**EverPrint**

**Prove**

EverPrint₀ = \{ \langle M \rangle \mid M \text{ is a TM and } M \text{ prints a 0 at some point} \}

is undecidable.

**Proof by reduction:** Good⊆₉ to EverPrint₀

**Initial Assumption:** EverPrint₀ is decidable. Thus there exists a TM, D₉₀⟨M⟩, that can decide EverPrint₀.

We will now build a decider for Good⊆₉ that takes input ⟨M⟩ and determines if M continues to print digits infinitely. This decider, D_{Good}⟨M⟩, will function as follows:

1.) Use decider D₉₀⟨M⟩ to build the decider Dₐ₀⟨M, x⟩ which decides the language:

AnotherPrint₀ = \{ ⟨M, x⟩ \mid M \text{ is a TM and } M \text{ prints another 0 after it has printed it's } x\text{th 0} \}

This is done by converting TM M into a new TM M' in the following way:

1.) Duplicate M’s transition table
2.) For states q₀ through qₙ in M’s table, create corresponding new states qₙ₊₁ through q₂ₙ₊₁
3.) In the duplicated instructions, replace each mention of state qᵢ with it’s corresponding new state qᵢ₊ₙ₊₁
4.) In the original instructions, replace every P₀ (print 0) instruction with an instruction to print a new symbol that is not currently in Γ
5.) Perform the above steps 1-4 x times
6.) Output the resulting machine as M'

The decider for AnotherPrint₀, Dₐ₀⟨M, x⟩, will simply call D₉₀⟨M'⟩

2.) Use decider Dₐ₀⟨M, x⟩ to build machine Calc₀⟨M⟩. The machine Calc₀⟨M⟩ will write out a number N such that the xth digit of N is equal to 1 if Dₐ₀⟨M, x⟩ accepts, and is equal to 0 if Dₐ₀⟨M, x⟩ rejects.

3.) Use machine Calc₀⟨M⟩ and decider D₉₀⟨M⟩ to build the decider D₁₀⟨M⟩ which decides the language:

InfinitePrint₀ = \{ ⟨M⟩ \mid M \text{ is a TM and } M \text{ prints an infinite number of 0’s} \}

The decider D₁₀⟨M⟩ simply calls ¬D₉₀⟨ Calc₀⟨M⟩ ⟩

4.) If we can construct machines D₉₀⟨M⟩, Dₐ₀⟨M, x⟩, Calc₀⟨M⟩, and D₁₀⟨M⟩, then we can construct the corresponding machines D₁¹⟨M⟩, Dₐ₁⟨M, x⟩, Calc₁⟨M⟩, and D₁₁⟨M⟩ which perform the analogous tasks pertaining to the symbol 1 rather than the symbol 0.
5.) Finally, the decider for Good\text{TM} can be constructed as follows:

\[ D_{\text{Good}}(M) = D_{I0}(M) \lor D_{I1}(M) \]

**Case 1:** \( \langle M \rangle \in \text{Good}_{\text{TM}} \) In this case \( M \) will print an infinite stream of digits. This must consist of an infinite number of either 0’s or 1’s, if not both. This means that either \( D_{A0}(M,x) \) or \( D_{A1}(M,x) \) will accept \( M \) for any value of \( x \). Thus the corresponding \( \text{Calc}_0(M) \) or \( \text{Calc}_1(M) \) will consist of an infinite stream of 1’s. Because of this, either \( D_{E0}(\text{Calc}_0(M)^{\langle M \rangle}) \) or \( D_{E0}(\text{Calc}_1(M)^{\langle M \rangle}) \) will reject causing \( D_{I0}(\langle M \rangle) \lor D_{I1}(\langle M \rangle) \) and thus \( D_{\text{Good}}(M) \) to accept.

**Case 2:** \( \langle M \rangle \notin \text{Good}_{\text{TM}} \) This means that \( M \) stops printing digits altogether at some point. In this case, both \( D_{A0}(M,x) \) and \( D_{A1}(M,x) \) will reject \( M \) at some value of \( x \), causing \( \text{Calc}_0(M) \) and \( \text{Calc}_1(M) \) to begin printing 0’s. Because of this, \( D_{E0}(\text{Calc}_0(M)^{\langle M \rangle}) \) and \( D_{E0}(\text{Calc}_1(M)^{\langle M \rangle}) \) will both accept causing \( D_{I0}(\langle M \rangle) \lor D_{I1}(\langle M \rangle) \) and thus \( D_{\text{Good}}(M) \) to reject.