Problem 1 *(Algebras and semantics)* (8 points)

Let $\Sigma = (\Omega, \Pi)$ be a signature, let $p(t_1, \ldots, t_n)$ be a ground Σ-atom, and let N be a set of ground Σ-clauses. Define

$$\begin{aligned} N^+ &= \{\, C \in N \mid p(t_1, \ldots, t_n) \text{ occurs positively in } C \,\} \\ N^- &= \{\, C \in N \mid p(t_1, \ldots, t_n) \text{ occurs negatively in } C \,\} \\ N^0 &= \{\, C \in N \mid p(t_1, \ldots, t_n) \text{ does not occur in } C \,\} \end{aligned}$$

Prove: If $N^- = \emptyset$, then N is satisfiable if and only if N^0 is satisfiable.

Problem 2 *(Resolution, model construction)* (3 + 3 = 6 points)

Let $\Sigma = (\Omega, \Pi)$ be a signature with $\Omega = \{a/0,\ f/1\}$ and $\Pi = \{p/1\}$. Suppose that the atom ordering $\succ$ is defined in such a way that $p(f^n(a)) \succ p(f^m(a))$ if and only if $n > m \geq 0$. Let N be the following set of clauses:

$$\begin{gathered} p(f(f(a))) \\ \neg p(x) \vee p(f(x)) \end{gathered}$$

Part (a)

Sketch how the set $G_\Sigma(N)$ of all ground instances of clauses in N looks like. How is it ordered with respect to the clause ordering $\succ_{\mathrm{C}}$?

Part (b)

Construct the candidate model $I^{\succ}_{G_\Sigma(N)}$ of the set of all ground instances of clauses in N.

Problem 3 *(Tableau calculus)* (8 points)

Use the tableau calculus to show that the following set of formulas is unsatisfiable:

$$\Big\{ \forall x\, p(x) \rightarrow p(f(x)), \quad \exists y\, p(y) \wedge \neg p(f(f(y))) \Big\}$$

Problem 4 *(Fixpoint theory)* (7 points)

Let $f : 2^U \rightarrow 2^U$ be a monotone function. Prove: If f has two fixpoints I and J such that $I \cap J = \emptyset$, then $\emptyset$ is a fixpoint of f.

Problem 5 *(Prolog)* (2 + 3 + 3 = 8 points)

Part (a)

Define a predicate `mapempty(`l`)` that succeeds if l is a list and each element of l is the empty list `[]`.

Part (b)

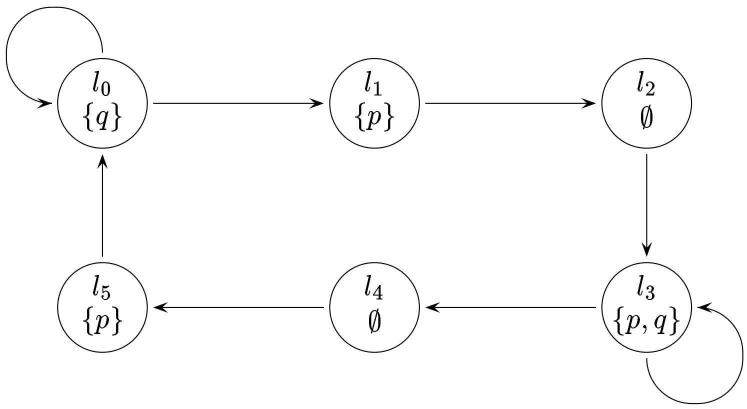
Define predicates `maphd(`l`,`l'`)` and `maptl(`l`,`l''`)` that take a list of lists $l = [l_1, \dots, l_n]$ and compute lists $l' = [l'_1, \dots, l'_n]$ and $l'' = [l''_1, \dots, l''_n]$, respectively, where l'_j is the head of l_j and l''_j is the tail of l_j. For instance, if l is the list `[[1,2,3],[a,b,c]]`, then `maphd` computes the list l' = `[1,a]` and `maptl` computes the list l'' = `[[2,3],[b,c]]`.

Part (c)

Let $l = [l_1, \dots, l_n]$ be a list of lists, such that all lists l_i have the same length m. The *transposition* of l is the list $l' = [l'_1, \dots, l'_m]$ where l'_j is the list of all j-th elements of $l_1, \dots, l_n$. (For instance, the transposition of `[[1,2,3],[a,b,c]]` is the list `[[1,a],[2,b],[3,c]]`.) Implement a Prolog predicate `tr(`l`,`l'`)` that computes the transposition l' of a list l. You can use the predicates defined in Part (a) and (b).

Problem 6 *(CTL)* (7 points)

Let $S = \{l_0, l_1, l_2, l_3, l_4, l_5\}$, let $\Pi = \{p, q\}$, and let $M = (S, R, L)$ be the following time structure (where R is represented by $\to$):



Compute $[\![\mathsf{EG}\,(p \to q)]\!]$.