**Aufgabe 1** *(Algebren und Semantik)* (8 Punkte)

Sei $\Sigma = (\Omega, \Pi)$ eine Signatur, sei $p(t_1, \ldots, t_n)$ ein $\Sigma$-Grundatom, und sei $N$ eine Menge von $\Sigma$-Grundklauseln. Wir definieren

$$\begin{aligned} N^+ &= \{\, C \in N \mid p(t_1, \ldots, t_n) \text{ kommt in } C \text{ positiv vor} \,\} \\ N^- &= \{\, C \in N \mid p(t_1, \ldots, t_n) \text{ kommt in } C \text{ negativ vor} \,\} \\ N^0 &= \{\, C \in N \mid p(t_1, \ldots, t_n) \text{ kommt nicht in } C \text{ vor} \,\} \end{aligned}$$

Beweisen Sie: Falls $N^- = \emptyset$, dann ist $N$ genau dann erfüllbar, wenn $N^0$ erfüllbar ist.

**Aufgabe 2** *(Resolution, Modellkonstruktion)* (3 + 3 = 6 Punkte)

Sei $\Sigma = (\Omega, \Pi)$ eine Signatur mit $\Omega = \{a/0,\ f/1\}$ und $\Pi = \{p/1\}$. Nehmen wir an, daß die Atomordnung $\succ$ so definiert ist, daß $p(f^n(a)) \succ p(f^m(a))$ genau dann gilt, wenn $n > m \geq 0$. Sei $N$ die folgende Klauselmenge:

$$p(f(f(a)))$$
$$\neg p(x) \vee p(f(x))$$

**Teil (a)**

Skizzieren Sie, wie die Menge $G_\Sigma(N)$ aller Grundinstanzen von Klauseln in $N$ aussieht. Wie ist sie bezüglich der Klauselordung $\succ_C$ geordnet?

**Teil (b)**

Konstruieren Sie das Kandidatenmodell $I^{\succ}_{G_\Sigma(N)}$ der Menge aller Grundinstanzen von Klauseln in $N$.

**Aufgabe 3** *(Tableaukalkül)* (8 Punkte)

Zeigen Sie mit Hilfe des Tableaukalküls, daß die folgende Formelmenge unerfüllbar ist:

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

**Aufgabe 4** *(Fixpunkttheorie)* (7 Punkte)

Sei $f : 2^U \rightarrow 2^U$ eine monotone Funktion. Zeigen Sie: Falls $f$ zwei Fixpunkte $I$ und $J$ hat, so daß $I \cap J = \emptyset$ gilt, dann ist $\emptyset$ ein Fixpunkt von $f$.

**Aufgabe 5** *(Prolog)* (2 + 3 + 3 = 8 Punkte)

**Teil (a)**

Definieren Sie ein Prolog-Prädikat `mapempty(`$l$`)`, das für $l$ gilt, falls $l$ eine Liste ist und jedes Element von $l$ die leere Liste `[]` ist.

**Teil (b)**

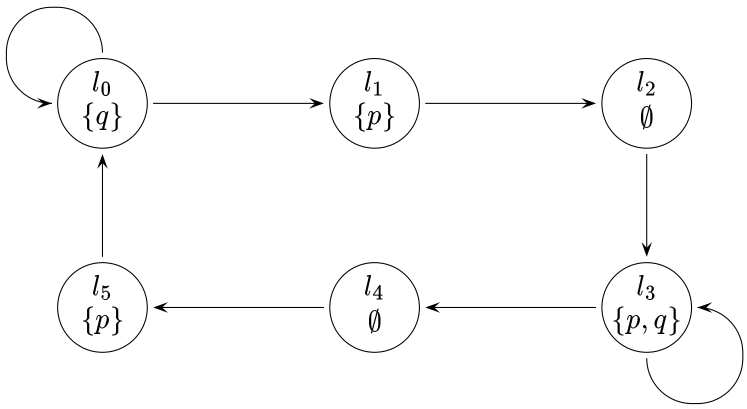
Definieren Sie Prolog-Prädikate `maphd(`$l$`,`$l'$`)` und `maptl(`$l$`,`$l''$`)`, die eine Liste von Listen $l = [l_1, \ldots, l_n]$ nehmen und Listen $l' = [l'_1, \ldots, l'_n]$ bzw. $l'' = [l''_1, \ldots, l''_n]$ berechnen, wobei $l'_j$ der Listenkopf von $l_j$ und $l''_j$ der Listenrumpf von $l_j$ ist. Falls beispielsweise $l$ die Liste `[[1,2,3],[a,b,c]]` ist, dann berechnet `maphd` die Liste $l'$ = `[1,a]` und `maptl` die Liste $l''$ = `[[2,3],[b,c]]`.

**Teil (c)**

Sei $l = [l_1, \ldots, l_n]$ eine Liste von Listen, so daß alle Listen $l_i$ die gleiche Länge $m$ besitzen. Die *Transposition* von $l$ ist die Liste $l' = [l'_1, \ldots, l'_m]$, wobei $l'_j$ die Liste aller $j$-ten Elemente von $l_1, \ldots, l_n$ ist. (Z.B. ist die Transposition von `[[1,2,3],[a,b,c]]` die Liste `[[1,a],[2,b],[3,c]]`.) Implementieren Sie ein Prolog-Prädikat `tr(`$l$`,`$l'$`)`, das die Transposition $l'$ einer Liste $l$ berechnet. Sie können die in Teil (a) und (b) definierten Prädikate benutzen.

**Aufgabe 6** *(CTL)* (7 Punkte)

Sei $S = \{l_0, l_1, l_2, l_3, l_4, l_5\}$, sei $\Pi = \{p, q\}$, und sei $M = (S, R, L)$ die folgende Zeitstruktur (wobei $R$ durch $\rightarrow$ repräsentiert wird):




Berechnen Sie $[\![\mathsf{EG}\,(p \rightarrow q)]\!]$.