Aufgabe 1

Zu zeigen ist die Transitivität von Polynomialzeitreduktionen: Aus $L \leq_p L'$ und $L' \leq_p L''$ folgt $L \leq_p L''$.

Es sei f_1 bzw. f_2 die Funktion, die $L \leq_p L'$ bzw. $L' \leq_p L''$ berechnet. Dann berechnet $f_2 \circ f_1$ eine Reduktion von L zu L'' (vgl. Übungsblatt 9, Aufgabe 2).

Zu zeigen bleibt: $f_2 \circ f_1$ ist in Polynomialzeit berechenbar. Die Funktion f_1 hat worst-case Laufzeit $c \cdot n^k$, für f_2 sei dies $d \cdot n^l$. Bei Eingabe eines Wortes w der Länge n benötigt die f_1 berechnende Turingmaschine höchstens $c \cdot n^k$ Zeit. Also hat $f_1(w)$ höchstens die Länge $c \cdot n^k$. Dieses Wort ist die Eingabe für f_2. Die Gesamtlaufzeit ist damit höchstens $d \cdot (c \cdot n^k)^l = dc^l \cdot n^{kl} =: e \cdot n^{kl}$, was ein Polynom in n ist. Also ist $f_1 \circ f_2$ in Polynomialzeit berechenbar. Insgesamt ist damit gezeigt, dass $\leq_p$ transitiv ist.

Aufgabe 2

(a) zu zeigen: $\lim_{n\to\infty} \frac{f(n)}{g(n)} = c, c \geq 0 \Rightarrow f(n) \in \mathcal{O}(g(n))$.

Sei $\lim_{n\to\infty} \frac{f(n)}{g(n)} = c, c \geq 0$. Dann gilt:

$$\forall \varepsilon > 0,\ \exists n_0 > 0,\ \forall n \geq n_0 :\ \left| \frac{f(n)}{g(n)} - c \right| < \varepsilon.$$

das gilt auch für ein $\varepsilon' > 0, \exists n_0 > 0, \forall n \geq n_0 :\ \varepsilon' - c < \frac{f(n)}{g(n)} < \varepsilon' + c$,
also $\exists c' = \varepsilon' + c > 0,\ \forall n \geq n_0 :\ f(n) < c' \cdot g(n)$,
es folgt $f(n) \in \mathcal{O}(g(n))$.

(b) zu zeigen: $\lim_{n\to\infty} \frac{f(n)}{g(n)} = 0 \Rightarrow f(n) \in o(g(n))$.

Sei $\lim_{n\to\infty} \frac{f(n)}{g(n)} = 0$. Dann gilt:

$$\forall \varepsilon > 0,\ \exists n_0 > 0,\ \forall n \geq n_0 :\ \frac{f(n)}{g(n)} < \varepsilon.$$

Das heisst, dass für $n \geq n_0, g(n) < g(n) \cdot \varepsilon$ ist.

Damit ist $f(n) \in o(g(n))$.

Aufgabe 3

(a) Es handelt sich nicht um eine Äquivalenz, da die Implikation
$f \in O(g) \Rightarrow lim_{n\to\infty} \frac{f(n)}{g(n)} = c \geq 0$
nicht zählt.
Gegenbeispiel:
$f(n) = sin(n) + 2$
$g(n) = 1$
$\Rightarrow f \in O(g)$, da $sin(n) + 2 \leq c \cdot 1$, für $c = 3$. Aber
$lim_{n\to\infty} \frac{sin(n)+2}{1} = lim_{n\to\infty} sin(n) + 2$ ist nicht definiert.
Um das Problem, dass der Grenzwert nicht definiert ist zu verhindern, benutze den limes superior anstatt des limes.
$\Rightarrow lim\ sup_{n\to\infty} \frac{f(n)}{g(n)} = c \geq 0 \Leftrightarrow f \in O(g)$

(b) Es handelt sich um eine Äquivalenz, da die Implikation
$f \in o(g) \Rightarrow lim_{n\to\infty} \frac{f(n)}{g(n)} = 0$
ebenfalls gilt.
Beweis:
$f \in O(g) \Rightarrow \forall c\ \exists n_0\ \forall n, n > n_0 : f(n) \leq c \cdot g(n)$
$f \in O(g) \Rightarrow \forall c\ \exists n_0\ \forall n, n > n_0 : \frac{f(n)}{g(n)} \leq c$

Da dies für beliebig kleine $c > 0$ gelten muss folgt, dass $\frac{f(n)}{g(n)}$ sich für große n beliebig nah an 0 nähert $\Rightarrow lim_{n\to\infty} \frac{f(n)}{g(n)} = 0$

Aufgabe 4

Um eine sinnvolle Hierarchie der Komplexität von Sprachen in Σ^* zu erreichen, wünschen wir das Reduktionsbeziehungen "$\preceq$" transitiv sind. Dies ist bei $\preceq_E$ nicht gegeben.

Seien $A \subset \Sigma^*$, $B \subset \Sigma^*$, und $C \subset \Sigma^*$ Sprachen mit $A \preceq_E B$, und $B \preceq_E C$. Weiter sei für ein Wort $x \in \Sigma^*$, $|x| = n$ und $t(f)$ die Laufzeit einer Funktion f.
Da $\preceq$ transitiv, gilt: Es gibt überall berechenbare Funktionen f, g sodass $x \in A \Leftrightarrow f(x) \in B \Leftrightarrow g(f(x)) \in C$. Weiter gilt f und g sind *exponentiell*.

$\Rightarrow \exists\ c, c' > 0 :\ f(x) \in O(2^{n^c}) \wedge\ g(x) \in O(2^{n^{c'}})$
$\Rightarrow \exists$ Konstanten $\ d, d' > 0 :\ t(f(x)) = d \cdot (2^{n^c})\ \wedge\ t(g(x)) = d' \cdot (2^{n^{c'}})$

Betrachte die Hintereinanderausführung von f und g, als

- lasse $f(x)$ für $d \cdot (2^{n^c})$ Schritte laufen, mit Resultat y.
- lasse $g(y)$ für $d' \cdot (2^{|y|^{c'}})$ Schritte Laufen, und gebe das Resultat aus.

Es gilt $|y| \leq f(x)| \leq d \cdot (2^{n^c})$. Somit ist im schlimmsten Fall $t(g(y)) = d' \cdot (2^{(d2^{n^c})^{c'}}) = d' \cdot 2^{O(2^n)}$.
Der Term 2^n wächst für grösserwerdendes n schneller als jedes Polynom in n.

$\Rightarrow \forall\ c > 0 : \limsup_{n\to\infty} \frac{2^{2^n}}{2^{n^c}} = \infty$
$\Rightarrow \nexists\ c > 0 : g(y) \in O(2^{n^c})$
$\Rightarrow g \circ f$ ist nicht exponentiell.

Damit ist $\preceq_E$ nicht transitiv.