A. Suppose (for the sake of contradiction) that A is context-free. Let p be the pumping length of A, and let w be the string

$$1^p 0^p = 1^p 0^p \cdot 1.$$

Then $w \in A$ and $|w| \geq p$, so the pumping lemma implies that $w = uvxyz$, where $vy \neq \epsilon$, $|vxy| \leq p$, and $uv^i xy^i z \in A$ for all $i \geq 0$. There are three basic cases. If the = is contained in vy, then pumping v and y produces a string with multiple =s, which is certainly not in the language. If the = is contained in uz, then pumping v and y alters one side of the equation without altering the other, and hence leaves the language.

The remaining case occurs when the = is contained in x. Recall that vy is nonempty. If either v or y is empty, then pumping v and y alters one side of the equation and not the other, and hence leaves the language. Because $|vxy| \leq p$, it must be true that v is a nonempty substring of 0^p on the left side of the equation, and y is a nonempty substring of 1^p on the right side of the equation. Therefore $v = 0^k$ and $y = 1^\ell$ for some $1 \leq k \leq p-2$ and $1 \leq \ell \leq p-2$, and uv^2xy^2z is the string

$$1^p 0^{p+k} = 1^{p+\ell} 0^p \cdot 1,$$

which is not in A. Thus, in all cases, the string w can be pumped to leave A. This contradiction implies that A is not context-free.

B. We first design a two-tape nondeterministic Turing machine N, whose language is L. The input string w arrives on the first tape. N begins by placing the start variable S on the second tape. N nondeterministically guesses a rule of the form $S \to u$, and rewrites the second tape to have contents u. Then N repeats these steps:

1. If the contents of the second tape equal the contents of the first tape, then N accepts.

2. Otherwise, N nondeterministically selects a rule $v \to t$ and a cell on the second tape. Starting at that cell, N examines the first $|v|$ cells on the second tape.

3. If the contents of those $|v|$ cells equal v, then v is a substring of the second tape, and N replaces v with t on the second tape (shifting the other contents of the tape as needed).

If the input string can be generated by the grammar through some sequence of steps, then N will find that sequence of steps, produce w on its second tape, realize that the second tape equals the first tape, and accept. If the input string cannot be generated, then it will never appear on the second tape, and N will loop. Hence the language of N is L.

Now we argue that N can be simulated by a one-tape nondeterministic Turing machine M. The argument is similar to the deterministic case. The TM M stores N's two tapes interleaved on its one tape, with the locations of the two tape heads marked. To simulate one step of N's computation, M first scans the tape to determine which symbols are under the two tape heads. It nondeterministically selects one of the possible transitions for its current state and those two tape symbols — just as N would. M then scans through the tape again, updating the contents of the marked cells and moving the marks left or right as appropriate.

Because the language of N is L, and N can be simulated by a one-tape nondeterministic Turing machine, which can be simulated by a Turing machine, we conclude that L is recognizable.

C. Let P be nontrivial, and assume (for the sake of contradiction) that

$$\{\langle M_1, M_2\rangle : P(M_1, M_2) = \text{True}\}$$

is decidable by some Turing machine D. Because P is nontrivial, there exist Turing machines M_1, M_2, M_1', M_2' such that $P(M_1, M_2) = \text{True}$ and $P(M_1', M_2') = \text{False}$.

Define a property P' of recognizable languages by $P'(M) = P(M, M_2)$. Because P is decidable, P' is also decidable. (Design a decider C for P' as follows. When C is given input $\langle M\rangle$, it runs D on $\langle M, M_2\rangle$, and outputs whatever D outputs.) Because $P'(M_1) = \text{True}$ and P' is decidable, Rice's theorem says that $P'(M) = \text{True}$ for all M. Thus $P'(M_1') = P(M_1', M_2) = \text{True}$.

Now define a new property P'' of recognizable languages by $P''(M) = P(M_1', M)$. Because P is decidable, so is P'' (by a similar argument to the one given above). Because $P''(M_2) = P(M_1', M_2) = \text{True}$ and P'' is decidable, Rice's theorem says that $P''(M) = \text{True}$ for all M. Thus $P''(M_2') = P(M_1', M_2') = \text{True}$. But we assumed that $P(M_1', M_2') = \text{False}$. This contradiction implies that $\{\langle M_1, M_2\rangle : P(M_1, M_2) = \text{True}\}$ is not decidable.