Some Answers on: Undecidability

G1: (a) Suppose we can decide if two PDAs accept the same language. Then on input PDA M, build a simple PDA N that accepts all strings of that alphabet. Then test whether M and N accept the same strings. This answers the question of whether M accepts all strings. Which contradicts what is known.

(b) Take N and convert to Chomsky Normal Form. Create all strings in increasing order of length, checking each to see if generated by N. Stop and accept if find a string ***not*** generated by N.

(c) No.

G2: (a) False. (S_{tm} is not even r.e.)

(b) True. (Since FAs are guaranteed to halt, we can simulate them.)

(c) True. (Since a program can be represented by a finite string.)

(d) True.

G3: Assume that L is recursive (but neither empty nor Σ^*). Then L is accepted by some TM M that always halts. Let x be any string in L, let y be any string in $\bar{L}$. Then the reduction is: "On input w, run M on w. If M accepts w then output y, else output x."

Assume that L is the language of TM M and that L reduces to $\bar{L}$ by reduction f. Then the decider for L is: "On input w, calculate $f(w)$. Run M on both w and $f(w)$ in parallel. If M accepts w then accept; if M accepts $f(w)$ then reject." This works, since if $w \in L$, then M will halt and accept w; and if $w \notin L$, then $f(w) \in L$, so M will halt and accept $f(w)$.

G4: (a) Acceptance problem for TM.
(b) Does not exist.
(c) The integers
(d) The real numbers
(e) $0^n 1^n 2^n$