Solution 3
by Rui Shi

[Question 1]:
Solution:
L′ be the language of matching a and b's, including the empty string. In the following, we are to prove that the grammar G exactly generate the language L′ as described.

(⇒) By induction on the derivation of w generated by G. Consider the first rule applied in the derivation:
   - S → ε. Straightforward.
   - S → aSb: We have that w = aw1b, where S → w1 is a derivable. By I.H. w1 ∈ L′, i.e. w1 is a legal word with matching parenthesis a and b's. Therefore, it implies aw1b ∈ L′ as well.
   - S → SS: We have that w = w1w2, where S → w1 and S → w2 are both derivable. By applying the I.H. twice we have w1 ∈ L′ and w2 ∈ L′, which implies w1w2 ∈ L′ as well.

(⇐) By induction of the length of w:
   - w = ε: Straightforward.
   - Let axb be the shortest prefix of w having a matched number of left and right a and b’s. Then w can be written as axby, where both x and y are balanced. By I.H., both of x and y are derivable from S. Namely, we can find a derivation of the form S → aSbS →* axbS →* axby = w.

[Question 2]:
Solution:
Since L_{half} is regular, there exists a DFA D which recognizes L_{half}. Then we can construct another DFA D_{half} which recognizes L_{half} as follows:
   - For each state q_i in D, there states in form of (q_i, S) in D_{half}, where S is a subset of states in D.
   - The initial state of D_{half} is (q_0, F), where F is the set which contains all the accept states of D.
   - For each transition (q_i, x, q_j) of D, we add a transition: ((q_i, S_1), x, (q_j, S_2)) into D_{half}, where S_2 contains all the states in D which could make a one step transition to some state in S_1.
   - For each accept state q_a in D, we mark the state (q_a, I) as an accept state, where I is a subset of states in D which contains the initial state q_0.
Intuitively, the first component of a state does exactly what a state in $D$ does and the second component $S$ keeps a set of states that keep track if there exists a string $y$ which can be accepted by $D$ within exactly same steps starting from one of state in $S$. If $D_{half}$ accepts a string $s$, it means there exists a string $y$ with same length which can be accepted by $D$. Therefore, $L_{half}$ is regular.

[Question 3]:

**Solution:**

The following grammar could generate the language described.

$$S \rightarrow D1S_0 \mid D0S_1 \mid D\# \mid \#D$$
$$S_0 \rightarrow 1S_01 \mid 1S_00 \mid 0S_01 \mid 0S_00 \mid \#D0$$
$$S_1 \rightarrow 1S_11 \mid 1S_10 \mid 0S_11 \mid 0S_10 \mid \#D1$$
$$D \rightarrow D1 \mid D0 \mid \epsilon$$

To construct a legal word $x\#y$ in the language, the basic idea is to make the $i$th digit of $x$ be different from that of $y$.

[Question 4]:

**Solution:**

The following grammar could generate the language described.

$$S \rightarrow S_0S_1 \mid S_1S_0$$
$$S_0 \rightarrow DS_0D \mid 0$$
$$S_1 \rightarrow DS_1D \mid 1$$
$$D \rightarrow 0 \mid 1$$

Similar techniques as the above apply.

The proof techniques of question 3 and 4 follows from those of question 1.