258. HaltVariation3

main3(p) {

if(selfHaltTest(p)) {

<loop>

} else {

<loop>

}

}

This version of the halting problem does not support a proof-by-contradiction.

Because of how main3 is constructed it can never halt on any input. This means that the statement in red can be ignored due to its absurdity. The second statement is not a contradiction, because it says that if main3 loops forever than main3 loops forever; so therefor, this example does not satisfy a proof-by-contradiction.

275. ApplyRice2

1. L = {p | p has even length}
   1. L is not a functional set of programs
   2. L is decidable
2. L = {p | the domain of pfn(p) has at least 10 strings}
   1. L is a functional set of programs
   2. L is not decidable
3. L = {p | the domain of pfn(p) ≠ 0}
   1. L is a functional set of programs
   2. L is decidable
4. L = {p | pfn(p) is the identity function}
   1. L is a functional set of programs
   2. L is not decidable
5. L = {p | pfn(p) is a constant function}
   1. L is a functional set of programs
   2. L is not decidable
6. L = {p | pfn(p) = pfn(q)}, where q is a fixed program that returns 0 on odd length strings and loops forever on even-length strings.
   1. L is a functional set of programs
   2. L is not decidable
7. L = {p | the domain of pfn(p) =Ø}
   1. L is not a functional set of programs
   2. L is decidable
8. L = {p | the domain of pfn(p) is finite.}
9. L = {p | there is some shorter program q computing the same function as p.}
   1. L is a functional set of programs
   2. L is not decidable
10. L = {p | the domain of pfn(p) is decidable}

280. DecClosure



288. Taxonomy

2. This is impossible because if L is decidable then must be decidable
3. This is impossible because L can only be decidable if and only if is semi-decidable.
4. This is impossible because can only be decidable if and only if L is semi-decidable.
5. This is impossible because L is decidable if both L and are semi-decidable.
7. can only be decidable if both and L are semi-decidable

290. SDSplitting