1 5.1

Solution. in this problem we consider problems related to context free grammars $\text{All}_{\text{CFG}} = \{\langle G\rangle | G$ is a CFG and L(G)=$\Sigma^*\}$
be the language of all CFG_s that accepts all strings Also let
$\text{EQ}_{\text{CFG}} = \{\langle < G_1 G_2 >\rangle | G_1 and G_2 are \text{CFG}_S$ and $L(G_1) = L(G_2)\}$
be the language of equivalentCFG_s
it is provided in the book that All_{CFG} is undecidable
Use the fact this fact that EQ_{CFG} is undecidable ■

2 5.4

Solution. No. For example, define the languages A =$\{0^n 1^n | n \geq 0\}$ and B = $\{1\}$,
both over the alphabet $\Sigma = \{0, 1\}$. Define the function f : $\Sigma^* -> \Sigma^*$ as
f(w) = 1 if w ∈ A, or 0 if w ∉ A Observe that A is a context-free language, so it is also Turing-decidable. Thus, f is a computable function. Also, w ∈ A if and only if f(w) = 1, which is true if and only if f(w) ∈ B. Hence, A$\leq_m$ B. Language A is nonregular, but B is regular since it is finite. ■

3 5.17

Solution. In the first stage, M checks for a single domino which forms a match. In the second stage, M looks for two dominos which form a match. If it finds such a pair, it can construct a match by picking $(b_j - a_j)$ copies of the ith domino, putting them together with $(a_i - b_i)$ copies of the jth domino. This construction has $a_i(b_j - a_j) + a_j(a_i - b_i) = a_i b_j - a_j b_i$ 1's on top, and $b_i(b_j - a_j) + b_j(a_i - b_i) = a_i b_j - a_j b_i$ 1's on the bottom. If neither stages of M accept, the problem instance contains dominos with all upper parts having more/less 1's than the lower parts. In such a case, no match exists and therefore M rejects. ■

4 5.24

Solution. Let A be the language $\{\langle M, x\rangle |$ M is a TM and M does not accept x}. It is easy to check that A is not Turing-recognizable (by reduction from A_{TM}).We first show how to reduce A to J. This is done by the reduction function f(w) = 1w, so that w is in A if and only if f(w) is in J. Obviously, the function f is computable. As A is not Turing-recognizable, J is not Turing-recognizable. We next show how to reduce reduce ATM to J. This is done by the reduction function g(w) = 0w, so that w is in ATM if and only if g(w) is in J. Again, the function g is computable. Since ATM is not Turing-recognizable, J is not Turing-recognizable. ■