# Questionnaire "Logic and Computability"

# Summer Term 2023

## Contents

# 1 Propositional Logic

## 1.1 Declarative Sentences

1.1.1 Look at the following statements and tick them if they are true.

- ☐ "*Give me the butter.*" is a declarative sentence.
- ☐ Questions are always declarative sentences.
- ☐ Declarative sentences can be true and false at the same time.
- ☒ "*My best friend is staying overnight.*" is a declarative sentence.

1.1.2 Model the following sentences as detailed as possible in propositional logic.

(a) Alice will either take the bike or the tram to get to the concert, not both.
(b) Students will have to take an exam at the end of the semester.
(c) If he is hungry and the fridge is not empty, he cooks for himself.

**Solution**

(a) $p$ : Alice will take the bike to get to the concert.
$q$ : Alice will take the tram to get to the concert.

$$(p \wedge \neg q) \vee (\neg p \wedge q)$$

(b) $p$ : Students will have to take an exam at the end of the semester.

$$p$$

(c) $p$ : He is hungry.
$q$ : The fridge is empty.
$r$ : He cooks for himself.

$$p \wedge \neg q \rightarrow r$$

1.1.3 Model the following sentences as detailed as possible in propositional logic.

(a) If the air temperature is above 30°C, then the water temperature is above 20°C and I am able to go for a swim.
(b) Your kid will be safe if and only if it learns to swim.
(c) What time is it?

**Solution**

(a) $p$ : The air temperature is above 30°C.
$q$ : The water temperature is above 20°C.
$r$ : I am able to go for a swim.

$$p \to q \land r$$

(b) $p$ : Your kid will be safe.
$q$ : Your kid learns to swim.

$$p \leftrightarrow q$$

(c) This is not a declarative sentence.

1.1.4 Model the following sentences as detailed as possible in propositional logic.

(a) Bob will win the lottery, if and only if he gets all the numbers right.
(b) Mozart was born in Salzburg, not in Innsbruck.
(c) If the year is a leap-year, then February will have 29 days.

**Solution**

(a) $b$ : Bob wins the lottery.
$n$ : Bob gets all the numbers right.

$$b \leftrightarrow n$$

(b) $S$ : Mozart was born in Salzburg.
$I$ : Mozart was born in Innsbruck

$$S \land \neg I$$

(c) $l$ : The year is a leap-year.
$f$ : Februar has 29 days.

$$l \to f$$

1.1.5 Model the following sentences as detailed as possible in propositional logic.

(a) Either Alice and Bob are going together to the conference or neither of them is going.
(b) Today is Friday, if and only if yesterday was Thursday and tomorrow is not Sunday.
(c) A model assigns truth values to variables of a propositional formula.

**Solution**

(a) $a$ : Alice is going to the conference.
$b$ : Bob is going to the conference.

$$a \land b \lor \neg a \land \neg b$$

(b) $S$ : Mozart was born in Salzburg.
$I$ : Mozart was born in Innsbruck

$$S \land \neg I$$

(c) $m$ : A model assigns truth values to variables of a propositional formula.

$$m$$

1.1.6 Model the following sentences as detailed as possible in propositional logic.

(a) If a formula is unsat, it cannot be valid.

(b) It can be proven that there exist an infinite number of primes.

(c) A sentence is called declarative, if and only if it can be assigned a truth value.

**Solution**

(a) $u$ : The formula is unsat.
$v$ : The formula is valid.

$$u \to \neg v$$

(b) $P$ : It can be proven that there exist an infinite number of primes.

$$P$$

(c) $d$ : The sentence is a declarative sentence.
$t$ : A truth value can be assigned to the sentence.

$$d \leftrightarrow t$$

1.1.7 Model the following sentences as detailed as possible in propositional logic.

(a) Today it will either be foggy or rainy, but not both.

(b) For any program we can prove that the program halts.

(c) The system will never crash, if it got completely verified at design time and is monitored during runtime.

**Solution**

(a) $f$ : Today it will be foggy.
$r$ : Today it will be rainy.

$$(f \wedge \neg r) \vee (\neg f \wedge r)$$

(b) $P$ : For any program we can prove that the program halts.

$$P$$

(c) $s$ : The system will never crash.
$v$ : The system has been verified during design time.
$m$ : The system is being monitored during runtime.

$$v \wedge m \rightarrow s$$

## 1.2 Syntax of Propositional Logic

1.2.1 Give the definition of well-formed formulas in propositional logic.

**Solution**

We give the definition of well-formed formulas in propositional logic using a grammar in Backus-Naur form (BNF) as:

$$\varphi := \ <\text{atomic proposition}> \mid \varphi \wedge \varphi \mid \varphi \vee \varphi \mid \neg\varphi \mid \varphi \rightarrow \varphi \mid \varphi \leftrightarrow \varphi \mid (\varphi)$$

1.2.2 How can you determine whether a string is a *well-formed formula* using a parse tree?

**Solution**

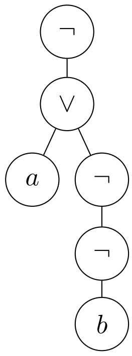
*There is no solution available for this question yet.*

1.2.3 Let $p, q$ and $r$ be a atomic propositions. Tick all statements that are true.

- ☐ "$\neg p \wedge \vee\ q$" is a propositional formula.
- ☒ "$(p \wedge q) \vee (r \rightarrow p)$" is a propositional formula.
- ☒ "$\neg p$" is a propositional formula.
- ☐ "$\vee$" is a propositional formula.
- ☒ "$p$" is a propositional formula.

1.2.4 Determine whether the string $\neg(a \vee \neg\neg b)$ is a well-formed formula using the parse tree. Explain your answer.

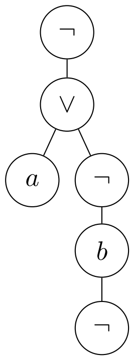
**Solution**




Every leaf is a atomic variable and the other nodes are labeled with logical operators, thus this is a well-formed formula.

1.2.5 Determine whether the string $\neg(a \vee \neg b \neg)$ is a well-formed formula using the parse tree. Explain your answer.

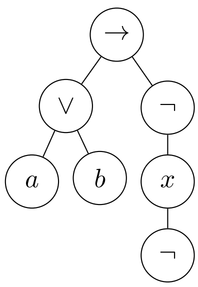
**Solution**




One leaf is labeled with a logical operator, which is not allowed. Thus this is not a well-formed formula.

1.2.6 Determine whether the string $(a \vee b) \rightarrow (\neg(x\neg))$ is a well-formed formula using the parse tree. Explain your answer.

**Solution**




One leaf is labeled with a logical operator, which is not allowed. Thus this is not a well-formed formula.

## 1.3 Semantics of Propositional Logic

1.3.1 Give the definitions of *syntax* of propositional logic and *semantics* of propositional logic. What is the difference between syntax and semantic?

**Solution**

Syntax refers to *grammar*, while semantics refers to *meaning*.
Syntax is the set of rules needed to ensure a formula is a well-formed formula; semantics assigns a truth value to formulas by assigning a truth value to the propositional variables used in the formula and by assigning the meaning via truth table to the logical operators.

1.3.2 Give the definition of the *semantics* of propositional logic.

**Solution**

The semantics of propositional logic define truth values to propositional variables and defines the rules for the propositional operators via their corresponding *truth tables*.

1.3.3 Consider a formula $\varphi$ in propositional logic. Let the number of propositional variables in $\varphi$ be $n$. How many rows does the truth table for $\varphi$ have?

**Solution**

$2^n$

1.3.4 Consider a truth table for a propositional formula $\varphi$ that has $R$ rows. How many propositional variables does $\varphi$ have?

**Solution**

$log_2(R)$

1.3.5 Why are truth tables, in general, not used to determine equivalence of large formulas?

**Solution**

*There is no solution available for this question yet.*

1.3.6 Give the definition of a model $\mathcal{M}$ of a formula in propositional logic? Explain the difference between a *full* and a *partial* model.

**Solution**

A *model* $\mathcal{M}$ of a propositional formula is an assignment of truth values to variables in the formula. A model $\mathcal{M}$ is called a *full* (or *total*) assignment if $\mathcal{M}$ assigns a truth value to each variable in the formula. Otherwise, $\mathcal{M}$ is called a *partial* model.

1.3.7 What is the difference between a *satisfying model* and a *falsifying model* of a formula in propositional logic? Give a satisfying and a falsifying model for the formula $\varphi = a \to b$.

Solution

*There is no solution available for this question yet.*

1.3.8 Consider the propositional formula $\varphi = (p \land q) \to (q \lor \neg r)$. Fill out the truth table for $\varphi$ and its subformulas.

| $p$ | $q$ | $r$ | $p \land q$ | $\neg r$ | $q \lor \neg r$ | $\varphi = (p \land q) \to (q \lor \neg r)$ |
|---|---|---|---|---|---|---|
| **F** | **F** | **F** | F | T | T | T |
| **F** | **F** | **T** | F | F | F | T |
| **F** | **T** | **F** | F | T | T | T |
| **F** | **T** | **T** | F | F | T | T |
| **T** | **F** | **F** | F | T | T | T |
| **T** | **F** | **T** | F | F | F | T |
| **T** | **T** | **F** | T | T | T | T |
| **T** | **T** | **T** | T | F | T | T |

1.3.9 Consider the propositional formula $\varphi = \neg(\neg p \lor q) \to (p \land \neg r)$.

Find a propositional formula $\psi$ that is syntactically different from $\varphi$, but semantically equivalent to $\varphi$. Show the semantic equivalence of $\varphi$ and $\psi$ using truth tables.

Solution

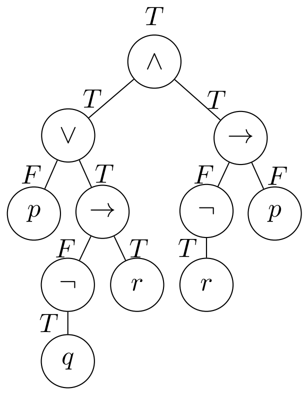
| $p$ | $q$ | $r$ | $\neg p \lor q$ | $\neg(\neg p \lor q)$ | $p \land \neg r$ | $\varphi = \neg(\neg p \lor q) \to (p \land \neg r)$ |
|---|---|---|---|---|---|---|
| **F** | **F** | **F** | T | F | F | T |
| **F** | **F** | **T** | T | F | F | T |
| **F** | **T** | **F** | T | F | F | T |
| **F** | **T** | **T** | T | F | F | T |
| **T** | **F** | **F** | F | T | T | T |
| **T** | **F** | **T** | F | T | F | F |
| **T** | **T** | **F** | T | F | T | T |
| **T** | **T** | **T** | T | F | F | T |

$$\psi = \neg p \lor q \lor \neg r$$

| $p$ | $q$ | $r$ | $\neg p$ | $\neg r$ | $\psi = \neg p \lor q \lor \neg r$ |
|---|---|---|---|---|---|
| **F** | **F** | **F** | T | T | T |
| **F** | **F** | **T** | T | F | T |
| **F** | **T** | **F** | T | T | T |
| **F** | **T** | **T** | T | F | T |
| **T** | **F** | **F** | F | T | T |
| **T** | **F** | **T** | F | F | F |
| **T** | **T** | **F** | F | T | T |
| **T** | **T** | **T** | F | F | T |

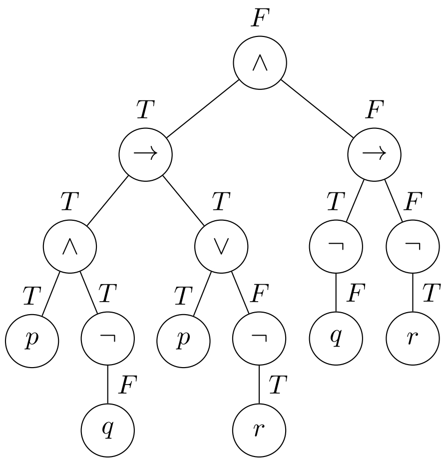
1.3.10 Given is a formula $\varphi = (p \lor (\neg q \to r)) \land (\neg r \to p)$ and a model $\mathcal{M} = \{p = F,\ q = T,\ r = T\}$. Determine the truth value of $\varphi$ for the given model $\mathcal{M}$ using its parse tree.

Solution



1.3.11 Given is a formula $\varphi = ((p \wedge \neg q) \to (p \vee \neg r)) \wedge (\neg q \to \neg r)$ and a model $\mathcal{M} = \{p = T,\ q = F,\ r = T\}$. Determine the truth value of $\varphi$ for the given model $\mathcal{M}$ using its parse tree.

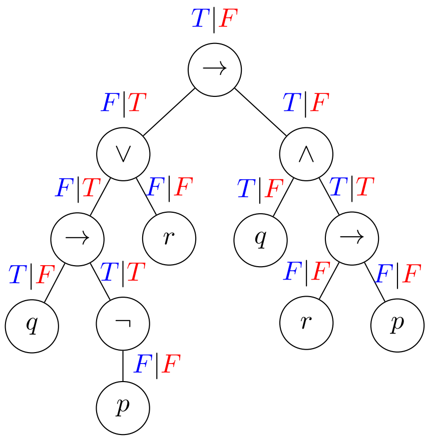
Solution



1.3.12 Given is a formula $\varphi = ((q \to \neg p) \vee r) \to (q \wedge (r \to p))$. Determine a satisfying model $\mathcal{M}_1$ and a falsifying model $\mathcal{M}_2$ using its parse tree.

Solution

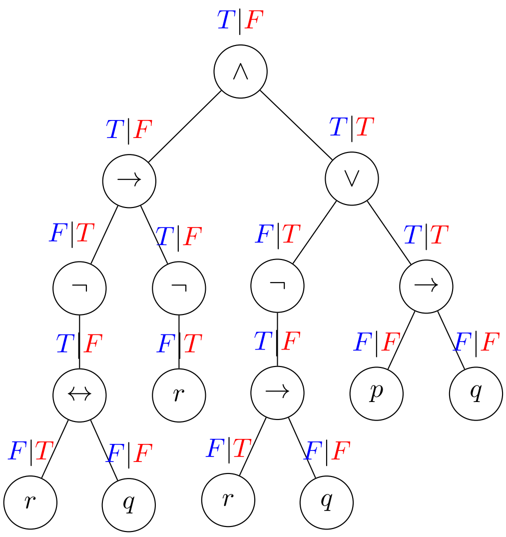
Let $\mathcal{M}_1 = \{p = \mathbf{F}, q = \mathbf{T}, r = \mathbf{F}\}$ and $\mathcal{M}_2 = \{p = \mathbf{F}, q = \mathbf{F}, r = \mathbf{F}\}$ and annotate the parse tree with in blue for $\mathcal{M}_1$ and red for $\mathcal{M}_2$.



1.3.13 Given is a formula $\varphi = (\neg(r \leftrightarrow q) \to \neg r) \wedge (\neg(r \to q) \vee (p \to q))$. Determine a satisfying model $\mathcal{M}_1$ and a falsifying model $\mathcal{M}_2$ using its parse tree.

Solution

Let $\mathcal{M}_1 = \{p = \mathbf{F}, q = \mathbf{F}, r = \mathbf{F}\}$ and $\mathcal{M}_2 = \{p = \mathbf{F}, q = \mathbf{F}, r = \mathbf{T}\}$ and annotate the parse tree with in blue for $\mathcal{M}_1$ and red for $\mathcal{M}_2$.



## 1.4 Semantic Entailment, Equivalence, Satisfiability and Validity

1.4.1 Give the definition of semantic entailment of two propositional formulas.

Solution

Let $\varphi$ and $\psi$ be formulas in propositional logic. We say that $\varphi \models \psi$ if and only if every model $\mathcal{M}$ that satisfies $\varphi$ ($\mathcal{M} \models \varphi$) also satisfies $\psi$ ($\mathcal{M} \models \psi$).

1.4.2 Give the definition of semantic equivalence of two propositional formulas.

**Solution**

Let $\varphi$ and $\psi$ be formulas in propositional logic. We say that $\varphi$ and $\psi$ are semantically equivalent if and only if $\varphi \models \psi$ and $\psi \models \varphi$ holds.
In that case we write $\varphi \equiv \psi$.

1.4.3 Give the definition of validity of a propositional formula.

**Solution**

Let $\varphi$ be a formula of propositional logic. We call $\varphi$ valid if $\models \varphi$ holds, i.e., any possible model for $\varphi$ is a satisfying model.

1.4.4 Give the definition of satisfiability and unsatisfiability of a propositional formula.

**Solution**

Given a formula $\varphi$ in propositional logic, we say that $\varphi$ is *satisfiable* if it has a model in which is evaluates to *true*. We say that $\varphi$ is *unsatisfiable* if there is no model under which $\varphi$ evaluates to *true*.

1.4.5 Consider a formula $\varphi$ in propositional logic. In the following list, tick all statements that are true.

- ☐ If $\varphi$ is a tautology, a falsifying model can be found.
- ☒ If $\varphi$ is equivalent to $\psi$, a satisfying model for $\varphi$ always satisfies $\psi$.
- ☐ If $\varphi$ has no satisfying model, it is called a tautology.
- ☐ If $\varphi$ semantically entails $\psi$, a satisfying model for $\psi$ always satisfies $\varphi$.

1.4.6 Consider a formula $\varphi$ in propositional logic. In the following list, tick all statements that are true.

- ☒ If $\varphi$ is not satisfiable, $\neg\varphi$ is valid.
- ☒ If $\varphi$ is valid, $\neg\varphi$ is not valid.
- ☒ If $\varphi$ is valid, $\neg\varphi$ is not satisfiable.
- ☒ If $\varphi$ is not valid, $\neg\varphi$ is satisfiable.

1.4.7 Consider a formula $\varphi$ in propositional logic. You want to test whether $\varphi$ is *valid*. However, you only have a procedure for checking satisfiability. Describe how to use this procedure to determine whether $\varphi$ is valid.

**Solution**

*There is no solution available for this question yet.*

1.4.8 Given are the truth tables for the propositional logic formulas $\varphi$ and $\psi$. Determine whether it holds that $\varphi \models \psi$, $\psi \models \varphi$, or neither.

| $p$ | $q$ | $r$ | $\varphi$ | $\psi$ |
|---|---|---|---|---|
| **F** | **F** | **F** | F | F |
| **F** | **F** | **T** | T | T |
| **F** | **T** | **F** | F | F |
| **F** | **T** | **T** | T | T |
| **T** | **F** | **F** | F | F |
| **T** | **F** | **T** | F | T |
| **T** | **T** | **F** | T | T |
| **T** | **T** | **T** | T | T |

Solution

It holds that $\varphi \models \psi$.

1.4.9 Consider the propositional formulas $\varphi = (p \to q) \vee \neg r$ and $\psi = (\neg r \wedge p) \vee (\neg q \to \neg r)$.

(a) Fill out the truth table for $\varphi$ and $\psi$ and their subformulas.

| $p$ | $q$ | $r$ | $\neg q$ | $\neg r$ | $p \to q$ | $\neg r \wedge p$ | $\neg q \to \neg r$ | $\varphi$ | $\psi$ |
|---|---|---|---|---|---|---|---|---|---|
| **F** | **F** | **F** | T | T | T | F | T | T | T |
| **F** | **F** | **T** | T | F | T | F | F | T | F |
| **F** | **T** | **F** | F | T | T | F | T | T | T |
| **F** | **T** | **T** | F | F | T | F | T | T | T |
| **T** | **F** | **F** | T | T | F | T | T | T | T |
| **T** | **F** | **T** | T | F | F | F | F | F | F |
| **T** | **T** | **F** | F | T | T | T | T | T | T |
| **T** | **T** | **T** | F | F | T | F | T | T | T |

(b) Which of the formulas is satisfiable?
Both of them are satisfiable.

(c) Which of the formulas is valid?
None of them are valid.

(d) Which of the two formulas $\varphi$ and $\psi$ entails the other?
It holds that $\psi \models \varphi$.

1.4.10 Consider the propositional formulas $\varphi = (p \vee q) \to r$, and $\psi = r \vee (\neg p \wedge \neg q)$.

(a) Fill out the truth table for $\varphi$ and $\psi$ (and their subformulas).

| $p$ | $q$ | $r$ | $\neg p$ | $\neg q$ | $p \vee q$ | $\neg p \wedge \neg q$ | $\varphi$ | $\psi$ |
|---|---|---|---|---|---|---|---|---|
| **F** | **F** | **F** | **T** | **T** | **F** | **T** | **T** | **T** |
| **F** | **F** | **T** | **T** | **T** | **F** | **T** | **T** | **T** |
| **F** | **T** | **F** | **T** | **F** | **T** | **F** | **F** | **F** |
| **F** | **T** | **T** | **T** | **F** | **T** | **F** | **T** | **T** |
| **T** | **F** | **F** | **F** | **T** | **T** | **F** | **F** | **F** |
| **T** | **F** | **T** | **F** | **T** | **T** | **F** | **T** | **T** |
| **T** | **T** | **F** | **F** | **F** | **T** | **F** | **F** | **F** |
| **T** | **T** | **T** | **F** | **F** | **T** | **F** | **T** | **T** |

(b) Which of the formulas is satisfiable?
Both are satisfiable.

(c) Which of the formulas is valid?
Neither are valid.

(d) Is $\varphi$ equivalent to $\psi$?
They are semantically equivalent.

(e) Does $\varphi$ semantically entail $\psi$?
Yes.

(f) Does $\psi$ semantically entail $\varphi$?
Yes.

1.4.11 Consider the propositional formula $\varphi = p \to (q \to r)$.

(a) Fill out the truth table for $\varphi$ and its subformulas.

| $p$ | $q$ | $r$ | $(q \to r)$ | $\varphi = p \to (q \to r)$ |
|---|---|---|---|---|
| **F** | **F** | **F** | **T** | **T** |
| **F** | **F** | **T** | **T** | **T** |
| **F** | **T** | **F** | **F** | **T** |
| **F** | **T** | **T** | **T** | **T** |
| **T** | **F** | **F** | **T** | **T** |
| **T** | **F** | **T** | **T** | **T** |
| **T** | **T** | **F** | **F** | **F** |
| **T** | **T** | **T** | **T** | **T** |

(b) Is $\varphi$ satisfiable?
Yes.

(c) Give a formula $\psi$ that is semantically equivalent to $\varphi$, but does not use the "$\to$" connective.
$\psi = \neg p \vee (\neg q \vee r)$

(d) How can you check whether $\psi$ is semantically equivalent to $\varphi$? Since both formulas are relatives compact, we can use their respective truth table to check whether they are semantically equivalent. We do this by checking whether they evaluate to **T** under the same models.

1.4.12 Consider the propositional formula $\psi = (p \to q) \wedge (q \to r) \wedge (\neg r \vee p)$.

(a) Fill out the truth table for $\varphi$ (and its subformulas).

| $p$ | $q$ | $r$ | $(p \to q)$ | $(q \to r)$ | $\neg r$ | $(\neg r \vee p)$ | $\psi$ |
|---|---|---|---|---|---|---|---|
| **F** | **F** | **F** | **T** | **T** | **T** | **T** | **T** |
| **F** | **F** | **T** | **T** | **T** | **F** | **F** | **F** |
| **F** | **T** | **F** | **T** | **F** | **T** | **T** | **F** |
| **F** | **T** | **T** | **T** | **T** | **F** | **F** | **F** |
| **T** | **F** | **F** | **F** | **T** | **T** | **T** | **F** |
| **T** | **F** | **T** | **F** | **T** | **F** | **T** | **F** |
| **T** | **T** | **F** | **T** | **F** | **T** | **T** | **F** |
| **T** | **T** | **T** | **T** | **T** | **F** | **T** | **T** |

(b) Is $\varphi$ satisfiable?
Yes.

(c) Is $\varphi$ valid?
No.

(d) Give a formula $\varphi$ that semantically entails $\psi$.
An equivalent formula $\varphi$ semantically entails $\psi$, therefore we let $\varphi = \psi$.

(e) How can you check, using a truth table, whether $\varphi$ semantically entails $\psi$?
We can check this by looking at the models that satisfy the two formulas. If every model that satisfies $\varphi$ also models $\psi$, then $\varphi$ semantically entails $\psi$.

1.4.13 Consider the propositional formula $\varphi = (\neg p \to r) \wedge (r \to \neg p) \wedge q$.

(a) Fill out the truth table for $\varphi$ (and its subformulas).

| $p$ | $q$ | $r$ | $\neg p$ | $(\neg p \rightarrow r)$ | $(r \rightarrow \neg p)$ | $\varphi$ |
|---|---|---|---|---|---|---|
| **F** | **F** | **F** | **T** | **F** | **T** | **F** |
| **F** | **F** | **T** | **T** | **T** | **T** | **F** |
| **F** | **T** | **F** | **T** | **F** | **T** | **F** |
| **F** | **T** | **T** | **T** | **T** | **T** | **T** |
| **T** | **F** | **F** | **F** | **T** | **T** | **F** |
| **T** | **F** | **T** | **F** | **T** | **F** | **F** |
| **T** | **T** | **F** | **F** | **T** | **T** | **T** |
| **T** | **T** | **T** | **F** | **T** | **F** | **F** |

(b) Is $\varphi$ satisfiable?
Yes.

(c) Is $\varphi$ valid?
No.

(d) Give a formula $\psi$ that semantically entails $\varphi$.
For any formula $\varphi$ it holds that $\bot \models \varphi$, we can therefore choose $\psi = \bot$. We could also represent $\varphi$ as DNF: $(\neg p \wedge q \wedge r) \vee (p \wedge q \wedge \neg r)$. This is an equivalent formula and therefore semantically entails $\varphi$.

(e) Give a formula $\psi$ such that $\varphi$ semantically entails $\psi$.
For any formula $\varphi$ it holds that $\varphi \models \top$. We could also again choose $\varphi$ in DNF.

## 1.5 Modelling Example

1.5.1 Use propositional logic to solve Sudoku. Rules: A Sudoku grid consists of a 9x9 square, which is partitioned into nine 3x3 squares. The goal of the game is to write one number from 1 to 9 in each cell in such a way, that each row, each column, and each 3x3-square contains each number exactly once. Usually several numbers are already given.

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | 6 | | 7 | | | 1 | 5 | |
| | | 3 | 9 | | | 8 | | |
| | | 2 | 3 | | | | 4 | 9 |
| | | 7 | | | 4 | | | |
| | 4 | | | 9 | | | 8 | |
| | | | 1 | | | 4 | | |
| 6 | 7 | | | | 9 | 3 | | |
| | | 9 | | | 2 | 5 | | |
| | 2 | 8 | | | 7 | | 6 | |

Sudoku

In order to model SUDOKU using propositional logic, we first need to define the propositional variables that we want to use in our formula. We define variables $x_{ijk}$ for every row $i$, for every column $j$, and for every value $k$. This encoding yields to 729 variables ranging from $x_{111}$ to $x_{999}$. Using this variables, define the constraints for the rows, the columns, the 3x3-squares and the predefined numbers.

Solution

- *Row-constraints: If* a cell in a row has a certain value, *then* no other cell in that row can have that value. For each $i$, and each $k$ we have:

$$x_{i1k} \to \neg x_{i2k} \wedge \neg x_{i3k} \wedge ... \wedge \neg x_{i9k}$$

$$x_{i2k} \to \neg x_{i1k} \wedge \neg x_{i2k} \wedge ... \wedge \neg x_{i9k}$$

$$\vdots$$

$$x_{i9k} \to \neg x_{i1k} \wedge \neg x_{i2k} \wedge ... \wedge \neg x_{i8k}$$

- *Column-constraints: If* a cell in a column has a certain value, *then* no other cell in that column can have that value. For each $j$, and each $k$ we have:

$$x_{1jk} \to \neg x_{2jk} \wedge \neg x_{3jk} \wedge ... \wedge \neg x_{9jk}$$

$$x_{2jk} \to \neg x_{1jk} \wedge \neg x_{2jk} \wedge ... \wedge \neg x_{9jk}$$

$$\vdots$$

$$x_{9jk} \to \neg x_{1jk} \wedge \neg x_{2jk} \wedge ... \wedge \neg x_{8jk}$$

- *Square-constraints: If* a cell in a 3x3 square has a certain value, *then* no other cell in that square can have that value. For the first square, we have for each $k$:

$$x_{11k} \to \neg x_{12k} \wedge \neg x_{13k} \wedge \neg x_{21k} \wedge \neg x_{22k} \wedge \neg x_{23k} \wedge \neg x_{31k} \wedge \neg x_{32k} \wedge \neg x_{33k}$$

$$\vdots$$

$$x_{33k} \to \neg x_{11k} \wedge \neg x_{12k} \wedge \neg x_{13k} \wedge \neg x_{21k} \wedge \neg x_{22k} \wedge \neg x_{23k} \wedge \neg x_{31k} \wedge \neg x_{32k}$$

  The constraints for the remaining squares are similar.

- *Predefined-number-constraints:* If a cell has a predefined value, we need to set the corresponding variable to true, e.g., the cell in the fifth row and the fifth column has the value 9. Therefore we have

$$x_{559}.$$

- *Cell-constraints:* Each cell must contain a number ranging from one to nine. For each $i$, and each $j$ we have

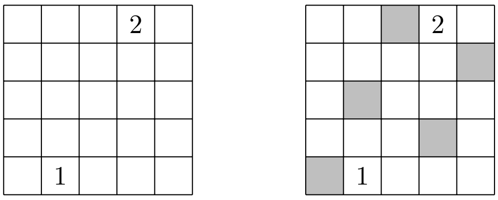
$$x_{ij1} \vee x_{ij2} \vee ... \vee x_{ij9}.$$

  On its own, this constraint would allow for a cell to have more than one value. However, this is not possible due to the other constraints.

To construct the final propositional formula, all constraints need to be connected via conjunctions. A satisfying assignment for the final formula represents one possible solution for the Sudoku puzzle. In case that there does not exists a solution, the SAT sovler would return UNSAT.

1.5.2 Describe the Latin Square Puzzle using propositional logic.

In the Latin Square Puzzle one has to color cells in an (n×n) grid such that there is exactly one colored cell in each row and each column. Furthermore, colored cells must not be adjacent to each other (also not diagonally). Numbers contained in certain cells of the grid indicate the exact number of colored cells that have to be adjacent (including diagonally) to it. Numbered cells can contain the numbers 0, 1, 2 and cannot be colored.



Example Latin Square Puzzle and its solution

Find propositional formulas which describe the puzzle and which could be used to solve it. Focus on explaining the concept of the formulas. You do not have to explicitly list all formulas and you do not have to solve the puzzle.

*Hints:* Use propositional atoms $c_{i,j}$, $c_{i,j,0}$, $c_{i,j,1}$,$c_{i,j,2}$ to represent each cell of the $(n \times n)$ game board. If $c_{i,j}$ has the value *True*, the cell $i, j$ is colored, otherwise it is not colored. If $c_{i,j,x}$ has the value *True*, the cell $i, j$ contains the number $x$.

Express the following constraints:

(a) There is exactly one colored cell in row $i$.

(b) No colored cells are adjacent to each other.

(c) No numbered cells can be colored.

(d) Numbered cells are adjacent to the indicated amount of colored cells.

Solution

*There is no solution available for this question yet.*