CS 4269/5469– Fundamentals of Logic In Computer Science

SEMESTER II, 2025-2026

Home Work-1

- You are welcome to consult any sources you find helpful—books, papers, websites, or even other people. However, you must clearly cite every source you use, and all submitted work must be written in your own words.
  - We strongly discourage the use of AI tools for this homework. The goal of the assignment is to help you engage deeply with the material discussed in class, and relying on AI tends to defeat this purpose. That said, if you do choose to use AI in any way, please acknowledge it and briefly explain how and where it was used. There will be no penalty for doing so; we are simply interested in understanding how these tools are being used.
- You may work either individually or in teams of two. Individual submissions receive a small bonus, but they are expected to require roughly twice the effort of a team submission. If you work in a team, please make sure that the workload is shared fairly.
- For team submissions, all members should be familiar with and have discussed the solutions to every question. This homework is meant to encourage independent thinking and discussion beyond what is covered in lectures.
- Please submit your solutions electronically as a PDF on Canvas. You are encouraged (but not required) to typeset your solutions in LaTeX. Handwritten submissions are also acceptable.
- The assignment is due on February 2, 2026 (11:59 pm). The grading policy is as follows:
  - Submissions before January 28, 2026 (11:59 pm): 30% extra credit
  - Submissions from January 29, 2026 (12:00 am) to January 31, 2026 (11:59 pm): 15% extra credit
  - Submissions from February 1, 2026 (12:00 am) to February 2, 2026 (11:59 pm): no penalty
  - Submissions from February 3, 2026 (12:00 am) to February 4, 2026 (11:59 pm): 25% penalty
  - Submissions from February 5, 2026 (12:00 am) to February 5, 2026 (11:59 pm): 50% penalty
  - Submissions after February 5, 2026: not accepted

  Additionally, individual (non-team) submissions receive a 10% bonus applied on top of the above scoring scheme.

Declaratively Defining Inductive Sets

In the first lecture, we dedicated a significant amount of time trying to formally define what well-formed formulae are. Indeed, translating an abstract concept, that exists only in our minds, into something mathematically precise can be challenging at first sight. But, at the same time, a clean and precise definition is not just very rewarding and good to look at, but also makes further analyses simple.

Let us recap the very first definition that we came up with. As you might recall, instead of characterizing 'whether a given string is a well-formed formula?', we presented a *declarative* definition for the set of all well-formed formulae. A declarative definition like the one we have seen in the lecture essentially presents the entire set *in one go*, unlike a definition which spells out the induction explicitly. Let us repeat the definition here for the sake of completeness.

Definition 1 (Proposition-Connective-Closed set)**.** Given a set of propositions $\mathcal{P}$ and the set of connectives $\mathcal{C} = \{\neg, \vee\}$, a set S of strings over the alphabet $\Sigma = \mathcal{C} \cup \mathcal{P} \cup \{`(', `)'\}$ is said to be $(\mathcal{P}, \mathcal{C})$-closed if the following hold:

1. $\mathcal{P} \subseteq S$.
2. for every string w, if $w \in S$, then $(\neg w) \in S$.
3. for every two strings w_1, w_2, if $w_1, w_2 \in S$, then $(w_1 \vee w_2) \in S$

In the above definition, we should ideally have been more precise and written "$(\cdot \neg \cdot w \cdot) \in S$" instead of "$(\neg w) \in S$" and "$(\cdot w_1 \cdot \vee \cdot w_2 \cdot)$" instead of "$(w_1 \vee w_2) \in S$", thereby explicitly spelling out the string concatenation operator '$\cdot$'. We however omit this pedantic detail in favor of readability.

Having defined such a closed set, the set of well-formed formulae was then defined to be the *smallest* set which was $(\mathcal{P}, \mathcal{C})$-closed. As we discussed, prior to stating such a definition, we must show that such a smallest set indeed exists. What does *smallest* mean, in the context of sets, anyway? We say that a set S_1 is smaller than another set S_2 if $S_1 \subseteq S_2$ (we allow the case when $S_1 = S_2$, in which case S_1 is smaller than S_2 and S_2 is smaller than S_1). So, when we say that a smallest set of a particular kind exists, this is what it really means : let C be the collection (or set) of all sets of the kind we are interested in; in our setting C will be the set of all sets which are $(\mathcal{P}, \mathcal{C})$-closed. A set $S \in C$ is then the smallest in C if $S \subseteq S'$ for every $S' \in C$. Now, in order to define something to be the smallest set in the collection C, we must be sure that the smallest set does exist. Notice that, this is more subtle compared to the case of, say natural numbers $\mathbb{N}$. For every set $C \subseteq \mathbb{N}$ of natural numbers, a unique smallest element is guaranteed to exist in C. This, however, is not true for a collection of sets. That is, the smallest set in a collection of set may not exist, in general. Consider, for example, the set of sets of natural numbers $C = \{\{1, 9\}, \{2, 8\}, \{3, 7\}\}$. Here, no set S in C is a subset of all other sets in C, and thus, in this example, the collection C does not have a smallest element. On the other hand, we will show that in the collection of all *closed* sets, a smallest member set does exist.

However, instead of showing the existence of the smallest set for the special case of $(\mathcal{P}, \mathcal{C})$-closed sets, we will prove a more general result as part of this exercise. Instead of restricting ourselves to the space of strings over some alphabet, we will work with a generic *universe* of elements, and then define when a subset of the universe is closed with respect to a fixed set of *operators*.

Definition 2 (Generic Closure). Let U ('*universe*') be a set of elements. Let $\mathcal{O}$ ('*operators*') be a set of functions whose domain is the set U and co-domain is the powerset[1] of U. The arity of a function $f \in \mathcal{O}$ is denoted by $\mathsf{arity}(f)$. Thus, a function $f \in \mathcal{O}$ of arity $r = \mathsf{arity}(f)$ is of the type $f : \underbrace{U \times U \times \cdots \times U}_{r \text{ times}} \to \mathscr{P}(U)$. Let $I \subseteq U$ be another set, called the *initial* set. We say that a set $S \subseteq U$ is $(I, \mathcal{O})$-closed if each of the following hold:

1. $I \subseteq S$.
2. For every function $f \in \mathcal{O}$, with $r = \mathsf{arity}(f)$, and for every sequence $e_1, e_2, \ldots, e_r \in U$ of r elements in U, if $e_1, e_2, \ldots, e_r \in S$, then $f(e_1, e_2, \ldots, e_r) \subseteq S$.

In words, a set S is $(I, \mathcal{O})$-closed if it includes the initial set I, and further, it includes the *closure* of every function in the set of operators. This means that, for example, if $f \in \mathcal{O}$ is a binary function that maps the elements e_1, e_2 to the set $\{e_3, e_4, e_5\}$, then whenever both e_1 and e_2 are in S, then the elements e_3, e_4 and e_5 must also be in S.

As an example, let us choose U to be the set of all strings over the alphabet $\Sigma = \mathcal{C} \cup \mathcal{P} \cup \{`(', `)'\}$ (i.e., $U = \Sigma^*$), the initial set to be $I = \mathcal{P}$ and the set of operators to be the set $\mathcal{O} = \{\mathsf{makeNegation}, \mathsf{makeDisjunction}\}$ of functions defined next. The unary function $\mathsf{makeNegation} : \Sigma^* \to \mathscr{P}(\Sigma^*)$ maps every string $w \in U$ to the singleton set $\mathsf{makeNegation}(w) = \{(\neg w)\}$. The binary function $\mathsf{makeDisjunction} : \Sigma^* \times \Sigma^* \to \mathscr{P}(\Sigma^*)$ maps every two strings $w_1, w_2 \in \Sigma^*$ to the singleton set $\mathsf{makeDisjunction}(w_1, w_2) = \{(w_1 \vee w_2)\}$ containing the larger string $(w_1 \vee w_2)$. Then, a set of strings $S \subseteq U$ will be $(I, \mathcal{O})$-closed exactly when it is $(\mathcal{P}, \mathcal{C})$-closed as defined earlier. That is, the definition of $(\mathcal{P}, \mathcal{C})$-closed set is an **instantiation** of the definition of $(I, \mathcal{O})$-closed set.

Having stated the definition of a (generic) closed set, we can establish that a smallest closed set indeed exists:

Lemma 1. *Let U be a universe, $\mathcal{O}$ be a set of operators and $I \subseteq U$ be a set of initial elements. There is a smallest $(I, \mathcal{O})$-closed set. That is, there is a set S such that S is $(I, \mathcal{O})$-closed, and further for every set $S' \subseteq U$ such that S' is $(I, \mathcal{O})$-closed, it must be that $S \subseteq S'$.*

In this exercise, we ask you to prove the above result. As a hint, you should be able to generalize the idea behind the proof of existence of the smallest $(\mathcal{P}, \mathcal{C})$-closed set, as discussed in the lecture.

Question 1. Prove Lemma 1.

---

[1] Recall that the powerset of a set S is the set of all of its subsets. That is $\mathscr{P}(S) = \{S' \mid S' \subseteq S\}$.

Having proved the existence of the smallest $(I, \mathcal{O})$-closed set, the existence of the smallest $(\mathcal{P}, \mathcal{C})$-closed set is automatically implied as the definition of $(\mathcal{P}, \mathcal{C})$-closed sets is an instantiation of the definition of $(I, \mathcal{O})$-closed sets. That is, we have the following.

Corollary 1. *There is a smallest* $(\mathcal{P}, \mathcal{C})$*-closed set.*

The above corollary now allows us to define the set of well-formed formulae using a declarative definition, which we have already seen in the lecture.

Definition 3 (Well-Formed Formulae)**.** The set of well-formed formulae, denoted FORM is the smallest $(\mathcal{P}, \mathcal{C})$-closed set of strings over the alphabet $\Sigma = \mathcal{C} \cup \mathcal{P} \cup \{`(', `)'\}$. A string $w \in \Sigma^*$ is said to be a well-formed formula iff $w \in \mathsf{FORM}$.

Question 2. For each of the following strings, prove or disprove that they are well-formed formulae:

(1) $w_1 = \epsilon$ (i.e., the empty string).
(2) $w_1 = (\neg p)$, where $p \in \mathcal{P}$ is some proposition.
(3) $w_2 = (p\neg)$, where $p \in \mathcal{P}$ is some proposition.

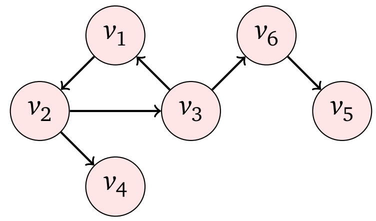
In the following, we ask you to express some sets of numbers as closed sets.

Question 3. Fix $U = \mathbb{R}$ to be the set of real numbers. In the following, state an appropriate initial set I and a set of operators $\mathcal{O}$ so that the smallest $(I, \mathcal{O})$-closed set is the set S defined below. Since there are multiple ways to define these sets, we ask you to state any one way to define I and $\mathcal{O}$ with the restriction that I is a *finite* subset of $\mathbb{R}$ and $\mathcal{O}$ has less than 10 operators.

(a) $S = \{0, 1, 2, 3, \ldots\}$. That is, S is the set of natural numbers.
(b) $S = \{1, 3, 5, 7, \ldots\}$. That is, S is the set of odd natural numbers.
(c) $S = \mathbb{Q}$, the set of all rational numbers. A real number r is a rational number if it can be expressed as $r = p/q$ or $-p/q$ for some natural numbers $p, q \in \mathbb{N}$, where $q \neq 0$.
(d) $S = \{\frac{1}{3}, \frac{2}{3}, \frac{4}{3}, \frac{5}{3}, \frac{7}{3}, \frac{8}{3}, \frac{10}{3} \ldots\}$. That is, S is the set of multiples of $\frac{1}{3}$ that are positive and not natural numbers.

Finite Directed Graphs. So far, all the examples of declarative definitions we have seen (well formed formulae, sets of real numbers etc.,) define sets which are infinite. However, declarative definitions can also very well be used to define sets which are *finite*. Graphs are a canonical example where such definitions are used. As a recap, a *directed graph* is a tuple $G = (V, E)$ where V is a finite set of nodes or *vertices* and E is a set of *edges* of G. An edge is an ordered pair of two vertices, i.e., $E \subseteq V \times V$. The successors of a vertex $u \in V$, denoted $\mathsf{succ}(u)$ is the set of vertices v such that $(u, v) \in E$, while the set of predecessors of a vertex u, denoted $\mathsf{pred}(u)$ is the set of those vertices that have an edge to u. The in-degree and out-degree of a vertex $u \in V$ is respectively $\mathsf{inDegree}(u) = |\mathsf{pred}(u)|$ and $\mathsf{outDegree}(u) = |\mathsf{succ}(u)|$.

As an example, consider the following pictorial representation of a graph G.



In the above graph G, the set of vertices is $V = \{v_1, v_2, v_3, v_4, v_5, v_6\}$ and the set of edges is $E = \{(v_1, v_2), (v_2, v_3), (v_3, v_1), (v_2, v_4), (v_6, v_5), (v_3, v_6)\}$. Here, Notice that, in this graph G, it is possible to *reach* vertex v_4 from vertex v_3 as there is a sequence of edges that start at v_3 and end in v_4 — $(v_3, v_1), (v_1, v_2), (v_2, v_4)$. We formally define this notion of reachability next.

Definition 4 (Graph reachability)**.** Let $G = (V, E)$ be a directed graph. Given vertices $s, t \in V$, a *path* from s to t is a sequence of vertices $\pi = u_1, u_2, \dots, u_k$ such that (i) $u_1 = s$, (ii) for every $i \in \{1, 2, \dots, k-1\}$, $(u_i, u_{i+1}) \in E$, and (iii) $u_k = t$. The length of such a path π is $k-1$. We say that t is reachable from s if either $s = t$ or there is a path from s to t. For the vertex t, the set of *ancestors* of t is defined to be the set

$$\mathsf{Reach}_t = \{v \mid t \text{ is reachable from } v\}.$$

In this exercise, we ask you to present an alternative definition for the set of ancestors in terms of a smallest *closed* set.

Question 4. Let $G = (V, E)$ be a directed graph and let $t \in V$ be a designated vertex. Fix the universe $U = V$ to be the set of vertices of G.

(a) Describe an appropriate initial set I of cardinality at most 2, a singleton set of operator $\mathcal{O} = \{f\}$ with arity$(f) = 1$, such that the smallest $(I, \mathcal{O})$-closed set is the set of vertices reachable from t.

(b) Let $d \in \mathbb{N} \setminus \{0\}$ be a positive natural number, and let $r \in \mathbb{N}$ be such that $r < d$. State an appropriate initial set I, a singleton set of operator $\mathcal{O} = \{f\}$ with arity$(f) = 1$, such that the smallest $(I, \mathcal{O})$-closed set is the set of vertices reachable from s with paths whose length ℓ is such that $\ell \equiv r \bmod d$.

We can extend the notion of reachable vertices, and define the set of all reachable pairs $\mathsf{AllReachPairs}$ for a graph G:

$$\mathsf{AllReachPairs} = \{(u, v) \mid v \text{ is reachable from } u\}.$$

As an example, for the example graph discussed previously, we have:

$$\begin{aligned}\mathsf{AllReachPairs} \;=\; & \{(v_1, v_1), (v_1, v_2), (v_1, v_3), (v_1, v_4), (v_1, v_5), (v_1, v_6), \\ & (v_2, v_1), (v_2, v_2), (v_2, v_3), (v_2, v_4), (v_2, v_5), (v_2, v_6), \\ & (v_3, v_1), (v_3, v_2), (v_3, v_3), (v_3, v_4), (v_3, v_5), \\ & (v_3, v_6), (v_4, v_4), (v_5, v_5), (v_6, v_5), (v_6, v_6)\}.\end{aligned}$$

In the following, we ask you to define this set declaratively as a smallest closed set.

Question 5. Let $G = (V, E)$ be a directed graph. Fix the universe $U = V \times V$ to be the set of all pairs of vertices of G. State an appropriate initial set I of cardinality at most $|V|$, a singleton set of operator $\mathcal{O} = \{f\}$ with $\mathsf{arity}(f) = 2$, such that the smallest $(I, \mathcal{O})$-closed set is $\mathsf{AllReachPairs}$.

Spelling Out the Induction Explicitly

In the second lecture, we saw another definition of the set of well-formed formulae that did not rely on the notion of a set being closed, or the smallest such set. This new definition instead spelled out the underlying induction explicitly by creating a sequence of successively larger sets, whose limit was then defined to be the set we wanted. Let us recap this definition here.

Definition 5 (Inductive-Well-Formed-Formulae)**.** Let $\mathcal{P}$ be a set of propositions and let $\mathcal{C} = \{\neg, \vee\}$ be the set of connectives. Let $\mathsf{FORM}_0, \mathsf{FORM}_1, \mathsf{FORM}_2, \ldots$ be a sequence of sets of strings over the alphabet $\Sigma = \mathcal{C} \cup \mathcal{P} \cup \{`(', `)'\}$ (i.e., each FORM_i is a set of strings) defined as follows.

- $\mathsf{FORM}_0 = \mathcal{P}$.
- for every $i \geq 0$,

$$\mathsf{FORM}_{i+1} = \mathsf{FORM}_i \cup \{(\neg w) \,|\, w \in \mathsf{FORM}_i\} \cup \{(w_1 \vee w_2) \,|\, w_1, w_2 \in \mathsf{FORM}_i\}$$

The set of inductive-well-formed formulae is then the following set

$$\mathsf{INDFORM} = \bigcup_{i \geq 0} \mathsf{FORM}_i$$

The above definition can, in fact, also be generalized so that it works for an arbitrary universe U, initial set I and operators $\mathcal{O}$. We will however skip this discussion as it should be fairly clear now.

Question 6. Let us first observe some basic properties of the above definition.

1. Show that for every $i \geq 0$, $\mathsf{FORM}_i \subseteq \mathsf{FORM}_{i+1}$. That is, the sequence

$$\mathsf{FORM}_0, \mathsf{FORM}_1, \mathsf{FORM}_2, \ldots$$

   is a *monotonic* sequence of sets, with respect to the subset ($\subseteq$) relation.
2. Prove or disprove that there is an $i \geq 0$ such that $\mathsf{FORM}_i = \mathsf{FORM}_{i+1}$.
3. Prove or disprove that for every $i \geq 0$, FORM_i is $(\mathcal{P}, \mathcal{C})$-closed.
4. Prove or disprove that there is an $i \geq 0$ for which FORM_i is $(\mathcal{P}, \mathcal{C})$-closed.

We can prove that the above 'inductive' definition is equivalent to the declarative definition we have seen earlier:

Lemma 2. $\mathsf{FORM} = \mathsf{INDFORM}$.

Question 7. Prove *Lemma* 2 by showing the following:

1. $\mathsf{INDFORM}$ is $(\mathcal{P}, \mathcal{C})$-closed.
2. For every $(\mathcal{P}, \mathcal{C})$-closed set S, we have $\mathsf{INDFORM} \subseteq S$. That is, $\mathsf{INDFORM}$ is the smallest $(\mathcal{P}, \mathcal{C})$-closed set. As a hint, you may want to prove this using induction on the subscript i (corresponding to the i^{th} set FORM_i).

Slightly Modifying the Inductive Definition Let us now look at a slightly different inductive definition of the set of well-formed formulae.

Definition 6 (Inductive-Well-Formed-Formulae (Alternative))**.** Let $\mathcal{P}$ be a set of propositions and let $\mathcal{C} = \{\neg, \vee\}$ be the set of connectives. Let $\mathsf{AltFORM}_0, \mathsf{AltFORM}_1, \mathsf{AltFORM}_2, \ldots$ be a sequence of sets of strings over the alphabet $\Sigma = \mathcal{C} \cup \mathcal{P} \cup \{\text{‘(’}, \text{‘)’}\}$ defined as follows.

- $\mathsf{AltFORM}_0 = \mathcal{P}$.
- for every $i \geq 0$,

$$\begin{aligned} \mathsf{AltFORM}_{i+1} \quad = \quad & \{(\neg w) \mid w \in \mathsf{AltFORM}_i\} \\ & \cup\{(w_1 \vee w_2) \mid w_1 \in \mathsf{AltFORM}_i \text{ and there is a } j \leq i \text{ such that } w_2 \in \mathsf{AltFORM}_j\} \\ & \cup\{(w_1 \vee w_2) \mid w_2 \in \mathsf{AltFORM}_i \text{ and there is a } j \leq i \text{ such that } w_1 \in \mathsf{AltFORM}_j\} \end{aligned}$$

The set of inductive-well-formed formulae is then the following set

$$\mathsf{AltINDFORM} = \bigcup_{i \geq 0} \mathsf{AltFORM}_i$$

Observe that the above alternative definition does not involve an explicit union with the previous set in the sequence, unlike the earlier inductive definition of FORM_i:

$$\mathsf{AltFORM}_{i+1} = \cancel{\mathsf{AltFORM}_i \cup} \{(\neg w) \mid w \in \mathsf{AltFORM}_i\} \cup \cdots$$

Despite this apparent difference, the two inductive definitions are equivalent.

Question 8. Your next task is to show that the two 'inductive' definitions are equivalent.

1. Show that, for every $i \geq 0$, $\mathsf{FORM}_i = \bigcup_{j \geq 0}^{i} \mathsf{AltFORM}_j$. Hint: Use Induction.
2. Show that $\mathsf{INDFORM} = \mathsf{AltINDFORM}$.

Formation Sequences. So far, we have been defining well-formed formulae by first defining the set of *all* well-formed formulae, and then implicitly declaring a string to be a well-formed formula if and only if it is in this set. An alternate, more intuitive approach, is to instead define when a single string is a well-formed formula. A crude definition will be along the lines of '*a well-formed formula is a string which is either a proposition or is of the form* $(\neg w)$ *or of the form* $(w_1 \vee w_2)$*, where* w, w_1 *and* w_2 *are well-formed formulae.*' Notice that this crude definition is using the term 'well-formed formula', rendering it a *chicken-and-egg* problem. In the next definition, we present a definition of the above kind, which nevertheless does not suffer from this problem.

Definition 7 (Formation Sequences)**.** Let $\mathcal{P}$ be a set of propositions and let $\mathcal{C} = \{\neg, \vee\}$ be the set of connectives. A $(\mathcal{P}, \mathcal{C})$—formation-sequence is a non-empty finite sequence of strings over the alphabet $\Sigma = \mathcal{C} \cup \mathcal{P} \cup \{`(', `)'\}$

$$\rho = w_1, w_2, \ldots, w_k$$

such that $k \geq 1$, and for every $1 \leq i \leq k$, one of the following holds:

- $w_i \in \mathcal{P}$, or
- there is a $j < i$ such that $w_i = (\neg w_j)$, or
- there are two indices j, k with $j < i$ and $k < i$ such that $w_i = (w_j \vee w_k)$.

The $(\mathcal{P}, \mathcal{C})$—formation sequence $\rho = w_1, w_2, \ldots, w_k$ is said to belong to the string w if $w_k = w$. A string w is said to be *formation-sequence-well-formed* if there is a formation sequence ρ that belongs to w.

Question 9. Give two different formation sequences that belong to the following string

$$w = \Big((\neg p_1) \vee \Big((\neg p_3) \vee \big(\neg(p_2 \vee (\neg p_1))\big)\Big)\Big)$$

Indeed, the set of well-formed formulae is exactly those that are formation-sequence-well-formed:

Lemma 3. $\mathsf{INDFORM} = \{w \in \Sigma^* \mid w \text{ is formation-sequence-well-formed}\}$

Question 10. Prove Lemma 3 by proving the following two:

1. Show that every string which is formation-sequence-well-formed is also well-formed. That is,

$$\{w \in \Sigma^* \mid w \text{ is formation-sequence-well-formed}\} \subseteq \mathsf{INDFORM}.$$

2. Show that for every well-formed formula w, w is formation-sequence-well-formed. That is,

$$\mathsf{INDFORM} \subseteq \{w \in \Sigma^* \,|\, w \text{ is formation-sequence-well-formed}\}.$$

We are now ready to prove the statement that we were (wrongly) willing to use as a definition for what a well-formed formula:

Lemma 4. *Let $\mathcal{P}$ be a set of propositions and let $\mathcal{C} = \{\neg, \vee\}$ be the set of connectives. Every well-formed formula ϕ over $\mathcal{P}$ and $\mathcal{C}$ is of one of the following forms*

1. *$\phi \in \mathcal{P}$, or*
2. *$\phi = (\neg\psi)$, where ψ is a well-formed formula, or*
3. *$\phi = (\psi_1 \vee \psi_2)$, where ψ_1 and ψ_2 are well-formed formula.*

Question 11. Prove Lemma 4. Hint: Use Lemma 3
