Logic for Knowledge Representation, Learning, and Inference

Luciano Serafini

serafini@fbk.eu

Contents

CHAPTER 1

Modelling in Propositional Logic

1. Logic Based Problem-solving

One of the most important applications of logic in artificial intelligence is in providing a general method for solving problems by modeling the problem in terms of logical formulas and finding the solution by applying some form of logical inference. This role of logic has been clearly identified by Adnan Darwiche in Darwiche 2020.

At an abstract level, a problem is specified by providing a set of *hypothesis* (e.g., input data, set of hypothesis, context, background knowledge, set of rules, "'. . .) and a *query*, whose answer should be inferred from the hypothesis. The main task in logic-based problem-solving is in *modeling* hypothesis in terms of a set of logical formulas so that the answer of the *query* can be obtained by some logical inference from such a set of formulas. The main schema is shown in Figure 1. A real world problem can be seen seen as a question to be answered given a set of data. For instance one would like to know who is the murder between a group of suspected persons and a set of clues. The data are the fact that the murdered is one among the suspected people and all the cues, the query is "who is the the murderer?", the (correct) answer will identify the person who actually committed the murder. Perhaps an example closer to real application, is the problem of finding the shortest path from one point to the other of a town. The hypothesis (data, background knowledge) are the street connections, the query is "find the shortest path from

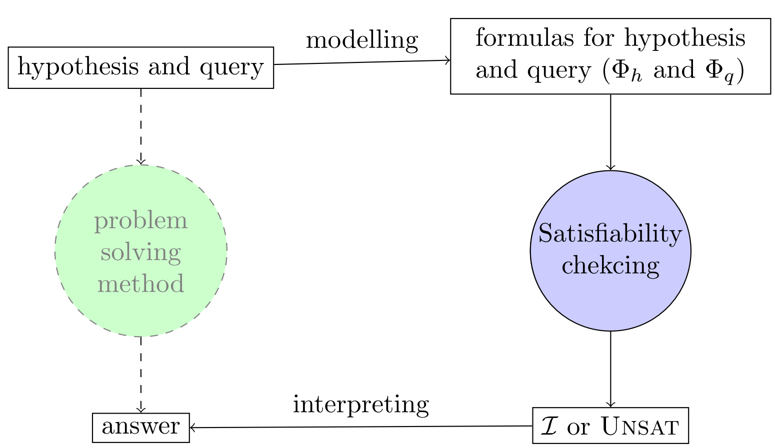


FIGURE 1. Schema of logic based problem solving method.

point a to point b" the answer is a sequence of connected streets that connets a and b with minimal length.

A general method for solving these classes of problems is by modelling the bypothesis and the query in a set of logic formulas Φ_h and Φ_q respectively and then apply some generic inference algorithm on Φ_h and Φ_q. Such an algorithm will provide an answer which is described in logical terms. Such an anwer need to be *iterpreted* to provide an answer to the real prolem.

The simplext logical inference task is *satisfiability*. I.e., check if a set of formula Φ is satisfiable by sone assignment to its propositional variables. In other words seach for an assignment $\mathcal{I}$ to the propositional variables of ϕ such that $\mathcal{I} \models \phi$. Algorithm for satisfiability provides two types of ansewrs when they are called with input Φ. The first answer is that Φ is or is not *unsatisfiable*; the second type of answer is an interpretation $\mathcal{I}$ that satisfies Φ (in case it is satisfiable). Therefore satisfiability algorithms can be used to answer two types of queries: *boolean queries* and *search queries*. Boolean queryies are queries of the form

"is it the case that a certain proposition is true/false?"

Example of these queries are "is John the murderer?", "is the murdered male or female". The answer to a boolean query is yes/no (true/false, 0/1, this is why they are called *boolean*). Search queries are queries of the form

"find an opbject that satisfies a certain proposition.

Examples of this type of queries are: "who is the mardered?". "find a path that connects location a to b", "find a path from a to b that passes through c". The answer to this type of queries is (the description of) a specific object.

2. Formalizing natural language (english) sentences

Natural [1] langauge is one of the most common way in which a problem can be specified; in the section, we discuss how to translate a variety of English statements into the language of propositional logic. From the viewpoint of sentential logic, there are five standard connectives – 'and', 'or', 'if...then', 'if and only if', and 'not'. corresponding to the connectives $\wedge$, $\vee$, $\rightarrow$, $\equiv$ and $\neg$. In addition to these standard connectives, there are in English numerous of other connectives, including 'unless', 'only if', 'neither...nor', among others.

To translate the description of a problem given in natural langauge text into a set of (propositional) logical formula we have to perform three basic steps.

(1) provide a set propositional variables corresponding to the simplest sentences of the text;
(2) compose the propositional variables in formula using the logical connectives in accordance to the natural language connectives that combine the atomic sentences

It is therefore of crucial importance to provide a correct way to tranlate the connectives in natural langauge, such as "not", "and", "although", ...into a suitable combination of the logical connectives $\wedge$, $\vee$,....

[1] The content of this section is a summary of a class by Gary Michael Hardegree, professor of Philosopy

2.1. Conjunction. Conjunction in enclish can be expressed by the connective “and”; there are however a set of alternative conjunctions that can be used. “but”, “yet”, “although”, “though”, “even though”, “moreover”, “furthermore”, , “however”, and “whereas” are all connectives that express some conjunctive information. Although these expressions have different connotations, they are all truthfunctionally equivalent to one another. For instance the sentences

(S1) it is raining, but I am happy
(S2) although it is raining, I am happy
(S3) it is raining, yet I am happy
(S4) it is raining and I am happy

are all true in the situations in which it is raining *and* i’m happy. Therefore they are truth-value equivalent (they capture the same proposition). They are all translated in $R \wedge H$, where H is the propositional variable that represent the proposition “its raining” and H the propositional variable corresponding to the proposition “I’m happy”. This does not mean that they convey the same information. Indeed, for instance (S1) and (S2) convey some contrastive relation between being happy and raining, which is not present in (S4). However, this additional information is not directly related to the truth value of the formula itself. Since propositional logic captures only the truth-functions of connective, such additional information cannot be captured by propositional logic formulas.

Conjunctive information can be provided also in additional form: The template

A and B are C

where A and B are individuals and C is a common name describing a quality, corresponds to the conjunction

A is C *and* B is C

So for instance “Cesare and Caligola are emperors” can be paraphrased in “Cesare is an emperor *and* Caligola is an emperor”. Similarly sentences that respect the pattern

A is B and C

where A is an individual and B and C describe some quality, can be paraphrased in

A is B *and* A is C

For instance “JS Bach is a composer and a musician” can be paraphrased in “JS Bac is a composer and JS Bach is a musician”. There are many other ways to express conjunctive information about the same individual, for instance by using relative pronouns like “who”. Often in this form the “and” is omitted and we have the pattern $AisaBC$, as for instance in

Charles Dickens is an English writer

meaning that Charles Dickens is English and Charles Dickens is a writer.

The pattern “A and B are C” can be used also in case in which C expresses some relation between the individuals A and B. In this case we cannot paraphrased the sentence as a conjunction of “A is C” and “B is C”. Consider for instance the sentence “Pierre and Marie are married”, as the intended meaning, if nothing else is added, is that “Pierre and Marie are married *each-other*”.

2.2. Disjunction. See slides

2.3. Implication. See slides

2.4. Negation. See slides

3. Formalizing constraints on possible worlds

In many situation we are in front of the problem of finding a set of formulas that "must" be true in all the possible configuration of a "world", so that you restrict to consider the interpretations that corresponds to the "possible worlds", and exclude the "impossible" worlds.

EXAMPLE 1.1. *Suppose that a robot can move around a flat that is composed of 25 cells, some of them are occupied by other objects and the robot cannot move into them. This situation is graphically represented in Figure 2 and it is called semantic grid occupancy map.*

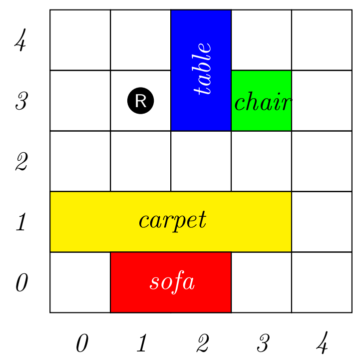


FIGURE 2. An example of occupancy 2D map

To model situation like the one shown in FIgure 2 you have to proceed follow two main steps (as in the case of translation from english)

(1) define the set of "atomic propositions" that are necessary to describe all the configurations (both the possible and the impossible worlds);
(2) write the formula that are true only in the "possible worlds" and are false in the impossible worlds".

EXAMPLE 1.2 (Cont'd). *The key aspect of scenario shown in Figure 2 is the fact that a certain object/robot occupies a cell.*

3.1. Graph coloring. Graph coloring problem is one of the basic problem in graph thery and it has a lot of aplications. In the following we will define the problem, describe it's formulation in propositional logic, and motivate it by means of an application.

DEFINITION 1.1. *A graph G is an ordered pair (V, E), where V is a finite set and $E \subset V \times V$, such that $(v, w) \in E$ implies that $v \neq w$. The set $f\,V$ is called the set of* vertices *and E is called the set of* edges *of G. G is* undirected *if $(v, w) \in V$ then $(w, v) \in V$. If $(v, w) \in E$ we way that v and w are adjacent vertices.*

DEFINITION 1.2. *A graph G is said to be k-colourable if each vertex can be assigned one of k colours so that adjacent vertices get different colours.*

An important problme is the following: given an undirected graph $G = (V, E)$ find the smallest k such that G is k-colourable. One possible method to solve this problem is to cast the problem of checking if a certain graph is k-colorable in a set of propositional formulas so that if they are satisfiable then the graph is k-colorable. In other words we have to define a propositional language and a set of axioms that formalize the graph k-coloring problem of a graph n nodes.

Let us first define the set of propositions that we need to axiomatize the graph k-coloring problem for a graph with n vertices.

- For each node $1 \leq i \leq n$ and color $1 \leq c \leq k$, color_{ic} is a propositional variable that represents the fact that the i-th node is colored with c-th color;
- For each pair of distinct nodes i, j such that $1 \leq i < j \leq n$, edge_{ij} is a propositional variable that represents the fact that there is an edge from node i to node j. Notice that we use only edge_{ij} for $i < j$ and we don't introduce edge_{ij} with $j < i$ since we have that the edges are symmetric and if htere is an edge from i to j there must be also an edge from j to i. This implies that the proposition edge_{ij} would be equivalent to edge_{ji}. Therefore, we need only one of the two.

Let us now introduce a set of axioms that imposes that the graph is crrectly colored.

- we first have to codigy the structure of the graph. for every $(i, j) \in V$ with $i < j$ we add edge_{ij}; furthermore for every $(i, j) \notin E$ with $i < j$ we add $\neg\mathsf{edge}_{ij}$;
- for each $1 \leq i \leq n$ we add the formula
$$\bigvee_{c=1}^{k} \mathsf{color}_{ic}$$
that formalizes the fact that each node has at least one color;
- for each $1 \leq i \leq n$ and $1 \leq c < c' \leq k$, we add the formula
$$\mathsf{color}_{ic} \rightarrow \neg\mathsf{color}_{ic'}$$
, which formalizes that every node has at most 1 color;
- for each $1 \leq i < j \leq n$ and $1 \leq c \leq k$ we add the formula
$$\mathsf{edge}_{ij} \rightarrow \neg(\mathsf{color}_{ic} \wedge \mathsf{color}_{jc})$$
that formalizes that adjacent nodes do not have the same color.

To conclude let's introducing a family of applications that involve avoiding some sort of clash, i.e. where some configuations shouldn't be allowed to happen in a world. A prototypical example is the following

EXAMPLE 1.3. *Suppose that a group of ministers serve on committees as described below:*

Committee	***Members***
Culture, Media & Sport	*Alexander, Burt, Clegg*
Defence	*Clegg, Djanogly, Evers*
Education	*Alexander, Gove*
Food & Rural Affairs	*Djanogly,Featherstone*
Foreign Affairs	*Evers, Hague*
Justice	*Burt, Evers, Gove*
Technology	*Clegg, Featherstone, Hague*

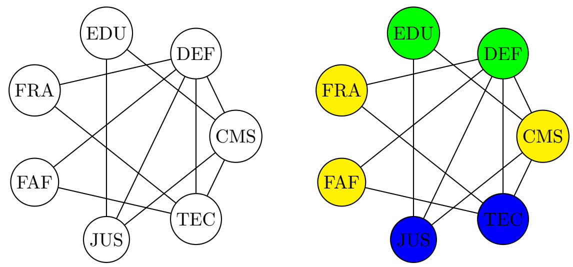


FIGURE 3. The graph at left has vertices labelled with abbreviated committee names and edges given by shared members. The graph at right is isomorphic, but has been redrawn for clarity and given a three-colouring, which turns out to be optimal.

What is the minimum number of time slots needed so that one can schedule meetings of these committees in such a way that the ministers involved have no clashes?

One can turn this into a graph-colouring problem by constructing a graph whose vertices are committees and whose edges connect those that have members in common: such committees can't meet simultaneously, or their shared members will have clashes. A suitable graph appears at left in Figure 3 where, for example, the vertex for the Justice committee (labelled JUS) is connected to the one for the Education committee (EDU) because Gove serves on both. The version of the graph at right in Figure 3 shows a three-colouring and, as the vertices CMS, EDU and JUS form a clique (i.e., totally connected graph). THerefore 3 is this is the smallest number of colours one can possibly use and so the chromatic number of the committee-and-clash graph is 3. This means that we need at least three time slots to schedule the meetings. To see why, think of a vertex's colour as a time slot: none of the vertices that receive the same colour are adjacent, so none of the corresponding committees share any members and thus that whole group of committees can be scheduled to meet at the same time. There are variants of this problem that involve, for example, scheduling exams so that no student will be obliged to be in two places at the same time or constructing sufficiently many storage cabinets in a lab so that chemicals that would react explosively if stored together can be housed separately.

4. Modelling Cardinality constraints

A very common class of constraints that we can encounter in modelling problems are the so called *cardinality constraints*, which impose limit on the number of proositional variables that are true. Let us consider the following simple example

EXAMPLE 1.4 (Crowded room). *Suppose that in a classroom there are k chairs, but there are $n > k$ students that attends the lecture. Suppose that you want to represent the fact that each single student stands or has found a seat. In this situation you have to impose that the maximum numner of students that seat are k. If seat_{ij} stands for the j-th students seats in place j, we have to impose that at most k propositional variables seat_{ij} are true, or equivalently at least $n-k$ propositional variables seat_{ij} are false. If, we want to impose that all the chairs are occupied then* *we need to require that exactly k variables seat_{ij} are true, or equivalently, exactly $n-k$ variables seat_{ij} are false.*

let us see how cardinality constraints can be encoded in propositional formuls.

4.1. At least k. Given a set of boolean variables $X = \{x_1, x_2, \ldots, x_n\}$, the constraint " *at least k propositional variables in X are true* is formalized by

$$\bigvee_{\substack{I\subseteq[n]\\|I|=k}} \bigwedge_{i\in I} x_i \qquad \text{where } [n] = \{1,2,3,\ldots n\} \tag{1}$$

In words, we consider all the possible subsets of $\{x_1, \ldots, x_n\}$ that contains k elements and require that for at least one of such subset (formalized by the outer disjunction) all the variables are true (formalized by the inner conjunction).

Example 1.5. *At least 2 among $X = \{a, b, c, d\}$. The subsets of X that contains two elements are $\{a, b\}, \{a, c\}, \{a, d\}, \{b, c\}, \{b, d\}$, and $\{c, d\}$. We have therefore that formula* (1) *becomes*

$$(a \wedge b) \vee (a \wedge c) \vee (a \wedge d) \vee (b \wedge c) \vee (b \wedge d) \vee (c \wedge d)$$

An alternative formulation of at least k among $X = \{x_1, \ldots, x_n\}$ variables, can be obtained using the following arguments. If I select $n-k+1$ variables in the set X then at least one of them must be true. We can put this in a propositional formula obtaining

$$\bigwedge_{\substack{I\subseteq[n]\\|I|=n-k+1}} \bigvee_{i\in I} x_i \tag{2}$$

For reasons that will be clarified later, the form (2) with ouyter conjunction and inner disjunction is preferrable then the form (1). The two forms anyway are equivalent. (prove it by exercize)

Example 1.6. *The at least 2 among X costaint in the form* (2) *is the following*

$$(a \vee b \vee c) \wedge (a \vee b \vee d) \wedge (b \vee c \vee d)$$

4.2. At most k. The constraint *at most k propositional variables in X are true* can be rephrased as *it is not the case that at least $k+1$ variables in X are true*. Therefore the *at most k* constraint can be formalized as the negation of *at least $k+1$*.

$$\neg\left(\bigvee_{\substack{I\subseteq[n]\\|I|=k+1}} \bigwedge_{i\in I} x_i\right) \qquad \text{which is equivalent to} \qquad \bigwedge_{\substack{I\subseteq[n]\\|I|=k+1}} \bigvee_{i\in I} \neg x_i \tag{3}$$

Example 1.7. *At most 2 among $X = \{a, b, c, d\}$ are true, can be formalized using the right formula of* (4.2) *as:*

$$(\neg a \vee \neg b \vee \neg c) \wedge (\neg a \vee \neg b \vee \neg d) \ \wedge (\neg a \vee \neg c \vee \neg d) \wedge (\neg b \vee \neg c \vee \neg d)$$

4.3. Exactly k. The constraint *exactly k propositional variables in X are true* can be rephrased as the conjunction of *at least k and at most k variables in X are true*. Therefore it can be formalized by the conjunction the of the formulas (2) and (4.2) obtaining

$$\bigwedge_{\substack{I\subseteq[n]\\|I|=n-k+1}} \bigvee_{i\in I} x_i \wedge \bigwedge_{\substack{I\subseteq[n]\\|I|=k+1}} \bigvee_{i\in I} \neg x_i \tag{4}$$

EXAMPLE 1.8 (Exactly k among $X = \{a, b, c, d\}$).

$$(a \vee b \vee c) \wedge (a \vee b \vee d) \wedge (b \vee c \vee d) \wedge (\neg a \vee \neg b \vee \neg c) \wedge$$
$$(\neg a \vee \neg b \vee \neg d) \wedge (\neg a \vee \neg c \vee \neg d) \wedge (\neg b \vee \neg c \vee \neg d)$$

A special case of *exactly k* cardinality constraint is whenpp $k = 1$. The cardinality constraint *exactly* 1 *among the variables in X are true* is formalized using (**??**) by the following formula

$$\bigvee_{i=1}^{n} x_i \wedge \bigwedge_{1\leq i<j\leq n} (\neg x_i \vee \neg x_j) \tag{5}$$

This formula can be read as: at least one variable among X must be true (first part of the conjunction) and for every pair ij with $i < j$ ar least one must be false. Notice that it is enough to consider $i < j$ because the case of $i > j$ will result in the same formula. (since $\vee$ is commutative).

4.4. Efficient representation of *exactly* k. Notice that the length of the formula that codifies the *exactly* k cardinality constaint is $(n-k+1)\binom{n}{k-1} + (k+1)\binom{n}{k+1}$, where $\binom{n}{k} = \frac{n!}{k!(n-k!)}$ is called the binomial coefficient (read "n choose k) is the number of distinct subsets of k elements choosen from a set of n elements. This can be done by encoding a procedure to produce the sets I. We can think that the selection of I is done by the following algorithm

Algorithm 1 Select k elements from X

1: $Y \leftarrow \emptyset$
2: **for** $o \leq i \leq k$ **do**
3:     $y \leftarrow$ select an element from X
4:     $Y \leftarrow I \cup \{y\}$
5:     $X \leftarrow I \setminus \{y\}$
6: **end for**
7: **return** Y

The efficient encoding is obtained by simulating the behavious of the algorithm 1 withing a set of propositional formulas, exploiting only the *exactly 1* cardinality constraint. For every $1 \leq i \leq k$ and for every $1 \leq j \leq n$ add the variable y_{ij} with the following intuitive meaning:

(1) x_j is true if it is selected in some iteration;
(2) At every iteration you select exactly 1 x_j
(3) If x_j is selected at one iteration it cannot be selected in the other iterations

The above statements can be formalized by the following formulas

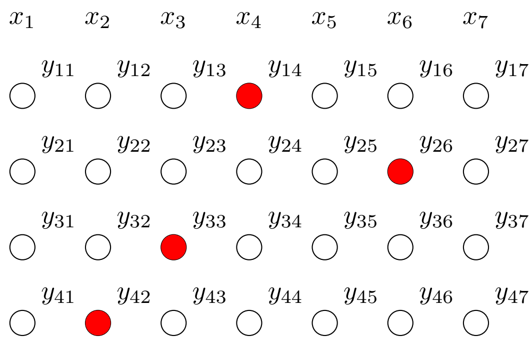


FIGURE 4. Every row represent an iteration, for every raw the selected variables is highighted in red.

(1) x_j is true if it is selected at least in one iteration

$$x_j \equiv \bigvee_{i=1}^{k} y_{ij}$$

(2) exactly 1 x_j is selected at every iteration i:

$$\bigwedge_{i=1}^{k} \left(\bigwedge_{j<j'=1}^{n} \neg(y_{ij} \wedge y_{ij'}) \wedge \bigvee_{j=1}^{n} y_{ij} \right)$$

(3) at most 1 x_j is selected in all the iterations:

$$\bigwedge_{j=1}^{n} \bigwedge_{i<i'=1}^{k} \neg(y_{ij} \wedge y_{i'j})$$

A graphical representation of what is happening is shown in Figure 4

5. exercises

Exercise 1:

Formalize the following english statements

(1) f Davide comes to the party then Bruno and Carlo come too
(2) Carlo comes to the party only if Angelo and Bruno do not come
(3) If Davide comes to the party, then, if Carlo doesn't come then Angelo comes

Solution We define the following propositional variables corresponding to the simple propositions.

A : Angela goes to the party
B : Bruno goes to the party
C : Carlo goes to the party
D : Davide goes to the party

(1) If Davide comes to the party then Bruno and Carlo come too
$$D \to B \wedge C$$
(2) Carlo comes to the party only if Angelo and Bruno do not come
$$C \to \neg A \wedge \neg B$$
(3) If Davide comes to the party, then, if Carlo doesn't come then Angelo comes
$$D \to (\neg C \to A)$$

□

Exercise 2:
Formalize the following english statements

(1) Carlo comes to the party provided that Davide doesn't come, but, if Davide comes, then Bruno doesn't come
(2) A necessary condition for Angelo coming to the party, is that, if Bruno and Carlo aren't coming, Davide comes
(3) Angelo, Bruno and Carlo come to the party if and only if Davide doesn't come, but, if neither Angelo nor Bruno come, then Davide comes only if Carlo comes

Solution We define the following propositional variables corresponding to the simple propositions.

A : Angela goes to the party

B : Bruno goes to the party

C : Carlo goes to the party

D : Davide goes to the party

(1) Carlo comes to the party provided that Davide doesn't come, but, if Davide comes, then Bruno doesn't come
$$(C \to \neg D) \wedge (D \to \neg B)$$
(2) A necessary condition for Angelo coming to the party, is that, if Bruno and Carlo aren't coming, Davide comes
$$A \to (\neg B \wedge \neg C \to D)$$
(3) Angelo, Bruno and Carlo come to the party if and only if Davide doesn't come, but, if neither Angelo nor Bruno come, then Davide comes only if Carlo comes
$$(A \wedge B \wedge C \leftrightarrow \neg D) \wedge (\neg A \wedge \neg B \to (D \to C))$$

□

Exercise 3:
Formalize the following constraing on the binary strings of n bits $(x_1, \ldots, x_n)$

(1) every sequence of k 1's is followed by a sequence of k 0's, with $k \geq 1$;
(2) the k-th digit is the product of the previous two digits (for $k \geq 2$;
(3) the k-th digit is the product of all the previous digits (the product of 0 digits is 1);
(4) the sequence is a palindrom;

(5) With n even: the second half of the string is a permutation of the first half;
(6) the string contains an even number of 0's.

Exercise 4:

Five friends (Abby, Heather, Kevin, Randy and Vijay) have access to an on-line chat room. We know the following are true:

(1) Either K or H or both are chatting.
(2) Either R or V but not both are chatting.
(3) If A is chatting, then R is chatting.
(4) V is chatting if and only if K is chatting.
(5) If H is chatting, then both A and K are chatting.

Represent the above facts in CNF (set of clauses) Notice that there are sentences that correspond to more than one clause.

Solution

(1) $K \vee H$,
(2) $R \vee V$, $\neg R \vee \neg V$,
(3) $\neg A \vee B$,
(4) $\neg V \vee K$, $V \vee \neg K$,
(5) $\neg H \vee A$, $\neg H \vee K$,

□

Exercise 5:

Translate each of the following statements into the language of sentential logic. Use the suggested abbreviations (capitalized words), if provided; otherwise, devise an abbreviation scheme of your own. In each case, write down what atomic statement each letter stands for, making sure it is a complete sentence. Letters should stand for positively stated sentences, not negatively stated ones; for example, the negative sentence 'I am not hungry' should be symbolized as '¬H' using 'H' to stand for 'I am hungry'.

(1) Although it is RAINING, I plan to go JOGGING this afternoon.
(2) It is not RAINING, but it is still too WET to play.
(3) JAY and KAY are Sophomores.
(4) It is DINNER time, but I am not HUNGRY.
(5) Although I am TIRED, I am not QUITTING.
(6) Jay and Kay are roommates, but they hate one another.
(7) Jay and Kay are Republicans, but they both hate Nixon.
(8) KEEP trying, and the answer will APPEAR.
(9) GIVE him an inch, and he will TAKE a mile.
(10) Either I am CRAZY or I just SAW a flying saucer.
(11) Either Jones is a FOOL or he is DISHONEST.
(12) JAY and KAY won't both be present at graduation.
(13) JAY will win, or KAY will win, but not both.
(14) Either it is RAINING, or it is SUNNY and COLD.
(15) It is RAINING or OVERCAST, but in any case it is not SUNNY.
(16) If JONES is honest, then so is SMITH.
(17) If JONES isn't a crook, then neither is SMITH.

(18) Provided that I CONCENTRATE, I will not FAIL.
(19) I will GRADUATE, provided I pass both LOGIC and HISTORY.
(20) I will not GRADUATE if I don't pass both LOGIC and HISTORY.
(21) Neither JAY nor KAY is able to attend the meeting.
(22) Although I have been here a LONG time, I am neither TIRED nor BORED.
(23) I will GRADUATE this semester only if I PASS intro logic.
(24) KAY will attend the party only if JAY does not.
(25) I will SUCCEED only if I WORK hard and take RISKS.
(26) I will go to the BEACH this weekend, unless I am SICK.
(27) Unless I GOOF off, I will not FAIL intro logic.
(28) I won't GRADUATE unless I pass LOGIC and HISTORY.
(29) In order to ACE intro logic, it is sufficient to get a HUNDRED on every exam.
(30) In order to PASS, it is necessary to average at least FIFTY.
(31) In order to become a PHYSICIAN, it is necessary to RECEIVE an M.D. and do an INTERNSHIP.
(32) In order to PASS, it is both necessary and sufficient to average at least FIFTY.
(33) Getting a HUNDRED on every exam is sufficient, but not necessary, for ACING intro logic.
(34) TAKING all the exams is necessary, but not sufficient, for ACING intro logic.
(35) In order to get into MEDICAL school, it is necessary but not sufficient to have GOOD grades and take the ADMISSIONS exam.
(36) In order to be a BACHELOR it is both necessary and sufficient to be ELIGIBLE but not MARRIED.
(37) In order to be ARRESTED, it is sufficient but not necessary to COMMIT a crime and GET caught.
(38) If it is RAINING, I will play BASKETBALL; otherwise, I will go JOGGING.
(39) If both JAY and KAY are home this weekend, we will go to the BEACH; otherwise, we will STAY home.
(40) JONES will win the championship unless he gets INJURED, in which case SMITH will win.
(41) We will have DINNER and attend the CONCERT, provided that JAY and KAY are home this weekend.
(42) If neither JAY nor KAY can make it, we should either POSTPONE or CANCEL the trip.
(43) Both Jay and Kay will go to the beach this weekend, provided that neither of them is sick.
(44) I'm damned if I do, and I'm damned if I don't.
(45) If I STUDY too hard I will not ENJOY college, but at the same time I will not ENJOY college if I FLUNK out.
(46) If you NEED a thing, you will have THROWN it away, and if you THROW a thing away, you will NEED it.
(47) If you WORK hard only if you are THREATENED, then you will not SUCCEED.

(48) If I do not STUDY, then I will not PASS unless the prof ACCEPTS bribes.
(49) Provided that the prof doesn't HATE me, I will PASS if I STUDY.
(50) Unless logic is very DIFFICULT, I will PASS provided I CONCENTRATE.
(51) Unless logic is EASY, I will PASS only if I STUDY.
(52) Provided that you are INTELLIGENT, you will FAIL only if you GOOF off.
(53) If you do not PAY, Jones will KILL you unless you ESCAPE.
(54) If he CATCHES you, Jones will KILL you unless you PAY.
(55) Provided that he has made a BET, Jones is HAPPY if and only if his horse WINS.
(56) If neither JAY nor KAY comes home this weekend, we shall not stay HOME unless we are SICK.
(57) If you MAKE an appointment and do not KEEP it, then I shall be ANGRY unless you have a good EXCUSE.
(58) If I am not FEELING well this weekend, I will not GO out unless it is WARM and SUNNY.
(59) If JAY will go only if KAY goes, then we will CANCEL the trip unless KAY goes.
(60) If KAY will come to the party only if JAY does not come, then provided we WANT Kay to come we should DISSUADE Jay from coming.
(61) If KAY will go only if JAY does not go, then either we will CANCEL the trip or we will not INVITE Jay.
(62) If JAY will go only if KAY goes, then we will CANCEL the trip unless KAY goes.
(63) If you CONCENTRATE only if you are INSPIRED, then you will not SUCCEED unless you are INSPIRED.
(64) If you are HAPPY only if you are DRUNK, then unless you are DRUNK you are not HAPPY.
(65) In order to be ADMITTED to law school, it is necessary to have GOOD grades, unless your family makes a large CONTRIBUTION to the law school.
(66) I am HAPPY only if my assistant is COMPETENT, but if my assistant is COMPETENT, then he/she is TRANSFERRED to a better job and I am not HAPPY.
(67) If you do not CONCENTRATE well unless you are ALERT, then you will FLY an airplane only if you are SOBER; provided that you are not a MANIAC.
(68) If you do not CONCENTRATE well unless you are ALERT, then provided that you are not a MANIAC you will FLY an airplane only if you are SOBER.
(69) If you CONCENTRATE well only if you are ALERT, then provided that you are WISE you will not FLY an airplane unless you are SOBER.
(70) If you CONCENTRATE only if you are THREATENED, then you will not PASS unless you are THREATENED – provided that CONCENTRATING is a necessary condition for PASSING.
(71) If neither JAY nor KAY is home this weekend, we will go to the BEACH; otherwise, we will STAY home.

Solution

(1) $R \wedge J$
(2) $\neg R \wedge W$
(3) $J \wedge K$
(4) $D \wedge \neg H$
(5) $T \wedge \neg Q$
(6) $R \wedge (J \wedge K)$
R: Jay and Kay are roommates
J: Jay hates Kay
K: Kay hates Jay
(7) $(J \wedge K) \wedge (H \wedge N)$
J: Jay is a Republican; K: Kay is a Republican
H: Jay hates Nixon; N: Kay hates Nixon
(8) $K \rightarrow A$
(9) $G \rightarrow T$
(10) $C \vee S$
(11) $F \vee D$
(12) $\neg(J \wedge K)$
(13) $(J \vee K) \wedge \neg(J \wedge K)$
(14) $R \vee (S \wedge C)$
(15) $(R \vee O) \wedge \neg S$
(16) $J \rightarrow S$
(17) $\neg J \rightarrow \neg S$
(18) $C \rightarrow \neg F$
(19) $(L \wedge H) \rightarrow G$
(20) $\neg(L \wedge H) \rightarrow \neg G$
(21) $\neg J \wedge \neg K[or : \neg(J \vee K)]$
(22) $L \wedge (\neg T \wedge \neg B)[or : L \wedge \neg(T \vee B)]$
(23) $\neg P \rightarrow \neg G$
(24) $\neg\neg J \rightarrow \neg K[J \rightarrow \neg K]$
(25) $\neg(W \wedge R) \rightarrow \neg S$
(26) $\neg S \rightarrow B$
(27) $\neg G \rightarrow \neg F$
(28) $\neg(L \wedge H) \rightarrow \neg G$
(29) $H \rightarrow A$
(30) $\neg F \rightarrow \neg P$
(31) $\neg(R \wedge I) \rightarrow \neg P$
(32) $(\neg F \rightarrow \neg P) \wedge (F \rightarrow P)$
(33) $(H \rightarrow A) \wedge \neg(\neg H \rightarrow \neg A)$
(34) $(\neg T \rightarrow \neg A) \wedge \neg(T \rightarrow A)$
(35) $(\neg(G \wedge A) \rightarrow \neg M) \wedge \neg[(G \wedge A) \rightarrow M]$
(36) $(\neg(E \wedge \neg M) \rightarrow \neg B) \wedge [(E \wedge \neg M) \rightarrow B]$
(37) $((C \wedge G) \rightarrow A) \wedge \neg[\neg(C \wedge G) \rightarrow \neg A]$
(38) $(R \rightarrow B) \wedge (\neg R \rightarrow J)$
(39) $((J \wedge K) \rightarrow B) \wedge [\neg(J \wedge K) \rightarrow S]$
(40) $(\neg I \rightarrow J) \wedge (I \rightarrow S)$
(41) $(J \wedge K) \rightarrow (D \wedge C)$
(42) $(\neg J \wedge \neg K) \rightarrow (P \vee C)$

(43) $(\neg S \wedge \neg T) \to (J \wedge K)$
S: Jay is sick; T: Kay is sick;
J: Jay will go to the beach; K: Kay will go to the beach.
(44) $(A \to D) \wedge (\neg A \to D)$
A: I do (what ever action is being discussed);
D: I am damned.
(45) $(S \to \neg E) \wedge (F \to \neg E)$
(46) $(N \to T) \wedge (T \to N)$
(47) $(\neg T \to \neg W) \to \neg S$
(48) $\neg S \to (\neg A \to \neg P)$
(49) $\neg H \to (S \to P)$
(50) $\neg D \to (C \to P)$
(51) $\neg E \to (\neg S \to \neg P)$
(52) $I \to (\neg G \to \neg F)$
(53) $\neg P \to (\neg E \to K)$
(54) $C \to (\neg P \to K)$
(55) $B \to [(W \to H) \wedge (\neg W \to \neg H)]$
(56) $(\neg J \wedge \neg K) \to (\neg S \to \neg H)$
(57) $(M \wedge \neg K) \to (\neg E \to A)$
(58) $\neg F \to [\neg(W \wedge S) \to \neg G]$
(59) $(\neg K \to \neg J) \to (\neg K \to C)$
(60) $(\neg\neg J \to \neg K) \to (W \to D)$
(61) $(\neg J \to \neg K) \to (C \vee \neg I)$
(62) $(\neg K \to \neg J) \to (\neg K \to C)$
(63) $(\neg I \to \neg C) \to (\neg I \to \neg S)$
(64) $(\neg D \to \neg H) \to (\neg D \to \neg H)$
(65) $\neg C \to (\neg G \to \neg A)$
(66) $(\neg C \to \neg H) \wedge (C \to [T \wedge \neg H])$
(67) $\neg M \to [(\neg A \to \neg C) \to (\neg S \to \neg F)]$
(68) $(\neg A \to \neg C) \to [\neg M \to (\neg S \to \neg F)]$
(69) $(\neg A \to \neg C) \to [W \to (\neg S \to \neg F)]$
(70) $(\neg C \to \neg P) \to [(\neg T \to \neg C) \to (\neg T \to \neg P)]$
(71) $((\neg J \wedge \neg K) \to B) \wedge [\neg(\neg J \wedge \neg K) \to S]$

$\square$

Exercise 6:

Translate each of the following statements into propositional logic. You have to specify the propositional variables you are using and their corresponding proposition in english (e.g., the translation of "I'm happy if Bob is at the party" is $B \to H$, where H stand for "I am happy", B stands for "Bob is at the party")

(1) If Ann will go to the party only if she has not to work, then, if John helps Ann in finishing the work Ann will come to the party;
(2) If neither Ann nor Bob is home this weekend, then we will go to the beach otherwise, we will stay home.
(3) I will be happy if at the end of the semester I will pass at least 2 exams among Deep Learning, Databases, and Probability.

Solution

(1)

$$
\begin{aligned}
A &= \text{Ann will go to the party} \\
W &= \text{Ann has to work} \\
H &= \text{John Helps Ann in finishing the work}
\end{aligned}
$$

$$(A \to \neg W) \to (H \to A)$$

A very common error is to translate "Ann will go to the party only if she has not to work" with the formula $\neg W \to A$. However this encodes the proposition "If Ann has not to work then she will go to the parti", Notice that the fact that Ann has not to work and She will go to the cinema, is consistent with the proposition "Ann will go to the party only if she has not to work".

(2)

$$
\begin{aligned}
A &= \text{Ann stays at homw this weekend} \\
B &= \text{Bob stays at homw this weekend} \\
G &= \text{we will go to the beach} \\
H &= \text{we will stay home}
\end{aligned}
$$

$$(\neg A \land \neg B \to G) \land (\neg(\neg A \land \neg B) \to H)$$

which is equivalent to

$$(\neg A \land \neg B \to G) \land (A \lor B \to H)$$

(3)

$$
\begin{aligned}
H &= \text{I will be happy} \\
D &= \text{I will pass Deep Learning} \\
B &= \text{I will pass Data Bases} \\
P &= \text{I will pass Probability}
\end{aligned}
$$

$$(D \land B) \lor (D \land F) \lor (B \land P) \to H$$

Some student proposed the more complex but equivalent translation.

$$(D \land B \land \neg P) \lor (D \land P \land \neg B) \lor (B \land P \land \neg D) \lor (D \land B \land P) \to H$$

This is ok, but in general it is a good practice to use the *simplest* formalization. □

5.1. Problem Solving with Propositional Logic. Exercise 7:

Determine the validity or invalidity of the following argument:

If Alice is elected class-president, then either Betty is elected vice-president, or Carol is elected treasurer but not both. Betty is elected vice-president. Therefore if Alice is elected class-president, then Carol is not elected treasurer. **Solution**

We use the following propositional variables for each atomic sentence.

A	-	Alice is elected class-president
B	-	Betty is elected vice - president
C	-	Carol is elected treasurer

The translation of each complex sentence of the argument is the following:

$A \to ((B \wedge \neg C) \vee (\neg B \vee C))$	If Alice is elected class-president, then either Betty is elected vice- president, or Carol is elected treasurer but not both.
B	Betty is elected vice-president
$A \to \neg C$	if Alice is elected class-president, then Carol is not elected treasurer.

The logical consequence corresponding to the argument is

$$A \to ((B \wedge \neg C) \vee (\neg B \vee C)), B \models A \to \neg C \tag{6}$$

In order to see if (6) holds, we can try to find an intepretation $\mathcal{I}$ for A,B and C that satisfies the premises and falsifies the conclusion Looking at the truth table of we have:

A	B	C	A	→	((B	∧	¬	C)	∨	(¬	B	∧	C))	B	A	→	¬	C
T	T	T	T	F	T	F	F	T	F	F	T	F	T	T	T	F	F	T
T	T	F	T	T	T	T	T	F	T	F	T	F	F	T	T	T	T	F
T	F	T	T	T	F	F	F	T	T	T	F	T	T	F	T	F	F	T
T	F	F	T	F	F	F	T	F	F	T	F	F	F	F	T	T	T	F
F	T	T	F	T	T	F	F	T	F	F	T	F	T	T	F	T	F	T
F	T	F	F	T	T	T	T	F	T	F	T	F	F	T	F	T	T	F
F	F	T	F	T	F	F	F	T	T	T	F	T	T	F	F	T	F	T
F	F	F	F	T	F	F	T	F	F	T	F	F	F	F	F	T	T	F

From the above truth table one can see that every time the two premises are true (highlighted in red background) the consequence is also true. This means that the logical argument is valid. □

Exercise 8:

Test the validity of the following arguments.

> James is either a policeman or a footballer (but not both). If he is a policeman, then he has big feet. James has not got big feet so he is a footballer.

Solution We use the following propositional variables for each atomic sentence.

p	-	James is a policeman
f	-	James is a footballer
b	-	James has big feet.

Then the argument is formalized by the following logical consequence:

$$(p \vee f) \wedge (\neg p \vee \neg f) \wedge (p \to b) \wedge \neg b \models f$$

Let us check the validity of the first argument by building a truth table for

b	f	p	p	∨	f	¬	p	∨	¬	f	p	→	b	¬	b	f
T	T	T	T	T	T	F	T	F	F	T	T	T	T	F	T	T
T	T	F	F	T	T	T	F	T	F	T	F	T	T	F	T	T
T	F	T	T	T	F	F	T	T	T	F	T	T	T	F	T	F
T	F	F	F	F	F	T	F	T	T	F	F	T	T	F	T	F
F	T	T	T	T	T	F	T	F	F	T	T	F	F	T	F	T
F	T	F	F	T	T	T	F	T	F	T	F	T	F	T	F	T
F	F	T	T	T	F	F	T	T	T	F	T	F	F	T	F	F
F	F	F	F	F	F	T	F	T	T	F	F	T	F	T	F	F

From the above truth table one can see that every time the two premises are true (highlighted in red background) the consequence is also true. This means that the logical argument is valid. □

Exercise 9:

Let p stand for the proposition "I bought a lottery ticket" and q for "I won the jackpot". Express the following as natural English sentences:

(1) $\neg p$
(2) $p \vee q$
(3) $p \wedge q$
(4) $p \to q$
(5) $\neg p \to \neg q$
(6) $\neg p \vee (p \wedge q)$

Exercise 10:

Formalise the following in terms of the propositional variables r, b, and w, first expressing in english that proposition they are intended to represent.

(1) Berries are ripe along the path, but rabbits have not been seen in the area.
(2) Rabbits have not been seen in the area, and walking on the path is safe, but berries are ripe along the path.
(3) If berries are ripe along the path, then walking is safe if and only if rabbits have not been seen in the area.
(4) It is not safe to walk along the path, but rabbits have not been seen in the area and the berries along the path are ripe.
(5) For walking on the path to be safe, it is necessary but not sufficient that berries not be ripe along the path and for rabbits not to have been seen in the area.
(6) Walking is not safe on the path whenever rabbits have been seen in the area and berries are ripe along the path.

Exercise 11:

Formalise these statements and determine (with truth tables or otherwise) whether they are consistent (i.e. if there are some truth assignment to the propositional variables that make all them true):

> *The system is in a multiuser state if and only if it is operating normally. If the system is operating normally, the kernel is functioning. Either the kernel is not functioning or the system is in interrupt mode. If the system is not in multiuser state, then it is in interrupt mode. The system is not in interrupt mode.*

Solution Let us find the propositions in the text and declare the propositional variables to represent them

> *The system is in a multiuser state if and only if it is operating normally.*
> *If the system is operating normally, the kernel is functioning.*
> *Either the kernel is not functioning or the system is in interrupt mode.*
> *If the system is not in multiuser state, then it is in interrupt mode.*
> *The system is not in interrupt mode.*

$$
\begin{aligned}
M &= \text{The system is in a multiuser state}\\
N &= \text{The system is operating normally}\\
K &= \text{Kernel is functioning}\\
I &= \text{The system is in interrupt mode}
\end{aligned}
$$

Let us now translate the above sentences in propositional logic:

- The system is in a multiuser state if and only if it is operating normally.

$$M \leftrightarrow N$$

- If the system is operating normally, the kernel is functioning.

$$N \rightarrow K$$

- Either the kernel is not functioning or the system is in interrupt mode.

$$\neg K \vee I$$

- If the system is not in multiuser state, then it is in interrupt mode.

$$\neg M \rightarrow I$$

- The system is not in interrupt mode.

$$\neg I$$

I	K	M	N	M	↔	N	N	→	K	¬	K	∨	I	¬	M	→	I	¬	I
T	T	T	T	T	T	T	T	T	T	F	T	T	T	F	T	T	T	F	T
T	T	T	F	T	F	F	F	T	T	F	T	T	T	F	T	T	T	F	T
T	T	F	T	F	F	T	T	T	T	F	T	T	T	T	F	T	T	F	T
T	T	F	F	F	T	F	F	T	T	F	T	T	T	T	F	T	T	F	T
T	F	T	T	T	T	T	T	F	F	T	F	T	T	F	T	T	T	F	T
T	F	T	F	T	F	F	F	T	F	T	F	T	T	F	T	T	T	F	T
T	F	F	T	F	F	T	T	F	F	T	F	T	T	T	F	T	T	F	T
T	F	F	F	F	T	F	F	T	F	T	F	T	T	T	F	T	T	F	T
F	T	T	T	T	T	T	T	T	T	F	T	F	F	F	T	T	F	T	F
F	T	T	F	T	F	F	F	T	T	F	T	F	F	F	T	T	F	T	F
F	T	F	T	F	F	T	T	T	T	F	T	F	F	T	F	F	F	T	F
F	T	F	F	F	T	F	F	T	T	F	T	F	F	T	F	F	F	T	F
F	F	T	T	T	T	T	T	F	F	T	F	T	F	F	T	T	F	T	F
F	F	T	F	T	F	F	F	T	F	T	F	T	F	F	T	T	F	T	F
F	F	F	T	F	F	T	T	F	F	T	F	T	F	T	F	F	F	T	F
F	F	F	F	F	T	F	F	T	F	T	F	T	F	T	F	F	F	T	F

Notice that there is no assignment that satisfy all the five formulas. Thererfore the set of formulas is inconsistent.

A shorter way to prove incinsistency would have been by propositional resolution. We first have to transform the four formulas in clauses, obtaining:

$$\{\neg M, N\}, \{\neg N, M\}, \{\neg N, K\}, \{\neg K, I\}, \{M, I\}, \{\neg I\}$$

By repeated applications of unit-propagation we can obtaine the empty clause

$$
\begin{array}{rl}
\multicolumn{2}{l}{\{\neg M, N\}, \{\neg N, M\}, \{\neg N, K\}, \{\neg K, I\}, \{M, I\}, \{\neg I\}} \\
\{\neg M, N\}, \{\neg N, M\}, \{\neg N, K\}, \{\neg K\}, \{M\} & \text{By unit propagation on } \{\neg I\} \\
\{N\}, \{\neg N, K\}, \{\neg K\} & \text{By unit propagation on } \{M\} \\
\{K\}, \{\neg K\} & \text{By unit propagation on } \{N\} \\
\{\} & \text{By unit propagation on } \{K\}
\end{array}
$$

Since we derive the empty clauses the initial set of clauses must be inconsistent. □

Exercise 12:

Five friends (Abby, Heather, Kevin, Randy and Vijay) have access to an on-line chat room. We know the following are true:

(1) Either K or H or both are chatting.
(2) Either R or V but not both are chatting.
(3) If A is chatting, then R is chatting.
(4) V is chatting if and only if K is chatting.
(5) If H is chatting, then both A and K are chatting.

Represent the above facts in CNF (set of clauses) Notice that there are sentences that correspond to more than one clause.

Solution

(1) $K \vee H$,
(2) $R \vee V$, $\neg R \vee \neg V$,
(3) $\neg A \vee B$,
(4) $\neg V \vee K$, $V \vee \neg K$,
(5) $\neg H \vee A$, $\neg H \vee K$,

□

Exercise 13:

Imagine that a logician puts four cards on the table in front of you. Each card has a number on one side and a letter on the other. On the uppermost faces, you can see E,K,4,and 7. He claims that if a card has a vowel on one side, then it has an even number on the other. Which cards do you have to turn over to check this? Explain why.

Solution To check that the logician states the truth we can check it for every single card The statemen is an implication

$$\textit{wowel} \rightarrow \text{even}$$

which is true if either theq premise is false or the conclusion is true. In the first card we see a vawel, therefore the premise is true (it is not false) and therefore we have to turn the card to check if the back is even. The second card shows a consonanto, which is not a vawel, which guarantees that $\textit{wowel} \rightarrow$ even. independently from what there is on the back. The third card shows a even number, this implies tha the conseuqence of $\textit{wowel} \rightarrow$ even is true,. This guarantee that $\textit{wowel} \rightarrow$ even holds without any further information. Finally, the forth card shows an odd number, therefore the conclusion of $\textit{wowel} \rightarrow$ even is false, to chek that the implication is true we have to see if the premise is also falsw, which means that we have to turn the card. □

Solution(alternative) To check if the formula *vowel* $\rightarrow$ *even* is true you have to check that in all the possible models, given what you know, the formula is true. If there are models in which the formula is false, you have to turn some card in order to acquire some knowledge that will exclude such a model.

Let V_i and E_i for $i \in \{1, 2, 3, 4\}$ be the propositional variables that express the fact that the i-th card has a vowel in one side and an even number on the other side.

The statement of the logician can be formalized by the formula:

$$\bigwedge_{i=1}^{4} V_i \rightarrow E_i$$

What you know on the basis on what you see on the table is

$$V_1 \wedge \neg V_2 \wedge E_3 \wedge \neg E_4 \tag{7}$$

The models that satisfies your knowledge i,e, (7) are:

V_1	E_1	V_2	E_2	V_3	E_3	V_4	E_4	$V_1 \rightarrow E_1$	$V_2 \rightarrow E_2$	$V_3 \rightarrow E_3$	$V_4 \rightarrow E_4$
1	0	0	0	0	1	0	0	0	1	1	1
1	0	0	0	0	1	1	0	0	1	1	0
1	0	0	0	1	1	0	0	0	1	1	1
1	0	0	0	1	1	1	0	0	1	1	0
1	0	0	1	0	1	0	0	0	1	1	1
1	0	0	1	0	1	1	0	0	1	1	0
1	0	0	1	1	1	0	0	0	1	1	1
1	0	0	1	1	1	1	0	0	1	1	0
1	1	0	0	0	1	0	0	1	1	1	1
1	1	0	0	0	1	1	0	1	1	1	0
1	1	0	0	1	1	0	0	1	1	1	1
1	1	0	0	1	1	1	0	1	1	1	0
1	1	0	1	0	1	0	0	1	1	1	1
1	1	0	1	0	1	1	0	1	1	1	0
1	1	0	1	1	1	0	0	1	1	1	1
1	1	0	1	1	1	1	0	1	1	1	0

From the truth table above, one can see that there are models that satisfies (7) (the knowledge we have) in which $V_1 \rightarrow E_1$ and $V_4 \rightarrow E_4$ are both true and false. This means that we are uncertain about the truth value of these formulas. since neither (7) $\models V_1 \rightarrow E_1$ nor (7) $\models \neg(V_1 \rightarrow E_1)$ (and the same for $V_4 \rightarrow E_4$).

If we would know the truth value of E_1 and V_4, we would be certain about the truth value of $V_1 \rightarrow E_1$ and $V_4 \rightarrow E_4$. Indeed notice that

- $V_1 \rightarrow E_1$ is true if and only if E_1 is true;
- $V_4 \rightarrow E_4$ is true if and only if V_4 is false.

Therefore to know if the logician said the truth we have to turn the first and the forth card.

An intuitive reasoning is the following. A statement $V_i \rightarrow E_i$ is true whenever the premise is true then the conclusion is also true, or equivalently that whenever the conclusion is false, also the premise is false. Therefore, to check that the logician says the truth, you have to turn the E (vowel is true) and the 7 (even is false) $\square$

Exercise 14:

Formalize the following puzzle in a set Γ of propositional formulas and show that the answer is a formula ϕ that logically follows from Γ.

> A very special island is inhabited only by knights and knaves. Knights always tell the truth, and knaves always lie. You meet two inhabitants: Marge and Zoey.
> (1) Marge says, "Zoey and I are both knights or both knaves."
> (2) Zoey claims, "Marge and I are the same."
> Can you determine who is a knight and who is a knave?

Solution We first define the propositional variables to represent the proposition we need to formalize the puzzle

M	Marge is a knight
Z	Zoey is a knight

Clearly since in the island if one person is not a knight it must be a knave, we have that $\neg M$ means that Marge is a knave and $\neg Z$ means that Zoey is a knave. Now we can formalize the knowledge encoded in the two sentences. We don't know if the two statements said by Marge and Zoey are true or false, but we know that if the speaker is a knight the sentence is true, and if the speaker is a knave the sentence must be false. So from the two sentences we can get the following facts:

$$M \rightarrow (M \wedge Z) \vee (\neg M \wedge \neg Z)$$
$$\neg M \rightarrow \neg((M \wedge Z) \vee (\neg M \wedge \neg Z))$$
$$Z \rightarrow (M \leftrightarrow Z)$$
$$\neg Z \rightarrow \neg(M \leftrightarrow Z)$$

We can simplify the above four formulas by reducing them in CNF

$$\{\neg M, Z\} \tag{8}$$
$$\{M, Z\} \tag{9}$$
$$\{\neg Z, M\} \tag{10}$$
$$\{Z, M\} \tag{11}$$

By applying the resolution rule to (8) and (9) we derive the clause $\{Z\}$, and then by unit propagation on (10) we obtain the clause $\{M\}$. WHich implies that both Marge and Zoey are knights. $\square$

Exercise 15:

Consider the set $S = \{0, 1, 2, 3\}^8$ of all the strings of length equal to 8 composed of $0, 1, 2, 3$, and the following subset $T \subset S$:

$$T = \left\{ \begin{array}{c} 00000000, 00000011, 00001111, 00001122, \\ 00111111, 00111122, 00112222, 00112233 \end{array} \right\}$$

(1) Define a set of propositional variable $\mathcal{P}$ such that every truth assigment $\mathcal{I}$ (interpretation) of $\mathcal{P}$ is one-to-one mapped into a string $s(\mathcal{I}) \in S$;
(2) What is the cardinality of $\mathcal{I}$ and the cardinality of S?
(3) Using the set of propositional variables in $\mathcal{P}$, write a formula ϕ such that $\mathcal{I} \models \phi$ if and only if $m(\mathcal{I}) \in T$.

Solution

(1) We have various possibilities here. A first alternative is to add the set of propositional variables:

$$p_{ij} \qquad \text{for } 1 \leq i \leq 8 \text{ and } 0 \leq j \leq 3$$

and interpret p_{ij} as "the digit j is in position i". and add the axioms

$$\bigwedge_{i=1}^{8} \bigvee_{j=0}^{3} p_{ij} \qquad \text{At every place } i \text{ there is at least one digit} \tag{12}$$

$$\bigwedge_{i=1}^{8} \bigwedge_{j=0}^{3} \bigwedge_{k=j+1}^{3} \neg(p_{ij} \wedge p_{ik}) \qquad \text{At every place } i \text{ there is at most one digit} \tag{13}$$

In this case the set $\mathbb{I}$ of interpretations must be restricted to the ones that satisfies (12) and (13).

Alternatively, we can introduce less propositions

$$p_i, q_i \qquad \text{for } 1 \leq i \leq 8$$

and interpred p_i as "in position i that is a digit j such that $j \div 2 = 1$ q_i as "in position i that is a digit j such that $i\%2 = 1$. Notice that with this interpretation we have that we can represent the fact that in position i there is a digit j as follows:

$\neg p_i \wedge \neg q_i$	In position i there is a 0
$\neg p_i \wedge q_i$	In position i there is a 1
$p_i \wedge \neg q_i$	In position i there is a 2
$p_i \wedge q_i$	In position i there is a 3

Since the above 4 formulas are exclusive (i.e. one and only one can be true in any interpretation) we don't need to add any additional axiom. In this case all the interpretations can be considered.

(2) Since we have to define a one to one mapping between $\mathbb{I}$ and S their cardinality must be the same, i.e, 4^8.

(3) If we use the first language we can represent the set T with the formula

$$\bigvee_{\boldsymbol{t} \in T} \bigwedge_{i=1}^{8} p_{it_i}$$

where t_i is the i-th digit of $\boldsymbol{t}$. If we use the second langauge we can represent the set T with the formula

$$\bigvee_{\boldsymbol{t} \in T} \bigwedge_{i=1} \circ_{t_i \div 2} p_i \wedge \circ_{t_i \% 2} q_i$$

where $\circ_0$ is equal to the emtpy string $\circ_1$ is equal to $\neg$.

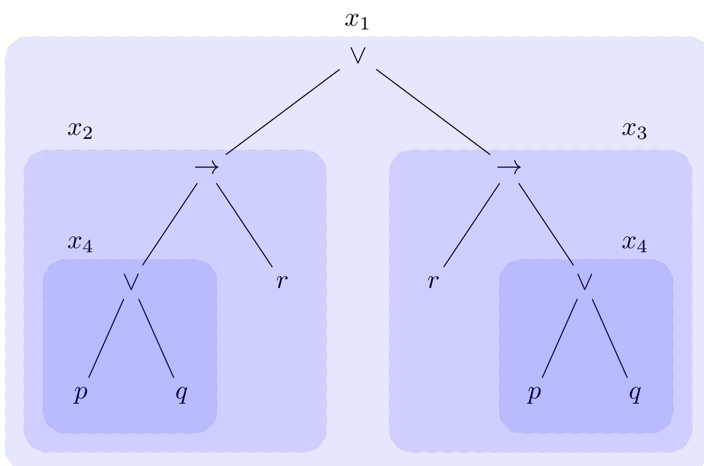
□

Exercise 16:

Provide the Tseitin's Transformation of the following formula:

$$((p \vee q) \rightarrow r) \vee (r \rightarrow (p \vee q))$$

Solution The Tseitin's tranformation introduces one new propositional variable for all the non atomic subformula of the original formula. The set of non atomic subformulas of $((p \vee q) \to r) \vee (r \to (p \vee q))$ (including itself) with the associated new propositional variables are:



$$
\begin{array}{lr}
x_1 & ((p \vee q) \to r) \vee (r \to (p \vee q)) \\
x_2 & ((p \vee q) \to r) \\
x_3 & (r \to (p \vee q)) \\
x_4 & (p \vee q)
\end{array}
$$

We then define the follwong clauses

$$
\begin{aligned}
x_1 &\equiv x_2 \vee x_3 \\
x_2 &\equiv x_4 \to r \\
x_3 &\equiv r \to x_4 \\
x_4 &\equiv p \vee q
\end{aligned}
$$

and tranform them in clausal form

$$
\begin{array}{l}
(\neg x_1, x_2, x_3), (\neg x_2, x_1), (\neg x_3, x_1) \\
(\neg x_2, \neg x_4, r), (x_4, x_2), (\neg r, x_2) \\
(\neg x_3, \neg r, x_4), (r, x_3), (\neg x_4, x_2) \\
(\neg x_4, p, q), (\neg p, x_4), (\neg q, x_4)
\end{array}
$$

$\square$

Exercise 17:

Alice (F), Bob (M), Craig (M), and Donna (F) are four friends that want to take a tour with their motorbikes. Everybody can decide either to go with somebody else or to ride a bike alone. Use propositional logic to formulate the problem of finding all the possible configurations for the tour.

Solution We introduce the following propositional variables

- $ride_X$ for X drives a bike for $X \in \{A, B, C, D\}$

- $pass_{XY}$ for X give a pass to Y, for $X, Y \in \{A, B, C, D\}$ nd $X \neq Y$.

We add the following axioms:

$$\bigwedge_X \left(ride_X \vee \bigvee_{Y \neq X} pass_{YX} \right) \quad \text{Each friend, either rides or get a lift}$$

$$\bigwedge_{X \neq Y} (pass_{XY} \rightarrow ride_X) \quad \text{If } Y \text{ get a lift from } X \text{ then } X \text{ rides the bike}$$

$$\bigwedge_{X \neq Y} (rides_X \rightarrow \neg pass_{YX}) \quad \text{If } X \text{ rides he/she does not get a lift}$$

$$\bigwedge_{X \neq X' \neq Y} (pass_{XY} \rightarrow \neg pass_{X'Y}) \quad Y \text{ can get a lift from at most one person}$$

$$\bigwedge_{X \neq Y \neq Y'} (pass_{XY} \rightarrow \neg pass_{XY'}) \quad X \text{ can give a lift to at most one person}$$

A python program that computes all the solutions is the following

Listing 1.1. Pysat code that compute all the configurations

```
from sympy import *

Friends = ["A","B","C","D"]

Rides = {X:Symbol(f"Rides_{X}") for X in Friends}
Pass = {(X,Y):Symbol(f"Pass_{X}{Y}") for X in Friends for Y in Friends if X != Y}

phi1 = And(*[Rides[X] | Or(*[Pass[Y,X] for Y in Friends if Y != X]) for X in Friends])
phi2 = And(*[Pass[X,Y] >> Rides[X] for X in Friends for Y in Friends if X != Y])
phi3 = And(*[Pass[X,Y] >> ~Rides[Y] for X in Friends for Y in Friends if X != Y])
phi4 = And(*[Pass[X,Y] >> ~Pass[X1,Y] for X in Friends for X1 in Friends for Y in Friends
             if X1 != X and Y != X1 and Y != X])
phi5 = And(*[Pass[X,Y] >> ~Pass[X,Y1] for X in Friends for Y in Friends for Y1 in Friends
             if X != Y and X != Y1 and Y != Y1])

for m in satisfiable(phi1 & phi2 & phi3 & phi4 & phi5, all_models=True):
    print(m)
```

□

Bibliography

Darwiche, Adnan (2020). “Three modern roles for logic in AI”. In: *Proceedings of the 39th ACM SIGMOD-SIGACT-SIGAI Symposium on Principles of Database Systems*, pp. 229–243.