Lecture 15
Automatic Structures

Introduction to Logic for Computer Science

Prof Hongseok Yang
KAIST

These slides are minor variants of those made by Prof Worrell and Prof Haase for their logic course at Oxford.

Overview

Today:

- Study another approach for proving the decidability of a theory.
- Structures whose universe and relations are regular languages.
- Automata-based decision procedures for the theories of those structures.

Relational structures

Definition

A σ-structure $\mathcal{A}$ is **relational** if σ only consists of relation symbols.

Relational structures

Definition

A σ-structure $\mathcal{A}$ is **relational** if σ only consists of relation symbols.

Will only consider relational structures in this lecture.

OK, because every structure $\mathcal{A}$ has a relational variant.

Relational structures

Definition

A σ-structure $\mathcal{A}$ is **relational** if σ only consists of relation symbols.

Will only consider relational structures in this lecture.

OK, because every structure $\mathcal{A}$ has a relational variant.

To get the variant, we replace every $f_{\mathcal{A}}\colon U_{\mathcal{A}}^{k} \to U_{\mathcal{A}}$ with relation

$$F_{\mathcal{A}} = \{(a_1, \ldots, a_k, b) \in U_{\mathcal{A}}^{k+1} \ : \ f_{\mathcal{A}}(a_1, \ldots, a_k) = b\},$$

and each constant $c_{\mathcal{A}}$ by unary relation

$$C_{\mathcal{A}} = \{a \in U_{\mathcal{A}} \ : \ a = c_{\mathcal{A}}\}.$$

Relational structures

Definition

A σ-structure $\mathcal{A}$ is **relational** if σ only consists of relation symbols.

Will only consider relational structures in this lecture.

OK, because every structure $\mathcal{A}$ has a relational variant.

To get the variant, we replace every $f_{\mathcal{A}} \colon U_{\mathcal{A}}^k \to U_{\mathcal{A}}$ with relation

$$F_{\mathcal{A}} = \{(a_1, \ldots, a_k, b) \in U_{\mathcal{A}}^{k+1} \ : \ f_{\mathcal{A}}(a_1, \ldots, a_k) = b\},$$

and each constant $c_{\mathcal{A}}$ by unary relation

$$C_{\mathcal{A}} = \{a \in U_{\mathcal{A}} \ : \ a = c_{\mathcal{A}}\}.$$

Ex: Construct a relational variant of the structure $(\mathbb{N}, 0, 1, +, \times)$.

Automatic structure infomally

Key concept of this lecture.

Informally refers to a structure $\mathcal{A} = (U_{\mathcal{A}}, R_1, \ldots, R_m)$ where $U_{\mathcal{A}}$ and the R_i can be represented by finite automata.

Theorem (Khoussainov & Nerode, Simpler Version)

$\mathrm{Th}(\mathcal{A})$ *is decidable for every automatic structure* $\mathcal{A}$.

Automatic structure infomally

Key concept of this lecture.

Informally refers to a structure $\mathcal{A} = (U_{\mathcal{A}}, R_1, \ldots, R_m)$ where $U_{\mathcal{A}}$ and the R_i can be represented by finite automata.

Theorem (Khoussainov & Nerode, Simpler Version)

$\mathrm{Th}(\mathcal{A})$ *is decidable for every automatic structure* $\mathcal{A}$.

Some terminology from formal language theory

An **alphabet** Σ is a finite set.

A **word** w over Σ is a finite sequence of elements in Σ.

A **language** L over Σ is a set of words. That is, $L \subseteq \Sigma^*$.

A language L over Σ is **regular** if it can be recognised by a finite state automaton.

Word convolutions

Given an alphabet Σ, we want to represent relations R on Σ^*, by words over another alphabet. **Word convolutions** let us do it.

Word convolutions

Given an alphabet Σ, we want to represent relations R on Σ^*, by words over another alphabet. **Word convolutions** let us do it.

Let $\# \notin \Sigma$. Define $\Sigma_\# := \Sigma \cup \{\#\}$.

Word convolutions

Given an alphabet Σ, we want to represent relations R on Σ^*, by words over another alphabet. **Word convolutions** let us do it.

Let $\# \notin \Sigma$. Define $\Sigma_\# := \Sigma \cup \{\#\}$.

For words $w_1, w_2, \ldots, w_n \in \Sigma^*$,

- let $w_i = a_{(i,1)} a_{(i,2)} \cdots a_{(i,\ell_i)}$, so that $|w_i| = \ell_i$;
- let $\ell = \max\{\ell_1, \ldots, \ell_n\}$;
- set $a_{(i,j)} := \#$ for all $\ell_i < j \leq \ell$ and $1 \leq i \leq n$.

Word convolutions

Given an alphabet Σ, we want to represent relations R on Σ^*, by words over another alphabet. **Word convolutions** let us do it.

Let $\# \notin \Sigma$. Define $\Sigma_\# := \Sigma \cup \{\#\}$.

For words $w_1, w_2, \ldots, w_n \in \Sigma^*$,

- let $w_i = a_{(i,1)} a_{(i,2)} \cdots a_{(i,\ell_i)}$, so that $|w_i| = \ell_i$;
- let $\ell = \max\{\ell_1, \ldots, \ell_n\}$;
- set $a_{(i,j)} := \#$ for all $\ell_i < j \leq \ell$ and $1 \leq i \leq n$.

The **convolution** of $w_1, \ldots, w_n$ is

$$
\begin{aligned}
w_1 \otimes w_2 \otimes \cdots \otimes w_n \in (\Sigma_\#^n)^* \\
:= (a_{(1,1)}, \ldots, a_{(n,1)})(a_{(1,2)}, \ldots, a_{(n,2)}) \cdots (a_{(1,\ell)}, \ldots, a_{(n,\ell)}).
\end{aligned}
$$

Word convolutions

Given an alphabet Σ, we want to represent relations R on Σ^*, by words over another alphabet. **Word convolutions** let us do it.

Let $\# \notin \Sigma$. Define $\Sigma_\# := \Sigma \cup \{\#\}$.

For words $w_1, w_2, \ldots, w_n \in \Sigma^*$,

- let $w_i = a_{(i,1)} a_{(i,2)} \cdots a_{(i,\ell_i)}$, so that $|w_i| = \ell_i$;
- let $\ell = \max\{\ell_1, \ldots, \ell_n\}$;
- set $a_{(i,j)} := \#$ for all $\ell_i < j \leq \ell$ and $1 \leq i \leq n$.

The **convolution** of $w_1, \ldots, w_n$ is

$$w_1 \otimes w_2 \otimes \cdots \otimes w_n \in (\Sigma_\#^n)^*$$
$$:= (a_{(1,1)}, \ldots, a_{(n,1)})(a_{(1,2)}, \ldots, a_{(n,2)}) \cdots (a_{(1,\ell)}, \ldots, a_{(n,\ell)}).$$

Ex: Compute $abba \otimes abaabba \otimes ac$.

Automatic relations

Definition

A relation $R \subseteq (\Sigma^*)^n$ is **automatic** if the following language over $\Sigma_\#^n$

$$L_R := \{w_1 \otimes w_2 \otimes \cdots \otimes w_n : (w_1, \ldots, w_n) \in R\}$$

is regular (i.e. it can be recognised by a finite automaton over $\Sigma_\#^n$).

Automatic relations

Definition

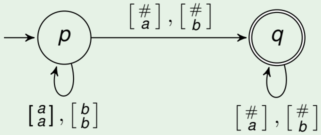
A relation $R \subseteq (\Sigma^*)^n$ is **automatic** if the following language over $\Sigma_\#^n$

$$L_R := \{w_1 \otimes w_2 \otimes \cdots \otimes w_n : (w_1, \ldots, w_n) \in R\}$$

is regular (i.e. it can be recognised by a finite automaton over $\Sigma_\#^n$).

Example

$R = \{(u, v) \in (\Sigma^*)^2 : u \text{ is a prefix of } v\}$ with $\Sigma = \{a, b\}$ is automatic:



Automatic structures

Definition

A relational structure $\mathcal{A} = (U_\mathcal{A}, R_1, \ldots, R_m)$ is **automatic** if there are a finite alphabet Σ and regular languages $L, L_1, \ldots, L_m$ such that

- $L = U_\mathcal{A}$;
- $L_i = L_{R_i}$ for all $1 \leq i \leq m$.

Automatic structures

Definition

A relational structure $\mathcal{A} = (U_\mathcal{A}, R_1, \ldots, R_m)$ is **automatic** if there are a finite alphabet Σ and regular languages $L, L_1, \ldots, L_m$ such that

- $L = U_\mathcal{A}$;
- $L_i = L_{R_i}$ for all $1 \leq i \leq m$.

A structure $\mathcal{A}$ has an **automatic presentation** if $\mathcal{A}$ is isomorphic to an automatic structure.

Note: Structures $\mathcal{A} = (U_\mathcal{A}, R_1, \ldots, R_m)$ and $\mathcal{B} = (U_\mathcal{B}, S_1, \ldots, S_m)$ are **isomorphic** if there is a bijection $f : U_\mathcal{A} \to U_\mathcal{B}$ such that

$$(a_1, \ldots, a_k) \in R_i \iff (f(a_1), \ldots, f(a_k)) \in S_i$$

for all $1 \leq i \leq m$ and all $a_1, \ldots, a_k \in U_\mathcal{A}$.

Theorem (Khoussainov & Nerode)

$\mathrm{Th}(\mathcal{A})$ *is decidable for every structure* $\mathcal{A}$ *with an automatic presentation.*

Reminder of Presburger arithmetic

$\mathrm{Th}(\mathbb{N}, 0, 1, +, =)$ is **Presburger arithmetic.**[1]

Theorem

Presburger arithmetic $\mathrm{Th}(\mathbb{N}, 0, 1, +, =)$ *is decidable.*

Last time we proved the theorem using quantifier elimination.

[1] In the last lecture, we included $<$ instead of $=$. But this does not make any difference.

Reminder of Presburger arithmetic

$\mathrm{Th}(\mathbb{N}, 0, 1, +, =)$ is **Presburger arithmetic.**[1]

Theorem

Presburger arithmetic $\mathrm{Th}(\mathbb{N}, 0, 1, +, =)$ *is decidable.*

Last time we proved the theorem using quantifier elimination.

We will give another proof here that uses Khoussainov & Nerode.

Theorem

The structure $(\mathbb{N}, 0, 1, +, =)$ *is automatic.*

[1] In the last lecture, we included $<$ instead of $=$. But this does not make any difference.

Presburger arithmetic has an automatic presentation

We will build an automatic structure isomorphic to $(\mathbb{N}, +)$.

Presburger arithmetic has an automatic presentation

We will build an automatic structure isomorphic to $(\mathbb{N}, +)$.

Set $N := (\{0,1\}^*1) \cup \{0\} \subseteq \{0,1\}^*$.

For $w = b_0 b_1 \cdots b_m \in N$, define $\mathrm{val}\colon N \to \mathbb{N}$ by

$$\mathrm{val}(w) := \sum_{i=0}^{m} 2^i \cdot b_i.$$

Set $A := \big\{(a,b,c) \in N^3 : \mathrm{val}(a) + \mathrm{val}(b) = \mathrm{val}(c)\big\} \subseteq N^3$.

Presburger arithmetic has an automatic presentation

We will build an automatic structure isomorphic to $(\mathbb{N}, +)$.

Set $N := (\{0,1\}^*1) \cup \{0\} \subseteq \{0,1\}^*$.

For $w = b_0 b_1 \cdots b_m \in N$, define $\mathrm{val}\colon N \to \mathbb{N}$ by

$$\mathrm{val}(w) := \sum_{i=0}^{m} 2^i \cdot b_i.$$

Set $A := \big\{(a,b,c) \in N^3 : \mathrm{val}(a) + \mathrm{val}(b) = \mathrm{val}(c)\big\} \subseteq N^3$.

Then, $(\mathbb{N}, +)$ is isomorphic to (N, A) by mapping $n \in \mathbb{N}$ to its unique minimal binary expansion $\mathrm{val}^{-1}(n)$.

Proposition

The structure (N, A) is automatic.

Presburger arithmetic has an automatic presentation

We will build an automatic structure isomorphic to $(\mathbb{N}, +)$.

Set $N := (\{0,1\}^* 1) \cup \{0\} \subseteq \{0,1\}^*$.

For $w = b_0 b_1 \cdots b_m \in N$, define $\mathrm{val}\colon N \to \mathbb{N}$ by

$$\mathrm{val}(w) := \sum_{i=0}^{m} 2^i \cdot b_i.$$

Set $A := \{(a,b,c) \in N^3 : \mathrm{val}(a) + \mathrm{val}(b) = \mathrm{val}(c)\} \subseteq N^3$.

Then, $(\mathbb{N}, +)$ is isomorphic to (N, A) by mapping $n \in \mathbb{N}$ to its unique minimal binary expansion $\mathrm{val}^{-1}(n)$.

Proposition

The structure (N, A) is automatic.

Ex: Prove the proposition.

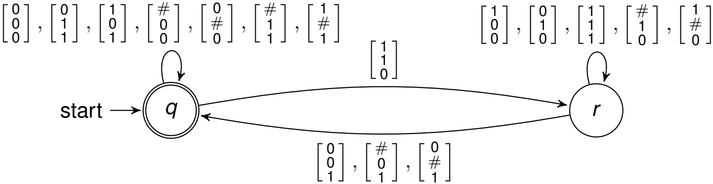
(N, A) is automatic

N is obviously regular.

(N, A) **is automatic**

N is obviously regular.

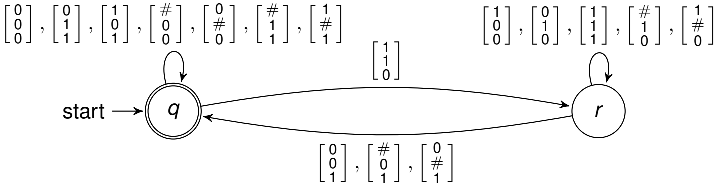
L_A is contained in the language of the following DFA:



(N, A) is automatic

N is obviously regular.

L_A is contained in the language of the following DFA:



Now intersect it with $\{a \otimes b \otimes c : a, b, c \in N\}$ to obtain a finite automaton for L_A.

Unbounded dense linear orders have automatic presentations

Theorem

Any countable structure $\mathcal{A} = (Q, <)$ that is a model of the unbounded dense linear order axioms has an automatic presentation.

Unbounded dense linear orders have automatic presentations

Theorem

Any countable structure $\mathcal{A} = (Q, <)$ that is a model of the unbounded dense linear order axioms has an automatic presentation.

Proof strategy:

- Show that any two countable unbounded dense linear orders are isomorphic.
- Show the statement for a suitable countable unbounded dense linear order.

Unbounded dense linear orders have automatic presentations

Theorem

Any countable structure $\mathcal{A} = (Q, <)$ that is a model of the unbounded dense linear order axioms has an automatic presentation.

Proof strategy:

- Show that any two countable unbounded dense linear orders are isomorphic.
- Show the statement for a suitable countable unbounded dense linear order.

Theorem (Cantor)

Any two countable unbounded dense linear orders are isomorphic.

We will assume the theorem, but I encourage you to try to prove it.

Unbounded dense linear orders have automatic presentations

Theorem

Any countable structure $\mathcal{A} = (Q, <)$ that is a model of the unbounded dense linear order axioms has an automatic presentation.

Proof strategy:

- Show that any two countable unbounded dense linear orders are isomorphic.
- Show the statement for a suitable countable unbounded dense linear order.

Theorem (Cantor)

Any two countable unbounded dense linear orders are isomorphic.

We will assume the theorem, but I encourage you to try to prove it.

An automatic unbounded dense linear order

Let $L = \{0,1\}^* \cdot 1$ and $<$ such that $x < y$ iff either

- $y = xu$ for some $u \in \{0,1\}^*$, or
- $x = z0u$ and $y = z1v$ for some $u, v, z \in \{0,1\}^*$.

Ex1: Prove that $(L, <)$ is automatic.

An automatic unbounded dense linear order

Let $L = \{0,1\}^* \cdot 1$ and $<$ such that $x < y$ iff either

- $y = xu$ for some $u \in \{0,1\}^*$, or
- $x = z0u$ and $y = z1v$ for some $u, v, z \in \{0,1\}^*$.

Ex1: Prove that $(L, <)$ is automatic.

Proposition

The structure $(L, <)$ is an automatic unbounded dense linear order (in short, UDLO).

Ex2: Prove the proposition. You need to show that $(L, <)$ is an UDLO.

$(L, <)$ is an UDLO.

No smallest element: for $u1 \in L$, have ??? $< u1$.

No largest element: for $u1 \in L$, have $u1 <$???.

Density: Let $x, y \in L$ with $x < y$. Should find z with $x < z < y$.

- Case $x = u1, y = u1v1$: Then, $x < z =$??? $< y$.
- Case $x = u0v1, y = u1w$: Then, $x < z =$??? $< y$.

$(L, <)$ is an UDLO.

No smallest element: for $u1 \in L$, have $u01 < u1$.

No largest element: for $u1 \in L$, have $u1 < u11$.

Density: Let $x, y \in L$ with $x < y$. Should find z with $x < z < y$.

- Case $x = u1, y = u1v1$: Then, $x < z = u10^{|v|+1}1 < y$.
- Case $x = u0v1, y = u1w$: Then, $x < z = u01^{|v|+2} < y$.

Structures with automatic presentations are decidable

Theorem (Khoussainov & Nerode)

$\mathrm{Th}(\mathcal{A})$ *is decidable for every structure* $\mathcal{A}$ *with an automatic presentation.*

Proposition

Let $\mathcal{A} = (L, R_1, \ldots, R_m)$ be an automatic σ-structure. Let F be a σ-formula with at most free variables $x_1, \ldots, x_n$. There is an effectively constructible regular language $L_F \subseteq (\Sigma_\#^*)^n$ such that

$$L_F = \left\{ w_1 \otimes \cdots \otimes w_n \ : \ \mathcal{A}_{[x_1 \mapsto w_1] \cdots [x_n \mapsto w_n]} \models F \right\}.$$

Ex1: Why does the proposition imply the theorem?

Ex2: Prove the proposition by induction.

Proof of the proposition: case $F = R_i(x_{i_1}, \ldots, x_{i_k})$ with $1 \le i_1, \ldots, i_k \le n$

Define homomorphism $h\colon (\Sigma_\#^n)^* \to (\Sigma_\#^k)^*$ such that for $a_1, \ldots, a_n \in \Sigma_\#$:

$$h(a_1, \ldots, a_n) = \begin{cases} \epsilon & \text{if } a_{i_1} = \cdots = a_{i_k} = \# \\ (a_{i_1}, \ldots, a_{i_k}) & \text{otherwise.} \end{cases}$$

By assumption, $L_{R_i} \subseteq (\Sigma_\#^k)^*$ is regular. Using the closure under inverse homomorphisms, obtain the regularity of

$$L_F = h^{-1}(L_{R_i}) \cap \{w_1 \otimes \cdots \otimes w_n \ : \ w_1, \ldots, w_n \in L\}.$$

Proof of the proposition: case $F = G \wedge H$, $F = G \vee H$, or $F = \neg G$

Induction hypothesis yields regular languages $L_G, L_H \subseteq (\Sigma_{\#}^{n})^*$.

The statement in the proposition follows from the closure of regular languages under intersection, union and complement.

Example: for $F = \neg G$, we have

$$L_F = \{w_1 \otimes \cdots \otimes w_n : w_1, \ldots, w_n \in L\} \setminus L_G.$$

Then, L_F is regular because of the closure of regular languages under complement.

Proof of the proposition: case $F = \exists x_{n+1}\, G$ with $x_1, \ldots, x_n, x_{n+1}$ free in G

Induction hypothesis yields regular languages L_G for G.

Define homomorphism $h\colon (\Sigma_{\#}^{n+1})^* \to (\Sigma_{\#}^{n})^*$ such that for $a_1, \ldots, a_n \in \Sigma_{\#}$,

$$h(a_1, \ldots, a_n, a_{n+1}) = \begin{cases} \epsilon & \text{if } a_1 = \cdots = a_n = \# \\ (a_1, \ldots, a_n) & \text{otherwise.} \end{cases}$$

Then, $L_F = h(L_G)$, which is regular because the homomorphic image of a regular language is regular.

Intractability

Theorem

There exists an automatic structure $\mathcal{A}$ with non-elementary complexity, i.e., no algorithm decides $F \in \mathrm{Th}(\mathcal{A})$ in time $2^{2^{\cdot^{\cdot^{\cdot^{2^n}}}}}$.

Proof idea.

This can be shown for the structure $\mathcal{A} = (\{0,1\}^*, S_1, S_2, \leq)$, where

- $S_0 = \{(w, w0) : w \in \{0,1\}^*\}$,
- $S_1 = \{(w, w1) : w \in \{0,1\}^*\}$,
- $\leq \ = \{(w, u) : w, u \in \{0,1\}^*\}$. $\square$

Proving Lagrange-style theorems automatically

Theorem (Lagrange, 1770)

Every natural number can be written as the sum of four integer squares.

Proving Lagrange-style theorems automatically

Theorem (Lagrange, 1770)

Every natural number can be written as the sum of four integer squares.

Call $n \in \mathbb{N}$ a **binary palindrome** if the string representing its binary presentation is a palindrome, e.g.,

$$27 = \mathrm{val}(11011).$$

Proving Lagrange-style theorems automatically

Theorem (Lagrange, 1770)

Every natural number can be written as the sum of four integer squares.

Call $n \in \mathbb{N}$ a **binary palindrome** if the string representing its binary presentation is a palindrome, e.g.,

$$27 = \mathrm{val}(11011).$$

Theorem (Rajasekaran, Shallit, Smith, 2017)

Every natural number can be written as the sum of four binary palindromes.

Proof idea.

Translate the statement into a suitably constructed nested-word automaton accepting all numbers that are the sum of four binary palindromes. Show that the automaton accepts all numbers. $\square$