Finite Automata

Regular Languages

Monadic Second Order Logic

Disjoint unions of structures, I

There are several ways of looking at disjoint unions of structures.

The most general might be:

$\mathcal{A}_0$ a τ_0-structure, $\mathcal{A}_1$ a τ_1-structure,
$\sigma = \tau_0 \sqcup \tau_1 \sqcup \{P_0, P_1\}$

$\mathcal{B} = \mathcal{A}_0 \sqcup \mathcal{A}_1$ is the σ-structure with

$B = A_0 \sqcup A_1$, $P_i(\mathcal{B} = A_i$ and
for $R \in \tau_i$, $R(\mathcal{B}) = R(\mathcal{A}_i)$

Remark: For $\tau_0 = \tau_1 = \tau$ one puts often
$R(\mathcal{B}) = R(\mathcal{A}_0) \sqcup R(\mathcal{A}_1)$
Sometimes the predicates P_1 are ommitted.
Only with the definition above are the parts $\mathcal{A}_i$ definable from the disjoint union.

Disjoint unions of structures, II

Theorem:(Feferman, Vaught, Ehrenfeucht)

If $\mathcal{A}_0 \sim_{q,v}^{MSOL} \mathcal{B}_0$ and $\mathcal{A}_1 \sim_{q,v}^{MSOL} \mathcal{B}_1$ so

$$\mathcal{A}_0 \sqcup \mathcal{A}_1 \sim_{q,v}^{MSOL} \mathcal{B}_0 \sqcup \mathcal{B}_1$$

If $h_{q,v}(\mathcal{A}_0) = h_{q,v}(\mathcal{B}_0)$ and $h_{q,v}(\mathcal{A}_1) = h_{q,v}(\mathcal{B}_1)$ so

$$h_{q,v}(\mathcal{A}_0 \sqcup \mathcal{A}_1) = h_{q,v}(\mathcal{B}_0 \sqcup \mathcal{B}_1)$$

In other words, the (q, v)-Hintikka sentence of a disjoint union is uniquely determined by the (q, v)-Hintikka sentence of its parts,

Concatenation, I

The concatenation of two words over an alphabet Σ is a special case of a disjoint union of *ordered structures*, where the second part follows the first.

We denote, for a word $w \in \Sigma^\star$ the corresponding structure by $\mathcal{A}_w$.

We denote by $\mathcal{A}_v \bullet \mathcal{A}_w$ the structure corresponding to the word vw.

Concatenation, II

Theorem:(Büchi, Ehrenfeucht)

If $\mathcal{A}_0 \sim_{q,v}^{MSOL} \mathcal{B}_0$ and $\mathcal{A}_1 \sim_{q,v}^{MSOL} \mathcal{B}_1$ so

$$\mathcal{A}_0 \bullet \mathcal{A}_1 \sim_{q,v}^{MSOL} \mathcal{B}_0 \bullet \mathcal{B}_1$$

If $h_{q,v}(\mathcal{A}_0) = h_{q,v}(\mathcal{B}_0)$ and $h_{q,v}(\mathcal{A}_1) = h_{q,v}(\mathcal{B}_1)$ so

$$h_{q,v}(\mathcal{A}_0 \bullet \mathcal{A}_1) = h_{q,v}(\mathcal{B}_0 \bullet \mathcal{B}_1) \qquad (+)$$

In other words, the (q, v)-Hintikka sentence of a concatenation is uniquely determined by the (q, v)-Hintikka sentence of its parts,

Finite Automata, I

We have **deterministic** and **non-deterministic** finite automata (Turing machines without work tape).

We **one-directional** and **two-directional** finite automata.

Let
$X \in \{(det, one), (n{-}det, one), (det, two), (n{-}det, two)\}$.
A language (set of words) L a $X - FA$, if it is accepted by some X finite automaton.

Theorem:(Rabin and Scott, 1959)

L is $X - FA$ iff L is $Y - FA$ for each $X, Y \in \{(det, one), (n{-}det, one), (det, two), (n{-}det, two)\}$.

The proof was given in the course Automata and Formal Languages

Finite Automata, II

We can also look at

- **multi-tape, k-tape** finite automata
  with one simultaneous head on the tapes.

- **multi-head, k-head** finite automata.

- k-**pebble** finite automata
  with pebbles (markers) on the tape.

Theorem:
A language L is k-tape $X - FA$ iff
L is 1-tape $X - FA$.

But there are **more** languages which are
2-head $X - FA$ than with one head.
The same with even one pebble.

Regular Languages, I

Let Σ be a finite alphabet.
λ denotes the empty word.
$\Sigma^\star$ is the set of all finite words (including λ).
Σ^+ is the set of all non-empty finite words, (excluding λ).

Regular Σ-expression are

- $\emptyset$, and a for each $a \in \Sigma$;

- if r, s are regular expressions, so are $(r \cup s), (rs)$ and r^+.

Regular Languages, II

For a regular expression r we define a language $Lang(r)$.

Assume $Lang(r) = R$ and $Lang(s) = S$.

- $Lang(\emptyset) = \emptyset$, $Lang(a) = \{a\}$ for $a \in \Sigma$.
- $Lang(r \cup s) = R \cup S$
- $Lang(rs) = \{uv : u \in R, v \in S\} = RS$
- We define $R^1 = R$ and $R^{n+1} = R^n R$, and $R^+ = \bigcup_{1 \leq n} R^n$.
- $Lang(r^+) = R^+$.

A language L is *regular* iff $L = Lang(r)$ for some Σ-regular expression r.

Regular Languages, III

Complementation:
For r we form the expression $\neg r$ with $Lang(\neg r) = \Sigma^{+} - Lang(r)$.

Theorem:
For every regular expression r $lang(\neg r)$ is regular.

A an expression is *regular plus-free* if it is defined inductively by

- $\emptyset$, $\{a\}$
- $(r \cup s), (rs), (\neg r)$

A regular language is *plus-free* if it is of the form $Lang(r)$ for some plus-free expression.

Finite Automata, III

Theorem:
(Kleene, 1953, Rabin and Scott 1959)

The following are equivalent for languages L:

- L is regular
- L is $(det, one) - FA$
- L is $(n - det, two) - FA$

and also for
$(det, two) - FA$ and $(n - det, one) - FA$.

The proof was given in the course
Automata and Formal Languages

Finite Automata, IV

Theorem:(Büchi-Trakhtenbrot)
A set of words L is regular iff the set of its structures K_L is definable in $MSOL$

Theorem:(McNaughton)
A set of words L is plus-free regular iff the set of its structures K_L is definable in FOL

Proof of Büchi's Theorem, I

Proof: If L is regular, it can be defined by a regular expression r.
We use induction.
For $\vee$, concatenation and complement, we use FOL operations. For $^+$ we quantify over sets of positions and relativize the formulas of the induction hypothesis.

Note that we did not use $(r^\star)$.
We avoid the empty word λ.

How could we include it?

Proof of Büchi's Theorem, II

Now assume that K_L is defined by $\phi \in Fm_{q,v}^{MSOL}(\tau)$.

We define the the automaton for L.

The states are $\mathcal{H}_{q,v}(\tau)$.

The transitions are given by $(+)$ of the previous theorem with the second word a singleton.

The accepting states are the (q, v)-Hintikka formulas the disjunction of which is equivalent to ϕ.

This works both for FOL and $MSOL$ with the according modifications.

Pumping Lemma, I

Theorem: Let A be a finite (deterministic, one-directional) finite automaton with n states and defining the language $L(A)$.

Let $w \in L(A)$ with length $\ell(w) \geq n$.
Then there exists words x, y, z such that

- $w = xyz$ and $y \neq \Lambda$ and
- for each $k \in \mathbb{N}$ $xy^k z \in L(A)$

A pumping lemma for **context free** languages was stated first in 1961 by Bar-Hillel, Perles, Shamir.

Pumping Lemma, II

We want to apply the Pumping Lemma to $MSOL$.

Theorem: Let ϕ be a $MSOL(\tau_{words(\Sigma)})$-sentence over words in Σ^+ with quantifier rank q and v variables and defining the language $L(\phi)$.

Let $\eta_{v,q,\Sigma} \leq \gamma_{v,q,\Sigma}$ be the number of Hintikka sentences in $Fm_{q,v}^{MSOL}(\tau(\Sigma))$.

Let $w \in L(\phi)$ with length $\ell(w) \geq \eta_{q,v,\Sigma}$. Then there exists words x, y, z such that

- $w = xyz$ and $y \neq \Lambda$ and
- for each $k \in \mathbb{N}$ $xy^k z \in L(\phi)$

Pumping Lemma, III
Examples

The following are not regular

- $\{a^i b^i : i \in \mathbb{N}\}$, $\{a^i b^i c^i : i \in \mathbb{N}\}$, $\{a^i b^j : i, j \in \mathbb{N}, i \leq j\}$,

- The set of prime numbers as binary words.
  This follows easily from a deep theorem on primes:

  Theorem: For every $n \in \mathbb{N}$ there are successive primes $p_{i(n)}, p_{i(n)+1}$ such that $p_{i(n)+1} - p_{i(n)} \geq n$.

  A direct proof is in

  Michael Harrison, Introduction to Formal Language Theory, Addison-Wesley 1978, chapter 2.2

A unary language L is regular iff
$X = \{i : a^i \in L\}$ is ultimately periodic.

$X \in \mathbb{N}$ (in increasing order) is ultimately periodic iff there is p such that for i large enough $x_{i+p} = x_i$.

Non-definability in $MSOL_1$, I

$MSOL_1$ is the $MSOL$ for structures which are graphs of the form $G = \langle V, E \rangle$ (E a binary relation).

The following are not $MSOL_1$-definable.

- HALF-CLIQUE: graphs with a clique of size at least $\frac{|V|}{2}$
- HAM: graphs which have a hamiltonian cycle.
- EULER: graphs which have an Eulerian circuit.

Non-definability in $MSOL_1$, II

Proof for HALF-CLIQUE:

Assume $\phi_{half-clique} \in MSOL_1$
defines HALF-CLIQUE.

For each word $w = a^i b^j, i, j \neq 0$ of length n
we define a graph G_w as follows:

$V = \{1, \ldots, n\}$
$E = \{(u, v) \subseteq V^2 : \psi(u, v) = P_b(u) \wedge P_b(v) \wedge u \neq v\}$

Clearly G_w in HALF-CLIQUE iff
$w = a^i b^j$ with $i \leq j$.

But then let Φ be the formula we obtain from substituting $E(x, y)$ in ϕ by $\psi(x, y)$.

$w \models \Phi$ iff $w = a^i b^j$ with $i \leq j$.

By Büchi's Theorem, this implies that
$\{a^i b^j : i \leq j\}$ is regular, a **contradiction**.

Non-definability in $MSOL_1$, III

Proof for HAM:

Assume $\phi_{ham} \in MSOL_1$ defines HAM.

For each word $w = a^i b^j, i, j \neq 0$ of length n we define a graph G_w as follows:

$V = \{1, \ldots, n\}$
$E = \{(u, v) \subseteq V^2 : \psi(u, v) = P_a(u) \wedge P_b(v)\}$

Clearly G_w in HAM iff
$w = a^i b^j$ with $i = j$.

But then let Φ be the formula we obtain from substituting $E(x, y)$ in ϕ by $\psi(x, y)$.

$w \models \Phi$ iff $w = a^i b^j$ with $i = j$.

By Büchi's Theorem, this implies that $\{a^i b^i : i \in \mathbb{N}\}$ is regular, a **contradiction**.

Non-definability in $MSOL_1$, IV

Proof for EULER:

A graph is eulerian iff it is connected and all vertices have even degree.
Hence, the complete graph K_n is eulerian iff $n = 2m + 1$.

For each word $w = a^i b^j, i, j \neq 0$ of length n we define a graph G_w as follows:

$V = \{1, \ldots, n\}$
$E = \{(u, v) \subseteq V^2 : \psi(u, v) = u \neq v\}$

Clearly G_w in EULER iff
$w = a^i b^j$ with $i + j = 2m + 1$.

Similarly as before, this implies that
$\{a^i b^j : i + j = 2m + 1\}$ is regular.
But it is regular.

THIS PROOF DOES NOT WORK !

Non-definability in $MSOL_1$, V

The proofs for HALF-CLIQUE and HAM actually show more:

Theorem:
HAM and HALF-CLIQUE are not $MSOL$-definable even on **ordered graphs**.

An ordered graph $G = \langle V, E, < \rangle$ is a graph with a linear order on the vertices.

But EULER is $MSOL$ definable on ordered graphs, because on linear orders there is a formula $\phi_{even}(X)$ which says that $\mid X \mid$ is even.

Note also that on unary words

$$\{a^i : i = 2m\}$$

is ultimately periodic and hence regular.

Non-definability in $MSOL_1$, V

Exercise:

To prove that
EULER is not $MSOL_1$-definable

Hint:
Use that sets of even cardinality are not $MSOL$-definable.

Translation schemes, I

In these proofs we used a technique which we will spell out in full generality:

- For a word $w \in L$ we **defined** a graph G_w
- **defined** by an $MSOL$-formula
  actually a FOL-formula ψ
- Then we assumed that the class of graphs K was definable by ϕ.
- Put $\Phi = subst_E(\phi, \psi(x, y))$
- Show that $w \in L$ iff $G_w \in K$
- Conclude that L is defined by Φ.

We shall develop a formalism for

Translation schemes

which will play a central rôle a
in the sequel of the course.