2. Hennessy-Milner-Logik

2.1. Syntax und Semantik

Hennessy-Milner-Logik (HML), auch als *multi-modale Logik K* bekannt, erweitert die Aussagenlogik um zwei Konstrukte ("*diamond*" und "*box*"), mit denen man über Nachfolger eines Zustandes in einem Transitionssystem quantifiziert.

Definition 2.1
Seien $\mathcal{P}$ und Σ Mengen von Propositionen bzw. Aktionennamen. Die Syntax von HML ist gegeben durch die folgende Grammatik.

$$\varphi \quad ::= \quad q \mid \varphi \vee \varphi \mid \varphi \wedge \varphi \mid \neg\varphi \mid \langle a \rangle \varphi \mid [a]\varphi$$

wobei $q \in \mathcal{P}$, $a \in \Sigma$. Für $|\Sigma| = 1$ heißt diese Logik auch *mono-modale Logik* oder einfach nur *Modallogik*. In diesem Fall schreiben wir nur $\Diamond\varphi$ und $\Box\varphi$. Als weitere Abkürzungen definieren wir $\mathtt{tt} := q \vee \neg q$ für ein $q \in \mathcal{P}$, $\mathtt{ff} := \neg\mathtt{tt}$, $\varphi \to \psi := \neg\varphi \vee \psi$.

Wir definieren die Semantik von HML über einem Transitionssystem $\mathcal{T} = (\mathcal{S}, \{\xrightarrow{a} \mid a \in \Sigma\}, \lambda, s_0)$ mit $\lambda : \mathcal{S} \to 2^{\mathcal{P}}$ wie folgt. Sei $s \in \mathcal{S}$.

$$\begin{aligned}
\mathcal{T}, s \models q \quad &\text{gdw.} \quad q \in \lambda(s) \\
\mathcal{T}, s \models \varphi \vee \psi \quad &\text{gdw.} \quad \mathcal{T}, s \models \varphi \text{ oder } \mathcal{T}, s \models \psi \\
\mathcal{T}, s \models \varphi \wedge \psi \quad &\text{gdw.} \quad \mathcal{T}, s \models \varphi \text{ und } \mathcal{T}, s \models \psi \\
\mathcal{T}, s \models \neg\varphi \quad &\text{gdw.} \quad \mathcal{T}, s \not\models \varphi \\
\mathcal{T}, s \models \langle a \rangle \varphi \quad &\text{gdw.} \quad \exists t \in \mathcal{S}, s \xrightarrow{a} t \text{ und } \mathcal{T}, t \models \varphi \\
\mathcal{T}, s \models [a]\varphi \quad &\text{gdw.} \quad \forall t \in \mathcal{S} : s \xrightarrow{a} t \text{ impliziert } \mathcal{T}, t \models \varphi
\end{aligned}$$

Ist klar, auf welches Transitionssystem Bezug genommen wird, dann schreiben wir auch einfach nur $s \models \varphi$.

Beispiel 2.1
Dieses Beispiel verdeutlicht bereits, dass HML keine wichtige Logik ist. Es ist schwierig, interessante Eigenschaften zu finden, die in HML ausdrückbar sind.

Formel	Bedeutung
$\bigwedge_{a \in \Sigma} [a]\mathtt{ff}$	es gibt keinen Nachfolger
$\langle a \rangle(\langle b \rangle \mathtt{tt} \wedge \langle c \rangle \mathtt{tt}) \wedge [a]\langle b \rangle \mathtt{tt} \wedge [a]\langle c \rangle \mathtt{tt}$	Bisimilarität (bzgl. $\{a, b, c\}$) zum linken Getränkeautomaten aus Bsp. 1.5.

Interessanter sind sogenannte *Axiomenschema* oder einfach nur *Axiome*. Die Bedeutung des Axioms $[a]\varphi \to \langle a\rangle\varphi$ ist so zu verstehen, dass für alle $\varphi \in \text{HML} \models [a]\varphi \to \langle a\rangle\varphi$ gelten soll. Dies ist z.B. der Fall, wenn die Transitionsrelation a total ist, d.h. für jeden Zustand s es ein t gibt mit $s \xrightarrow{a} t$.

Axiom	Bedeutung
$[a]\varphi \to \langle a\rangle\varphi$	die Relation $\xrightarrow{a}$ ist total
$\langle a\rangle\varphi \to [a]\varphi$	die Relation $\xrightarrow{a}$ ist funktional
$[a]\varphi \to [a][a]\varphi$	die Relation $\xrightarrow{a}$ ist transitiv
$\langle a\rangle\varphi \to [a]\langle a\rangle\varphi$	die Relation $\xrightarrow{a}$ ist symmetrisch

Definition 2.2
Die Menge aller *Unterformeln* einer HML-Formel φ ist $Sub(\varphi)$, definiert als

$$\begin{aligned} Sub(q) &:= \{q\} \\ Sub(\varphi \vee \psi) &:= \{\varphi \vee \psi\} \cup Sub(\varphi) \cup Sub(\psi) \\ Sub(\varphi \wedge \psi) &:= \{\varphi \wedge \psi\} \cup Sub(\varphi) \cup Sub(\psi) \\ Sub(\neg\varphi) &:= \{\neg\varphi\} \cup Sub(\varphi) \\ Sub(\langle a\rangle\varphi) &:= \{\langle a\rangle\varphi\} \cup Sub(\varphi) \\ Sub([a]\varphi) &:= \{[a]\varphi\} \cup Sub(\varphi) \end{aligned}$$

Wir definieren außerdem die Größe einer Formel φ durch die Anzahl ihrer Unterformeln: $|\varphi| := |Sub(\varphi)|$.

Definition 2.3
Eine HML-Formel φ ist in *positiver Normalform*, wenn das Negationssymbol nur noch direkt vor Propositionen vorkommt, also wenn für alle $\psi \in Sub(\varphi)$ gilt: $\psi = \neg\psi'$ für ein ψ' impliziert $\psi' \in \mathcal{P}$.

Sei $\text{HML}^+ := \{\varphi \in \text{HML} \mid \varphi \text{ ist in positiver Normalform }\}$.

Definiere außerdem HML^- als die Menge aller HML-Formeln, die nur aus Propositionen, Disjunktionen, Negationen und dem Diamond-Operator bestehen.

Lemma 2.1
$\text{HML}^+ \equiv \text{HML} \equiv \text{HML}^-$

Beweis Übung. ∎

Es gilt sogar nicht nur, dass zu jeder Formel φ aus HML es Formeln $\varphi^+ \in \text{HML}^+$ und $\varphi^- \in \text{HML}^-$ gibt, so dass $\varphi \equiv \varphi^+ \equiv \varphi^-$, sondern darüberhinaus auch noch $|\varphi^-| = O(|\varphi|)$ und $|\varphi^+| = O(|\varphi|)$.

2.2. Bisimulationsinvarianz

Satz 2.1
Für alle Transitionssysteme $\mathcal{T}$ mit Zuständen $s, t \in \mathcal{S}$ gilt: Falls $s \sim t$ dann gilt für alle $\varphi \in \text{HML}$: $s \models \varphi$ gdw. $t \models \varphi$.

Beweis Per Induktion über den Formelaufbau. Wegen Lemma 2.1 beschränken wir uns auf Propositionen und die Konstrukte $\vee$, $\neg$ und $\langle a \rangle$.

Angenommen, $s \sim t$. Also existiert eine Bisimulation B mit $(s,t) \in B$. Offensichtlich gilt für alle $(s',t') \in B$: $s' \sim t'$.

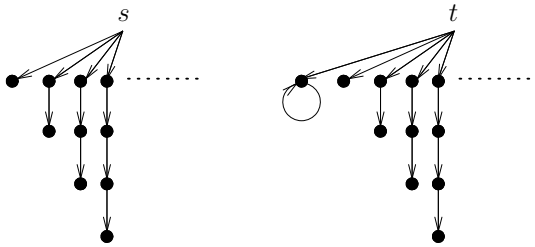
Fall $\varphi = q$. Aus $s \sim t$ folgt $\lambda(s) = \lambda(t)$. Somit auch $s \models q$ gdw. $q \in \lambda(s)$ gdw. $q \in \lambda(t)$ gdw. $t \models q$.

Fall $\varphi = \psi_1 \vee \psi_2$. $s \models \varphi$ gdw. es ein $i \in \{0,1\}$ gibt, so dass $s \models \psi_i$. Nach Voraussetzung ist dies der Fall, gdw. es ein i gibt, so dass $t \models \psi_i$. Dies ist der Fall, gdw. $t \models \varphi$.

Fall $\varphi = \neg\psi$. $s \models \varphi$ gdw. $s \not\models \psi$ gdw. (nach der Hypothese) $t \not\models \psi$ gdw. $t \models \varphi$.

Fall $\varphi = \langle a \rangle \psi$. Angenommen $s \models \varphi$. D.h. es gibt ein $s' \in \mathcal{S}$, so dass $s \xrightarrow{a} s'$ und $s' \models \psi$. Wegen $s \sim t$ gibt es auch ein $t' \in \mathcal{S}$ mit $t \xrightarrow{a} t'$ und es gilt $s' \sim t'$. Nach Induktionsvoraussetzung gilt also $s' \models \psi$ gdw. $t' \models \psi$. Somit also auch $t \models \varphi$. Die Umkehrung wird auf dieselbe Art bewiesen. ■

Die Umkehrung von Satz 2.1 gilt nicht. Betrachte die beiden Transitionssysteme



Es gilt $s \not\sim t$, aber es gibt kein $\varphi \in$ HML, so dass $s \models \varphi$ und $t \not\models \varphi$.

Korollar 2.1
HML hat die Baummodelleigenschaft.

Beweis Sei $\varphi \in$ HML erfüllbar. Also existiert ein Transitionssystem $\mathcal{T}$ mit einem Zustand s, so dass $\mathcal{T}, s \models \varphi$ gilt. Betrachte nun die Baumabwicklung $\mathcal{R}_{\mathcal{T}}(s)$ von $\mathcal{T}$ bzgl. s. Laut Lemma 1.2 gilt: $\mathcal{T}, s \sim \mathcal{R}_{\mathcal{T}}(s), s$. Laut Satz 2.1 gilt somit auch $\mathcal{R}_{\mathcal{T}}(s), s \models \varphi$. Somit hat φ ein Baummodell. ■

2.3. Entscheidungsverfahren und Komplexität

2.3.1. Das Model Checking Problem

Satz 2.2
Das Model Checking Problem für HML ist in P.

Beweis Dies zu zeigen gelingt am einfachsten durch Angabe eines alternierenden Algorithmus. Laut Lemma 2.1 reicht es aus, das Fragment HML^+ zu betrachten. Sei $\mathcal{T} = (\mathcal{S}, \{\xrightarrow{a} \mid a \in \Sigma\}, \lambda)$ ein Transitionssystem. Algorithmus MC-HML nimmt ein $s \in \mathcal{S}$ und ein $\varphi \in \text{HML}$ und entscheidet, ob $\mathcal{T}, s \models \varphi$ gilt oder nicht.

Algorithmus MC-HML(φ,s)
`case` φ `of`

q :	`if` $q \in \lambda(s)$ `then` *akzeptiere* `else` *verwerfe*
$\neg q$:	`if` $q \notin \lambda(s)$ `then` *akzeptiere* `else` *verwerfe*
$\psi_1 \vee \psi_2$:	wähle existentiell ein $i \in \{1,2\}$;
	MC-HML(ψ_i,s)
$\psi_1 \wedge \psi_2$:	wähle universell ein $i \in \{1,2\}$;
	MC-HML(ψ_i,s)
$\langle a \rangle \psi$:	wähle existentiell ein $t \in \mathcal{S}$;
	`if not` $s \xrightarrow{a} t$ `then` *verwerfe*;
	MC-HML(ψ,t)
$[a]\psi$:	wähle universell ein $t \in \mathcal{S}$;
	`if not` $s \xrightarrow{a} t$ `then` *akzeptiere*;
	MC-HML(ψ,t)

Dieser alternierende Algorithmus ist $O(\log n)$-platzbeschränkt. In jedem rekursiven Aufruf müssen lediglich die beiden Argumente gespeichert werden. Diese sind ein Zustand des Transitionssystems sowie eine Unterformel der Eingabe. Beides kann durch einen Zeiger logarithmischer Grösse realisiert werden.

Also ist das Model Checking Problem für HML in ALOGSPACE und somit, laut Satz A.3 auch in P. ∎

Als nächstes zeigen wir, dass dies bereits optimal ist.

Satz 2.3
Das Model Checking Problem für HML ist P-hart.

Beweis Von folgendem Problem ist bekannt, dass es P-hart ist: Gegeben ein gerichteter Graph $G = (V, E)$ und zwei Knoten $s, t \in V$, so dass V disjunkt zerlegt ist in $V_I \uplus V_{II}$. Zwei Spieler (I und II) verschieben nun einen Spielstein, der zu Beginn auf s liegt, entlang der Kanten E durch G. Liegt der Spielstein auf einem $v \in V_p$, dann darf Spieler p ihn weiterschieben. Spieler II gewinnt, wenn der Spielstein t erreicht. Spieler I gewinnt, wenn die Partie unendlich lang ist und dabei nie t erreicht, oder eine Sackgasse erreicht ist, die nicht t ist. Es ist P-hart zu entscheiden, ob Spieler II dieses Spiel gewinnen kann, d.h. immer t erreichen kann, egal wie Spieler I zieht. Dieses Problem wird auch *Alternating Directed Graph Reachability* (ADGR) gennant.

Wir reduzieren nun das Model Checking Problem für HML auf ADGR. Sei $G = (V, E)$ ein gerichteter Graph mit $s, t \in V$ und $V = V_I \uplus V_{II}$. Fasse diesen als knotenbeschriftetes Transitionssystem $\mathcal{T}_G = (V, \rightarrow, \lambda, s)$ über der Menge $\mathcal{P} = \{q_I, q_{II}, t\}$ auf, wobei für alle $v \in V$:

- $t \in \lambda(v)$ gdw. $v = t$,
- $q_I \in \lambda(v)$ gdw. $v \in V_I$,
- $q_{II} \in \lambda(v)$ gdw. $v \in V_{II}$.

Dann gilt: Spieler II gewinnt das Spiel auf G gdw. $\mathcal{T}_G, s \models \psi_{|V|}$, wobei

$$\begin{aligned} \psi_1 &:= q_t \\ \psi_{n+1} &:= q_t \vee \big((q_I \to \Box\psi_n) \wedge (q_{II} \to \Diamond\psi_n)\big) \end{aligned}$$

Dies basiert darauf, dass Spieler II jede Partie bereits mit höchstens $|V|$ vielen Zügen gewinnen kann, wenn er überhaupt gewinnen kann.

Da $\mathcal{T}_G$ und $\psi_{|V|}$ mit zusätzlichem Platz, der höchstens $O(\log |G|)$ ist, konstruiert werden können, ist das Model Checking Problem für HML ebenfalls P-hart. ∎

Korollar 2.2
Das Model Checking Problem für HML ist P-vollständig.

2.3.2. Das Erfüllbarkeitsproblem

Lemma 2.2
Für alle $\varphi, \psi_1, \psi_2 \in$ HML gilt: $(\psi_1 \vee \psi_2) \wedge \varphi$ erfüllbar gdw. es ein $i \in \{1, 2\}$ gibt, so dass $\psi_i \wedge \varphi$ erfüllbar ist.

Beweis Dies folgt sofort aus der Äquivalenz $(\psi_1 \vee \psi_2) \wedge \varphi \equiv (\psi_1 \wedge \varphi) \vee (\psi_2 \wedge \varphi)$. ∎

Definition 2.4
Ein HML-Literal ist jede Formel der Form q oder $\neg q$, wobei $q \in \mathcal{P}$. Eine Menge L von Literalen heißt *konsistent*, wenn es kein $q \in \mathcal{P}$ gibt, so dass $\{q, \neg q\} \subseteq L$.

Lemma 2.3
Sei $l_1, \ldots, l_k$ eine konsistente Menge von HML-Literalen und $n, m, k \in \mathbb{N}$. Für alle $\varphi_i, \psi_i \in$ HML gilt:

$$\langle a_1\rangle\varphi_1 \wedge \ldots \langle a_n\rangle\varphi_n \wedge [b_1]\psi_1 \wedge \ldots [b_m]\psi_m \wedge l_1 \ldots l_k$$

ist erfüllbar gdw. $\varphi_i \wedge \bigwedge\{\psi_j \mid b_j = a_i\}$ für alle $i = 1, \ldots, n$ erfüllbar ist.

Beweis Sei $\Phi := \langle a_1\rangle\varphi_1 \wedge \ldots [b_1]\psi_1 \wedge \ldots l_1 \ldots$, $L = \{l_1, \ldots, l_k\}$ und $\Phi_i := \varphi_i \wedge \bigwedge\{\psi_j \mid b_j = a_i\}$.

($\Rightarrow$). Angenommen Φ is erfüllbar, hat also ein Modell $\mathcal{T}, s$. Laut der Semantik von HML gibt es dann $t_1, \ldots, t_n$, so dass für alle $i = 1, \ldots, n$:

- $s \xrightarrow{a_i} t_i$,
- $\mathcal{T}, t_i \models \varphi_i$,
- $\mathcal{T}, t_i \models \psi_j$ für alle $j = 1, \ldots, m$ mit $b_j = a_i$.

Dann gilt aber auch $\mathcal{T}, t_i \models \Phi_i$ für alle $i = 1, \ldots, n$, was zu zeigen ist.

($\Leftarrow$). Sei nun Φ_i erfüllbar für alle $i = 1, \ldots, n$. D.h. es gibt Transitionssysteme $\mathcal{T}_i = (\mathcal{S}_i, \{\xrightarrow{a}_i \mid a \in \Sigma\}, \lambda_i, t_i)$, so dass $\mathcal{T}, t_i \models \Phi_i$ für alle $i = 1, \ldots, n$ gilt. Konstruiere nun ein Transitionssystem $\mathcal{T} = (\mathcal{S}, \{\xrightarrow{a} \mid a \in \Sigma\}, \lambda, s)$ wie folgt.

- $\mathcal{S} = \{s\} \uplus \biguplus_{i=1}^{n} \mathcal{S}_i$,
- $\xrightarrow{a} = \bigcup_{i=1}^{n} \xrightarrow{a}_i \ \cup \ \{(s, t_i) \mid a = a_i\}$,
- $\lambda(v) = \lambda_i(v)$, falls $v \in \mathcal{S}_i$, und $\lambda(s) = L \cap \mathcal{P}$.

Man sieht leicht, dass $\mathcal{T}, s \models \Phi$ gilt. Dies benutzt die Tatsache, dass L konsistent ist. ■

Satz 2.4
Das Erfüllbarkeitsproblem für HML ist in PSPACE.

Beweis Wir beschränken uns wieder auf das Fragment HML$^+$ und geben einen Algorithmus an, der zu einer gegebenen Formel $\varphi \in$ HML entscheidet, ob diese erfüllbar ist. Damit dieser rekursiv arbeiten kann, muss er ein generelleres Problem lösen. Algorithmus SAT-HML nimmt eine Menge Φ von HML-Formeln und entscheidet, ob $\bigwedge \Phi$ erfüllbar ist. Ausserdem ist er alternierend.

Algorithmus SAT-HML(Φ)
`case` Φ `of`
$\{q, \neg q\} \cup \Phi'$: *verwerfe*
$\{\psi_1 \wedge \psi_2\} \cup \Phi'$: SAT-HML($\{\psi_1, \psi_2\} \cup \Phi'$)
$\{\psi_1 \vee \psi_2\} \cup \Phi'$: wähle existentiell ein $i \in \{1, 2\}$;
SAT-HML($\{\psi_i\} \cup \Phi'$)
$\{\langle a_1\rangle\varphi_1, \ldots, \langle a_n\rangle\varphi_n, [b_1]\psi_1, \ldots, [b_m]\psi_m, l_1, \ldots, l_k\}$:
wähle universell ein $i \in \{1, \ldots, n\}$;
SAT-HML($\{\varphi_i\} \cup \{\psi_j \mid b_j = a_i\}$)
$\{[a_1]\psi_1, \ldots, [a_m]\psi_m, l_1, \ldots, l_k\}$:
akzeptiere

Dabei sind $l_1, \ldots, l_k$ Literale, und es gilt $n \geq 1$, $m \geq 0$ und $k \geq 0$.

Wir zeigen zuerst, dass SAT-HML(Φ) immer terminiert. Sei $\|\Phi\| := \sum_{\psi \in \Phi} \|\psi\|$, wobei

$$\begin{aligned} \|q\| = \|\neg q\| &:= 0 \qquad \text{für alle } q \in \mathcal{P} \\ \|\psi_1 \vee \psi_2\| = \|\psi_1 \wedge \psi_2\| &:= 1 + \|\psi_1\| + \|\psi_2\| \\ \|\langle a\rangle\psi\| = \|[a]\psi\| &:= 1 + \|\psi\| \end{aligned}$$

Dann gilt: Wenn $\|\Phi\| = 0$, dann trifft der letzte Fall zu, es sei denn, vorher trifft der erste Fall zu. Für alle rekursiven Aufrufe von SAT-HML(Φ') gilt: $0 \leq \|\Phi'\| < \|\Phi\|$. Da offensichtlich $\|\Phi\| \leq |\Phi|$ gilt, ist somit gleich auch gezeigt, dass SAT-HML in Polynomialzeit terminiert.

Als nächstes zeigen wir Vollständigkeit: Wenn SAT-HML(Φ) verwirft, dann ist $\bigwedge \Phi$ unerfüllbar. Dies geschieht durch Induktion über die Anzahl der $r(\Phi)$ Rekursionsschritte auf Eingabe Φ. Die Behauptung gilt offensichtlich für $r(\Phi) = 0$, denn dies ist der Fall gdw. $\Phi = \{q, \neg q\} \cup \Phi'$, womit $\bigwedge \Phi$ unerfüllbar ist. Sei nun $r(\Phi) > 0$ und die Aussage für $r' = r - 1$ bereits bewiesen. Dann gibt es drei Fälle für rekursive Aufrufe.

Konjunktion. In diesem Fall folgt die Behauptung trivialerweise.

Disjunktion. In diesem Fall wird existentiell gewählt. Daher wird nur verworfen, wenn beide mögliche rekursive Aufrufe auch verwerfen. Angenommen, dies ist der Fall. Nach der Induktionshypothese gilt dann: $\psi_1 \wedge \bigwedge \Phi'$ ist unerfüllbar und $\psi_2 \wedge \bigwedge \Phi'$ ist unerfüllbar. Nach Lemma 2.2 ist dann auch $(\psi_1 \vee \psi_2) \wedge \bigwedge \Phi'$ unerfüllbar.

Modale Formeln. In diesem Fall wird universell gewählt. Daher wird verworfen, wenn es eine mögliche Wahl gibt, bei der verworfen wird. Sei i diese Wahl. Nach der Hypothese gilt somit $\varphi_i \wedge \bigwedge\{\psi_j \mid b_j = a_i\}$ ist unerfüllbar. Laut Lemma 2.3 ist dann aber auch Φ unerfüllbar.

Dann bleibt noch Korrektheit zu zeigen: Wenn SAT-HML(Φ) akzeptiert, dann ist Φ erfüllbar. Dies geschiet auf dieselbe Art und Weise wie bei der Vollständigkeit. Bei dem Induktionsanfang sei bemerkt, dass SAT-HML die Formelmenge $\Phi = l_1, \ldots, l_k$ nur akzeptiert, wenn es kein $q \in \mathcal{P}$ gibt, so dass $\{q, \neg q\} \subseteq \Phi$. Ansonsten hätte die erste Klausel der Fallunterscheidung gegriffen.

Somit ist das Erfüllbarkeitsproblem für HML in APTIME. Laut Satz A.3 ist dies jedoch dasselbe wie PSPACE. ■

Beispiel 2.2
Als Anwendungsbeispiel zeigen wir, dass die Formel $[a](\varphi \to \psi) \to ([a]\varphi \to [a]\psi)$ ein Axiom ist, d.h. für alle $a \in \Sigma$ und alle HML-Formeln ψ_1 und ψ_2 allgemeingültig ist. Dazu definieren wir zuerst für jedes $\varphi \in \text{HML}^+$ ein $\overline{\varphi} \in HML^+$, so dass $\varphi \equiv \neg\overline{\varphi}$ gilt.

$$\begin{aligned}
\overline{q} &:= \neg q \\
\overline{\neg q} &:= q \\
\overline{\varphi_1 \vee \varphi_2} &:= \overline{\varphi_1} \wedge \overline{\varphi_2} \\
\overline{\varphi_1 \wedge \varphi_2} &:= \overline{\varphi_1} \vee \overline{\varphi_2} \\
\overline{\langle a \rangle \varphi} &:= [a]\overline{\varphi} \\
\overline{[a]\varphi} &:= \langle a \rangle \overline{\varphi}
\end{aligned}$$

Als nächstes zeigen wir – wieder durch Induktion über den Formelaufbau – dass SAT-HML jede Formelmenge Φ verwirft, wenn es ein φ gibt mit $\{\varphi, \overline{\varphi}\} \subseteq \Phi$. Für $\varphi = q$ oder $\varphi = \neg q$ ist dies offensichtlich der Fall.

Sei nun $\varphi = \varphi_1 \vee \varphi_2$, also $\{\varphi_1 \vee \varphi_2, \overline{\varphi_1} \wedge \overline{\varphi_2}\} \subseteq \Phi$. Bevor SAT-HML Φ akzeptieren könnte, müssen zuerst die Disjunktion und die Konjunktion aufgelöst werden. Dies erzeugt – je

nach existentieller Wahl – ein Argument $\Phi' = \{\varphi_i, \overline{\varphi_1}, \overline{\varphi_2}, \ldots\}$ für ein $i \in \{1, 2\}$. Nach Induktionsvoraussetzung werden diese jedoch beide verworfen. Der Fall von $\varphi = \varphi_1 \wedge \varphi_2$ wird genauso bewiesen.

Sei nun $\varphi = \langle a \rangle \psi$, also $\{\langle a \rangle \psi, [a]\overline{\psi}\} \subseteq \Phi$. Wie im vorigen Fall könnte Φ erst akzeptiert werden, wenn die modalen Operatoren abgestreift sind. Beachte, dass dies durch eine universelle Wahl geschieht, welche ein Argument $\Phi' = \{\psi, \overline{\psi}, \ldots\}$ erzeugt. Nach Voraussetzung wird Φ' jedoch verworfen, somit auch Φ. Der Fall von $\varphi = [a]\psi$ wird wieder genauso bewiesen.

Jetzt gilt $\models [a](\varphi \to \psi) \to ([a]\varphi \to [a]\psi)$ gdw. $\varphi' := [a](\overline{\varphi} \vee \psi) \wedge [a]\varphi \wedge \langle a \rangle \overline{\psi}$ unerfüllbar ist. Dazu reicht es laut Satz 2.4 zu zeigen, dass SAT-HML die Formel φ' verwirft. Zuerst werden die Konjunktionen ersetzt zu $\{[a]\overline{\varphi} \vee \psi), [a]\varphi, \langle a \rangle \overline{\psi}\}$. Dann bleibt nur ein universelle Wahl zu $\{\overline{\varphi} \vee \psi, \varphi, \overline{\psi}\}$. Damit SAT-HML dies verwirft, müssen sowohl $\{\overline{\varphi}, \varphi, \overline{\psi}\}$ als auch $\{\psi, \varphi, \overline{\varphi}\}$ verworfen werden, da bei Disjunktionen existentiell gewählt wird. Dies ist aber nach voriger Überlegung der Fall.

Abschließend zeigen wir wieder, dass obiges Komplexitätsresultat optimal ist.

Definition 2.5
Eine *quantifizierte, boolesche Formel* in Normalform über einer Menge $x_1, \ldots, x_n$ von Variablen ist ein

$$\Phi \;:=\; Q_n x_n. \ldots \exists x_3. \forall x_2. \exists x_1. \bigwedge_{i=1}^{m} \bigvee_{j=1}^{k_i} l_{i,j}$$

wobei die $l_{i,j}$ Literale über den Variablen $x_1, \ldots, x_n$ sind und $Q_n = \exists$, falls n ungerade, $Q_n = \forall$ sonst.

Die Gültigkeit eines solches Φ ist induktiv definiert durch:

0 und $\neg 1$ sind nicht gültig,

1 und $\neg 0$ sind gültig,

$\bigvee l_i$ ist gültig, wenn ein l_i gültig ist,

$\bigwedge \Phi_i$ ist gültig, wenn alle Φ_i gültig sind,

$\exists x_i.\Phi$ ist gültig, wenn $\Phi[0/x_i]$ oder $\Phi[1/x_i]$ gültig ist,

$\forall x_i.\Phi$ ist gültig, wenn $\Phi[0/x_i]$ und $\Phi[1/x_i]$ gültig sind.

Dabei bezeichnet $\Phi[b/x_i]$ die Formel, die aus Φ entsteht, wenn jedes Vorkommen von x_i in ihr durch b ersetzt wird.

Sei QBF := $\{\Phi \mid \Phi \text{ ist gültig }\}$.

Satz 2.5 (ohne Beweis)
Die Sprache QBF ist PSPACE-hart.

Satz 2.6
Das Erfüllbarkeitsproblem für HML ist PSPACE-hart.

Beweis Wir zeigen dies durch Reduktion auf QBF. Sei Φ eine quantifizierte, boolesche Formel

$$\Phi \;:=\; Q_n x_n. \ldots \exists x_3. \forall x_2. \exists x_1. \bigwedge_{i=1}^{m} \bigvee_{j=1}^{k_i} l_{i,j}$$

Dann konstruieren wir daraus eine HML-Formel φ über der Menge $\mathcal{P} = \{x_1, \ldots, x_n\}$ von Propositionen. Setze $\varphi := \psi_n \wedge \alpha$, wobei

$$\psi_i \;:=\; \begin{cases} \Diamond\big((x_i \vee \neg x_i) \wedge \psi_{i-1}\big) \;, & \text{falls } i \text{ ungerade} \\ \Diamond(x_i \wedge \psi_{i-1}) \wedge \Diamond(\neg x_i \wedge \psi_{i-1}) \;, & \text{falls } i \text{ gerade} \end{cases} \quad \text{für } i = 1, \ldots, n$$

$$\psi_0 \;:=\; \bigwedge_{i=1}^{m} \Diamond \psi_i'$$

$$\psi_i' \;:=\; \bigvee_{j=1}^{k_i} l_{i,j} \qquad \text{für alle } i = 1, \ldots, m$$

$$\alpha \;:=\; \bigwedge_{i=1}^{n} \bigwedge_{j=i}^{n+1} \Box^j\big((x_i \to \Box x_i) \wedge (\neg x_i \to \Box \neg x_i)\big)$$

$$\Box^0 \beta \;:=\; \beta$$

$$\Box^{k+1} \beta \;:=\; \Box \Box^k \beta$$

Zuerst sei bemerkt, dass die syntaktische Länge von φ zwar exponentiell in $|\Phi|$ ist, aber φ nur aus polynomiell in $|\Phi|$ vielen Unterformeln besteht.

Ein *Zeuge* für solch ein Φ ist ein Baum $\mathcal{T}$ der Tiefe $n+2$, dessen Knoten mit Mengen von Propositionen beschriftet sind. Wir schreiben $\lambda(s)$ für die Beschriftung eines Knotens s. Der Einfachheit halber gehen wir davon aus, dass der Baum die Ebenen $n, \ldots, -1$ hat mit der Wurzel auf der Ebene n und den Blättern auf der Ebene -1. Folgendes gilt:

1. Für alle ungeraden $i = 1, 3 \ldots, n$: Ein Knoten s auf Ebene i hat genau einen Nachfolger t und entweder $x_i \in \lambda(t)$ oder $x_i \notin \lambda(t)$ (was natürlich selbstverständlich ist).

2. Für alle geraden $i = 2, 4 \ldots, n$: Ein Knoten s auf Ebene i hat genau zwei Nachfolger t_1 und t_2 und $x_i \in \lambda(t_1)$ und $x_i \notin \lambda(t_2)$.

3. Ein Knoten auf Ebene 0 hat genau m Nachfolger $t_1, \ldots, t_m$. Für diese gilt: $\lambda(t_i) \supseteq \{x_h \mid x_h = l_{i,j} \text{ für ein } j \in \{1, \ldots, k_i\}\}$.

4. Für alle $i = 1, \ldots, n$ und alle $j = 0, \ldots, n-i$ und alle Knoten s auf Ebene j gilt: wenn $x_i \in \lambda(s)$, dann $x_i \in \lambda(t)$, und wenn $x_i \notin \lambda(s)$, dann $x_i \notin \lambda(t)$, für alle Nachfolger t von s.

Man überzeugt sich leicht, dass gilt: Wenn Φ gültig ist dann gibt es einen Zeugen von Φ. Ein solcher Zeuge $\mathcal{T}$ mit Wurzel s ist ein Transitionssystem und $\mathcal{T}, s \models \varphi$. Also ist φ erfüllbar.

Umgekehrt: Wenn φ erfüllbar ist, dann hat es ein Modell $\mathcal{T}, s$. Laut Korollar 2.1 kann man annehmen, dass $\mathcal{T}$ ein Baum mit Wurzel s ist. Aus diesem lässt sich dann ein Zeuge für Φ gewinnen, womit Φ gültig sein muss. ∎

2.4. Ausdrucksstärke

Wir gesehen haben, dass HML bereits einige gute Eigenschaften hat: Bisimulationsinvarianz, polynomielles Model Checking Problem, nicht allzu schweres Erfüllbarkeitsproblem, etc. Dennoch hat HML einen entschiedenen Nachteil: Die Logik ist viel zu ausdrucksschwach. Wir werden zeigen, dass bereits sehr elementare, aber wichtige Eigenschaften nicht in HML ausdrückbar sind. Der Grund dafür ist, dass eine HML-Formel nur "beschränkt tief in ein Transitionssystem schauen kann".

Definition 2.6
Die *modale Tiefe* $md(\varphi)$ einer HML-Formel φ ist induktiv definiert als:

$$\begin{aligned} md(q) &:= 0 \\ md(\varphi \vee \psi) = md(\varphi \wedge \psi) &:= \max\{md(\varphi), md(\psi)\} \\ md(\neg\varphi) &:= md(\varphi) \\ md(\langle a\rangle\varphi) = md([a]\varphi) &:= 1 + md(\varphi) \end{aligned}$$

Satz 2.7
Sei $q \in \mathcal{P}$ und $M := \{(\mathcal{T}, s) \mid \mathcal{T} = (\mathcal{S}, \to, \lambda, s_0)$ und es gibt einen Pfad $s_0, s_1, \ldots$ durch $\mathcal{T}$ mit $s_0 = s$ und ein $n \in \mathbb{N}$, so dass $q \in \lambda(s_n)\}$. Es gibt keine HML-Formel φ, so dass $[\![\varphi]\!] = M$.

Beweis Wir definieren zuerst zwei Familien $\mathcal{T}_i, \mathcal{T}_i'$, $i \in \mathbb{N}$, von Transitionssystemen. $\mathcal{T}_i = (\{0, \ldots, i{+}1\}, \to, \lambda, 0)$, wobei $j \to j{+}1$ für alle $j \leq i$ und $\lambda(i{+}1) = \{q\}$, $\lambda(j) = \emptyset$ für alle $j \leq i$. $\mathcal{T}_i'$ ist definiert wie $\mathcal{T}_i$ mit dem einzigen Unterschied, dass dort gilt $\lambda'(i{+}1) = \emptyset$.

Offensichtlich gelten die folgenden beiden Aussagen für alle $i \in \mathbb{N}$ und alle $j \leq i$.

1. $(\mathcal{T}_i, 0) \in M$ und $(\mathcal{T}_i', 0) \notin M$,
2. $\mathcal{T}_i, 0 \sim \mathcal{T}_{i+1}, 1$ und $\mathcal{T}_i', 0 \sim \mathcal{T}_{i+1}', 1$.

Als nächstes zeigen durch Induktion über den Formelaufbaeu von HML: Für alle $i \in \mathbb{N}$ und alle $\varphi \in$ HML mit $md(\varphi) = i$ gilt: $\mathcal{T}_i, 0 \models \varphi$ gdw. $\mathcal{T}_i', 0 \models \varphi$.

Fall $\varphi = p$ für ein $p \in \mathcal{P}$. Offensichtlich ist $md(p) = 0$. Dann gilt $\mathcal{T}_0, 0 \not\models p$ und $\mathcal{T}_0', 0 \not\models p$ wegen $\lambda(0) = \lambda'(0) = \emptyset$.

Fall $\varphi = \psi_1 \vee \psi_2$. Sei $i := md(\varphi)$. Dann gilt: $\mathcal{T}_i, 0 \models \varphi$ gdw. $\mathcal{T}_i, 0 \models \psi_1$ oder $\mathcal{T}_i, 0 \models \psi_2$ gdw. $\mathcal{T}_i', 0 \models \psi_1$ oder $\mathcal{T}_i', 0 \models \psi_2$ gdw. $\mathcal{T}_i', 0 \models \varphi$.

Die Fälle $\varphi = \psi_1 \wedge \psi_2$ und $\varphi = \neg\psi$ werden genauso gezeigt.

Fall $\varphi = \Diamond\psi$. Sei $i = md(\varphi)$, also $md(\psi) = i - 1$. Angenommen, $\mathcal{T}_i, 0 \models \varphi$. Da $0 \to 1$ in $\mathcal{T}_i$ gilt: $\mathcal{T}_i, 1 \models \psi$. Aber $\mathcal{T}_i, 1 \sim \mathcal{T}_{i-1}, 0$ laut Aussage (2) oben. Nach Satz 2.1 gilt somit: $\mathcal{T}_{i-1}, 0 \models \psi$. Die Induktionshypothese liefert $\mathcal{T}_{i-1}', 0 \models \psi$, und wiederum wegen (2) gilt $\mathcal{T}_i', 1 \models \psi$. Dann gilt aber auch $\mathcal{T}_i', 0 \models \psi$. Die Rückrichtung wird genauso gezeigt.

Der Fall wird $\varphi = [a]\psi$ wird genauso gezeigt wie der vorige Fall.

Die Aussage des Satzes ist dann einfach zu beweisen. Angenommen, es gäbe eine HML-Formel φ mit $[\![\varphi]\!] = M$. Dann hätte diese eine feste, modale Tiefe, z.B. $md(\varphi) = i$. Also gilt $\mathcal{T}_i, 0 \models \varphi$ und $\mathcal{T}_i', 0 \not\models \varphi$. Dies wiederspricht aber der eben bewiesenen Aussage, dass φ nicht $\mathcal{T}_i, 0$ und $\mathcal{T}_i', 0$ unterscheiden kann. ∎

2.5. Infinitäre Modallogik

Zur Erinnerung: Die Modellklasse einer HML-Formel ist unter Bisimulation abgeschlossen, aber nicht alle Modelle einer solchen Klasse sind bisimilar. Laut der Bemerkung nach Satz 2.1 und dem Beweis von Satz 2.7 ist das Problem, dass eine Formel nur "beschränkt tief in ein Transitionssystem schauen kann". Diese Beschränkung ließe sich leicht mithilfe von unendlich großen Formeln umgehen. So wird "es gibt einen Pfad, auf dem irgendwann einmal q gilt" z.B. durch die Formel $\bigvee_{i\in\mathbb{N}} \Diamond^i q$ ausgedrückt.

Selbst wenn man Formeln dieser Form in HML zuließe, würde die Rückrichtung von Satz 2.1 immer noch nicht gelten. Man braucht in der Tat *beliebig* große Disjunktionen und Konjunktionen. So erhält man die für praktische Zwecke wertlose, aber in der Theorie nützliche, infinitäre Modallogik ∞-HML. Ihre Syntax ist definiert durch

$$\varphi \quad ::= \quad q \mid \bigvee_{i\in I} \varphi_i \mid \bigwedge_{i\in I} \varphi_i \mid \langle a\rangle\varphi \mid [a]\varphi$$

wobei $q \in \mathit{Prop}$, $a \in \Sigma$ und I eine beliebige Indexmenge ist. Die Semantik einer ∞-HML-Formel φ ist definiert wie die einer herkömmlichen HML-Formel.
Wir betrachten die Logik ∞-HML nur aus Gründen der Bisimulationsinvarianz. Der erste Teil – "∞-HML-Formeln können bisimilare Strukturen nicht unterscheiden" wird genauso induktiv über den Formelaufbau wie Satz 2.1 bewiesen. Der trans-finite Fall der booleschen Operatoren macht dabei keine Schwierigkeiten.

Satz 2.8
Für alle Transitionssysteme $\mathcal{T}$ mit Zuständen $s, t \in \mathcal{S}$ gilt: Falls $s \sim t$ dann gilt für alle $\varphi \in \infty$-HML: $s \models \varphi$ gdw. $t \models \varphi$.

Mithilfe der infinitären Operatoren läßt sich jetzt auch die Rückrichtung dieses Satzes zeigen.

Satz 2.9
Für alle Transitionssysteme $\mathcal{T}$ mit Zuständen $s, t \in \mathcal{S}$ gilt: Wenn für alle $\varphi, \psi \in \infty$-HML gilt, dass $s \models \varphi$ gdw. $t \models \varphi$, dann gilt auch $s \sim t$.

Beweis Angenommen, $s \models \varphi$ gdw. $t \models \varphi$ für alle $\varphi \in$ HML. Wir konstruieren nun eine binäre Relation B auf $\mathcal{S}$ wie folgt.

$$B \quad := \quad \{(s', t') \mid \forall \varphi \in \infty\text{-HML} : s' \models \varphi \text{ gdw. } t' \models \varphi\}$$

Wir zeigen, dass B eine Bisimulation ist. Dafür nehmen wir das Gegenteil an. Es gebe also $(s', t') \in B$, so dass es ein $a \in \Sigma$ und ein $s'' \in \mathcal{S}$ gibt mit $s' \xrightarrow{a} s''$. Jetzt gibt es zwei Möglichkeiten, die Eigenschaften einer Bisimulation zu verletzen.

(1) Angenommen, es gibt kein t'' mit $t' \xrightarrow{a} t''$. Aber $s' \models \langle a \rangle \mathtt{tt}$ und somit auch $t' \models \langle a \rangle \mathtt{tt}$, was ein Widerspruch ist.

(2) Sei die Menge $T := \{t'' \mid t' \xrightarrow{a} t''\}$ nicht leer, aber $\{(s'', t'') \mid t'' \in T\}$ leer. D.h. für jedes $t'' \in T$ existiert eine Formel $\varphi_{t''} \in \infty$-HML, so dass nicht gilt: $s'' \models \varphi_{t''}$ gdw. $t'' \models \varphi_{t''}$. Da ∞-HML unter Negation abgeschlossen ist, können wir $s'' \models \varphi_{t''}$ und $t'' \not\models \varphi_{t''}$ voraussetzen. Betrachte jetzt die Formel $\psi := \bigwedge_{t'' \in T} \varphi_{t''}$. Offensichtlich gilt $s'' \models \psi$, aber $t'' \not\models \psi$ für jedes $t'' \in T$. Somit gilt auch $s' \models \langle a \rangle \psi$, aber $t' \not\models \langle a \rangle \psi$ im Widerspruch zu der Annahme, das $(s', t') \in B$.

Somit muss B eine Bismulation sein. Ausserdem gilt $(s, t) \in B$ wegen der Annahme, dass $s \models \varphi$ gdw. $t \models \varphi$ für alle $\varphi \in$ HML. Also $s \sim t$. ∎