PY4612 PROJECT 2

Read through all of these questions before attempting to answer them. Write your answers as clearly and explicitly as you can and explain all your working. ¶ Please do not discuss this project with any other students. ¶ Submit your answers as a PDF file on MMS by the due date: **Monday April 15, 2024**.

Q1: SHORT QUESTIONS

Answer two of the following three questions:

Q1A: RECURSIVE FUNCTIONS (5 POINTS)

A two-place relation R on $\omega \times \omega$ is recursive if and only if its characteristic function $f_R : \omega \times \omega \to \{0, 1\}$ is recursive (where, as usual, $f_R(x, y) = 1$ if and only if Rxy, for each $x, y \in \omega$).

If R is a two-place relation, its *slice* (in its first place) is the one-place relation S defined by setting Sy iff $\exists x\, Rxy$.

First, give one example of a recursive relation on $\omega \times \omega$ such that its slice is also recursive. *Then*, give *two* different examples of recursive relations R on $\omega \times \omega$ that have *non-recursive* slices.

Be careful to explain your reasoning in each case. For the relations with non-recursive slices, one of your examples should be taken from the domain of register machines and their properties, and the other from the domain of theories of arithmetic.

Q1B: RECURSIVE SETS IN ARITHMETIC (5 MARKS)

Let T be any deductively defined theory extending Robinson's Arithmetic (Q). Consider the two sets: P^+, consisting of the Gödel numbers of formulas provable in T, and P^-, consisting of the Gödel numbers of formulas refutable in T (that is, formulas whose negations are provable in T). Prove that no set S that contains every member of P^+ and contains no member of P^- is recursive.

Q1C: INCOMPLETENESS & UNDECIDABILITY (5 MARKS)

Consider the theory P in the language of arithmetic without the multiplication function symbol, given by the axioms Q1 to Q5 (the Robinson's Arithmetic axioms governing successor and addition), together with the axiom scheme of induction. It is known that

(a) P is *consistent*,

(b) P does *not* have the finite model property,

(c) P is *complete*, and

(d) P is *decidable*.

In your own words, explain why (a) and (b) are both true, and then explain, in detail, why (c) and (d) do *not* conflict with the results we have proved in this module.

That is, you are to first prove that the theory P defined above is consistent and that it does not have the finite model property. Then, you are to explain, in your own words, and in detail, where the proofs of Gödel's *First Incompleteness Theorem*, and the *undecidability of arithmetic*, break down when we attempt to apply them to the theory P.

Q2: ESSAY (10 POINTS)

This question asks you to explore one or more of the limitative results for predicate logic we have considered in the subject, and to demonstrate your understanding and appreciation of the power and limits of logic.

Choose one of these limitative results that we have studied over this class:

- The Compactness Theorem
- The Downward Löwenheim–Skolem Theorem
- The Undecidability of First Order Predicate Logic
- Tarski's Indefinability Theorem
- Gödel's Incompleteness Theorem

Write a short essay of no more than 1000 words that

1. Clearly *states* the result you will discuss.
    - So, you don't just *name* the result, you state the theorem clearly and precisely, showing that you understand what it is and what it is not.
2. Explains why this result marks some kind of *limit* on logic.
    - To do this, you should try to explain why the "limit" you mention is significant. Explain why this limit, in some sense, puts bounds on what we can do with logic, and why we might have reasonably thought we could go beyond that limit.
3. Explores the connection between this limitation and some field connected to logic, such as *epistemology* (what we can know, and our criteria for knowledge), *metaphysics* (what there is, and its fundamental nature and structure), *semantics* (what our words, sentences, etc. mean, and how our language gets those meanings), *mathematics*, *computer science*, or some other field you consider relevant.
    - *Please check with me if you want to write about some field I haven't mentioned. I will probably say that's great, but I'll probably be able to give you some extra tips for what to explore and maybe what traps to avoid.*

GRADING CRITERIA

You have *a lot* of room to move in the essay component of this project. Here are the criteria I will be using to assess your work. I am looking for essays that we can describe using these six adjectives.

[CREATIVE] *Be original*! Don't just reproduce something that you've read elsewhere. You want to refer to things you've seen and read, but this is your opportunity to make something new out of what you have learned.

[FITTING] With that said, your topic has to *fit well* with what we've explored in the subject. Don't just use it as a launching pad to talk about *something else*. A good essay will explore the significance of one of our limitative results, rather than using it as an excuse to focus on another topic.

[DEEP] A good submission will *go deep* with the results you discuss. You'll take the time to your understanding of some of the difficult things we've proved, and not just stay on the surface. You'll show that you've taken some time to absorb and understand the hard things we've worked with.

[ACCURATE] You'll need to *get things right*. Don't paraphrase something at a superficial level to give an *impression* of a theorem without also giving the details. The details matter, and it matters to get them right. Only this way will you be able to consider what a theorem says and what it doesn't. You can't sacrifice accuracy in order to be creative. A great essay will be creative and get the details right.

[CLEAR] A good essay will be *well structured*. You don't have lots of room to move. You need to put in what you need to address the questions you're planning to explore, and leave out what does not help with your goals. Don't leave it to the last minute to write: read your work over, asking yourself whether you've made yourself clear. Can someone who hasn't studied the subject at least follow the higher level outline of your essay? If they can, you're doing a good job of being clear.

[SCHOLARLY] *Cite the literature* generously, clearly and liberally. In writing your essay you're taking part in a wider conversation. We have (online) access to a library which gives you access to the conversation. Use it! Many people have written about these results, and you should show some awareness of this wider literature.

This creative essay is not exactly the *usual* sort of essay we ask for in philosophy subjects, but it is in the ballpark. If you haven't before, or even if you have, it is a good idea to read through our *Philosophy Essay Resources*. They will help you reflect on what you're doing when you're writing.

If you have any questions, don't hesitate to contact me.

Good luck!

PY4612 Advanced Logic

Project 2

21st April 2024

I hereby declare that the attached piece of written work is my own work and that I have not reproduced, without acknowledgement, the work of another.

QUESTION 1A: RECURSIVE FUNCTIONS

where Rxy iff $x \leq y$ (& not $y \leq x$?)

$\leqslant \subseteq \omega \times \omega$ is a recursive relation because it can be defined by $\mathrm{IsZero}(x \dot{-} y)$ which is composed of two recursive functions $\mathrm{IsZero}(x)$ and truncated subtraction $\dot{-}$. For why they are recursive functions, see (Incompleteness and Computability ver F19, Section 2.4-2.8). Every natural number has a number that is smaller or equal to it, so the slice of $\leqslant$ defined by $\{y \in \omega \mid \exists x \in \omega(x \leqslant y)\}$ is just the set of natural numbers ω. We know ω is a recursive set, so is the slice.

the gn of a   of a formula with gn

From (Incompleteness and Computability ver F19, Proposition 3.19), we know that $\mathrm{Proof}_\Gamma(x, y) \subseteq \omega \times \omega$, i.e., x is a proof of y from undischarged assumptions in Γ, is a recursive relation if Γ is a recursive set of sentences. Q (Robinson arithmetic) is a recursive set of sentences, so $\mathrm{Proof}_Q(x, y)$ is a recursive relation. From (Incompleteness and Computability ver F19, Theorem 4.29), we know that the slice, y is provable in Q

$$\mathrm{Prov}_Q(y) \iff \exists x \mathrm{Proof}_Q(x, y)$$

is not recursive since Q is undecidable.

given input...

Inspired by (Computability and Logic 5ed, Chapter 8), let $\mathrm{Halt}(m, t)$ expressing "register machine with code number m halts at step t", and it is recursive because it can be defined by other recursive functions constructed in the chapter. Its slice $\exists t\, \mathrm{Halt}(m, t)$ means a register machine m has the property that it eventually halts at certain step t. This slice corresponds to the halting set for register machines, which is known to be non-recursive.

given what input?

Very swift treatment.

We proved that $H(m,m) = \begin{cases} 1 & \text{if } Run_m \text{ halts on input } m \\ 0 & \text{o/w} \end{cases}$ is not recursive.

What is the input for your function?

Basically OK, but a small slip in third part.

QUESTION 1B: RECURSIVE SETS IN ARITHMETIC

We wish to show that there is no recursive set S such that both $P^+ \subset S$ and $P^- \cap S = \emptyset$.

Proof. Suppose S is recursive, then there is a formula $F(x)$ that defines S in Q:

$$n \in S \iff Q \vdash F(\underline{n}) \tag{1}$$
$$n \notin S \iff Q \vdash \neg F(\underline{n}) \tag{2}$$

where $\underline{n}$ is the numeral of n. By the diagonalisation lemma, we can find a sentence G such that

$$T \vdash G \leftrightarrow \neg F(\ulcorner G \urcorner) \tag{3}$$

gn(G) ⌜G⌝ is the numeral.

Suppose G is a theorem of T, then $\cancel{\ulcorner G \urcorner} \in P^+$ by the definition of P^+. Since $P^+ \subset S$ according to the definition of S, we have $\ulcorner G \urcorner \in S$. The left-to-right direction of (1) gives $Q \vdash F(\ulcorner G \urcorner)$. Since T extends Q, we can weaken the last conclusion to $T \vdash F(\ulcorner G \urcorner)$. Consequently, the left-to-right direction of the contraposition of (3)

× No. P^- isn't the non theorems. It's what you can refute.

gn(G)

$$T \vdash F(\ulcorner G \urcorner) \leftrightarrow \neg G$$

tells us $T \vdash \neg G$. This implies $\cancel{\ulcorner G \urcorner} \in P^-$ by the definition of P^-. According to the definition of S, $P^- \cap S = \emptyset$, so $\ulcorner G \urcorner \notin S$. But this contradicts $\ulcorner G \urcorner \in S$.

Suppose G is not a theorem of T, then by the definition of P^-, $\ulcorner G \urcorner \in P^-$. Since $P^- \cap S = \emptyset$ according to the definition of S, we have $\ulcorner G \urcorner \notin S$. The left-to-right direction of (2) gives $Q \vdash \neg F(\ulcorner G \urcorner)$. Since T extends Q, we can weaken the last conclusion to $T \vdash \neg F(\ulcorner G \urcorner)$. Consequently, by the right-to-left direction of (3), $T \vdash G$. This implies $\ulcorner G \urcorner \in P^+$ by the definition of P^+. According to the definition of S, $P^+ \subset S$, so $\ulcorner G \urcorner \in S$. But this contradicts $\ulcorner G \urcorner \notin S$.

Since both possibilities lead to a contradiction, we conclude that S cannot be recursive. ∎

OK, but you have assumed that if $T \nvdash G$ then $T \vdash \neg G$, which is to assume that T is complete – which it isn't.

QUESTION 2: ESSAY

Theorem 1 (Compactness theorem for first-order logic). *Let Γ be a set of well-formed formula in the language of first-order logic. If Γ is finitely satisfiable, then it is satisfiable.*

One purpose for employing logic is to characterise types of structures. For instance, we can define the domain of a structure as a set containing only two elements:

$$|D|_{=2} \equiv \exists y_1 \exists y_2 \forall x (y_1 \neq y_2 \wedge (y_1 = x \vee y_2 = x))$$

Likewise, we can also express that the cardinality of a domain is greater than n:

$$\begin{aligned}
|D|_{>1} &\equiv \neg \exists y_1 \forall x (y_1 = x) \\
|D|_{>2} &\equiv \neg \exists y_1 \exists y_2 \forall x (y_1 = x \vee y_2 = x)) \\
|D|_{>3} &\equiv \neg \exists y_1 \exists y_2 \exists y_3 \forall x (y_1 = x \vee y_2 = x \vee y_3 = x)) \\
&\vdots \\
|D|_{>n} &\equiv \ldots
\end{aligned}$$

Naturally, we might wonder whether we can use the language of first-order logic to characterise the structure of natural numbers. However, the Compactness theorem tells us that the language of first-order logic cannot precisely characterise the natural number structure. Suppose we have a first-order theory T, we can add additional formulae to it to obtain a new theory:

$$T^* = T \cup \{\mathbf{c} \neq \underline{n} \mid n \in \omega\}$$

For every finite subset of T^*, we can find a model with a domain being the natural number set ω that satisfies it. According to the Compactness theorem, the entire T^* has a model whose domain contains a constant c that is not a natural number. Since any model of $T*$ is a model of T, it follows that T has a model that does not have the structure of the natural numbers.

What we initially hoped for was that T could only be satisfiable by models possessing the structure of natural numbers. However, the Compactness theorem predicts that any T also have a model that do not possess the structure of natural numbers yet still make it satisfiable. The language of first-order logic lacks the expressibility to exclude this unwanted models. Hence, we cannot precisely describe the natural number structure within the first-order language.

Similarly, we also cannot precisely characterise the real number structure within first-order language. We can introduce a strange constant just as we did in the reasoning above for the natural

Give references

number structure:

$$T^* = T \cup \{\underline{0} < \mathbf{c} < \underline{1/n} \mid n \in \omega\}$$

This constant was called *infinitesimal* in the history of calculus development. It was once considered a concept lacking a rigorous definition and was later replaced by the more rigorously defined concept called limits in the further development of calculus. However, although introducing this constant reveals that the first-order language cannot precisely characterise the real number structure due to the Compactness theorem, more importantly, the theorem predicts the existence of a model that includes infinitesimal. This implies that infinitesimal can be rigorously treated, much like limits. We should not dismiss it so easily; perhaps calculus based on infinitesimal could be better than calculus based on limits.

Well, there are many limitations of non-standard analysis, including the inability to express the standard/nonstandard distinction

Compactness theorem is indeed a limitative result, as it implies that the expressibility of first-order logic is limited because it cannot precisely characterise some important structures we concerned. However, it is also a powerful result because it allows us to infer the existence of peculiar models, even if we do not yet know what these models look like.

Let's revisit the second example mentioned at the beginning of the essay. If we apply the Compactness theorem to it, we will find that if a formula is true in all models with a finite domain, it's true in some models with an infinite domain. There is no formula true in all and only the finite models. At first glance, this says that first-order language cannot distinguish between finitude and infinitude. However, looking at it from another perspective, this implies that the Compactness theorem establishes a bridge between finitude and infinitude.

For example, sometimes we wish to prove that a certain goal is unattainable. We wish prove that the goal cannot be achieved even using infinite resources. However, the difficulty in proving that a goal cannot be achieved with infinite resources can be significantly different from proving that it cannot be achieved with finite resources. Since we have the Compactness theorem, we can instead prove that it cannot be achieved with every possible finite resources. Then, according to the theorem, we can conclude that even with infinite resources, we also cannot achieve the goal. Conversely, if we prove that we can achieve a goal using infinite resources, then the Compactness theorem suggests that we can also achieve it using finite resources. In this sense, "compactness" refers to a kind of "finiteness" in the quantity of necessary resources.

Perhaps it is partly because of the power of the Compactness theorem that we find it difficult to abandon first-order logic despite its limitations.

Nice essay — would be good to cite some references & point to wider discussion.

PY4612 Assignment 2 Solutions and Marking Guide

Recursive Functions

The task is to give three examples of relations (on $\omega \times \omega$) and their slices. (Given a binary relation R, its slice is the relation S, defined by setting Sy iff $\exists$x Rxy.) The first example should be such that R and its slice are both recursive. The second and third should be such that R is recursive and S is not recursive. Of the second and the third cases for R, one should be defined in terms of register machines, and the other should be defined in terms of theories of arithmetic.

Possible answers for case 2 and case 3 are:

(2) Given a computable enumeration of the set of all register machines, defining for each n the register machine RM(n), the relation R defined by setting Rxy iff RM(y) halts within x steps on input y. This R is recursive (simply simluate RM(y) for x steps of computation and check whether it has terminated or not), while to determine whether or not its slice S holds of a given y is to decide whether RM(y) halts (at all) on input y, which, as we have seen in the proof of the undecidability of the halting problem, is not a recursive relation.

(3) Given a Gödel numbering not only of formulas but of proofs (explained in class), define the relation R by setting Rxy iff x is the Gödel number of a proof in Q of a formula with Gödel number y. This is recursive. (Q is finitely axiomatised, and so, checking, of a given number, that it is a gn of a Q-proof of a given formula is recursive.) But its slice, S is true of a number iff that number is the gn of a Q- theorem, and this set is not recursive, as proved in class.

A good answer

- specifies the relations clearly, and correctly identifies the slice, given the original relation.
- gives some correct explanation as to why the relation is recursive. (This could be an intuitive appeal to algorithm to decide the relation, or specifying a register machine to compute its characteristic function, or showing that it is representable in Q.)
- demonstrates why the slice is not recursive. To show that the slice is not recursive, it is not enough to define an algorithm intended to compute the slice, but to either refer to a proof of non-recursiveness given in class, to reduce your example to an example given in class (to show that if you could decide your S you could decide the halting problem, or theoremhood in Q, etc.), or to prove directly that your S is not recursive, using diagonalisation or some other technique.

Recursive sets in arithmetic

For this question, the student has to show that, whenever T is a deductively defined (that is, recursively axiomatised) extension of Q, if P+ is the set of all Gödel numbers of theorems of T and P- is the set of all Gödel numbers of anti-theorems of T (that is, x in P- iff x = gn(A) for some A where T $\vdash \neg$A), then no set S contains every member of P+ and excludes every member of P- is recursive.

To do this, the most direct way is to show that if such a set S were recursive, then there's some formula σ(y) that represents S in T: that is if $n \in S$ then T $\vdash \sigma(n)$ and if $n \notin S$, then T $\vdash \neg\sigma(n)$. Then use diagonalisation to find a formula G such that T $\vdash G \equiv \neg\sigma([G])$, and appeal to the facts that if T $\vdash$

G then gn(G) $\in$ S, and if T $\vdash \neg$G then gn(G) $\notin$ S, while mimicking the structure of the usual proof of the undefinability of theoremhood in (extensions of) Q.

That is: If T |- G, then gn(G) in S, and so, T |- σ([G]) and hence T |- ¬G. But then, gn(G) is not in S, which means T |- ¬σ([G]), which again by the biconditional means T |- G. Which would mean that T is inconsistent, but that is impossible, given the existence of S, which contains all theorems but avoids all refutable formulas, so the result is that no such S is recursive, since no such S is representable in T, while T (being an extension of Q) represents all recursive functions

A good answer

- somehow deals with the case where T is inconsistent, either implicitly or explicitly. (That is, it does not simply assume that T is consistent and ignore the limit case where T is not. In that case, P+ = P- = the whole language, and there is no set S at all that contains every member of P+ and excludes every member of P-.)
- Does not assume that S = P+, or that to decide membership of S you must decide membership of P+. (To see why, consider the related case: let E+ be the set of (Gödel numbers of) theorems of T with an even number of symbols, and let O- be the set of (gns of) anti-theorems of T with an odd number of symbols. Both E+ and O- fail to be recursive. The set of (gns of) formulas with even numbers of symbols contains every member of E+ and excludes every member of O-, and it is clearly a recursive set.)

Incompleteness and Undecidability

This question asks the student to consider Presburger Arithmetic (not given under that name), and to show that the theory is (a) consistent, (b) does not have the finite model property, and to explain why, the facts that (c) it is complete and (d) it is decidable, do not conflict with the results we have shown in this module.

For (a), the simplest way to answer this is to provide a model for the theory. We have shown, in class, that the standard model (in which the domain is ω, and the non-logical symbols in the language—the term 0, the function symbols for successor, sum and product—are given their standard interpretations on ω) is a model for Peano Arithmetic. So, since the axioms for Preseburger arithmetic (henceforth, Presb) are a proper subset of those for PA (we eliminate any formulas involving the product funcion symbol), the axioms remain true in the standard model.

For (b), it suffices to show that Presb has no finite models. In class, we proved that the axioms Q1 and Q2 of Robinson's Arithmetic (and hence, of PA) have no finite models. If they state that result without proof, citing that it was a class result, that's OK. If they elaborate, though, and do so in some inaccurate way, they cannot get full marks for this question.

For (c) and (d) the cruical steps in the failure of the proofs for incompleteness and undecidability to apply to Presb are the appeals to the result that in Robinson's Arithmetic (and and any extensions), we can represent all recursive functions, and hence, we can represent the diagonal function. If we could not represent diag, the standard proof of Gödel's first incompleteness theorem (and the undecidability of arithmetic) breaks down at the point where we appeal to the diagonal lemma (the existence, for any predicate B(y), a sentence G such that Presb $\vdash$ G ≡ B([G])). We have no assurance that this is true. We do know, though, that if Presb represents a set, that set is recursive, and so, if B were a predicate that represented the (gns of) theorems of Presb, then the standard argument would show that the diagonal lemma must fail for the predicate ¬B. The other option is that Presb does not represent the set of (gns of) Presb-theorems, and so this is where the gap between the sets Presb represents and the class of recursive sets. Both are consistent with the results we have proved in this class.

A good answer

- has explicit and clear proofs for (a) consistency and (b) failure of the finite model property.
- points out that Presb is, by design, less expressive than PA, but does not, without at least some explanation, move from “there is no multiplication symbol in the language of Presb” to “multiplication is not representable in Presb.” We have seen many examples of functions not given explicitly in the language of arithmetic which are nonetheless representable in it (an easy example is the predecessor function, where pred(0) = 0 and pred(y+1) = y. (it is represented by the predicate ((x=0 ∧ y=0) ∨ (x+1 = y)), as we've seen in class). So, a good answer would make explicit that since not all recursive sets are representable, and if multiplication were represented in Presb, we'd be able to represent everything in Presb that we can in PA, but we can't, so multiplication is not only absent from the primitive syntax, but also not representable.
- doesn’t make the mistake of saying that there’s no Gödel coding for the language of Presb: clearly there is, since the language is a proper subset of the language of arithmetic, which itself has a Gödel coding. Yes, Gödel's example coding used powers of primes, and without multiplication, those are unlikely to be definable in the theory. But that's irrelevant for the proof. The notion of a Gödel coding is assumed in the notion of the representation of a set of formulas as a set of numbers, and that makes sense, even for formulas in a language that is a proper subset of the language of arithmetic.

General Grading Criteria

Correctness: Good answers will be correct. Make true claims and avoid error.

Completeness: Good answers will be complete. If I ask you to show something, don’t just say that it’s obvious, but demonstrate how the answer to the question follows from the definitions of the concepts involved. You do not need to re- prove anything proved in our lectures or class notes, but if you appeal to those results, say so at the point where you do, and show how your answer builds on the results we have explored in class. A good answer leaves out no step of the explanation.

Concision: Good answers include only what is relevant. If you get straight to the point in your explanation, you show that you understand what is needed, and what is not. Do not pad your answer with extraneous material.

Clarity: Good answers explain things, clearly. Write in sentences. Use words. Do not put formulas on the page and expect me to understand what you are doing with them, if you do not explain yourself. Do not submit your rough working.

Citation: If you appeal to any material other than the class notes, cite it appropriately, giving full references with some standard referencing convention.