Tutorial solutions (Part-II)

Om Swostik Mishra, Anish Kulkarni

Contents

1 Tutorial-1

1:

Given $\phi = \forall x \exists y\, R(x,y) \;\wedge\; \exists y \forall x\, \neg R(x,y)$.

Take the model to be the set of naturals $\mathbb{N}$ with $<$ relation.

Then, $m \models \phi$ (Why?)

(**Hint**: $\mathbb{N}$ is an well-ordered set (i.e has a minimum) and isn't bounded above)

2:

$\varphi_B(x,y) = \exists z(P(z,x) \wedge P(z,y)) \wedge \neg F(x)$

$\varphi_A(x,y) = \exists z(P(z,y) \wedge \varphi_S(x,z))$ $\qquad (\varphi_S(x,y) = \exists z(P(z,x) \wedge P(z,y)) \wedge F(x)$, x is sister of y)

$\varphi_C(x,y) = \exists z(\varphi_A(z,x) \wedge P(z,y))$

$\varphi_O(x) = \forall z, y(P(z,y) \wedge P(z,x) \Rightarrow (x = y))$

The spousal relationship cannot be defined (Why?)

3:

$Zero(x) = \;+(x,x) = x$

$One(x) = \forall y(\times(x,y) = y)$

$Two(x) = \exists z((+(z,z) = x) \wedge One(z))$

$Even(x) = \exists z, y((\times(z,y) = x) \wedge Two(y))$

$Odd(x) = \neg Even(x)$

$Prime(x) = \neg One(x) \wedge (\neg \exists w, y((\times(w,y) = x) \wedge (\neg One(w) \wedge \neg One(y))))$

Goldbach conjecture in FO:

$\forall x(\neg One(x) \wedge \neg Two(x) \wedge Even(x) \Rightarrow \exists z, w(Prime(z) \wedge Prime(w) \wedge +(z,w) = x))$

4:

Encoding associativity of $+$: $\forall x, y, z(+(x, +(y,z)) = +(+(x,y), z))$

Encoding the right identity as 0: $\forall x(+(x,0) = x)$

Encoding right inverse: $\forall x \exists y(+(x,y) = 0)$

Encoding A(4): $\forall x, y, z(+(x,z) = +(y,z) \Rightarrow x = y)$

Here we have used the signature $\tau = (0, +)$.

5:

(i) Consider the set of integers $\mathbb{Z}$ with the induced relation $+_Z$ referring to the usual addition in $\mathbb{Z}$. The constant 0_Z refers to 0 in $\mathbb{Z}$. Observe that addition is associative and admits both left and right inverses. Also 0 is a identity for addition. We can conclude the τ-structure $\mathbb{Z}$ satisfies ψ.

(ii) Consider the set $\mathbb{N}_0$ of whole numbers and the corresponding induced relation being addition and the constant being 0 (in $\mathbb{N}_0$). This τ-structure doesn't satisfy ψ as φ_3 fails to be true (non-zero elements in $\mathbb{N}_0$ don't have inverses).

(iii) Consider the set of all $n \times n$ invertible matrices with complex values, $GL_n(\mathbb{C})$. Let the induced binary operation be matrix multiplication and let the constant 0 map to the identity $n \times n$ matrix.

It's clear that the τ-structure $GL_n(\mathbb{C})$ satisfies ψ, however, it doesn't satisfy $\forall x, y(+(x, y) = +(y, x))$ (Why?).

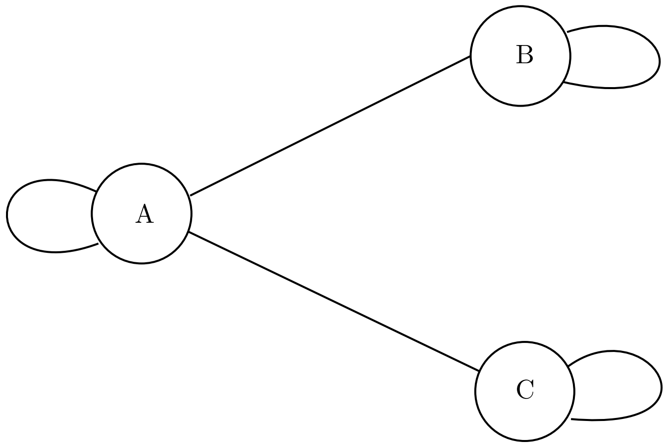
(iv) As before, consider the set $\mathbb{N}_0$ of whole numbers with the usual addition. This satisfies $\varphi_1 \wedge \varphi_2$ but doesn't satisfy φ_3.

Consider the set of non-negative reals $\mathbb{R}_{\geq 0}$, with the binary operation defined as $+(a, b) = |a - b|$ and the constant mapping to 0. Show that this structure satisfies $\varphi_2 \wedge \varphi_3$ but fails to satisfy φ_1.

Consider $\mathbb{Z}$ with the usual addition and the constant 0 mapping to 1 (in $\mathbb{Z}$). This satisfies $\varphi_1 \wedge \varphi_3$ but fails to satisfy φ_2.

We can conclude that ψ isn't equivalent to any of $\varphi_1 \wedge \varphi_2$, $\varphi_2 \wedge \varphi_3$ or $\varphi_1 \wedge \varphi_3$.

7:



Consider the undirected graph $\mathcal{G}$ above (with loops). This (with its natural edge relation) satisfies the second formula but not the first.

8:

$\exists^{\geq n}x(x = x) \wedge \neg\exists^{\geq n+1}x(x = x)$ is true for all models whose universe has exactly n elements.

Let $\varphi = \exists x_1, x_2 \dots x_n(\wedge_{i \neq j}(x_i \neq x_j))$.

$\varphi \equiv \exists^{\geq n}x(x = x)$ (Why?)

9:

Using counting quantifiers, we can write,

$\varphi = \exists^{\geq n}x(x = x) \wedge \neg\exists^{\geq m+1}x(x = x)$

φ evaluates to true only over models with atleast n and atmost m elements.

2 Tutorial-2

1:

(a) Observe that this is the same as the set of the words that start and end in the same letter! (Try and see how. A hint: An "ab" occurrence can be seen as "switching" from a to b while parsing the word from left to right. Similar for "ba".) We'll define the following functions w.r.t the word signature - they'll help us out throughout the tutorial:

$first(x) = \forall y.(x < y \vee x = y)$ $\qquad$ $last(x) = \forall y.(x > y \vee x = y)$

So, $\varphi_1 = \forall x.(x \neq x) \vee \exists x.(\ \exists y.(\ first(x) \wedge last(y) \wedge \neg(Q_a(x) \wedge Q_b(y)) \wedge \neg(Q_b(x) \wedge Q_a(y))\ )\ )$ is such that $L(\varphi_1) = L$.

(b) $\varphi_2 = \exists x.(\ Q_{\#}(x) \wedge \forall y.(\ (x < y \Rightarrow Q_b(y)) \wedge (y < x \Rightarrow Q_a(y))\ )\ )$ is such that $L(\varphi_2) = L$.

Note - for any letter in our alphabet we have its corresponding Q-function - hence, $Q_{\#}$ in our φ_2.

(c) Either there is no b or the only b's in the word come at the end.

Hence, $\varphi_3 = \forall x \forall y.(\ (S(x,\, y) \wedge Q_b(x)) \Rightarrow Q_b(y)\ )$ is such that $L(\varphi_3) = L$.

(d) $\varphi_4 = \exists x \exists y.(\ first(x) \wedge last(y) \wedge \forall z.(\ (S(x, z) \vee S(z, y)) \Rightarrow Q_0(z)\ )\ )$ is s.t. $L(\varphi_4) = L$.

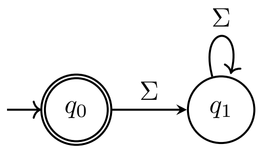
(e) Say we parse the word letters from left to right. In the beginning, the top and bottom entries of each letter may or may not be the same. However, if we want the top row to be larger than the bottom row, then the moment where the entries first differ will be $\binom{1}{0}$.

Hence, $\varphi_5 = \exists x.(\ Q_{\binom{1}{0}}(x) \wedge \forall y.(\ y < x \Rightarrow (Q_{\binom{0}{0}}(y) \vee Q_{\binom{1}{1}}(y))\ )\ )$ is s.t. $L(\varphi_5) = L$.

2:

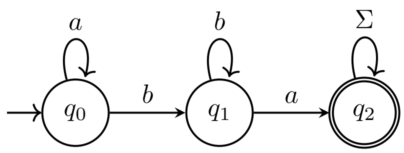
Note that for all φ, $L(\varphi)$ is by definition FO-definable, and hence regular. This is as *FO-definable languages* $\subseteq$ *regular languages*. (How? Given an FO formula, can we find an algorithm to construct a DFA for its language?). Also, as regular languages are closed under complementation, $\overline{L(\varphi)}$ will also be regular.

(1) $L(\varphi) = \epsilon$. The DFA is given by:



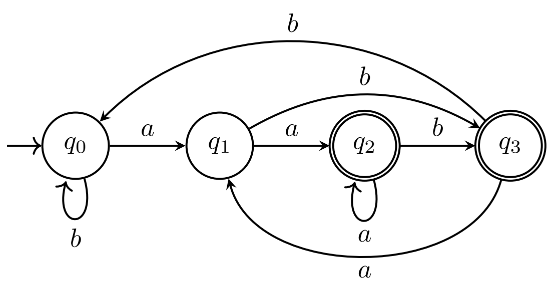
Naturally, only a structure (model) with an empty universe (domain) can satisfy φ here.

(2) $L(\varphi) = \Sigma^* b a^* a \Sigma^*$. The DFA is given by:



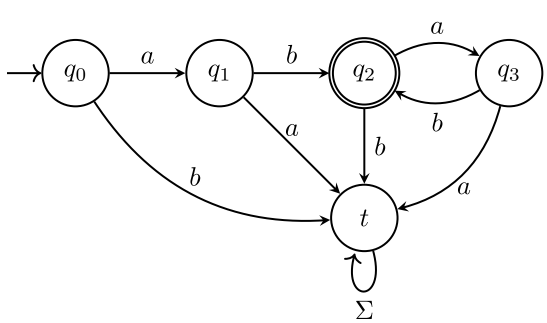
Observe that we need to encode *an* occurrence of ba^*a. With the first 2 states, we are basically encoding the $first$ such occurrence.

(3) $L(\varphi) = \Sigma^* a \Sigma$. The DFA is given by:



Basically, the second-last letter (has to exist and) has to be a. Despite being a pretty simple condition, the DFA ends up rather convoluted due to having to satisfy conditions of determinism.

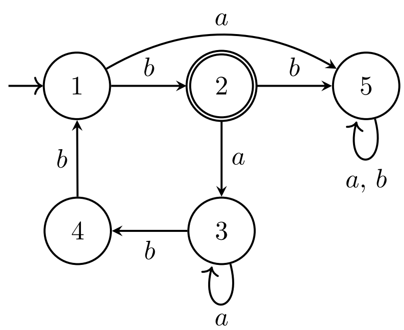
(4) $L(\varphi) = ab(ab)^*$. The DFA is given by:



We want words starting with a, ending with b, and having an alternating a-b pattern. We don't want the empty word though; hence, we add q_0 and q_1.

3:

The automaton given is:



5 is clearly a trap state. Let's ignore the paths leading to it. The first accepted word is "b", and to get other words we need to go through the $2 \to 3 \to 4 \to 1 \to 2$ cycle once ($(abbb)^*$).

However, we can go through the self-loop on 3 an arbitrary amount (a^*).

So, $L = b(aa^*bbb)^*$.

Define the following:

$\forall x, y, z \in Vars,\ \varphi_{bbb}(x, y, z) = S(x, y) \wedge S(y, z) \wedge Q_b(x) \wedge Q_b(y) \wedge Q_b(z)$

$\varphi_1 = \exists x.(first(x) \wedge Q_b(x))$ (since the word starts with a "b").

$\varphi_2 = (\exists w.(first(w) \wedge \neg\, last(w))) \Rightarrow \exists x \exists y \exists z.(last(z) \wedge \varphi_{bbb}(x,\ y,\ z))$ (since if the word size is > 1, the word has to end in a "bbb").

$\varphi_3 = \forall x \forall y \forall z.(\ \varphi_{bbb}(x,\ y,\ z) \Rightarrow \exists w.(S(w, x) \wedge Q_a(w))\ )$ (Immediately before every occurrence of "bbb", there is a non-empty series of "a"s).

$\varphi_4 = \forall w.(\ Q_a(w) \Rightarrow \forall x.(\ (Q_b(x) \wedge x < w) \Rightarrow$

$(\ first(x) \vee \exists y \exists z.(\varphi_{bbb}(x, y, z) \vee \varphi_{bbb}(y, x, z) \vee \varphi_{bbb}(y, z, x)\ )\ )\ )$ (Before every "a", every occurrence of "b" before it is either the first letter or part of a "bbb").

With all these, we claim our L is exactly the language of $\varphi = \varphi_1 \wedge \varphi_2 \wedge \varphi_3 \wedge \varphi_4$! Try and figure out how - each φ_i was coded such that together they provide us with all the necessary and sufficient properties to make L; however, saying that is rather vague.

Playing around with each φ_i might provide you with a better understanding: just for an example, try and see how $\varphi_4' = \forall w.(\ Q_a(w) \Rightarrow \exists x.(\ Q_b(x) \wedge x < w \wedge (\ first(x) \vee \exists y \exists z.(\varphi_{bbb}(x,y,z) \vee \varphi_{bbb}(y,x,z) \vee \varphi_{bbb}(y,z,x))\ )\ )\ )$ won't work, even though it seems to be arguing something similar. Or, try to see how things change if, in our sub-formulae, we want to encode that immediately *after* (not before) every "bbb" not at the end, there are a non-empty number of 'a's (and vice versa).

3 Tutorial-3

1-4:

Refer to the solutions sent on the group, for now.

5:

Given n, consider the set of all words in $\{0, 1\}$ with length $< n$. Consider any DFA $A = (Q, \{0, 1\}, \delta, q_0, F)$ accepting L_n. Suppose there are distinct words v and w that end in the same state $q \in Q$.

Let us first assume $v, w \neq \epsilon$. Suppose the leftmost position at which v and w differ is the kth position. We can assume w/o loss of generality, that v has 0 at that position, and w has 1. Let x be any word of length k. Due to the determinism of A, x has exactly one run in the automaton starting at q, which ends at $q' \in Q$, say. Then, the run of vx and wx in A must end in q'.

vx has nth bit from the right $= 0 \implies vx \notin L_n$. Thus, $q' \notin F$.

wx has nth bit from the right $= 1 \implies vx \in L_n$. Thus, $q' \in F$.

Contradiction!

As a result, every word with length $n - 1$ has its run ending in a different state in A. There are 2^{n-1} such words, and so at least $2^n - 1$ states in A.