5.1 Vi vet at ALL_{CFG} er uavgjørbart (Teorem 5.13)
Vi viser at hvis EQ_{CFG} er avgjørbart, er ALL_{CFG} også avgjørbart.

Anta at EQ_{CFG} er avgjørbart, og la D være en Turingmaskin som avgjør EQ_{CFG}. Vi konstruerer en Turingmaskin E som avgjør ALL_{CFG}.

E = "På input $\langle G \rangle$:
1. Konstruer en CFG H slik at $L(H) = \Sigma^*$
2. Kjør D på $\langle G, H \rangle$.
3. Hvis D aksepterer; aksepter. Hvis ikke; forkast."

Nå aksepterer E alle input $\langle G \rangle$ slik at $L(G) = L(H) = \Sigma^*$, altså er $L(E) = ALL_{CFG}$, så ALL_{CFG}. Siden dette er en motsigelse, er EQ_{CFG} uavgjørbart.

5.2 La A være en Turingmaskin som avgjør A_{CFG}.
Vi konstruerer B som gjenkjenner $\overline{EQ_{CFG}}$.

B = "1. Hvis input ikke er på formen $\langle G, H \rangle$ der G og H er CFG-er, aksepter.
2. Hvis input er $\langle G, H \rangle$:
3. For $i = 1, 2, \ldots$
4. Kjør A på $\langle G, s_i \rangle$ og $\langle H, s_i \rangle$.
5. Hvis A ga forskjellig resultat; aksepter."

B aksepterer $\langle G, H \rangle$ nøyaktig hvis det finnes w slik at $w \in L(G) \iff w \notin L(H)$, så $L(G) \neq L(H)$, så $\langle G, H \rangle \in \overline{EQ_{CFG}}$.

5.17 La $P = \{\langle M \rangle \mid M$ er en TM$\}$. Da er P en egenskap ved Turingmaskiners språk, men triviell. P er åpenbart avgjørbar.

La A være en Turingmaskin, og la $P = \{\langle A \rangle\}$. P er ikke-triviell, men er ikke en egenskap ved Turingmaskiners språk. Siden P er endelig er P avgjørbar.

Dermed vil du ikke nødvendigvis få et uavgjørbart språk hvis du fjerner ett av kravene i Problem 5.16, så begge kravene er nødvendige.

5.19 Vi lager først en Turingmaskin som sjekker om, gitt x, prosessen til slutt når 1.

M_1 = "På input x:
1. Sett $n = x$
2. Gjenta:
3. Hvis $n = 1$, aksepter.
4. Sett $n = f(n)$."

Vi konstruerer så en Turingmaskin som bruker H og M_1 til å sjekke om det finnes en n slik at prosessen aldri når 1.

M_2 = "Ignorer input
1. For $i = 1, 2, \ldots$
2. Kjør H på input $\langle M_1, i \rangle$.
3. Hvis H forkaster, aksepter."

Vi bruker så H og M_2 til å lage en avgjører.

M_3 = "Ignorer input:

1. Kjør H på $\langle M_2, \varepsilon \rangle$.
2. Hvis H aksepterer; forkast.
   Hvis H forkaster; aksepter."

M_3 aksepterer hvis og bare hvis for alle n, prosessen som begynner med n alltid når 1.

5.33 Anta at vi har en instans $\left\{ \left[\frac{a_1}{b_1} \right], \ldots, \left[\frac{a_k}{b_k} \right] \right\}$ av PCP over alfabetet $\Sigma = \{1\}$. Da er $a_i = 1^{n_i}$, $b_i = 1^{m_i}$ for noen bestemte n_i, m_i. Vi har fire tilfeller:

1. Det finnes en j slik at $n_j = m_j$.
2. For alle i er $n_i > m_i$.
3. For alle i er $m_i > n_i$.
4. Det finnes j_1 og j_2 slik at $n_{j_1} < m_{j_1}$ og $m_{j_2} < n_{j_2}$.

I tilfelle 1 er $\left[\frac{a_j}{b_j} \right]$ en match.

I tilfellene 2 og 3 finnes det åpenbart ingen match.

I tilfelle 4, sett $k_1 = m_{j_1} - n_{j_1}$ og $k_2 = n_{j_2} - m_{j_2}$.

Da er $\left[\frac{a_{j1}}{b_{j1}} \right]^{k_2} \left[\frac{a_{j2}}{b_{j2}} \right]^{k_1} = \left[\frac{1^{n_{j1}}}{1^{n_{j1}+k_1}} \right]^{k_2} \left[\frac{1^{m_{j2}+k_2}}{1^{m_{j2}}} \right]^{k_1}$

$$= \left[\frac{1^{n_{j1}k_2 + m_{j2}k_1 + k_2 k_1}}{1^{n_{j1}k_2 + k_1 k_2 + m_{j2} k_1}} \right],$$ så dette er en match.

Siden det er åpenbart avgjørbart å avgjøre hvilket tilfelle instansen oppfyller, er PCP avgjørbart over $\Sigma = \{1\}$.