1. **5.1.** Show that $EQ_{CFG}$ is undecidable.  
Assume it is decidable. Then there exists a TM $R$ that decides $EQ_{CFG}$. A construction of TM $S$ that uses $R$ to decide $ALL_{CFG}$ follows. 

$S = \text{On input } \langle G \rangle$, where $G$ is a CFG:

1. Run $R$ on input $\langle G, G_1 \rangle$ where $G_1$ is a CFG that generates $\Sigma^*$.
2. If $R$ accepts, accept; if $R$ rejects, reject.

If $R$ decides $EQ_{CFG}$, $S$ decides $ALL_{CFG}$. But $ALL_{CFG}$ is undecidable, so $EQ_{CFG}$ must also be undecidable.

2. **5.2.** Show that $EQ_{CFG}$ is co-Turing-recognizable. This implies that we must show that $\overline{EQ_{CFG}}$ is Turing recognizable. Let TM $R$ be a decider for the known decidable language $A_{CFG}$. We use this fact to construct a TM $M$ to recognize $\overline{EQ_{CFG}}$. 

$M = \text{On input } \langle G, G_1 \rangle$ where $G$ and $G_1$ are CFGs:

1. Convert $G$ and $G_1$ to equivalent CNF CFGs $C$ and $C_1$, respectively.
2. For $i \leftarrow 0, \infty$:
   For each string $s$ of length $i$ generated by $C$:
   1. Run $R$ on input $\langle C_1, s \rangle$.
   2. If $R$ rejects, accept.

3. **5.3.**

4. **5.4.** - Answer: no. Consider the non-regular language $A = \{0^n1^n|n \geq 0\}$ and the regular language $B = \epsilon$. The map $f : \Sigma^* \rightarrow \Sigma^*$ is defined as follows: for $x \in \Sigma^*$, if $x = 0y1$ then strip the leading 0 and trailing 1, and continue recursively until either $y$ does not satisfy this form or $y = \epsilon$. Then the map $f$ satisfies the reducibility definition and maps a non-regular language into a regular one.

5. **5.9.** For each $M$ and $w$ construct a TM $S$ so that : for every input other than $w$ or $w^R$, $S$ rejects. For input $x = w$ or $w^R$ it accepts if $M$ accepts $w$. This machine accepts a string iff it accepts the reverse of the string, so its language can be decided by the assumed decider, say $T$. If $T$ accepts, $S$ accepts; if $T$ rejects, $S$ rejects. We now have a decider for $A_{TM}$ - contradiction.

6. **5.12.** First modify $M$ so that all of its transitions that WRITE a blank over a non-blank are replaced by writing a new, different symbol. Then the modified $M$ will accept exactly the same set of strings without ever writing a blank over a non-blank. The "encompassing TM" - $S$ of the last problem must:

   a) change each $M$ as indicated,
   b) run $M$ and at all acceptances make sure that a blank is now written over a non-blank: you may have to move left one or more positions, but, since the initial input was NOT blank and you never overwrote a blank with a non-blank, there must be at least one non-blank for you to overwrite.
The submachine that carries out a) and b) is the desired machine $R$. Now run this machine through the $T$ that is supposed to decide the blank/non-blank. Any machine $R$ that writes a blank must have resulted from a modified $M$ stopping; any machine $R$ that is rejected (did not write a blank) must have come from an $M$ that did NOT stop. So we have a decision for $A_{TM}$.

7. **5.17.** Clearly, if you have at least one domino with equal top as bottom, the PCP is decidable for that instance of PCP (justs pick that domino). If all tops are larger than all bottoms (or vice versa), you can also reject (why?). The only case of interest is when at least one domino has a larger top than bottom, and at least one has larger bottom than top. Think in terms of taking enough multiples of each so that the total bottoms equal the total tops - easily shown (so... show: you don’t need to construct a TM to do so, although it can be done).