7.1 TRUE: a, d, e, f. FALSE: de rest.

Nota bij (e): $3^n = 2^{O(n)}$ omdat $3 = 2^c$ waarbij $c = \log 3$, en dus $3^n = (2^c)^n = 2^{cn} = 2^{O(n)}$.

7.6 Te bewijzen: als A en B in P zitten, dan ook $A \cup B$, $A \cdot B$, en $\Sigma^* - A$.

$A \cup B$: op input x, test of $x \in A$ of $x \in B$. Beide testen kunnen in polynomiale tijd, het geheel is dus een som van twee polynomen, wat terug polynomiaal is.

$A \cdot B$: op input $x = x_1 \dots x_n$, test voor elke $i \in \{0, 1, \dots, n\}$ of $x_1 \dots x_i \in A$ en $x_{i+1} \dots x_n \in B$. Dit zijn $O(n)$ testen, elke test is polynomiaal, het geheel is dus $O(n) \times$ polynomiaal, dit is polynomiaal.

$\Sigma^* - A$: op input x, test of $x \in A$, indien ja, reject, indien nee, aanvaard. Aangezien $x \in A$ polynomiaal is, is dit polynomiaal.

7.13 Volgend algoritme is een polynomiaal algoritme voor A^*. Op input $y = y_1 \dots y_n$, bouwen we een booleaanse tabel $T[i, j]$ op voor $1 \leqslant i \leqslant j \leqslant n$, zodat $T[i, j] =$ "$y_i \dots y_j \in A^*$". Na afloop kan het antwoord dan afgelezen worden in $T[1, n]$. De opbouw verloopt als volgt:

1. Voor elke i, $T[i, i] :=$ true als de 1-letter string $y_i \in A$, anders $T[i, i] :=$ false. Inderdaad, een 1-letter string kan enkel in A^* zitten als hij in A zit.

2. We weten nu voor elke substring van lengte 1 of hij in A^* zit. Nu gaan we dit bepalen voor elke substring van lengte 2: we gaan dus $T[i, j]$ invullen voor alle (i, j) zodat $j = i + 1$. Daarna doen we het dan voor elke van lengte 3 (dus $j = i + 2$), en zo voort, tot we aan $T[1, n]$ zitten.

3. De algemene regel om $T[i, j]$ in te vullen is als volgt. Zoals zopas uitgelegd zorgen we ervoor dat $T[k + 1, j]$ reeds is ingevuld voor elke $k \in \{i, \dots, j - 1\}$. Merk nu op dat $y_i \dots y_j \in A^*$ als en slechts als $y_i \dots y_j \in A$, of er een $k \in \{i, \dots, j - 1\}$ bestaat zodat $y_i \dots y_k \in A$ en $y_{k+1} \dots y_j \in A^*$. Dit laatste kunnen we simpelweg zien in $T[k + 1, j]$. Er zijn $O(n)$ mogelijkheden voor k, en de testen in A zijn polynomiaal. Elke invulling is dus polynomiaal.

4. We vullen $O(n^2)$ cellen van de tabel in, en $O(n^2) \times$ polynomiaal is opnieuw polynomiaal.