Homework 3
CS 474

Alexandra Levinshteyn
(Discussed with Rachel Dickinson and Aarthy Padmanaban)

October 26th, 2024

Problem 1

Part (a)

1.

This is asking if there exists some value y such that $y^2 = 2$.

- Naturals: False - The value $y = \sqrt{2}$ doesn't exist in the naturals.
- Rationals: False - The value $y = \sqrt{2}$ doesn't exist in the rationals.
- Reals: True - The value $y = \sqrt{2}$ exists in the reals.

2.

This is asking if for all values x, there is some additive inverse you can add to it to make 0.

- Naturals: False - Set $x = 1$. There is no y such that $x + y = 0$ since -1 isn't in the naturals.
- Rationals: True - For any x, we can choose $y = -x$ so that $x + y = x - x = 0$ and it will still be in the rationals.
- Reals: True - For any x, we can choose $y = -x$ so that $x + y = x - x = 0$ and it will still be in the reals.

3.

This is asking if for all values x and y, as long as $y \neq 0$, we can choose some z such that $x \cdot y = x + z$.

- Naturals: True - Set $z = x \cdot y - x = x(y-1)$ so that $x+z = x+xy-x = x \cdot y$. Since $y \geq 1$ and $x \geq 0$, and both x and y are integers, $z = x(y-1)$ is too a nonnegative integer, that is, a natural number and thus exists.
- Rationals: True - Set $z = x \cdot y - x = x(y-1)$ so that $x+z = x+xy-x = x \cdot y$. As x and y are rational, the resulting z we chose will also be rational and thus exist.
- Reals: True - Set $z = x \cdot y - x = x(y-1)$ so that $x+z = x+xy-x = x \cdot y$. As x and y are real, the resulting z we chose will also be real and thus exist.

4.

This is asking if there exist x and y such that $x + 1 = 0$ (so $x = -1$) and $y^2 = x$ (so $y^2 = -1$).

- Naturals: False - There isn't even an x such that $x + 1 = 0$, so this automatically can't be true.
- Rationals: False - We are forced to set $x = -1$ to satisfy the first statement but $y^2 = -1$ can never be satisfied among the rationals as i isn't rational, so both statement can't be true at once.
- Reals: False - We are forced to set $x = -1$ to satisfy the first statement but $y^2 = -1$ can never be satisfied among the reals as i isn't real, so both statements can't be true at once.

Part (b)

Task 1

We define the following formula:

$$gt_{\mathbb{N}}(x, y) := \exists z. \left((y + z = x) \land \neg(z = 0)\right).$$

Intuitively, if $x > y$ and x and y are both integers, there is some positive integer you can add to y to make x, that is, $z = y - x$. Of course, since this is greater than and not greater than or equal to, we need to specify that $z \neq 0$.

If $gt_{\mathbb{N}}(x, y)$ is true, then we have some $z \neq 0$, that is, positive integer z such that $y + z = x$. This forces $x > y$.

If $x > y$, we can set $z = x - y$, which must be a positive (nonzero) integer, that is, a nonzero natural number and plug it into $gt_{\mathbb{N}}(x, y)$, which will then return true.

Therefore, $gt_{\mathbb{N}}(x, y)$ is true precisely when x and y are substituted by natural numbers m and n such that $m > n$.

Task 2

We define the following formula:

$$gt_{\mathbb{R}}(x, y) := \exists z. \left((y + z \cdot z = x) \land \neg(z = 0)\right).$$

We can use similar intuition here. If $x > y$ and both x and y are real numbers, there is some positive real number $z = x - y$ that we can add to y to obtain x. In order to force positivity of the element we're adding, we use $z \cdot z$, which is always nonnegative. Furthermore, if w is a positive real number, there is some real number z such that $z * z = w$, that is, $\sqrt{w}$.

If $gt_{\mathbb{R}}(x, y)$ is true, then we have some $z \neq 0$, that is, real nonzero number z such that $y + z * z = x$. So, we are adding a positive number z^2 to y to obtain x. This forces $x > y$.

If $x > y$, we can set $z = \sqrt{x - y}$, which must be real as x and y are real and $x - y$ is a positive real number. This satisfies $y + z \cdot z = y + \sqrt{x - y} \cdot \sqrt{x - y} = y + x - y = x$. As $x \neq y$, $z \neq 0$, so this z satisfies the formula.

Therefore, $gt_{\mathbb{R}}(x, y)$ is true precisely when x and y are substituted by real numbers m and n such that $m > n$.

Extra Credit

We begin by stating a fascinating result.

Lagrange's Four-Square Theorem: Every nonnegative integer can be represented a sum of four nonnegative integer squares.

Consider the following macro formula:

$$\begin{aligned}Positive(x) := \exists n_1, n_2, n_3, n_4, d_1, d_2, d_3, d_4.\\ &((x \cdot (d_1 \cdot d_1 + d_2 \cdot d_2 + d_3 \cdot d_3 + d_4 \cdot d_4) = n_1 \cdot n_1 + n_2 \cdot n_2 + n_3 \cdot n_3 + n_4 \cdot n_4)\\ &\wedge (d_1 \cdot d_1 + d_2 \cdot d_2 + d_3 \cdot d_3 + d_4 \cdot d_4 \neq 0)\\ &\wedge (n_1 \cdot n_1 + n_2 \cdot n_2 + n_3 \cdot n_3 + n_4 \cdot n_4 \neq 0)).\end{aligned}$$

I claim that this formula captures exactly when the rational number x is positive, that is, it returns TRUE if and only if x is positive.

First, we note that $n_1, n_2, n_3, n_4, d_1, d_2, d_3, d_4$ could be any rational numbers (not necessarily integers). However, since the squares of rational numbers are positive and the sum of positive numbers is positive, we have that

$$n = n_1 \cdot n_1 + n_2 \cdot n_2 + n_3 \cdot n_3 + n_4 \cdot n_4 \geq 0$$

and

$$d = d_1 \cdot d_1 + d_2 \cdot d_2 + d_3 \cdot d_3 + d_4 \cdot d_4 \geq 0.$$

If $Positive(x)$ returns TRUE, we have some rational numbers $n_1, n_2, n_3, n_4, d_1, d_2, d_3, d_4$ such that

$$x \cdot (d_1 \cdot d_1 + d_2 \cdot d_2 + d_3 \cdot d_3 + d_4 \cdot d_4) = n_1 \cdot n_1 + n_2 \cdot n_2 + n_3 \cdot n_3 + n_4 \cdot n_4,$$

that is,

$$x \cdot d = n$$

and

$$d = d_1 \cdot d_1 + d_2 \cdot d_2 + d_3 \cdot d_3 + d_4 \cdot d_4 \neq 0$$

and also

$$n = n_1 \cdot n_1 + n_2 \cdot n_2 + n_3 \cdot n_3 + n_4 \cdot n_4 \neq 0.$$

Due to the second condition, we can divide both sides of the first equation by d, so we get that

$$x = \frac{n}{d},$$

where n and d are both positive rational numbers. Notably, neither n nor d are 0, so $x \neq 0$. This means that x is a positive rational number.

Now, we show the other direction. If x is a positive rational number, we can represent it as

$$x = \frac{p}{q}$$

for some positive integers p and q. Notably, $q \neq 0$ and as $x \neq 0$, $p \neq 0$.

By Lagrange's Four-Square Theorem, we can write any nonnegative integer as the sum of four nonnegative integer squares. So, for some nonnegative integers p_1, p_2, p_3, p_4 and q_1, q_2, q_3, q_4, we have that

$$p = p_1 \cdot p_1 + p_2 \cdot p_2 + p_3 \cdot p_3 + p_4 \cdot p_4$$

and

$$q = q_1 \cdot q_1 + q_2 \cdot q_2 + q_3 \cdot q_3 + q_4 \cdot q_4.$$

Since $p \neq 0$ and $q \neq 0$, we also have that

$$p = p_1 \cdot p_1 + p_2 \cdot p_2 + p_3 \cdot p_3 + p_4 \cdot p_4 \neq 0$$

and

$$q = q_1 \cdot q_1 + q_2 \cdot q_2 + q_3 \cdot q_3 + q_4 \cdot q_4 \neq 0.$$

As $q \neq 0$, our original definition of x can be rewritten so that we have

$$x \cdot q = p,$$

that is,

$$x \cdot (q_1 \cdot q_1 + q_2 \cdot q_2 + q_3 \cdot q_3 + q_4 \cdot q_4) = p_1 \cdot p_1 + p_2 \cdot p_2 + p_3 \cdot p_3 + p_4 \cdot p_4.$$

Thus, as the integers $p_1, p_2, p_3, p_4, q_1, q_2, q_3, q_4$ are also rational numbers, we can plug in $n_i = p_i$ for $i = 1, 2, 3, 4$ and $d_j = q_j$ for $j = 1, 2, 3, 4$ to show that $Positive(x)$ returns TRUE.

We define the following formula:

$$gt_{\mathbb{Q}}(x, y) := \exists z. \left((y + z = x) \wedge \mathit{Positive}(z)\right).$$

We can use similar intuition here. If $x > y$ and both x and y are rational numbers, there is some positive rational number $z = x - y$ that we can add to y to obtain x. In order to force positivity of the element we're adding, we use the macro defined above, which guarantees that z is positive.

If $gt_{\mathbb{Q}}(x, y)$ is true, then we have some $z > 0$ (guaranteed by $\mathit{Positive}(z)$), that is, positive rational number z such that $y + z = x$. So, we are adding a positive number z to y to obtain x. This forces $x > y$.

If $x > y$, we can set $z = x - y$, which must be a positive rational number. This satisfies $y + z = y + x - y = x$. As $z > 0$ is a rational positive number, $\mathit{Positive}(z)$ returns TRUE, so the formula will be true.

Therefore, $gt_{\mathbb{Q}}(x, y)$ is true precisely when x and y are substituted by rational numbers m and n such that $m > n$.

Part (c)

Professor Moriarty is being ridiculous. We can just negate Formula 1 from Part (a). Consider the formula

$$\phi := \forall y. \neg(y \cdot y = 1 + 1).$$

This formula asks if for all y, $y^2 \neq 2$. This is, of course, true over the natural numbers. For every natural number y, y^2 cannot possibly be 2. However, this is not the case over the real numbers. If we set $y = \sqrt{2}$, a real number, we clearly get that $y \cdot y = 1 + 1$, so ϕ is false.

Problem 2

Part (a)

Task 1

We have the following formula:

$$\begin{aligned}&\forall z.\,((l_1 < z \wedge z < u_1 \wedge l_2 < z \wedge z < u_2)\\&\quad\implies (\exists w. l_1 < w \wedge w < u_1 \wedge l_2 < w \wedge w < u_2 \wedge w \neq z))\,.\end{aligned}$$

Consider the inner formula:

$$\exists w. l_1 < w \wedge w < u_1 \wedge l_2 < w \wedge w < u_2 \wedge w \neq z.$$

We first expand it by converting $w \neq z \equiv \neg(w = z) \equiv (w < z) \vee (z < w)$:

$$\begin{aligned}&\exists w. l_1 < w \wedge w < u_1 \wedge l_2 < w \wedge w < u_2 \wedge (w < z \vee z < w)\\&\equiv \exists w.\,((l_1 < w \wedge w < u_1 \wedge l_2 < w \wedge w < u_2 \wedge w < z) \vee (l_1 < w \wedge w < u_1 \wedge l_2 < w \wedge w < u_2 \wedge z < w))\\&\equiv (\exists w. l_1 < w \wedge w < u_1 \wedge l_2 < w \wedge w < u_2 \wedge w < z) \vee (\exists w. l_1 < w \wedge w < u_1 \wedge l_2 < w \wedge w < u_2 \wedge z < w)\,.\end{aligned}$$

We use one of the quantifier elimination techniques to write this as follows by setting the values lesser than w to be less than those greater than w:

$$\begin{aligned}&(\exists w. l_1 < w \wedge w < u_1 \wedge l_2 < w \wedge w < u_2 \wedge w < z) \vee (\exists w. l_1 < w \wedge w < u_1 \wedge l_2 < w \wedge w < u_2 \wedge z < w)\\&\equiv (l_1 < u_1 \wedge l_1 < u_2 \wedge l_1 < z \wedge l_2 < u_1 \wedge l_2 < u_2 \wedge l_2 < z)\\&\quad \vee (l_1 < u_1 \wedge l_1 < u_2 \wedge l_2 < u_1 \wedge l_2 < u_2 \wedge z < u_1 \wedge z < u_2)\end{aligned}$$

Inside the very outside universal quantifier, we now have the following:

$$\begin{aligned}&(l_1 < z \wedge z < u_1 \wedge l_2 < z \wedge z < u_2)\\\implies &((l_1 < u_1 \wedge l_1 < u_2 \wedge l_1 < z \wedge l_2 < u_1 \wedge l_2 < u_2 \wedge l_2 < z)\\&\vee (l_1 < u_1 \wedge l_1 < u_2 \wedge l_2 < u_1 \wedge l_2 < u_2 \wedge z < u_1 \wedge z < u_2))\,.\end{aligned}$$

We can convert the implication as follows, getting close to DNF form:

$$\begin{aligned}&\neg\,(l_1 < z \wedge z < u_1 \wedge l_2 < z \wedge z < u_2)\\&\vee ((l_1 < u_1 \wedge l_1 < u_2 \wedge l_1 < z \wedge l_2 < u_1 \wedge l_2 < u_2 \wedge l_2 < z)\\&\quad\vee (l_1 < u_1 \wedge l_1 < u_2 \wedge l_2 < u_1 \wedge l_2 < u_2 \wedge z < u_1 \wedge z < u_2))\,.\end{aligned}$$

We could propagate the negation symbol but instead we will first transform the universal qualifier as follows:

$$
\begin{aligned}
&\neg\exists z.\neg \\
&(\neg\,(l_1 < z \wedge z < u_1 \wedge l_2 < z \wedge z < u_2) \\
&\vee ((l_1 < u_1 \wedge l_1 < u_2 \wedge l_1 < z \wedge l_2 < u_1 \wedge l_2 < u_2 \wedge l_2 < z) \\
&\quad \vee (l_1 < u_1 \wedge l_1 < u_2 \wedge l_2 < u_1 \wedge l_2 < u_2 \wedge z < u_1 \wedge z < u_2)))\,.
\end{aligned}
$$

We then get that

$$
\begin{aligned}
&\neg\exists z. \\
&((l_1 < z \wedge z < u_1 \wedge l_2 < z \wedge z < u_2) \\
&\wedge (\neg\,(l_1 < u_1 \wedge l_1 < u_2 \wedge l_1 < z \wedge l_2 < u_1 \wedge l_2 < u_2 \wedge l_2 < z) \\
&\quad \wedge\neg\,(l_1 < u_1 \wedge l_1 < u_2 \wedge l_2 < u_1 \wedge l_2 < u_2 \wedge z < u_1 \wedge z < u_2)))\,.
\end{aligned}
$$

Ignoring the existential quantifier for now, we have that

$$
\begin{aligned}
&(l_1 < z \wedge z < u_1 \wedge l_2 < z \wedge z < u_2) \\
&\wedge (l_1 = u_1 \vee u_1 < l_1 \vee l_1 = u_2 \vee u_2 < l_1 \vee l_1 = z \vee z < l_1 \\
&\quad \vee l_2 = u_1 \vee u_1 < l_2 \vee l_2 = u_2 \vee u_2 < l_2 \vee l_2 = z \vee z < l_2) \\
&\wedge (l_1 = u_1 \vee u_1 < l_1 \vee l_1 = u_2 \vee u_2 < l_1 \vee l_2 = u_1 \vee u_1 < l_2 \vee l_2 = u_2 \vee u_2 < l_2 \vee z = u_1 \\
&\quad \vee u_1 < z \vee z = u_2 \vee u_2 < z).
\end{aligned}
$$

We can distribute the first clause over the second and third clauses, so that we get a disjunction of a bunch of conjunctions of the first clause with one of the atomic formulas from the second clause and one of the atomic formulas from the third clause. The first clause directly contradicts any of the atomic formulas containing z from the second or third clause, which will convert those to false. We can then refactor/undo the distribution and we get the following:

$$
\begin{aligned}
&(l_1 < z \wedge z < u_1 \wedge l_2 < z \wedge z < u_2) \\
&\wedge (l_1 = u_1 \vee u_1 < l_1 \vee l_1 = u_2 \vee u_2 < l_1 \vee l_2 = u_1 \vee u_1 < l_2 \vee l_2 = u_2 \vee u_2 < l_2) \\
&\wedge (l_1 = u_1 \vee u_1 < l_1 \vee l_1 = u_2 \vee u_2 < l_1 \vee l_2 = u_1 \vee u_1 < l_2 \vee l_2 = u_2 \vee u_2 < l_2).
\end{aligned}
$$

Now, let's add back the existential quantifier, ignoring the negation outside it for now. Noting that z now only appears in the first clause, we can take the second and third clauses out of the existential quantifier:

$$
\begin{aligned}
&(\exists z.\, (l_1 < z \wedge z < u_1 \wedge l_2 < z \wedge z < u_2)) \\
&\wedge (l_1 = u_1 \vee u_1 < l_1 \vee l_1 = u_2 \vee u_2 < l_1 \vee l_2 = u_1 \vee u_1 < l_2 \vee l_2 = u_2 \vee u_2 < l_2) \\
&\wedge (l_1 = u_1 \vee u_1 < l_1 \vee l_1 = u_2 \vee u_2 < l_1 \vee l_2 = u_1 \vee u_1 < l_2 \vee l_2 = u_2 \vee u_2 < l_2).
\end{aligned}
$$

The second and third clauses are now equivalent, so we write

$$
\begin{aligned}
&(\exists z.\, (l_1 < z \wedge z < u_1 \wedge l_2 < z \wedge z < u_2)) \\
&\wedge (l_1 = u_1 \vee u_1 < l_1 \vee l_1 = u_2 \vee u_2 < l_1 \vee l_2 = u_1 \vee u_1 < l_2 \vee l_2 = u_2 \vee u_2 < l_2).
\end{aligned}
$$

Let's only consider the existential part for now:

$$\exists z.\, (l_1 < z \wedge z < u_1 \wedge l_2 < z \wedge z < u_2)\,.$$

Using the same quantifier elimination technique as before, we can write that this is equivalent to

$$l_1 < u_1 \wedge l_1 < u_2 \wedge l_2 < u_1 \wedge l_2 < u_2.$$

So, ignoring the very outside negation, we now have

$$
\begin{aligned}
&(l_1 < u_1 \wedge l_1 < u_2 \wedge l_2 < u_1 \wedge l_2 < u_2) \\
&\wedge (l_1 = u_1 \vee u_1 < l_1 \vee l_1 = u_2 \vee u_2 < l_1 \vee l_2 = u_1 \vee u_1 < l_2 \vee l_2 = u_2 \vee u_2 < l_2).
\end{aligned}
$$

We can once again distribute so that we have disjunctions of conjunctions of the first clause with an atomic element from the second clause. Every single atomic element from the second clause directly contradicts one of the atomic elements from the first clause, so we get the disjunction of a bunch of FALSEs, which is, of course, just FALSE. So, the entire formula from the beginning can now be written as

$$\neg\text{FALSE} \equiv \text{TRUE}.$$

Therefore, the q.e. procedure covered in class for Dense Linear Orders Without Endpoints converts the original formula

$$
\begin{aligned}
\phi := \forall z.\, ((&l_1 < z \wedge z < u_1 \wedge l_2 < z \wedge z < u_2) \\
&\implies (\exists w. l_1 < w \wedge w < u_1 \wedge l_2 < w \wedge w < u_2 \wedge w \neq z))\,.
\end{aligned}
$$

to the quantifier-free formula

$$\text{TRUE},$$

which means that this formula is actually a theorem.

Task 2

Z3 gives the following result, which matches with mine. There's not much to this code, we just encode the formula ψ and apply q.e. to it.

```
[12] # Problem 2 Part (a) Task 2
     z = Real("z")
     l1 = Real("l1")
     u1 = Real("u1")
     l2 = Real("l2")
     u2 = Real("u2")
     w = Real("w")
     psi = ForAll(z, Implies(And(l1 < z, z < u1, l2 < z, z < u2),
                             Exists(w, And(l1 < w, w < u1, l2 < w, w < u2, w != z))))
     t = Tactic('qe')
     result = t(psi).as_expr()
     print(result)
```

```
True
```

Part (b)

Task 1

Consider any graph $G = (V, E)$ with vertices $v_1, \ldots, v_n$ and edges (v_i, v_j) between them. Intuitively, we want to assign an interval to each of the vertices and guarantee that any two vertices connected by an edge have overlapping intervals and any two vertices not connected by an edge don't have overlapping intervals.

Define the following macro formula:

$$NoOverlap(l_1, u_1, l_2, u_2) := (u_2 < l_1) \lor (u_1 < l_2).$$

This formula returns TRUE if and only if the intervals (l_1, u_1) and (l_2, u_2) don't overlap. This is because two intervals don't overlap if and only if one starts after the other ends. This formula "assumes" that the intervals (l_1, u_1) and (l_2, u_2) are valid intervals, that is, $l_1 < u_1$ and $l_2 < u_2$.

Then, define the following macro formula:

$$Overlap(l_1, u_1, l_2, u_2) := \neg NoOverlap(l_1, u_1, l_2, u_2)$$

that captures exactly when the intervals (l_1, u_1) and (l_2, u_2) do overlap.

So, our formula α_G should capture the idea that we can create an interval for each vertex $v \in V$ such that two intervals overlap if and only if the two corresponding vertices are adjacent. Consider the following:

$$\begin{aligned}\alpha_G := \exists l_1, u_1, \ldots, l_n, u_n.& \\ &(l_1 < u_1 \land l_2 < u_2 \land \cdots \land l_n < u_n) \\ &\land_{i=1, j>i | (v_i, v_j) \in E}^{n} Overlap(l_i, u_i, l_j, u_j) \land_{i=1, j>i | (v_i, v_j) \notin E}^{n} NoOverlap(l_i, u_i, l_j, u_j).\end{aligned}$$

Then, a graph G with vertices numbered $v_1, \ldots, v_n$ is an interval graph if and only if its vertices satisfy α_G.

Proof: Assume that a graph G is an interval graph. Then, you can map each vertex in the graph to an interval on the real line such that two distinct vertices have an edge between them if and only if the corresponding intervals intersect. Choose l_i and u_i to be the upper and lower coordinates of the interval corresponding to v_i. Then, these values can be plugged into the existential quantifier and will satisfy all the needed statements. All intervals will be valid so $l_i < u_i$. Pairs of vertices with edges between them will have overlapping intervals, satisfying the first looping conjunction, and pairs of vertices without edges won't have overlapping intervals, satisfying the second looping conjunction. Thus, an interval graph G will force α_G to be true and valid.

Assume that a graph G satisfies α_G. Then, choose the values $l_1, u_1, \ldots, l_n, u_n$ that work and assign (l_i, u_i) as the corresponding interval for v_i. All of these intervals will be valid since α_G guarantees that $l_i < u_i$. The first looping conjunction will force adjacent vertices to have overlapping intervals. The second looping conjunction will force non-adjacent vertices to have non-overlapping intervals. Thus, a graph G such that α_G is true and valid will be an interval graph.

Let's now consider the specific graph shown. Label the vertices on the left v_1, v_2, and v_3 from top to bottom. Label the vertex on the right v_4. Then, we have the following adjacent pairs of vertices:

1. (v_1, v_2)
2. (v_1, v_4)
3. (v_2, v_3)
4. (v_3, v_4)

We also have the following non-adjacent pairs of vertices:

1. (v_1, v_3)
2. (v_2, v_4)

So, the corresponding α_G will be

$$\begin{aligned}
\alpha_G := \exists l_1, u_1, l_2, u_2, l_3, u_3, l_4, u_4.& \\
& l_1 < u_1 \wedge l_2 < u_2 \wedge l_3 < u_3 \wedge l_4 < u_4 \\
& \wedge Overlap(l_1, u_1, l_2, u_2) \wedge Overlap(l_1, u_1, l_4, u_4) \wedge Overlap(l_2, u_2, l_3, u_3) \wedge Overlap(l_3, u_3, l_4, u_4) \\
& \wedge NoOverlap(l_1, u_1, l_3, u_3) \wedge NoOverlap(l_2, u_2, l_4, u_4)
\end{aligned}$$

with the justification (done in the more general case) above. The macros *Overlap* and *NoOverlap* can just be directly plugged in but for clarity, I will write it as above. The formula α_G is valid if and only if G is an interval graph.

Task 2

We use Z3 to determine the validity of α_G below. We find that α_G is not valid, so G is not an interval graph. There's not much to this code, we just encode α_G and check if it is valid by checking if its negation is satisfiable.

```
[20] # Problem 2 Part (b) Task 2

# Define NoOverlap 'macro'
def NoOverlap(lower1, upper1, lower2, upper2):
  return Or(upper2 < lower1, upper1 < lower2)

# Define Overlap 'macro'
def Overlap(lower1, upper1, lower2, upper2):
  return Not(NoOverlap(lower1, upper1, lower2, upper2))

# Prepare variables
l1 = Real("l1")
u1 = Real("u1")
l2 = Real("l2")
u2 = Real("u2")
l3 = Real("l3")
u3 = Real("u3")
l4 = Real("l4")
u4 = Real("u4")

# Checking that the intervals are valid
intervals = And(l1 < u1, l2 < u2, l3 < u3, l4 < u4)

# Checking that the adjacent vertices have overlapping intervals
adjacent = And(Overlap(l1, u1, l2, u2), Overlap(l1, u1, l4, u4),
                       Overlap(l2, u2, l3, u3), Overlap(l3, u3, l4, u4))

# Checking that the nonadjacent vertices have nonoverlapping intevals
nonadjacent = And(NoOverlap(l1, u1, l3, u3), NoOverlap(l2, u2, l4, u4))

# Creating the entire formula
alpha = Exists([l1, u1, l2, u2, l3, u3, l4, u4],
               And(intervals, adjacent, nonadjacent))

# Checking if its negation is satisfiable
s = Solver()
s.add(Not(alpha))

result = s.check()

if result == unsat:
    print("The formula is valid and G is an interval graph.")
else:
    print("The formula is not valid and G is not an interval graph.")
```

```
The formula is not valid and G is not an interval graph.
```

Problem 3

Task 1

We start with the following formula:

$$\psi \equiv \forall x . \exists y . \left((2y > 3x) \land (4y < 8x + 10) \right).$$

Consider the inner formula

$$\exists y . \left((2y > 3x) \land (4y < 8x + 10) \right).$$

We will begin by solving this for y:

$$\exists y . \left(\left(y > \frac{3x}{2} \right) \land \left(y < 2x + \frac{5}{2} \right) \right).$$

Let

$$\phi := \left(y > \frac{3x}{2}\right) \wedge \left(y < 2x + \frac{5}{2}\right).$$

Define

$$\phi_r = \phi[y \mapsto r].$$

By the Ferrante-Rackoff procedure, these are now the only intervals and values to consider, as we need to consider very small values, very large values, each of the interval points, and the center of each interval:

1. $r = -\infty$
2. $r = \infty$
3. $r = \frac{3x}{2}$
4. $r = 2x + \frac{5}{2}$
5. $r = \frac{\frac{3x}{2} + 2x + \frac{5}{2}}{2} = \frac{7x}{4} + \frac{5}{4}$

We then need to calculate ϕ_r for each of the values above replacing r. We have that

$$\phi_{-\infty} \equiv \left(-\infty > \frac{3x}{2}\right) \wedge \left(-\infty < 2x + \frac{5}{2}\right) \equiv \bot \wedge \top \equiv \bot$$
$$\phi_{\infty} \equiv \left(\infty > \frac{3x}{2}\right) \wedge \left(\infty < 2x + \frac{5}{2}\right) \equiv \top \wedge \bot \equiv \bot$$
$$\phi_{\frac{3x}{2}} \equiv \left(\frac{3x}{2} > \frac{3x}{2}\right) \wedge \left(\frac{3x}{2} < 2x + \frac{5}{2}\right) \equiv \bot \wedge \left(\frac{3x}{2} < 2x + \frac{5}{2}\right) \equiv \bot$$
$$\phi_{2x+\frac{5}{2}} \equiv \left(2x + \frac{5}{2} > \frac{3x}{2}\right) \wedge \left(2x + \frac{5}{2} < 2x + \frac{5}{2}\right) \equiv \left(2x + \frac{5}{2} > \frac{3x}{2}\right) \wedge \bot \equiv \bot$$
$$\phi_{\frac{7x}{4}+\frac{5}{4}} \equiv \left(\frac{7x}{4} + \frac{5}{4} > \frac{3x}{2}\right) \wedge \left(\frac{7x}{4} + \frac{5}{4} < 2x + \frac{5}{2}\right) \equiv (x > -5) \wedge (x > -5) \equiv x > -5.$$

By the procedure, we know that

$$\exists y.\phi \equiv \phi_{-\infty} \vee \phi_{\infty} \vee \phi_{\frac{3x}{2}} \vee \phi_{2x+\frac{5}{2}} \vee \phi_{\frac{7x}{4}+\frac{5}{4}} \equiv \bot \vee \bot \vee \bot \vee \bot \vee (x > -5) \equiv x > -5.$$

Thus, we can now write that our original formula is

$$\psi \equiv \forall x.\exists y.\phi \equiv \forall x.x > -5.$$

We now write the universal quantifier in terms of the existential quantifier, so that

$$\psi \equiv \neg\exists x.\neg(x > -5) \equiv \neg\exists x.((x = -5) \vee (x < -5)).$$

Let

$$\omega := (x = -5) \vee (x < -5).$$

Define

$$\omega_r = \omega[x \mapsto r].$$

By the Ferrante-Rackoff procedure, these are now the only intervals and values to consider, as we need to consider very small values, very large values, each of the interval points, and the center of each interval:

1. $r = -\infty$
2. $r = \infty$
3. $r = -5$

We then need to calculate ω_r for each of the values above replacing r. We have that

$$\begin{aligned}
&\omega_{-\infty} \equiv (-\infty = -5) \vee (-\infty < -5) \equiv \bot \vee \top \equiv \top \\
&\omega_{\infty} \equiv (\infty = -5) \vee (\infty < -5) \equiv \bot \vee \bot \equiv \bot \\
&\omega_{-5} \equiv (-5 = -5) \vee (-5 < -5) \equiv \top \vee \bot \equiv \top.
\end{aligned}$$

By the procedure, we know that

$$\exists x.\omega \equiv \omega_{-\infty} \vee \omega_{\infty} \vee \omega_{-5} \equiv \top \vee \bot \vee \top \equiv \top.$$

Thus, we can now write that our original formula is

$$\psi \equiv \neg\exists x.\omega \equiv \neg\top \equiv \bot.$$

Therefore, the q.e. procedure due to Ferrante and Rackoff shows that

$$\psi \equiv \bot,$$

so it is false and not valid.

Note: I'm fairly certain I'm allowed to convert formulas such as $2y < 3x$ to $y < \frac{3x}{2}$ as dividing by a positive constant is allowed according to the notes. I assumed arithmetic simplifications referred to directly combining two atomic formulas together, which I did not do.

Task 2

We use Z3 to determine the validity of ψ below. We find that ψ is not valid because its negation is satisfiable. There's not much to this code, we just encode ψ and check if it is valid by checking if its negation is satisfiable.

```
[48] # Problem 3 Task 2

     # Defining rational x in terms of numerator and denominator
     num_x = Int("num_x")
     denom_x = Int("denom_x")
     denom_check_x = denom_x != 0
     x = num_x/denom_x

     # Defining rational y in terms of numerator and denominator
     num_y = Int("num_y")
     denom_y = Int("denom_y")
     denom_check_y = denom_y != 0
     y = num_y/denom_y

     # Defining psi in terms of numerators and denominators
     # Ensuring that the x denominator isn't 0 by adding an implies clause
     # This implies clause functions exactly like a such that
     psi = ForAll([num_x, denom_x], Implies(denom_check_x, Exists([num_y, denom_y],
                                            And(And(2*y > 3*x, 4*y < 8*x + 10),
                                                denom_check_y))))

     # Checking if its negation is satisfiable
     s = Solver()
     s.add(Not(psi))

     result = s.check()

     if result == unsat:
         print("The formula is valid.")
     else:
         print("The formula is not valid.")
```

```
The formula is not valid.
```