5.10 Anta at A er gjenkjennbart. Da finnes det en Turingmaskin M som gjenkjenner A.
Definer f ved
$$f(w) = \langle M, w \rangle.$$
Det er åpenbart at f er beregnbar, og vi har at $w \in A \iff w \in L(M) \iff \langle M, w \rangle \in A_{TM} \iff f(w) \in A_{TM}$, som viser at $A \leq_m A_{TM}$.

Anta at $A \leq_m A_{TM}$. Siden A_{TM} er gjenkjennbart har vi, per Teorem 5.28, at A er gjenkjennbart.

5.11 Anta at A er avgjørbart. Definer f ved
$$f(w) = \begin{cases} 01 & \text{hvis } w \in A \\ 10 & \text{hvis } w \notin A \end{cases}$$
Siden A er avgjørbart, er f en beregnbar funksjon.
Hvis $w \in A$ er $f(w) = 01 \in 0^*1^*$, og hvis $w \notin A$ er $f(w) = 10 \notin 0^*1^*$, så f er en reduksjon fra A til 0^*1^*, så $A \leq_m 0^*1^*$.

Anta at $A \leq_m 0^*1^*$. Siden 0^*1^* er avgjørbart har vi, per teorem 5.22, at A er avgjørbart.

7.1 a) Sant c) Usant e) Sant
b) Usant d) Sant f) Sant

7.6 Vi viser at P er lukket under konkatenering.
La A og B være språk i P, med
$A \in TIME(n^{k_1})$ og $B \in TIME(n^{k_2})$.
Vi lager en Turingmaskin M som avgjør AB.
M = "På input $w = s_1 \dots s_n$:
1. For $i = 0, 1, \dots, n$
2. Test om $s_1 \dots s_i \in A$ og $s_{i+1} \dots s_n \in B$.
3. Hvis dette er tilfellet, aksepter.
4. Hvis dette steget nås, forkast."
(Obs: Når $i=0$ er $s_1 \dots s_i = \varepsilon$ og når $i=n$ er $s_{i+1} \dots s_n = \varepsilon$.)

Dette kjører i polynom tid siden steg 2 gjentas maksimalt $n+1 = O(n)$ ganger, og steg 2 tar maksimalt $O(n^{k_1}) + O(n^{k_2})$ steg, så M har en kjøretid på $O(n(n^{k_1} + n^{k_2}))$ steg. Siden $n(n^{k_1} + n^{k_2})$ er et polynom er $AB \in P$, så P er lukket under konkatenering.

7.13 Vi gir først noen definisjoner.
Vi sier at $\mathcal{D} = \{D \mid D$ er en disjunksjon av litteraler$\}$ er tilfredsstillbar hvis det finnes en tildeling slik at hvert element i $\mathcal{D}$ inneholder et sant litteral.

Det er tydelig at hvis $() \in \mathcal{D}$ er $\mathcal{D}$ ikke tilfredsstillbar og at φ er tilfredsstillbar hvis og bare hvis $\mathcal{C} = \{C_i \mid C_i$ er en klausul i $\varphi\}$ er tilfredsstillbar.

a) Vi viser først at resolusjon er sunn (sound).
Anta at $\mathcal{D}$ er tilfredsstillbar, og la
$D_a = (x \vee y_1 \vee \dots \vee y_k)$, $D_b = (\bar{x} \vee z_1 \vee \dots \vee z_\ell)$.
La ℓ være den tilfredsstillende tildelingen.
Hvis $\ell(x) = \text{TRUE}$, vil det finnes en i slik at $\ell(z_i) = \text{TRUE}$, så $\mathcal{D} \cup \{(y_1 \vee \dots \vee y_k \vee z_1 \vee \dots \vee z_\ell)\}$ vil være tilfredsstilt av ℓ. Tilsvarende hvis $\ell(x) = \text{FALSE}$.
Altså hvis en mengde $\mathcal{D}$ er tilfredsstillbar, vil man fortsatt ha en tilfredsstillbar mengde etter et resolusjonssteg.

Så hvis φ er tilfredsstillbar vil $\mathcal{C}$ være tilfredsstillbar, og etter å gjøre et vilkårlig antall resolusjonssteg vil vi fortsatt ha en tilfredsstillbar mengde som ikke inneholder $()$, så φ vil ikke bli deklarert utilfredsstillbar.

For komplettheitsbeviset, se detaljer på nettsiden. Idéen går ut på å vise at en mengde klausuler som er utilfredsstillbar der du ikke kan gjøre flere resolusjonssteg må inneholde ().

b) Hvis du gjør et resolusjonssteg med maksimalt to litteraler i hver klausul vil den nye klausulen også ha maksimalt to litteraler. Så ved å gjøre resolusjonssteg på 2SAT vil alle klausuler ha lengde ≤ 2. La n være lengden på input φ. Da har φ $k \leq n$ variable. Antall klausuler av lengde 2 er $(2k)(2k-1)/2 = O(k^2)$. Antall av lengde 1 er $2k = O(k)$. Antall av lengde 0 er 1. Totalt er det $O(k^2)$ klausuler som kan bli lagt til. Siden hvert resolusjonssteg legger til en ny klausul vil det være $O(k^2)$ resolusjonssteg. Siden et resolusjonssteg kan gjøres ved å velge to klausuler ($O(k^4)$), beregne resolusjonen ($O(1)$), og sjekke om den nye klausulen finnes i mengden ($O(k^2)$), vil dette ha en kjøretid på $O(k^8)$, så algoritmen kjører i polynomtid, så $2SAT \in P$.