1. Satisfiability, Safety, and Containment

These are sample solutions to some of the exercises that were given as coursework. They are not intended as models but show each one way to approach the problem set in the exercise.

1. Finite vs. Infinite Satisfiability

We consider first-order sentences (= closed formulas), possibly with constants and equality, but without function symbols. A sentence is *finitely satisfiable* if it is satisfied by some interpretation with a finite domain. It is *infinitely satisfiable* if it is satisfied by some interpretation with an infinite domain.
For each of the two cases below, write down a sentence ϕ, ψ, respectively, with the required property:

1. ϕ is finitely satisfiable, but not infinitely satisfiable;
2. ψ is infinitely satisfiable, but not finitely satisfiable.

Explain why your formulas have the required property.

Sample solution (by Ario Santoso and Nhung Ngo).

The following formula is finitely satisfiable, but not infinitely satisfiable:

$$\Phi_1 = \forall x, y\ (x = y).$$

Clearly, Φ_1 is satisfied by an interpretation if and only if the domain of the interpretation has exactly one element.

The following formula is infinitely satisfiable but not finitely satisfiable:

$$\begin{aligned}\Phi_2\ =\ & (\forall x \exists y\ \mathsf{LessThan}(x, y)) \land (\neg \exists x\ \mathsf{LessThan}(x, x)) \land \\ & (\forall x, y, z\ \mathsf{LessThan}(x, y) \land \mathsf{LessThan}(y, z) \rightarrow \mathsf{LessThan}(x, z)).\end{aligned}$$

We have to show that Φ_2 is infinitely satisfiable but not finitely satisfiable. The interpretation $\mathbf{I}$ with the domain $\mathbb{N}$ where

$$(x, y) \in \mathsf{LessThan}^{\mathbf{I}} \quad \text{iff} \quad x < y,$$

is obviously an interpretation with infinite domain which satisfies Φ_2. Hence Φ_2 is infinitely satisfiable. It remains to show that Φ_2 is not finitely satisfiable.

To show that Φ is not finitely satisfiable, suppose by contradiction there exists an interpretation $\mathbf{J}$ with a finite domain $\Delta = \{ d_0, \ldots, d_n \}$, which satisfies Φ_2. Hence, because of

$$(\forall x \exists y \ \mathsf{LessThan}(x, y)) \tag{1}$$

and

$$(\neg \exists x \ \mathsf{LessThan}(x, x)), \tag{2}$$

we have that for all $d_i \in \Delta$ there exists $d_j \in \Delta$ such that $(d_i, d_j) \in \mathsf{LessThan}^{\mathbf{J}}$ and $d_i \neq d_j$, where $0 \leq i \leq j \leq n$. By the transitivity imposed by

$$(\forall x, y, z \ \mathsf{LessThan}(x, y) \wedge \mathsf{LessThan}(y, z) \rightarrow \mathsf{LessThan}(x, z)) \tag{3}$$

we have that if $(d_i, d_j) \in \mathsf{LessThan}^{\mathcal{J}}$ and $(d_j, d_k) \in \mathsf{LessThan}^{\mathcal{J}}$ then $(d_i, d_k) \in \mathsf{LessThan}^{\mathcal{J}}$, where $0 \leq i \leq j \leq k \leq n$. Now let's construct the interpretation $\mathbf{J}$ iteratively from d_0. Because of (1) and (2), we need to have $d \in \Delta$ such that $(d_0, d) \in \mathsf{LessThan}^{\mathbf{J}}$ and $d \neq d_0$. W.l.o.g. suppose $d = d_1$. Again by (1) and (2), we need to have $d' \in \Delta$ such that $(d_1, d') \in \mathsf{LessThan}^{\mathbf{J}}$ and $d' \neq d_1$. Because of (3) and (1) we can't also have $d' = d_0$. W.l.o.g. suppose $d' = d_2$. Because of the transitivity imposed by (3), we must also have $(d_0, d_2) \in \mathsf{LessThan}^{\mathbf{J}}$. Repeating the same argument, it is easy to see that at some point we will reach d_n and we cannot find $d'' \in \Delta$ such that $(d_n, d'') \in \mathsf{LessThan}^{\mathbf{J}}$ because the domain is finite. That is, it requires an interpretation with infinite domain. Hence we cannot have an interpretation $\mathbf{J}$ with a finite domain that satisfies Φ_2. Hence we have a contradiction. Therefore it's impossible for Φ_2 to have a finite model.

2. Finite vs. Database Satisfiability

We consider now first-order sentences that may have constants, but no function symbols and no equality or disequality atoms. We say that a sentence is *database satisfiable* if it is satisfied by an interpretation that is a database instance. (Note that database and finite satisfiability are not necessarily the same since every database instance has the domain **dom**, which is infinite.) Do there exist sentences ϕ, ψ such that

1. ϕ is finitely satisfiable, but not database satisfiable;
2. ψ is database satisfiable, but not finitely satisfiable?

For each of the two cases, if there exists such a formula, write one down. If there does not exist such a formula, write down a proof for your claim. (That is, explain why a sentence is finitely satisfiable if it is database satisfiable, or why it is database satisfiable if it is finitely satisfiable.)

Sample solution.

The answer to the first question is positive.

Proposition 1. *There is a sentence that is finitely satisfiable, but not database satisfiable.*

Proof. Consider the sentence $\phi = \forall x \ P(x)$. Clearly, there exists a finite interpretation where all elements of the domain are in the relation P. However, the sentence is not database satisfiable, because the domain of every database interpretation is infinite, while the number of domain elements occurring in any relation is finite. $\square$

The answer to the second question is negative.

Theorem 2. *Every database-satisfiable sentence is finitely satisfiable.*

To prove the theorem, we need some auxiliary results. For the proof we want to show that for every first-order sentence ψ and every instance $\mathbf{I}$ satisfying ψ we can construct a finite interpretation $\mathbf{I}_0$ that satisfies ψ. The idea is to leave the interpretation of the relation symbols unchanged, but to keep only one constant that occurs neither in a relation instance nor in the query. Since the formulas in question do not contain equality or other built-in predicates, we cannot distinguish between the domain elements that do not occur in any relation or the query. (If we had built-in atoms, then we could achieve the same effect by keeping as many constants outside the active domain as can be distinguished by the sentence and its subformulas. One can show that it would be sufficient to keep as many such constants as there are variables in the sentence.)
Let D be the active domain of $\mathbf{I}$ and ψ, and let d_0 be a fresh constant, not occurring in $\mathbf{I}$ or ψ. The constant d_0 is intended to represent all elements outside of D. Define $\Delta_0 := D \cup \{ d_0 \}$. Let the domain of $\mathbf{I}_0$ be Δ_0 and let the interpretation of each relation symbol be the same as in $\mathbf{I}$. We want to show that $\mathbf{I}_0$ satisfies ψ.
Without loss of generality we can assume that ψ is constructed using only the operators $\wedge$, $\neg$, and existential quantification. (Note that this assumption simplifies the induction step in our proof, since each operator constitutes one case to be considered.)
Let $\gamma\colon \mathbf{dom} \to \Delta_0$ be the function that maps every constant $d \in D$ to itself and every constant $d \in \mathbf{dom} \setminus D$ to d_0. Then, for every assignment α, the assignment $\gamma \circ \alpha$, the composition of γ and α, maps a variable x to $\alpha(x)$ if $\alpha(x) \in D$, and it maps x to d_0 if $\alpha(x)$ is outside D.
As a preparation for our proof we show the following lemma, which says that for checking whether a formula is satisfied by an instance it is sufficient to consider assignments that map variables to constants in the active domain of the instance and the formula and possibly one distinguished constants not in the active domain.

Lemma 3. *For every formula ψ' with $\boldsymbol{adom}(\psi') \subseteq D$ and every assignment $\alpha\colon \boldsymbol{var} \to \boldsymbol{dom}$ we have that*

$$\mathbf{I}, \alpha \models \psi' \quad \textit{if and only if} \quad \mathbf{I}, (\gamma \circ \alpha) \models \psi'.$$

Proof. The claim is shown by induction over the structure of ψ'. Clearly, the claim holds for atoms. It can also be shown in a straightforward way for conjunctive formulas. Let us next have a look at negated formulas. Suppose $\mathbf{I}, \alpha \models \neg\psi''$. This holds if and only if $\mathbf{I}, \alpha \not\models \psi''$. By induction, this holds if and only if $\mathbf{I}, (\gamma \circ \alpha) \not\models \psi''$, which is equivalent to $\mathbf{I}, (\gamma \circ \alpha) \models \neg\psi''$.
Let us finally have a look at existentially quantified subformulas. Suppose $\mathbf{I}, \alpha \models \exists x \psi''$. Then there is an element $d \in \mathbf{dom}$ such that $\mathbf{I}, \alpha[x/d] \models \psi''$. From the induction hypothesis we conclude that $\mathbf{I}, \gamma \circ (\alpha[x/d]) \models \psi''$. Noting that $\gamma \circ (\alpha[x/d]) = (\gamma \circ \alpha)[x/\gamma(d)]$, we infer that $\mathbf{I}, (\gamma \circ \alpha)[x/\gamma(d)] \models \psi''$ and thus that $\mathbf{I}, (\gamma \circ \alpha) \models \exists x \psi''$. Conversely, suppose that $\mathbf{I}, (\gamma \circ \alpha) \models \exists x \psi''$. Then there is an element $d \in \mathbf{dom}$ such that $\mathbf{I}, (\gamma \circ \alpha)[x/d] \models \psi''$. Again, since $\gamma \circ ((\gamma \circ \alpha)[x/d]) = (\gamma \circ \alpha)[x/\gamma(d)]$, the induction hypothesis implies that $\mathbf{I}, (\gamma \circ \alpha)[x/\gamma(d)] \models \psi''$. Since $(\gamma \circ \alpha)[x/\gamma(d)] = \gamma \circ (\alpha[x/d])$, we conclude by the induction hypothesis that $\mathbf{I}, \alpha[x/d] \models \psi''$ and hence that $\mathbf{I}, \alpha \models \exists x \psi''$. $\square$

We now show a lemma that together with the preceding lemma implies our theorem.

Lemma 4. *For every formula ψ' with **adom**$(\psi') \subseteq D$ and every assignment α: **var** $\to D_0$ it holds that*

$$\mathbf{I}, \alpha \models \psi' \quad \textit{if and only if} \quad \mathbf{I}_0, \alpha \models \psi'.$$

Proof. Again, the claim is shown by induction over the structure of ψ'. By definition of $\mathbf{I}_0$, the claim holds if ψ' is an atom.
Suppose now that ψ' is a conjunction, that is $\psi' = \psi'_1 \wedge \psi'_2$. Then for every assignment $\alpha\colon \mathbf{var} \to D_0$ we have

$$\begin{aligned} \mathbf{I}, \alpha \models \psi'_1 \wedge \psi'_2 \quad & \text{iff} \quad \mathbf{I}, \alpha \models \psi'_1 \text{ and } \mathbf{I}, \alpha \models \psi'_2 \\ & \text{iff} \quad \mathbf{I}_0, \alpha \models \psi'_1 \text{ and } \mathbf{I}_0, \alpha \models \psi'_2 \\ & \text{iff} \quad \mathbf{I}_0, \alpha \models \psi'_1 \wedge \psi'_2. \end{aligned}$$

Suppose next that ψ' is a negated formula, that is $\psi' = \neg\psi''$. Then for every assignment $\alpha\colon \mathbf{var} \to D_0$ we have

$$\begin{aligned} \mathbf{I}, \alpha \models \neg\psi' \quad & \text{iff} \quad \mathbf{I}, \alpha \not\models \psi'' \\ & \text{iff} \quad \mathbf{I}_0, \alpha \not\models \psi'' \\ & \text{iff} \quad \mathbf{I}_0, \alpha \models \neg\psi''. \end{aligned}$$

Suppose finally that ψ' is an existentially quantified formula, that is $\psi' = \exists x \psi''$. Let $\alpha\colon \mathbf{var} \to D_0$ be an assignment. We note in passing that $\alpha = \gamma \circ \alpha$.
Let $\mathbf{I}, \alpha \models \exists x \psi''$. Then there exists an element $d \in$ **dom** such that $\alpha[x/d], \mathbf{I} \models \psi''$. Using Lemma 3, we conclude that $\mathbf{I}, \gamma \circ (\alpha[x/d]) \models \psi''$. Since $\alpha = \gamma \circ \alpha$, we have $\gamma \circ (\alpha[x/d]) = \alpha[x/\gamma(d)]$ and thus $\mathbf{I}, \alpha[x/\gamma(d)] \models \psi''$. By the induction hypothesis it follows that $\mathbf{I}_0, \alpha[x/\gamma(d)] \models \psi''$ and hence $\mathbf{I}_0, \alpha \models \exists x \psi''$.
Conversely, let $\mathbf{I}_0, \alpha \models \exists x \psi''$. Then there exists a $d' \in D_0$ such that $\mathbf{I}_0, \alpha[x/d'] \models \psi''$. By our induction hypothesis, it follows that $\mathbf{I}, \alpha[x/d'] \models \psi''$, which implies that $\mathbf{I}, \alpha \models \exists x \psi''$. □

This now proves the theorem: For an arbitrary sentence ψ satisfied by some instance $\mathbf{I}$ we have constructed a finite interpretation $\mathbf{I}_0$ such that $\mathbf{I}_0$ satisfies ψ, too.

3. Positive Queries

We consider formulas without built-in predicates such as "$=$, $\leq$, or $\neq$." A first-order logic formula is *positive* if it contains only the logical symbols "$\wedge$", "$\vee$", and "$\exists$". A relational calculus query Q_ϕ is *positive* if the defining formula ϕ is positive.

1. Is satisfiability of positive queries decidable? If yes, what does an algorithm look like? If not, how can one prove undecidability?

2. Are positive queries safe?

3. Can one represent positive queries in relational algebra? If one can, explain how. If not, provide a proof.

Sample solution.

Regarding the first question, we show that every positive formula is satisfiable. Therefore, every positive query is satisfiable. Since positivity of a formula can be verified by a straightforward syntactic check, the set of satisfiable positive formulas is decidable.
let ϕ be a positive formula and d be a fresh constant not occurring in ϕ. Let $Atom_\phi$ be the set of all atoms occurring ϕ and let $\mathbf{I}_\phi$ be obtained from $Atom_\phi$ by substituting d for every variable occurring in an atom of ϕ. Let α_d be the assignment that maps every variable to d. Then one can show in a straightforward manner that $\mathbf{I}_\phi, \alpha_d \models \phi$.

Regarding the second claim, consider the formula

$$\phi(x, y) \;=\; P(x) \vee R(y).$$

Then for every instance $\mathbf{I}$ we have that

$$Q_\phi(\mathbf{I}) = \{\, (d, e) \mid d \in P^{\mathbf{I}} \text{ and } e \in \mathbf{dom} \,\} \cup \{\, (d, e) \mid d \in \mathbf{dom} \text{ and } e \in R^{\mathbf{J}} \,\},$$

which is an infinite set. This example shows that also disjunction can give rise to unsafe queries.

The fact that positive queries may be unsafe precludes the possibility to represent positive queries in relational algebra.

4. Query Semantics and Integrity Constraints

Let Σ be the signature with the schemas

$$\mathsf{S}(\mathsf{theater}, \mathsf{mtitle}), \qquad \mathsf{M}(\mathsf{title}, \mathsf{director})$$

Intuitively S stands for "schedule" and M stands for "movie". Both attributes, title and mtitle, refer to the title of a movie.
Consider the following two first-order formulas with free variable t:

$$\phi_1 = \exists m\, \mathsf{S}(t, m) \wedge \forall m' \,(\mathsf{S}(t, m') \to \mathsf{M}(m', \text{'Tarantino'}))$$

$$\phi_2 = \exists m\, \mathsf{S}(t, m) \wedge \forall m', d\, (\mathsf{S}(t, m') \wedge \mathsf{M}(m', d) \to d = \text{'Tarantino'}).$$

and the corresponding two relational calculus queries

$$Q_1 = \{\, t \mid \phi_1 \,\}$$

$$Q_2 = \{\, t \mid \phi_2 \,\}.$$

You may remember that we discussed the two queries in the lab and were wondering whether they were equivalent.

1. Is one of the two queries Q_1, Q_2 contained in the other?

   If you claim that $Q_i \sqsubseteq Q_j$, provide an argument (not necessarily a formal proof). If you claim that $Q_i \not\sqsubseteq Q_j$, give a database instance $\mathbf{I}$ such that $Q_i(\mathbf{I}) \not\subseteq Q_j(\mathbf{I})$.

Consider in addition the following two integrity constraints:

$$\gamma_K = \forall\, m, d, d' \,(\mathsf{M}(m,d) \wedge \mathsf{M}(m,d') \rightarrow d = d')$$

$$\gamma_{FK} = \forall\, t, m\, \exists\, d \,(\mathsf{S}(t,m) \rightarrow \mathsf{M}(m,d)).$$

Clearly, γ_K is a primary key constraint that states that title is the primary key of M, while γ_{FK} is a foreign key constraint that states that the mtitle attribute of S refers to the key attribute title of M.

2. What can you say about the containment of the two queries if you consider only instances that satisfy one or both of the constraints γ_K and γ_{FK}?

   Note that these are three cases. For each case, two possible containments have to be considered.

Sample solution to Question 1.

If the first query retrieves a theater, then for any movie shown in the theater Tarantino is a director. If the second query retrieves a theater, then for any movie shown in the theater no director other than Tarantino exists. Thus, if theater t_1 shows a movie m_1 with two directors, one of which is Tarantino, then t_1 is retrieved by query Q_1, but not by by query Q_2. However, if theater t_2 shows a movie m_2 without director, then t_2 is retrieved by query Q_2, but not by by query Q_1. An instance serving as a counterexample to both containments is thus $\mathbf{I}_1$, defined as follows:

$$\begin{aligned}\mathbf{I}_1 = \{\, & \mathsf{S}(\text{‘Roxy’}, \text{‘Reservoir Cats’}),\\ & \mathsf{M}(\text{‘Reservoir Cats’}, \text{‘Tarantino’}),\ \mathsf{M}(\text{‘Reservoir Cats’}, \text{‘Torontono’}),\\ & \mathsf{S}(\text{‘Palace’}, \text{‘The Great Gatsby’})\,\}.\end{aligned}$$

With this definition we have

$$Q_1(\mathbf{I}_1) = \{\, \text{‘Roxy’} \,\}$$
$$Q_2(\mathbf{I}_1) = \{\, \text{‘Palace’} \,\}.$$

Thus, none of the two queries is contained in the other.

Sample solution to Question 2.

If we consider only instances that satisfy γ_K, then we have $Q_1 \sqsubseteq Q_2$, but not vice versa. To see this, consider an instance $\mathbf{I}$ that satisfies γ_K. Suppose that $t \in Q_1(\mathbf{I})$. Then for any movie m on schedule at t, there is a tuple $\mathsf{M}(m, \text{'Tarantino'})$ in $\mathbf{I}$. Since the key constraint γ_K holds over $\mathbf{I}$, this is the only M record for such an m. Thus t satisfies also Q_2. That Q_2 is not contained in Q_1 is shown by the instance $\mathbf{I}_K = \{\, \mathsf{S}(\text{‘Palace’}, \text{‘The Great Gatsby’}) \,\}$, which satisfies γ_K and for which we have $Q_1(\mathbf{I}_K) = \emptyset$, while $Q_2(\mathbf{I}_K) = \{\, \text{‘Palace’} \,\}$.
If we consider only instances that satisfy γ_{FK}, then we have $Q_2 \sqsubseteq Q_1$, but not vice versa. To see this, consider an instance $\mathbf{I}$ that satisfies γ_{FK}. Suppose that $t \in Q_2(\mathbf{I})$. Let $\mathsf{S}(t,m)$ be an S-record for t in $\mathbf{I}$. Since $\mathbf{I}$ satisfies γ_{FK}, there is a corresponding M record $\mathsf{M}(m,d)$. Since t is returned by Q_2, we have that $d = \text{'Tarantino'}$. Thus, t is also returned by Q_1. That Q_1 is not contained in Q_2 is shown by the instance

$$\begin{aligned}\mathbf{I}_{FK} = \, & \mathsf{S}(\text{‘Roxy’}, \text{‘Reservoir Cats’}),\\ & \mathsf{M}(\text{‘Reservoir Cats’}, \text{‘Tarantino’}), \mathsf{M}(\text{‘Reservoir Cats’}, \text{‘Torontono’})\,\},\end{aligned}$$

which satisfies γ_{FK} and for which we have $Q_2(\mathbf{I}_{FK}) = \emptyset$, while $Q_1(\mathbf{I}_{FK}) = \{$ 'Roxy $\}$.
If we consider only instances that satisfy both constraints, then Q_1 and Q_2 are mutually contained and thus equivalent.