Decidable languages

1) • Convert PDA to CFG
• p = pumping length of CFG
• Create regular expression to hold all string greater than p
• Take intersection of the CFG & the regular language this will yeild a CFG (G)
• Check $L(G) = \emptyset$ using decider L
If L accepts, then reject
else accept

2) • R & S are regular expressions
• $L(R)$ and $L(S)$ will be regular
• if $L_n \subseteq L_m$ then $L_n - L_m$ is an empty language
• $L(R) - L(S) = L(R) \cap L(S)'$
• $L(R) - L(S)$ is regular since regular langages are closed under complemetation & intersection
• There exists a DFA for $L(R) - L(S)$
Therefore A is decidable since DFA can recognize $\{\emptyset\}$

2) Let P be the set of all PDAs
Now we must construct a language L, to check for useless states

$L = \{w \mid w \in P$ where w is a PDA with useless states$\}$

We can construct a Turing machine where a PDAs are taken as the inputs & checks for useless states.

Since Turing machine wont enter a loop state the machine can check each input PDA for useless states.

Since we can use a Turing machine to check for useless states it is decidable

3) A is Turing recognizable

E = Enumerator that enumerates A

M_i = i^{th} output of E

let $S_1, S_2, S_3, S_4, S_5 \ldots S_k$ be strings in $\{0,1\}^*$

Turing maching T' will be defined as:

If input $w \notin \{0,1\}^*$ then reject,
else $w = S_i$
enumerate M on E
Run M_i on w
If M_i accepts then reject
  else accept

Let T = decider
Let D = language T decides

Since D behaves opposite to M_i for all i D is a decidable language not in A