ASSIGNMENTS

Problem set 6

AI504 — Knowledge Representation

Simon Holm · Johannes Rothe · Shuagib Ibrahim · Anne Sofie Høj · Daniel Nissen

May, 2026

SDU
DEPARTMENT OF MATHEMATICS
AND COMPUTER SCIENCE

Contents

Problem 1

We are going to describe a translation from the syntax of $\mathcal{A}(\mathcal{RC})$ to the syntax of propositional logic. First of all, if (P, R) is a signature of $\mathcal{A}(\mathcal{RC})$ then just map it to the propositional logic signature $P \cup R$. (In other words, either a noun or a verb in my source signature can be used as a sentence letter in my target signature.)

Now map terms t of $\mathcal{A}(\mathcal{RC})$ to formulas t^* of propositional logic as follows. Each noun p becomes the propositional formula p. Then "r all (...)" becomes "(...) $\to r$," so that, e.g., the term see `love` all `dogs` becomes (`dogs` $\to$ `love`) $\to$ `see`. In the latter formula `see`, `love`, `dogs` are just boolean-valued sentence letters; we have forgotten the noun/verb distinction.

Finally, the translation of the sentence "t all s" is $t^\star \to s^\star$, where $t^\star$ and $s^\star$ is the mapping on terms defined above. (Call this map $\phi \mapsto \phi^\star$, reusing the symbol $*$.) So, for example

$$(\text{all } (\texttt{see} \text{ all } \texttt{love} \text{ all } \texttt{dogs}) \ (\texttt{love} \text{ all } \texttt{birds}))^\star$$

(a) Suppose that Γ is a set of sentences and ϕ is a sentence in $\mathcal{A}(\mathcal{RC})$, and suppose that $\Gamma \vDash \phi$. Prove that $\Gamma^\star \vDash \phi^\star$, as a formulas of propositional logic.

Solution:
This is proved by induction in an $\mathcal{A}(\text{RC})'$.

Proof by induction

Goal: For all T, Γ, ϕ where T is a proof tree s.t $\Gamma \vdash \phi$, then there exists a proof tree $T^\star$ s.t. $\Gamma^\star \vDash \phi^\star$.

Base case

If T is a single sentence (the whole tree is a leaf), then either ϕ is an axiom such that $\phi = \phi^\star$ or $\phi \in \Gamma$ which means that $\phi^\star \in \Gamma^\star$

Then since all leaves are $\in \Gamma^\star$, $\Gamma^\star \vdash \phi^\star$ and from soundness of propositional logic, $\Gamma^\star \vDash \phi^\star$

Inductive hypothesis

For all proofs T where $\Gamma \vdash \phi$, there exists another proof $T^\star$ from $\Gamma^\star$ such that $\Gamma^\star \vdash \phi^\star$

Inductive step

Suppose T is not a leaf, then it is either an instance of ANTI or BARBARA.

Case 1 ANTI:

If T is not a single leaf but an instance of ANTI, then it is a sentence proved by one subtree $\mathcal{T}$. This can be written up as follows:

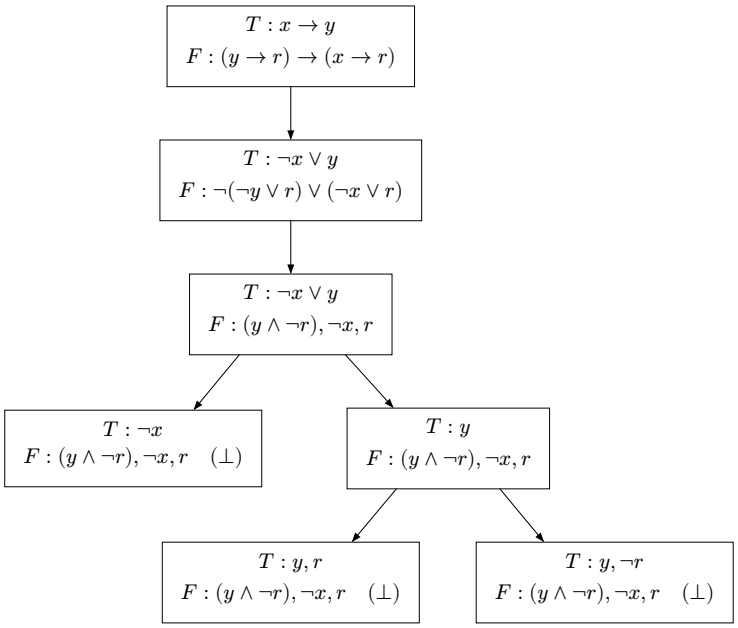
$$\frac{\overbrace{\text{all } x \ y}^{\mathcal{T}}}{\underbrace{\text{all } (r \text{ all } y) \ (r \text{ all } x)}_{\phi}} \text{ANTI}$$

Applying the conversion rules for $\mathcal{A}(\text{RC})$ to propositional logic yields the converted subtree $\mathcal{T}^\star$ and sentence $\phi^\star$:

$$\frac{\overbrace{x \to y}^{\mathcal{T}^\star}}{\underbrace{(y \to r) \to (x \to r)}_{\phi^\star}}\,\text{ANTI}$$

Then to show that $x \to y \Longrightarrow (y \to r) \to (x \to r)$.

This can be done by assuming that $(y \to r) \to (x \to r)$ is false and exploring all nodes in an analytic tableau-style table to ensure that all ends are contradictions:



Since all leaves lead to a contradiction denoted as $(\bot)$. This proves $\phi^\star$

By the basis of the inductive hypothesis applied to $\mathcal{T}$, it follows that $\Gamma^\star \vdash \mathcal{T}^\star$, and by using the ANTI rule, it follows that $\Gamma^\star \vdash \phi^\star$.

Case 2: BARBARA

If T is not a single leaf, but an instance of BARBARA, then it is a sentence proved by 2 subtrees $\mathcal{T}_0, \mathcal{T}_1$ which can be written as follows:

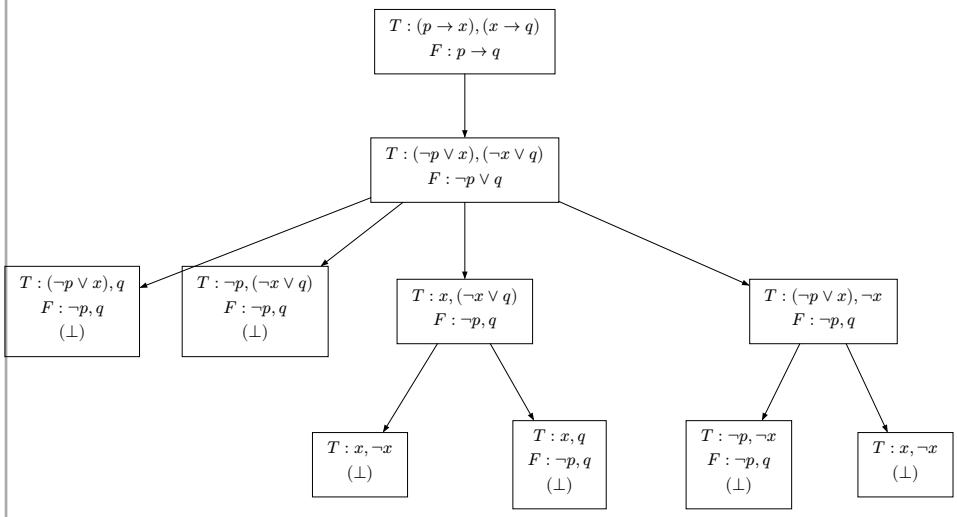
$$\frac{\overbrace{(\text{all } p\ x)}^{\mathcal{T}_0} \qquad \overbrace{(\text{all } x\ q)}^{\mathcal{T}_1}}{\underbrace{\text{all } p\ q}_{\phi}}\,\text{BARBARA}$$

Where $T^\star$ would be:

Applying the conversion rules for $\mathcal{A}$(RC) to propositional logic yields the converted subtrees $\mathcal{T}_0^\star, \mathcal{T}_1^\star$ and sentence $\phi^\star$:

$$\frac{\overbrace{(p \to x)}^{\mathcal{T}_0^\star} \qquad \overbrace{(x \to q)}^{\mathcal{T}_1^\star}}{\underbrace{p \to q}_{\phi^\star}}\,\text{BARBARA}$$

23 Using the same tableu method as before, it can be shown that the two subtrees prove $\phi^\star$:



24

25 Therefore, all proofs T from $\mathcal{A}(RC)'$ where $\Gamma \vdash \phi$, proves the existence of $\Gamma^\star \vdash \phi^\star$, and from propositional logic soundness, $\Gamma^\star \vDash \phi^\star$

(b) Show that the converse is not true. In other words, come up with a concrete sentences Γ, ϕ of $\mathcal{A}(\mathcal{RC})$ such that $\Gamma^\star \vDash \phi^\star$ in propositional logic but $\Gamma \nvDash \phi$ in $\mathcal{A}(\mathcal{RC})$.

Take the sentence $\phi = \text{all } x \ (z \text{ all } y)$ and $\Gamma = \{\text{all } y \ (z \text{ all } x)\}$. To show that $\Gamma \nvDash \phi$, a concrete model will be used as a counterexample. Let:

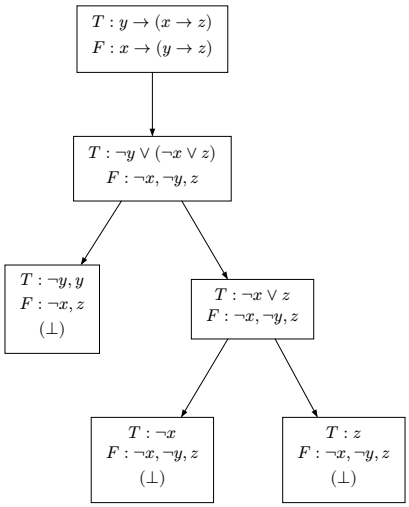
$$[\![x]\!] = \{1\}$$
$$[\![y]\!] = \{2\}$$
$$[\![z]\!] = \{(2, 1)\}$$

Then:

$$[\![z \text{ all } x]\!] = \{2\}$$
$$[\![z \text{ all } y]\!] = \emptyset$$

Since $\{1\} \notin \emptyset$, this counter model shows that $\Gamma \nvDash \phi$.

By the rules stated in Section 1, ϕ becomes $\phi^\star = x \to (y \to z)$ and $\Gamma^\star = \{y \to (x \to z)\}$ Now lets use tableau-style proof, to show that $\Gamma^\star \vdash \phi^\star$.



This shows that in all possible models of $\Gamma^\star$, $\Gamma^\star \vDash \phi^\star$

Problem 2

If ϕ is a sentence in propositional logic, let $\phi^\dagger$ be obtained from ϕ by negating all of the sentence letters. For example,

$$\big((p \vee (q \implies r))^\dagger \iff \neg(r \wedge p)\big)^\dagger = (\neg p \vee (\neg q \implies \neg r)) \iff \neg(\neg r \wedge \neg p)$$

(a) For any satisfiable propositional formula ϕ, must $\phi^\dagger$ also be satisfiable? Either prove or give a counterexample.

Solution:

The goal is to show that for any satisfiable propositional formula ϕ, $\phi^\dagger$ is also satisfiable.

ϕ is satisfiable if there is some function v such that finds a truth assignment such that the sentence evaluates to true:

$$\exists v.P \to Bool : [\![\phi]\!]_v = T$$

$\phi^\dagger$ is obtained from ϕ by negating all of the sentence letter. For it to be satisfiable, there must also be a function v' $s.t.$

$$\exists v'.P \to Bool : [\![\phi^\dagger]\!]_{v'} = T$$

v' being a function that assigns the opposite truth value to each sentence letter from v:

$$v'(x) = \begin{cases} F & v(x) = T \\ T & v(x) = F \end{cases}$$

Since $\phi^\dagger$ is obtained by negating every sentence letter in ϕ and v' negates the truth value of each sentence letter. Evaluating $\phi^\dagger$ with v' is the same as negating each letter and flipping it assigned truth value. This can be shown using induction:

Proof by induction

Goal: Show that $\left[\!\left[\phi^{\dagger}\right]\!\right]_{v'} = [\![\phi]\!]_v$ for all sentences ϕ

Base case

Case 1

Let ϕ be an atomic sentence p

Then $[\![\neg p_i]\!]_{v'} = [\![p_i]\!]_v$,

$\neg[\![p_i]\!]_{v'} = [\![p_i]\!]_v$

And so $\neg v'(p_i) = v(p_i)$ is trivial by v' definition

Case 2

Let ϕ be T or F:

Then $\left[\!\left[T^{\dagger}\right]\!\right]_{v'} = [\![T]\!]_v$

$[\![\neg T]\!]_{v'} = [\![T]\!]_v$

$\neg[\![T]\!]_{v'} = [\![T]\!]_v$

$\neg v'(T) = v(T)$ which is trivial by v' definition, and holds for F aswell.

Inductive hypothesis

Assume that $S(\phi) \stackrel{\text{def}}{=} \left[\!\left[\phi^{\dagger}\right]\!\right]_{v'} = [\![\phi]\!]_v$

Inductive step

ϕ can either be negated ($\neg$) or used with any other operator ($\vee, \wedge, \Longrightarrow$ or $\Longleftrightarrow$)

Case 1 (negation):

Show that $S(\phi) \Longrightarrow S(\neg\phi)$

By **IH**, this is:

$$\left[\!\left[\phi^{\dagger}\right]\!\right]_{v'} = [\![\phi]\!]_v \Longrightarrow \left[\!\left[\neg\phi^{\dagger}\right]\!\right]_{v'} = [\![\neg\phi]\!]_v$$

And thus:

$$\left[\!\left[\phi^{\dagger}\right]\!\right]_{v'} = [\![\phi]\!]_v \Longrightarrow \neg\left[\!\left[\phi^{\dagger}\right]\!\right]_{v'} = \neg[\![\phi]\!]_v$$

$$\left[\!\left[\phi^{\dagger}\right]\!\right]_{v'} = [\![\phi]\!]_v \Longrightarrow \neg\left[\!\left[\phi^{\dagger}\right]\!\right]_{v'} = \neg[\![\phi]\!]_v$$

$$[\![\neg\phi]\!]_{v'} = [\![\phi]\!]_v \Longrightarrow \neg[\![\neg\phi]\!]_{v'} = \neg[\![\phi]\!]_v$$

$$\neg[\![\phi]\!]_{v'} = [\![\phi]\!]_v \Longrightarrow [\![\phi]\!]_{v'} = \neg[\![\phi]\!]_v$$

$$\neg[\![\phi]\!]_{v'} = [\![\phi]\!]_v \Longrightarrow [\![\phi]\!]_{v'} = \neg[\![\phi]\!]_v$$

$$\neg v'(\phi) = v(\phi) \Longrightarrow v'(\phi) = \neg v(\phi)$$

Which is trivially true by v' definition

Case 2 (other operators)

Let $\odot$ be any of the following logical operators $\wedge, \vee, \Longrightarrow$ or $\Longleftrightarrow$.

Show that if $S(\phi) \odot S(\psi) \Longrightarrow S(\phi \odot \psi)$

Then

$$\left(\left[\!\left[\phi^{\dagger}\right]\!\right]_{v'} = \left[\!\left[\phi\right]\!\right]_{v}\right) \odot \left(\left[\!\left[\psi^{\dagger}\right]\!\right]_{v'} = \left[\!\left[\psi\right]\!\right]_{v}\right) \Longrightarrow \left[\!\left[(\phi \odot \psi)^{\dagger}\right]\!\right]_{v'} = \left[\!\left[\phi \odot \psi\right]\!\right]_{v'}$$

$$\Longrightarrow \left[\!\left[\phi^{\dagger}\right]\!\right]_{v'} \odot \left[\!\left[\psi^{\dagger}\right]\!\right]_{v'} = \left[\!\left[\phi\right]\!\right]_{v} \odot \left[\!\left[\psi\right]\!\right]_{v}$$

$$(\left[\!\left[\neg\phi\right]\!\right]_{v'} = \left[\!\left[\phi\right]\!\right]_{v}) \odot (\left[\!\left[\neg\psi\right]\!\right]_{v'} = \left[\!\left[\psi\right]\!\right]_{v}) \Longrightarrow \left[\!\left[\neg\phi\right]\!\right]_{v'} \odot \left[\!\left[\neg\psi\right]\!\right]_{v'} = \left[\!\left[\phi\right]\!\right]_{v} \odot \left[\!\left[\psi\right]\!\right]_{v}$$

$$(\neg\left[\!\left[\phi\right]\!\right]_{v'} = \left[\!\left[\phi\right]\!\right]_{v}) \odot (\neg\left[\!\left[\psi\right]\!\right]_{v'} = \left[\!\left[\psi\right]\!\right]_{v}) \Longrightarrow \neg\left[\!\left[\phi\right]\!\right]_{v'} \odot \neg\left[\!\left[\psi\right]\!\right]_{v'} = \left[\!\left[\phi\right]\!\right]_{v} \odot \left[\!\left[\psi\right]\!\right]_{v}$$

$$(\neg v'(\phi) = v(\phi)) \odot (\neg v'(\psi) = v(\psi)) \Longrightarrow \neg v'(\phi) \odot \neg v'(\psi) = v(\phi) \odot v(\psi)$$

27 This is trivial by definition of v' and applies to all logical operators.

The proof by induction shows that:

$$\left[\left[\phi^{\dagger}\right]\right]_{v'} = \left[\left[\phi\right]\right]_{v}$$

And from the definition of satisfiability:

$$\exists v : \left[\left[\phi\right]\right]_{v} = T \Longrightarrow \exists v' : \left[\left[\phi^{\dagger}\right]\right]_{v} = T$$

(b) Find a formula ϕ such that ϕ is neither a tautology nor a contradiction and ϕ and $\phi^{\dagger}$ are logically equivalent? Find one or prove that none exists.

Assume that ϕ is satisfiable and falsifiable which means:

$$\exists v.P \to \text{Bool} : \left[\left[\phi\right]\right]_{v} = T$$
$$\exists f.P \to \text{Bool} : \left[\left[\phi\right]\right]_{f} = F$$

From definition 5.8, two sentences ϕ, ψ are logically equivalent if:

$$\forall v : \left[\!\left[\phi\right]\!\right]_{v} = \left[\!\left[\psi\right]\!\right]_{v}$$

Let's now assign

$$\phi := p \Longleftrightarrow q$$
$$\phi^{\dagger} := \neg p \Longleftrightarrow \neg q$$

and from the formulas truth table of possible models:

p	q	$\neg p$	$\neg q$	$p \iff q$	$\neg p \iff \neg q$
T	T	F	F	T	T
T	F	F	T	F	F
F	T	T	F	F	F
F	F	T	T	T	T

Table 1: truth table of ϕ and $\phi^\dagger$,
showing that both ϕ and $\phi^\dagger$ is satisfiable

From Table 1 it can be seen that for both sentences, there exits both a v, f such that the models evaluate to T and F. For all possible models of assignment for each sentence letter, both sentences evaluates to the same, therefore $\phi \equiv \phi^\dagger$.

Problem 3

Work in the signature that has sentence letters $p_{i,j}$ for $0 \leq i, j \leq 9$ (one hundred letters in all). Consider the formula Φ which is the conjunction of all of the following clauses:

- $p_{0,9} \wedge p_{9,0}$,
- $p_{i,j} \Longrightarrow p_{i-1,j}$ for each $1 \leq i \leq 9$ and $0 \leq j \leq 9$, and
- $p_{i,j} \Longrightarrow p_{i,j-1}$ for each $0 \leq i \leq 9$ and $1 \leq j \leq 9$.

How many models satisfy Φ? (*Hint.* If this seems too daunting, replace 9 with something smaller.)

Solution

First of all to complete this problem it is necessary to figure out what the first possible state of the space is based on the given clauses. It is known from the clauses that both $p_{0,9}$ and $p_{9,0}$ need to be filled out, as this is the first given clause, the next two clauses then also state something that based on the two original points will create a change in the starting space. The 2nd clause states that every cell to the left of a filled out cell, will also need to be filled out, and the third clause states that every cell below any colored cell will also need to be colored.

Then the first state of all the cells can be visualized in a table form like this, where the red cells are the ones filled by the first clause:

x									
x									
x									
x									
x									
x									
x									
x									
x									
x	x	x	x	x	x	x	x	x	x

Then the next step would be to figure out how much is needed to actually fill all the cells, when looking at the build of the table and the clauses, we can sort of imagine that some steps, either normal, steep or very wide would happen when attempting to fill out in every different combination of cells. And since it is a lot of combinations, the logical way forwards would be using combinatorics.

We can see that the total amount of cells from all the way left to right or top to bottom is 10, so essentially the grid is 10×10 but since the first row and column is filled by the clauses we know that it is know a 9×9 grid, and due to the 2nd and 3rd clause we know that they will fill out in a way that fills a lot of cells and combinations at a time, meaning that we can essentially only take 9 steps to the right or 9 steps up.

Now it can just be entered into the combination formula given on form:

$$^{n}C_{r} = \frac{n!}{r! \cdot (n-r)!}$$

Where n is the total amount of moves that can be done, and r is the max amount of moves that can be done in one step.

Then we can add the numbers from our actual task where we know that there is essentially a total of 18 steps and 9 steps that can be taken in one step.

$$^{18}C_{9} = \frac{18!}{9! \cdot (18-9)!} = 48620$$

Meaning that there is a total of 48620 models that satisfy the given clauses.