1. a) \[
\begin{bmatrix}
00 & 1 & 0 & 0 & 1 \\
00 & 1 & 1 & 0 & 0 \\
00 & 1 & 0 & 1 & 0 \\
1 & 1 & 0 & 1 & 0 \\
1 & 0 & 1 & 1 & 0
\end{bmatrix}
\]

b) i) Match:
\[
\begin{bmatrix}
1 \\
101 \\
1 \\
011 \\
1
\end{bmatrix}
\]

ii) This instance has the following property for all dominoes \[
\begin{bmatrix}
X \\
Y
\end{bmatrix}
\]

x is not a suffix of y and y is not a suffix of x.
The instance has no match because no domino could end a match.
Consider PCP instance \((u_1, v_1), \ldots, (u_m, v_m)\)

Define

\[ L_u = \{ u_1, \ldots, u_m \}^+, \quad L_v = \{ v_1, \ldots, v_m \}^+ \]

The instance has a non-constrained solution iff

\[ L_u \cap L_v \neq \emptyset \]

Decision for \(L_{ncpcp}\):

\[ Q = "\text{On input }\langle (u_1, v_1), \ldots, (u_m, v_m) \rangle:\]

1. Using the regex-to-DFA conversion algorithm, construct DFA \(A\) for \(L_u\) and DFA \(B\) for \(L_v\).

2. Construct DFA \(C\) such that \(L(C) = L(A \cup \overline{B})\)

3. Using decision for DFA-emptiness decide whether \(L(C) = \emptyset\).
   - If "yes" reject, and if "no" accept"
3. a) Yes, $A = \text{ECFG}_A \leq_{\text{m}} \text{ECFG}$ via the identity function. ECFG is decidable.

b) Yes, $B = \text{ECFG}_B$. ECFG $\leq_{\text{m}} B$ via the identity function.

c) No. ECFG is decidable so, according to Theorem 5.22, $C \leq_{\text{m}} \text{ECFG}$ implies that C is decidable.

d) Yes. ECFG $\leq_{\text{m}} \text{ATM}$ using following reduction $f$. Let $M_1$ be a TM that accepts string $b$ and rejects $bb$. Define $f$ by setting:

$$f(\langle G \rangle) = \begin{cases} \langle M_1, b \rangle & \text{if } L(G) = \emptyset \\ \langle M_1, bb \rangle & \text{if } L(G) \neq \emptyset \end{cases}$$

$f$ is a computable function because ECFG is decidable.

$f$ mapping reduces ECFG to ATM because

$$\begin{cases} L(G) = \emptyset \text{ implies } f(\langle G \rangle) \in \text{ATM} \\ L(G) \neq \emptyset \text{ implies } f(\langle G \rangle) \notin \text{ATM} \end{cases}$$

(*) If $w$ does not encode a grammar, define $f(w) = \langle M_1, bb \rangle$. 
4.  

a) TRUE

b) FALSE

c) TRUE

d) TRUE

e) FALSE

f) NO (Explanation: $2^{n+5} = 32 \cdot 2^n$ so $2^n$ is not asymptotically less than $2^{n+5}$.)

g) NO

h) YES

i) YES

j) YES
5. Decide for ALL DFA:

\[ Q = \text{"On in } \langle A \rangle \text{ where } A \text{ is a DFA } \]

1. Using graph reachability algorithm find the set of reachable states of \( A \).

2. If some reachable state is non-final or some reachable state has an undefined transition, reject. Otherwise accept.

The algorithm works because a DFA accepts all strings \( \text{iff} \) (all reachable states are final and all transitions from reachable states are defined).

The algorithm works in polynomial time because graph reachability is in \( \mathbb{P} \).

(Theorem 7.14)
Decider for TRIANGLE:

\[ M = \text{"On input } G = (V, E) \text{ where } E \text{ contains ordered pairs of vertices :}

1. For all \((x, y, z) \in V \times V \times V:\)
   \[ (x, y, z) \in E \text{ and } (y, z, x) \in E \text{ if and only if } (x, y) \in E \text{ and } (y, z) \in E \text{ and } (x, z) \in E, \text{ then accept.}

2. If did not accept up to this line, reject."}

Time used:
The for-loop repeats \( |V|^3 \) times when \(|V| \) is at most the length of input. Each iteration of the for-loop can be done by scanning through the list of edges, and checking equality of names of some vertices.

On a TM, \( |V|^3 \) can be done in time \( O(n^2) \) where \( n \) is the length of the input. (More precise: \( O(n \cdot \log n) \)).
The total running time is \( O(n^5) \).
6. b) Using a similar algorithm for CLIQUE means that we have to go through $k$-tuples of elements of $V$ where $k$ is given as input. The for-loop repeats $|V|^{k^2}$ times and this is not polynomial as a function of the input length.