Exercise 9.1. Show that any Datalog program can be expressed as a safe Datalog program that is polynomial in size of the original program and given schema.

Exercise 9.2. Assume that the database uses a binary EDB predicate `edge` to store a directed graph. Try to express the following properties in semi-positive Datalog programs with a successor ordering, or explain why this is not possible.

1. The database contains an even number of elements.
2. The graph contains a node with two outgoing edges.
3. The graph is 3-colourable.
4. The graph is not connected (*)
5. The graph does not contain a node with two outgoing edges.
6. The graph is a chain.

Exercise 9.3. A Horn logic program is in `propHorn2` if every rule it contains is of the form `H ←` or `H ← B_1 ∧ B_2`.

It was claimed that entailment checking in `propHorn2` is P-hard. To support this claim, explain how entailment in propositional Horn logic can be reduced to entailment in `propHorn2`. Argue how this reduction can be accomplished in logarithmic space.

Exercise 9.4. Prove that entailment checking in propositional Horn logic is P-hard.

Hint: Modify the `EXPTime TM` simulation from the lecture to simulate a `PTime TM` instead.

Exercise 9.5. Show that the following property cannot be expressed in Datalog:

The `edge` predicate has a proper cycle, i.e., a cycle that is not of the form `edge(a, a)`.

Can you express this property using …

1. … a successor ordering?
2. … atomic EDB negation?
3. … an equality predicate `≈` with the obvious semantics?
4. … an inequality predicate `≉` with the obvious semantics?