a$, and
    - for any $b \in B$, if $a \vee \mathbf{X}b \leq b$, then $\mathbf{F}a \leq b$.

- T is the theory of the linear temporal algebras.

- **Theorem.** $T^* = \mathrm{Th}(\mathcal{P}(\omega))$ is the model companion of T.
  - Co-theories: a non-trivial **exercise**.
  - Model completeness of T^*: automata!

Recall our first example

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



Answer yes iff A accepts w.

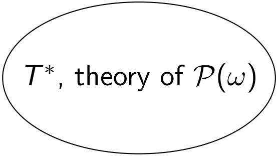
- Solution 3: a predicate formula φ describing A:

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

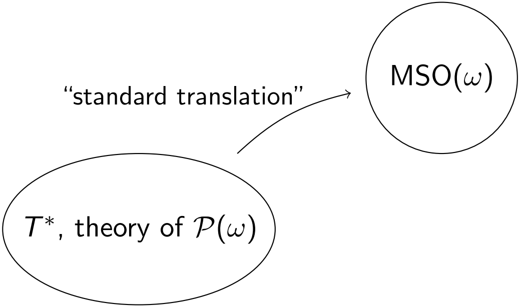
$\forall x [\texttt{0}(x) \wedge Q_0(x) \rightarrow Q_0(\texttt{S}x)] \wedge [\texttt{1}(x) \wedge Q_0(x) \rightarrow Q_1(\texttt{S}x)] \wedge \dots).$

Answer yes iff w satisfies the formula φ.

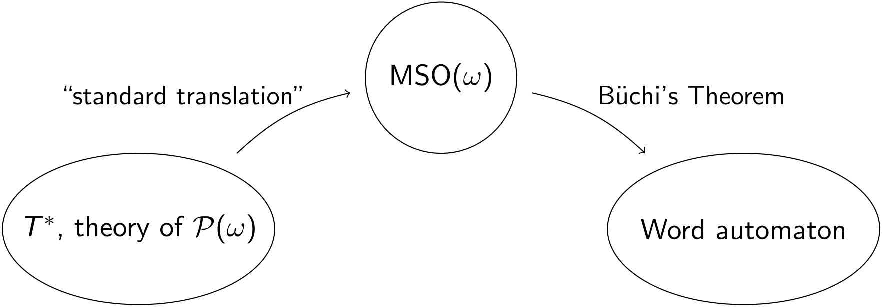
Proving model completeness with automata



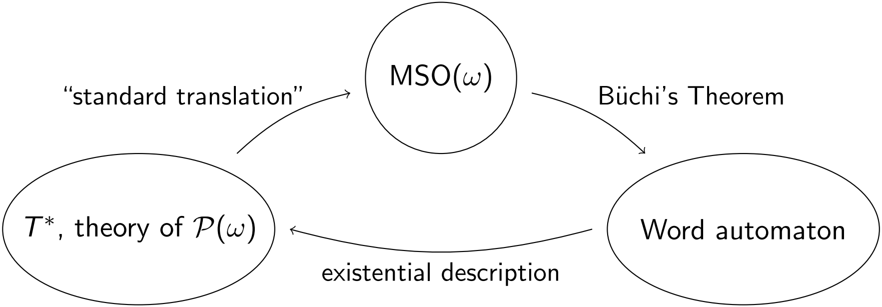
Proving model completeness with automata



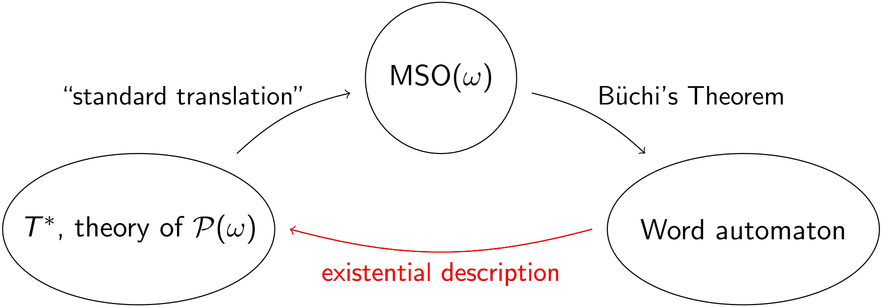
Proving model completeness with automata



Proving model completeness with automata



Proving model completeness with automata



An existential 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 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 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 $\Sigma = \{0, \dots, s\}$, $Q = \{0, \dots, m\}$, $q_0 = 0$.
- A word $W\colon \omega \to \Sigma$ is a partition $(W_0, \dots, W_s)$ of ω; $W_j = W^{-1}(j)$.

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

An existential 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:

$$\exists q_0, \ldots, q_m (\text{“the } q_i \text{ partition } \omega\text{''} \,\&\, \mathop{\&}_{\substack{0 \le i \le m \\ 0 \le j \le s}} \left(q_i \wedge w_j \le \bigvee_{k \in \delta(i,j)} \mathbf{X} q_k \right)$$

An existential 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:

$$\exists q_0, \ldots, q_m (\text{"the } q_i \text{ partition } \omega\text{"} \,\&\, \mathop{\&}_{\substack{0 \le i \le m \\ 0 \le j \le s}} \left(q_i \wedge w_j \le \bigvee_{k \in \delta(i,j)} \mathbf{X} q_k \right)$$

$$\&\ \ I \le q_0 \,\&\, \mathbf{F} \left(\textstyle\bigvee_{i \in F} q_i \right) = \top).$$

Proof outline: T^* is model complete

- Any first order formula φ in the temporal algebra $\mathcal{P}(\omega)$ translates to an MSO formula Φ in logic on ω-words.

Proof outline: T^* is model complete

- Any first order formula φ in the temporal algebra $\mathcal{P}(\omega)$ translates to an MSO formula Φ in logic on ω-words.

- This MSO formula Φ defines a regular language L_Φ.

Proof outline: T^* is model complete

- Any first order formula φ in the temporal algebra $\mathcal{P}(\omega)$ translates to an MSO formula Φ in logic on ω-words.
- This MSO formula Φ defines a regular language L_Φ.
- Build an automaton A for L_Φ.

Proof outline: T^* is model complete

- Any first order formula φ in the temporal algebra $\mathcal{P}(\omega)$ translates to an MSO formula Φ in logic on ω-words.
- This MSO formula Φ defines a regular language L_Φ.
- Build an automaton A for L_Φ.
- Describe the automaton A with an existential first order formula φ' in the temporal algebra $\mathcal{P}(\omega)$.

Proof outline: T^* is model complete

- Any first order formula φ in the temporal algebra $\mathcal{P}(\omega)$ translates to an MSO formula Φ in logic on ω-words.
- This MSO formula Φ defines a regular language L_Φ.
- Build an automaton A for L_Φ.
- Describe the automaton A with an existential first order formula φ' in the temporal algebra $\mathcal{P}(\omega)$.
- **Conclusion.** $\mathcal{P}(\omega)$ believes that any first order formula φ is equivalent to an existential formula φ'.

Model companions and languages

Theorem.

The first order theory T^* of an algebra for word languages, $\mathcal{P}(\omega)$,

is the model companion of

a theory T of algebras for a linear temporal logic.

Ghilardi & G. JSL 2017

Model companions and languages

Theorem.

The first order theory T^* of an algebra for tree languages, $\mathcal{P}(2^*)$,

is the model companion of

a theory T of algebras for a fair computation tree logic.

Ghilardi & G. LICS 2016

Overview

Tree automata

- Suppose (T, R) is a tree with root t_0, and let $\sigma \colon T \to \Sigma$.

Tree automata

- Suppose (T, R) is a tree with root t_0, and let $\sigma\colon T \to \Sigma$.
- A tree automaton $\mathcal{A}$ is a tuple $(Q, \Sigma, \delta, q_0, \Omega)$, where $\delta\colon Q \times \Sigma \to \mathcal{P}\mathcal{P}(Q)$, $q_0 \in Q$, and $\Omega\colon Q \to \omega$.

Tree automata

- Suppose (T, R) is a tree with root t_0, and let $\sigma\colon T \to \Sigma$.
- A tree automaton $\mathcal{A}$ is a tuple $(Q, \Sigma, \delta, q_0, \Omega)$, where $\delta\colon Q \times \Sigma \to \mathcal{PP}(Q)$, $q_0 \in Q$, and $\Omega\colon Q \to \omega$.
- A run of $\mathcal{A}$ on (T, R, t_0, σ) is a function $r\colon T \to Q$ such that $r(t_0) = q_0$ and $\{r(t') \mid tRt'\} \in \delta(r(t), \sigma(t))$ for all $t \in T$.

Tree automata

- Suppose (T, R) is a tree with root t_0, and let $\sigma\colon T \to \Sigma$.
- A tree automaton $\mathcal{A}$ is a tuple $(Q, \Sigma, \delta, q_0, \Omega)$, where $\delta\colon Q \times \Sigma \to \mathcal{PP}(Q)$, $q_0 \in Q$, and $\Omega\colon Q \to \omega$.
- A run of $\mathcal{A}$ on (T, R, t_0, σ) is a function $r\colon T \to Q$ such that $r(t_0) = q_0$ and $\{r(t') \mid tRt'\} \in \delta(r(t), \sigma(t))$ for all $t \in T$.
- The automaton $\mathcal{A}$ accepts (T, R, t_0, σ) if there is a run r such that, for every infinite path $(t_i)_{i=1}^{\omega}$ in T, the number

$$\min\{\Omega(q) \mid r(t_i) = q \text{ for infinitely many } i\}$$

is even.

Rabin-Janin-Walukiewicz theorem

Theorem (Rabin 1969, Janin & Walukiewicz 1995)

For any MSO formula $\Phi(\overline{p})$, there exists an automaton $\mathcal{A}_\Phi$ on the alphabet $\Sigma = 2^{\overline{p}}$ such that, for any tree (T, R, t_0) and colouring $\sigma\colon T \to \Sigma$,

$$T_\omega, \sigma_\omega \models \Phi(\overline{p}) \iff \mathcal{A}_\Phi \text{ accepts } (T_\omega, \sigma_\omega).$$

Rabin-Janin-Walukiewicz theorem

Theorem (Rabin 1969, Janin & Walukiewicz 1995)

For any MSO formula $\Phi(\overline{p})$, there exists an automaton $\mathcal{A}_\Phi$ on the alphabet $\Sigma = 2^{\overline{p}}$ such that, for any tree (T, R, t_0) and colouring $\sigma\colon T \to \Sigma$,

$$T_\omega, \sigma_\omega \models \Phi(\overline{p}) \iff \mathcal{A}_\Phi \text{ accepts } (T_\omega, \sigma_\omega).$$

Here, T_ω is the ω-unravelling of the tree T.

$\mathcal{P}(T)$ as an algebra for fair CTL

- Problem: what is the appropriate enrichment of Boolean algebras to capture the acceptance condition of tree automata?

$\mathcal{P}(T)$ as an algebra for fair CTL

- Problem: what is the appropriate enrichment of Boolean algebras to capture the acceptance condition of tree automata?
- Solution: a version of CTL with local fairness constraints.

$\mathcal{P}(T)$ as an algebra for fair CTL

- Problem: what is the appropriate enrichment of Boolean algebras to capture the acceptance condition of tree automata?
- Solution: a version of CTL with local fairness constraints.
- For $P, Q \subseteq T$, define
  - $\mathbf{EU}(P, Q) := \{t \in T \mid$ there exists a finite path starting in t and ending in Q, which stays in P until (excl.) the end$\}$.

$\mathcal{P}(T)$ as an algebra for fair CTL

- Problem: what is the appropriate enrichment of Boolean algebras to capture the acceptance condition of tree automata?
- Solution: a version of CTL with local fairness constraints.
- For $P, Q \subseteq T$, define
  - $\mathbf{EU}(P, Q) := \{t \in T \mid$ there exists a finite path starting in t and ending in Q, which stays in P until (excl.) the end$\}$.
  - $\mathbf{AF}(P, Q) := \{t \in T \mid$ for every infinite path starting in t, if the path visits $T \setminus Q$ infinitely often, then some node on the path is in $P\}$.

$\mathcal{P}(T)$ as an algebra for fair CTL

- Problem: what is the appropriate enrichment of Boolean algebras to capture the acceptance condition of tree automata?
- Solution: a version of CTL with local fairness constraints.
- For $P, Q \subseteq T$, define
  - $\mathbf{EU}(P, Q) := \{t \in T \mid$ there exists a finite path starting in t and ending in Q, which stays in P until (excl.) the end$\}$.
  - $\mathbf{AF}(P, Q) := \{t \in T \mid$ for every infinite path starting in t, if the path visits $T \setminus Q$ infinitely often, then some node on the path is in $P\}$.
- A run r with associated partition $\overline{q}$ of T will be accepting iff,

$\mathcal{P}(T)$ as an algebra for fair CTL

- Problem: what is the appropriate enrichment of Boolean algebras to capture the acceptance condition of tree automata?
- Solution: a version of CTL with local fairness constraints.
- For $P, Q \subseteq T$, define
  - $\mathbf{EU}(P, Q) := \{t \in T \mid$ there exists a finite path starting in t and ending in Q, which stays in P until (excl.) the end$\}$.
  - $\mathbf{AF}(P, Q) := \{t \in T \mid$ for every infinite path starting in t, if the path visits $T \setminus Q$ infinitely often, then some node on the path is in $P\}$.
- A run r with associated partition $\overline{q}$ of T will be accepting iff, for every odd n in the range of Ω,

$$\mathbf{AF}\left(\bigvee_{\Omega(q')<n} q', \neg\left[\bigvee_{\Omega(q)=n} q\right]\right) = \top.$$

Axiomatizing fair CTL

- Boolean algebra axioms,
- Standard axioms for $\Diamond$ and **I**,
- Fixpoint axioms and rules for **EU**:

$$a \vee (b \wedge \Diamond \mathbf{EU}(a, b)) \leq \mathbf{EU}(\varphi, \psi)$$

$$\frac{a \vee (b \wedge \Diamond c) \leq c}{\mathbf{EU}(a, b) \leq c}$$

- and for **EG**:

$$\mathbf{EG}(a, b) \leq a \wedge \Diamond \mathbf{EU}(b \wedge \mathbf{EG}(a, b), a)$$

$$\frac{c \leq a \wedge \Diamond \mathbf{EU}(b \wedge c, a)}{c \rightarrow \mathbf{EG}(a, b)}$$

- **AF** is an abbrevation, $\mathbf{AF}(\varphi, \psi) := \neg \mathbf{EG}(\neg \varphi, \neg \psi)$.

Algebras for fair CTL

- A fair CTL algebra is a tuple $(A, \mathbf{I}, \Diamond, \mathbf{EG}, \mathbf{EU})$ such that
  - A is a Boolean algebra,
  - $\mathbf{I}$ is a constant symbol,
  - $\Diamond$ is a unary operation,
  - $\mathbf{EG}$ and $\mathbf{EU}$ are binary operations,
  - the universal axioms on the previous slide are satisfied.

Algebras for fair CTL

- A fair CTL algebra is a tuple $(A, \mathbf{I}, \Diamond, \mathbf{EG}, \mathbf{EU})$ such that
  - A is a Boolean algebra,
  - $\mathbf{I}$ is a constant symbol,
  - $\Diamond$ is a unary operation,
  - **EG** and **EU** are binary operations,
  - the universal axioms on the previous slide are satisfied.
- **Theorem.** A fair CTL formula φ is true in all trees iff φ evaluates to $\top$ under any assignment to a fair CTL algebra.

Algebras for fair CTL

- A fair CTL algebra is a tuple $(A, \mathbf{I}, \Diamond, \mathbf{EG}, \mathbf{EU})$ such that
  - A is a Boolean algebra,
  - $\mathbf{I}$ is a constant symbol,
  - $\Diamond$ is a unary operation,
  - **EG** and **EU** are binary operations,
  - the universal axioms on the previous slide are satisfied.
- **Theorem.** A fair CTL formula φ is true in all trees iff φ evaluates to $\top$ under any assignment to a fair CTL algebra.
  - The proof uses a non-trivial tableau construction, and mimicks Walukiewicz' proof of completeness of the modal μ calculus.

Algebras for fair CTL

- A fair CTL algebra is a tuple $(A, \mathbf{I}, \Diamond, \mathbf{EG}, \mathbf{EU})$ such that
  - A is a Boolean algebra,
  - $\mathbf{I}$ is a constant symbol,
  - $\Diamond$ is a unary operation,
  - **EG** and **EU** are binary operations,
  - the universal axioms on the previous slide are satisfied.
- **Theorem.** A fair CTL formula φ is true in all trees iff φ evaluates to $\top$ under any assignment to a fair CTL algebra.
  - The proof uses a non-trivial tableau construction, and mimicks Walukiewicz' proof of completeness of the modal μ calculus.
  - There are interesting open questions here.

Algebras for fair CTL

- A fair CTL algebra is a tuple $(A, \mathbf{I}, \Diamond, \mathbf{EG}, \mathbf{EU})$ such that
  - A is a Boolean algebra,
  - $\mathbf{I}$ is a constant symbol,
  - $\Diamond$ is a unary operation,
  - **EG** and **EU** are binary operations,
  - the universal axioms on the previous slide are satisfied.
- **Theorem.** A fair CTL formula φ is true in all trees iff φ evaluates to $\top$ under any assignment to a fair CTL algebra.
  - The proof uses a non-trivial tableau construction, and mimicks Walukiewicz' proof of completeness of the modal μ calculus.
  - There are interesting open questions here.
- Given this result, the tree automata from the Rabin-Janin-Walukiewicz theorem can be used to prove that the theory of fair CTL algebras has a model companion.

Overview

So what about Heyting algebras?

- The theory of Heyting algebras also has a model companion, but for (apparently) very different reasons.

So what about Heyting algebras?

- The theory of Heyting algebras also has a model companion, but for (apparently) very different reasons.
- Pitts' Uniform Interpolation Theorem (1992) shows that Intuitionistic Propositional Logic (IPL) can interpret monadic second order quantifiers.

So what about Heyting algebras?

- The theory of Heyting algebras also has a model companion, but for (apparently) very different reasons.
- Pitts' Uniform Interpolation Theorem (1992) shows that Intuitionistic Propositional Logic (IPL) can interpret monadic second order quantifiers.
- More precisely, for every Heyting term $\varphi(\overline{x}, y)$, there exist Heyting terms $\varphi_y(\overline{x})$ and $\varphi^y(\overline{x})$, effectively computable from φ,

So what about Heyting algebras?

- The theory of Heyting algebras also has a model companion, but for (apparently) very different reasons.
- Pitts' Uniform Interpolation Theorem (1992) shows that Intuitionistic Propositional Logic (IPL) can interpret monadic second order quantifiers.
- More precisely, for every Heyting term $\varphi(\overline{x}, y)$, there exist Heyting terms $\varphi_y(\overline{x})$ and $\varphi^y(\overline{x})$, effectively computable from φ, such that, for any ψ and θ not containing q,

$$HA \models \varphi \leq \psi \iff HA \models \varphi^y \leq \psi,$$

$$HA \models \theta \leq \varphi \iff HA \models \theta \leq \varphi_y.$$

- For a different, topological proof of Pitts' theorem, see my paper with Reggio, Top. Appl. 2018.

The model companion of Heyting algebras

- Ghilardi and Zawadowski, 1995, use Pitts' theorem to give an effective (infinite) first order axiomatization of the class of existentially closed Heyting algebras.

The model companion of Heyting algebras

- Ghilardi and Zawadowski, 1995, use Pitts' theorem to give an effective (infinite) first order axiomatization of the class of existentially closed Heyting algebras.

- For a Heyting algebra A, the solvability of a quantifier-free formula in an extension of A can be expressed using Pitts' operators $(-)^y$ and $(-)_y$.

The model companion of Heyting algebras

- Ghilardi and Zawadowski, 1995, use Pitts' theorem to give an effective (infinite) first order axiomatization of the class of existentially closed Heyting algebras.
- For a Heyting algebra A, the solvability of a quantifier-free formula in an extension of A can be expressed using Pitts' operators $(-)^y$ and $(-)_y$.
- Pitts' operators thus roughly play the role for Heyting algebras that automata played in our results on LTL and CTL.

The model companion of Heyting algebras

- Ghilardi and Zawadowski, 1995, use Pitts' theorem to give an effective (infinite) first order axiomatization of the class of existentially closed Heyting algebras.
- For a Heyting algebra A, the solvability of a quantifier-free formula in an extension of A can be expressed using Pitts' operators $(-)^y$ and $(-)_y$.
- Pitts' operators thus roughly play the role for Heyting algebras that automata played in our results on LTL and CTL.
- Can we say more about this analogy?

Summary of Part II: Model companions

- Model companions are a logical way to think about existentially closed structures; the canonical example is algebraically closed fields.
- Logical algebras, in particular those for linear temporal logic and computation tree logic, admit model companions.
- Automata are crucially used in the proofs, to *eliminate alternations of quantifiers*.
- Heyting algebras have a model companion too, albeit for an (apparently) different reason.

Summary

- Part I: How algebraic methods can play a role in logic

Summary

- Part I: How algebraic methods can play a role in logic
- Part II: How model theory interacts with automata theory

Summary

- Part I: How algebraic methods can play a role in logic
- Part II: How model theory interacts with automata theory
- Thank you!