

G5029 Limits of Computation

03/06/2020

Candidate: 181509

Questions Chosen: 1 & 2

1-) a-) Binary trees that only contain leaves with atom nil are sufficient to define the notion of computability, because other data types can be encoded by using only the atom nil, such as booleans, natural numbers and lists. Therefore, the one built-in data type, the type of binary trees where leaves contain only atom nil, is sufficient to solve certain classes of problems, when used with the expressions and statements of the WHILE-language.

b-) i-)
$$\llbracket \text{foo} \rrbracket^{\text{WHILE}}('1') = \langle \text{nil}, \text{nil} \rangle$$
$$\llbracket \text{foo} \rrbracket^{\text{WHILE}}('1') = \text{nil}$$

ii-) For all input, the program foo. decides whether the input is ~~a list or a~~ ~~single element~~ a natural number or not, by returning nil if it is.

c-) WHILE-expressions define and manipulate the variables of a program. However, WHILE-commands are the decision procedures based on the WHILE-expressions.

d-)
$$\begin{array}{l} P \text{ read } X \{ \\ \quad \text{while } X \{ \\ \quad \quad Y := \text{cons hd } X \text{ nil}; \\ \quad \quad X := tl } X \\ \quad \} \\ \} \\ \text{write } Y \end{array}$$

e-) It is important to have a computational model that supports programs-as-data, because this allows the build of compilers, self-interpreters and Program specialisers. These programs extend the programming language's functionality, hence, increasing the number of problems that can be solved by this computational model.

03/06/2020

Candidate: 181509

f-) i-)

A-) Yes, a finite number of instructions will be run, n times, to result into a definite number.

B-) Yes, the function would always result in a definite number, based on program p .

C-) No, there is a chance that program p would not terminate.

D-) No, all $n \in \mathbb{N}$ are finite and definite, therefore, it cannot diverge.

E-) Yes, the function would always result in a definite number, based on program p .

ii-) Based on Church-Turing Thesis, all reasonable computation models are equivalent. Therefore, the computability of such functions do not depend on whether they are implemented in WHILE or RAM.

iii-) No, it wouldn't be possible to compute the same functions, if the conditional statement is removed from the pure WHILE-language. Not only because some of these functions rely on the conditional statement, but also the language would not be a reasonable computation model anymore, which means it would have lost the ability to solve certain problems rely on this statement.

2-) a-) i-) Not semi-decidable, it is decidable.

ii-) It is semidecidable.

iii-) Not semi-decidable, it is decidable.

iv-) Not semi-decidable.

b-) i-) decidable, all inputs must be finite, therefore, can be encoded, then to be decided.

ii-) Undecidable, as it would not terminate, because of its endless loop.

iii-) Undecidable, in the case that one of the inputs does not ~~encode~~ encode to a number.

c-) A-) False, a HALT is undecidable.

B-) True, ~~everything~~ ^{every number} that is not ~~an~~ even, will be odd and can be reduced by a function.

C-) False, HALT is undecidable.

D-) False, HALT is undecidable.

E-) True, complement of HALT decidable and can be reduced ^{by a} ~~to~~ function.

ii-) Effective problem reduction, is used to reduce problems by the use of a function to compute their decidability, whereas polynomial time reduction is for the complexity.

03/06/2020

Candidate: 181509

- d-) i-) True, as WHILE-languages accept programs-as-data, they can also be computed by another WHILE-language programs.
- ii-) False, Java has a Reflection library that allows recursion theorem.
- iii-) True, because all of these ~~programs~~ ^{programming languages} are considered to be already closed under recursion.
- e-) i-) ~~No, because the complements of A and B are~~
Yes, because the same complements of A and B that cause the semi-decidability will also cause the semi-decidability of C.
- ii-) Yes, because all $a \in A$ are finite and definite, which results the set of all identical pairs of numbers will be finite and definite as well.
- iii-) No, as the number inputted to Busy-Beaver will increase, it will become non-computable, therefore, undecidable.