Aufgabe 1:

a) $q_0 11001 \vdash 1 q_1 1001$

$\vdash 11 q_0 001$

$\vdash 110 q_1 01$

$\vdash 1100 q_0 1$

$\vdash 11001 q_1$

$\vdash 11001 q_2 1$

b) Die TM hängt an jedes Wort genau 1 weiteres Zeichen an, bei Wörter gerader Länge eine "0" und bei Wörter mit ungerader Länge eine "1".

$$f: \Sigma^* \to \Sigma^*, \; f(w) = \begin{cases} w0, & \text{falls } |w| \text{ gerade} \\ w1, & \text{falls } |w| \text{ ungerade} \end{cases}$$

Aufgabe 2:

Die TM kopiert zunächst die Eingabe auf das zweite Band (Zustand q_0). Dann wird der Kopf auf dem ersten Band auf das erste Zeichen positioniert (Zustand q_1). Auf dem ersten Band lesen wir das Eingabewort von links nach rechts, auf dem zweiten Band von rechts nach links und vergleichen zeichenweise (Zustand q_2).

$M = (Q, \Sigma, \Gamma, q_0, F, \delta)$, $Q = \{q_0, q_1, q_2, q_3\}$
$\Sigma = \{0,1\}$, $\Gamma = \{0,1,B\}$, $F = \{q_3\}$

$$\delta(q_0, (x,B)) = (q_0, (x,x), (R,R))$$
$$\delta(q_0, (B,B)) = (q_1, (B,B), (L,L))$$
$$\delta(q_1, (x,y)) = (q_1, (x,y), (L,N))$$
$$\delta(q_1, (B,y)) = (q_2, (B,y), (R,N))$$
$$\delta(q_2, (x,x)) = (q_2, (x,x), (R,L))$$
$$\delta(q_2, (B,B)) = (q_3, (B,B), (N,N))$$

für alle $x \in \{0,1\}$
$y \in \{0,1,B\}$

In allen anderen Situation stoppt die TM ($\delta(q_i, (a,b)) = (q_i, (a,b), (N,N))$) und akzeptiert nicht.

Aufgabe 3:

Annahme: H_S ist rekursiv

$\Rightarrow$ ex. stets haltende TM M die H_S entscheidet

Konstruiere TM M', die wie folgt arbeitet:

M' soll nicht halten, falls M akzeptiert und

M' soll akzeptieren, falls M nicht akzeptiert

Es gilt nun:

M' angesetzt auf $\langle M' \rangle$ hält

$\Leftrightarrow$ M akzeptiert $\langle M' \rangle$ nicht

$\Leftrightarrow$ $\langle M' \rangle \notin H_S$

$\Leftrightarrow$ M' angesetzt auf $\langle M' \rangle$ hält nicht $\lightning$ Widerspruch.

alternativ: $H_\varepsilon \leq H_S$

oder $H \leq H_S$

falsch: $H_S \leq H_\varepsilon$

$H_S \leq H$

Aufgabe 4:

a) falsch: $L_1 = \mathbb{B}^*$, $L_2 = H$

L_1 ist rekursiv, $L_2 \subseteq L_1$ nicht

b) wahr: U ist rekursiv aufzählbar, aber nicht rekursiv $\overset{2.7.3}{\Longrightarrow}$ $\overline{U}$ nicht rekursiv aufzählbar

c) falsch: Beispiel 2.8.3 ist nicht lösbar, die Sprache mit nur dieser Instanz ist aber entscheidbar

alternativ: Die Aussage macht keinen Sinn, da der Begriff der Entscheidbarkeit nicht für Instanzen, sondern für Sprachen (= Mengen von Instanzen) definiert ist.

d) falsch: $S := \{f: \Sigma^* \to \Sigma^* \mid f \text{ ist bijektiv}\}$

$\Rightarrow L(S) = L$, $(S \neq \emptyset, S \neq \Sigma^*)$

$\overset{\text{Rice}}{\Longrightarrow}$ L ist nicht rekursiv

e) wahr: $2^{2\log_2 n} = \left(2^{\log_2 n}\right)^2 = n^2 \in O(n^2)$.

Aufgabe 5:

Folgender Algorithmus berechnet eine nicht vergrößerbare Clique V' in $G=(V,E)$.

$V' := \emptyset$

$K := V$

while $K \neq \emptyset$ do

sei v ein beliebiger Knoten aus K (1 Schritt)

entferne v aus K $(K := K \setminus \{v\})$ (1 Schritt)

teste, ob $V' \cup \{v\}$ eine Clique ist (*)

falls ja: $V' := V' \cup \{v\}$

Laufzeit: sei $n := |V|$

– n Schleifendurchläufe

zu (*): $|V' \cup \{v\}| \leq n$

d.h. es werden maximal n^2 Knotenpaare getestet.

$\Rightarrow$ insgesamt $O(n^3)$ Tests, ob Knotenpaare durch Kanten verbunden sind

$\Rightarrow$ Laufzeit polynomial in n und damit polynomial in der Eingabelänge $S \in O(n^2)$.

Aufgabe 6:

Die NTM M' läuft zum rechten Ende der Eingabe w und rät eine mögliche Fortsetzung $\tilde{w}$, so dass $w\tilde{w} \in L$ (Zustand q_0). Danach läuft die NTM zum linken Ende der Eingabe (Zustand q_1) und verhält sich wie M.

Seien q_0 und q_1 zwei neue Zustände, q_0 ist der neue Anfangstand. Sei q_s der alte Anfangszustand.

$$\left.\begin{array}{l} (q_0, 0, q_0, 0, R) \in \delta \\ (q_0, 1, q_0, 1, R) \in \delta \end{array}\right\} \text{ zum rechten Ende laufen}$$

$$\left.\begin{array}{l} (q_0, B, q_0, 0, R) \in \delta \\ (q_0, B, q_0, 1, R) \in \delta \end{array}\right\} \tilde{w} \text{ raten}$$

$$(q_0, B, q_1, B, L) \in \delta \quad \text{fertig mit raten}$$

$$\left.\begin{array}{l} (q_1, 0, q_1, 0, L) \in \delta \\ (q_1, 1, q_1, 1, L) \in \delta \end{array}\right\} \text{ zum linken Ende laufen}$$

$$(q_1, B, q_s, B, R) \in \delta \quad \text{Kopf auf erstes Zeichen von } w \text{, und weiterarbeiten wie } M$$

Ansonsten bleibt δ unverändert.

Aufgabe 7:

Eine NTM M akzeptiert Wörter aus FÄRBBARKEIT wie folgt:

1) M rät zunächst eine Funktion $f: V \to \{1, \dots, k\}$
   (z.B. $|V|$ Zahlen aus $\{1, \dots, k\}$, getrennt durch #)

2) $\forall v \in V$:
   $\forall \tilde{v} \in V, \tilde{v} \neq v$:
   teste, ob $f(v) \neq f(\tilde{v})$ oder $(v, \tilde{v}) \notin E$

Falls alle Tests positiv ausgehen, wird die Eingabe akzeptiert.

Rechenzeit:

1) $|V|$ Zahlen à $\lfloor \log k \rfloor + 1$ Bits
   $\Rightarrow \quad \leq |V|^2$ Bits (k $\leq |V|$ kann man annehmen, sonst trivial)

2) $\leq |V|^2$ Tests,
   jeder Test in sicher in polynomialer Zeit möglich

insgesamt: Rechenzeit polynomial in $|V|$ und damit auch in der Eingabelänge $s \in O(n^2)$

$\Rightarrow$ FÄRBBARKEIT $\in NP$.

Aufgabe 8:

Sei $\mathcal{F} = \{F_1, \dots F_m\}$. $(F_i \subseteq U)$

Sei $V = \{v_1, \dots v_n\}$.

- SET COVER $\in NP$

Die NTM rät eine Indexmenge $I \subseteq \{1, \dots, m\}$ und testet, ob $|I| = k$ und $\bigcup_{i \in I} F_i = U$. Dies ist in polynomialer Zeit möglich.

- VERTEX COVER $\leq_p$ SET COVER

Sei $G = (V, E)$, $k \in \mathbb{N}$ eine Instanz für VERTEX COVER. Konstruiere eine Instanz für SET COVER wie folgt:

$m := n$, k bleibt unverändert, $U := E$,

$F_i := \{e \in E \mid e \text{ ist zu } v_i \text{ inzident}\}$, $1 \leq i \leq n$

(F_i enthält gerade die Kanten von v_i zu Knoten in der Adjazenzliste von v_i). Diese Transformation ist in Polynomialzeit machbar.

Korrektheit: $\forall w \in \Sigma^*: w \in V.C. \Leftrightarrow f(w) \in S.C.$

Zu geg. V' wähle $I := \{i \in \{1, \dots, m\} \mid i \in V'\}$.

Zu geg. I wähle $V' := \{v_i \in V \mid i \in I\}$. Dann gilt:

V' ist ein VERTEX COVER

$\Leftrightarrow \forall e \in E$: e ist zu einem Knoten aus V' inzident

$\Leftrightarrow \forall e \in U$: e ist in einer Menge F_i, $i \in I$ enthalten

$\Leftrightarrow I$ ist ein SET COVER

alternativ: 3-SAT $\leq_p$ SET COVER $\blacksquare$