Question 9

(i) (a)

Next Instruction	R_1	R_2	R_3
1	2	0	0
2	2	0	1
3	2	0	1
4	2	1	1
5	2	1	2
2	2	1	2
6	2	1	2
STOP	1	1	2

(i)(b)

P does not compute the function f when the input is zero. In this case the contents of register 1 (0) is never equal to the contents of register 3 (≥ 1) when instruction 2 is executed. Therefore the program never terminates as the loop consisting of instructions 2-5 is executed ad-infinitum..

$f_P{}^1: \mathbb{N} \to \mathbb{N}$ is defined by $f_P{}^1(n) = \begin{cases} n-1, & \text{if } n > 0 \\ \text{undefined}, & \text{otherwise} \end{cases}$.

$f_P{}^2: \mathbb{N}^2 \to \mathbb{N}$ is defined by $f_P{}^2(n, m) = \begin{cases} n+m-1, & \text{if } n > 0 \\ \text{undefined}, & \text{otherwise} \end{cases}$.

[[Check that when we put m = 0 in $f_P{}^2$ we get $f_P{}^1$.]]

(ii)

[[We need to put in a test for n = 0 to ensure that the program works in this case. As an instruction is added at the start the following statement numbers change so we also need to change the instruction numbers in the jump instructions.]]

1 J(1, 3, 7) [[Check for 0 and adjust statement nos. in Jumps]]
2 S(3)
3 J(1, 3, 7)
4 S(2)
5 S(3)
6 J(1, 1, 3)
7 C(2, 1). [[See Unit 1, Problem 2.2]]

Question 10

(i) [[See Unit 2, Problem 1.9 for <.]]

Let $\chi_<: \mathbb{N}^2 \to \mathbb{N}$ be defined by $\chi_<(n, m) = sg\,(m \dot{-} n)$.

If $n < m$ then $m \dot{-} n > 0$ and $sg\,(m \dot{-} n) = 1$. So $\chi_<(n, m) = 1$ in this case.
If $n \geq m$ then $m \dot{-} n = 0$ and so $\chi_<(n, m) = 0$.

Since $\chi_<$ is obtained by substitution from the total primitive recursive functions sg and $\dot{-}$ then $\chi_<$ is also a total primitive recursive function.

Therefore the relation < is primitive recursive.

(ii) [[See proof of Unit 2, Theorem 1.3.]]

Since R_1 and R_2 are primitive recursive relations then their characteristic functions χ_{R_1} and χ_{R_2}, respectively, are primitive recursive functions.

Consider the function

$$f(n_1, n_2) = g_1(n_1, n_2)\chi_{R_1}(n_1, n_2) + g_2(n_1, n_2)\chi_{R_2}(n_1, n_2).$$

If $\chi_{R_1}(n_1, n_2) = 1$ then, since the relations R_1 and R_2 are mutually exclusive and exhaustive, then $\chi_{R_2}(n_1, n_2) = 0$. So, in this case, $f(n_1, n_2) = g_1(n_1, n_2)$.

If $\chi_{R_1}(n_1, n_2) = 0$ then, since the relations R_1 and R_2 are mutually exclusive and exhaustive, then $\chi_{R_2}(n_1, n_2) = 1$. So, in this case, $f(n_1, n_2) = g_2(n_1, n_2)$.

Since the function f is obtained from the primitive recursive functions mult, add, χ_{R_1} and χ_{R_2} by substitution then f is primitive recursive.

(iii) [[Use of Unit 2 Theorem 1.5. Similar to Unit 2, Additional Exercise, Sect. 1, Qu. 4]]

Define the functions

$g_1(n, m) = m + n = add(m, n)$,
$g_1(n, m) = m^2 = exp(m, 2)$
$g_3(n, m) = 5 = C_5^2\,(n, m)$,

and the relations

$R_1(n, m) \Leftrightarrow \max\,(n, 2m) < 100$,
$R_2(n, m) \Leftrightarrow 2n + 3m = 360$,
$R_3(n, m) \Leftrightarrow$ not $R_1(n, m)$ and not $R_2(n, m)$.

Then we can write

$$f(n, m) = \begin{cases} g_1(n,m) & \text{if } R_1(n,m) \\ g_2(n,m) & \text{if } R_2(n,m) \\ g_3(n,m) & \text{if } R_3(n,m) \end{cases}$$

As g_1, g_2, and g_3 can be written using the primitive recursive functions C_5^2, add, and exp using constants then g_1, g_2, and g_3 are primitive recursive functions.

The characteristic function of the relation R_1, $\chi_{R1}(n,m) = \chi_<(\max(n, 2m), 100)$. As χ_{R_1} is obtained by substitution from the primitive recursive functions max, mult, and $\chi_<$ using constants, then it is a primitive recursive function. Hence R_1 is a primitive recursive relation.

The characteristic function of the relation R_2, $\chi_{R2}(n,m) = \chi_{eq}(2n + 3m, 360)$. As χ_{R_2} is obtained by substitution from the primitive recursive functions χ_{eq}, mult and add using constants, then it is a primitive recursive function. Hence R_2 is a primitive recursive relation.

Using the result of Unit 2, Problem 1.10, then R_3 is also a primitive recursive relation.

From the definition of R_3 it follows that the set of relations R_1, R_2, and R_3 are exhaustive.

If the relation R_1 holds then $n < 100$ and $m < 50$. So $2n + 3m < 200 + 150 < 360$. Therefore R_1 and R_2 are mutually exclusive. From the definition of R_3, if the relation R_3 holds then neither R_1 or R_2 holds. Therefore R_1, R_2 and R_3 are mutually exclusive.

Since all the conditions required for the use of Theorem 1.5 of Unit 2 hold then it follows that f is primitive recursive.

Question 11

(i)(a) [[Similar to Unit 2, Problem 1.3]]

$g(n_1, n_2, n_3) = f\,(U^3_2(n_1, n_2, n_3),\ U^3_1(n_1, n_2, n_3),\ U^3_3(n_1, n_2, n_3),\ )$.

As g is obtained by substitution from the primitive recursive functions f, U^3_1, U^3_2, and U^3_3, then g is a primitive recursive function.

(i)(b) [[Unit 2, Example 1.5.]]

See 2007 solution.

(ii) [[Similar to Unit 2, Problem 3.4.]]

Let R be the relation given by $R(n, y) \Leftrightarrow n < 2^y$.

The characteristic function for this relation is $\chi_R(n, y) = \chi_<(n, \exp(2, y))$. This is obtained by substitution from the primitive recursive functions $\chi_<$ and exp using constants. Hence it is primitive recursive [HB p21 result of problem 1.10].

By Theorem 3.5 [HB p23] the function $g : \mathbb{N}^2 \to \mathbb{N}$ given by
$g(n, z) = \mu y \leq z\ R(n, y)$
is primitive recursive.

As $n < 2^y$ then a suitable bound on y is n as $n < 2^n$.

For all n we have
$$f(n) = \mu y\ R(n, y) = g(n, n).$$
As f is obtained from the primitive recursive function g by substitution then f is primitive recursive.

Question 12

(i) [[Similar to Unit 3, Problem 3.3.]]

For any natural number n, define a URM program by

(1) C(1, 1)
⋮
(n) C(n, n)
(n+1) Z(1)
(n+2) S(1)

This implements the constant function C_1 since it halts with the value 1 in register 1. As a program exists for each $n \in \mathbb{N}$, where $n > 0$, then there are infinitely many such programs.

(ii) [[See Unit 3, Problem 3.4.]]

The program P^* can be created by concatenating the programs
P, and
(1) Z(1).
(2) S(1).

If f_P^1 is total, then P* executes P. As f_P^1 is total, the last two instructions of P* are also executed. These set the value of register 1 to 1. Therefore P* computes the constant function C_1. If f_P^1 is not total, then P will not halt for some input n, and neither will P* for the same input. So P* computes the constant function C_1 function precisely when the function f_P^1 is total.

(iii) [[See Unit 3, Problem 3.4.]]

Assume the set X of code numbers which computes C_1 is recursive. We will show that is X is recursive then so is *Tot*, the set of numbers that code URM programs which compute a total function of one variable.

Let e be any code number. Initially check to see whether e codes a URM program. This is possible since the set of code numbers, *Prog*, is primitive recursive [[HB p21]]. If e does not code a URM program then $e \notin$ *Tot*.

If e does code a URM program then the instructions of the program P can be recovered from e and the program P^* which computes the function C_2 can then be created as described in part (ii). The code number e^* of P^* can then be determined. As the set X is recursive then there is an algorithm for deciding if a number $e^* \in X$. As $e^* \in X$ if and only if $e \in$ *Tot* then we can determine whether $e \in$ *Tot*.

Theorem 3.2 of Unit 3 states that there is no algorithm which determines whether $e \in$ Tot. As we have found one then the assumption that the set X is recursive must be false.

Question 13

(i) [[See Unit 4 Example 3.3 and Problems 3.2 and 3.3]]

Let φ be the subformula $\exists y\ y = z$; ψ be $\exists y\ (y = z \vee y = \mathbf{0}')$; and χ be $\exists y\ y = \mathbf{0}'$.

The given formula can then be written as $((\neg\varphi \,\&\, \psi) \rightarrow (\chi \vee (\varphi \rightarrow \chi)))$

A truth table for this formula is

φ	ψ	χ	$((\neg\varphi \,\&\, \psi) \rightarrow (\chi \vee (\varphi \rightarrow \chi)))$
1	1	1	0 1 0 1 1 1 1 1 1 1
1	1	0	0 1 0 1 1 0 0 1 0 0
1	0	1	0 1 0 0 1 1 1 1 1 1
1	0	0	0 1 0 0 1 0 0 1 0 0
0	1	1	1 0 1 1 1 1 1 0 1 1
0	1	0	1 0 1 1 1 0 1 0 1 0
0	0	1	1 0 0 0 1 1 1 0 1 1
0	0	0	1 0 0 0 1 0 1 0 1 0
			(1) (2) (3) (2) (1)

Since column 3 is all ones then the formula takes the truth value 1 under all interpretations then the formula is a tautology.

(ii)

(ii)(a)

Line	1	2	3	4	5	6	7	8
Ass.	1	2	3	3	2,3	2, 3	1,3	3

(ii)(b)

$((\phi \,\&\, (\phi \rightarrow \psi)) \rightarrow \psi)$.

(ii)(c)

NO. ψ may contain free occurrences of x so the EH rule cannot be used.

(iii)

In order to use the UI rule on line 2 then the formula on line 1 must contain no free occurrences of x. Since it does then the use of the UI rule is invalid.

Take the standard interpretation $\mathcal{N}$ with domain $\mathbb{N}$. In this interpretation the formula $x = \mathbf{0}'$ is true but $\forall x\ x = \mathbf{0}'$ is false as not every number equals $\mathbf{0}'$. Therefore $\forall x\ x = \mathbf{0}'$ is not a logical consequence of $x = \mathbf{0}'$.

Question 14

(i) [[Only free occurrences of x are in the term $\exists y\,(y+x) = x$]]

(a) NO [[z becomes bound]]
(b) YES
(c) YES

(ii)(a) [[I find Unit 6, Techniques for finding formal proofs useful]]

1	(1)	$\exists x\,(\neg\phi \,\&\, \psi)$	Ass
2	(2)	$\forall x\,((\psi \,\&\, \theta) \rightarrow \phi)$	Ass
2	(3)	$((\psi \,\&\, \theta) \rightarrow \phi)$	UE, 2
4	(4)	$(\neg\phi \,\&\, \psi)$	Ass
5	(5)	$\forall x\,\theta$	Ass
5	(6)	θ	UE, 5
4	(7)	ψ	Taut, 4
4, 5	(8)	$(\psi \,\&\, \theta)$	Taut, 6, 7
2, 4, 5	(9)	ϕ	Taut, 3, 8
4	(10)	$\neg\phi$	Taut, 4
2, 4, 5	(11)	$(\phi \,\&\, \neg\phi)$	Taut, 9, 10
2, 4	(12)	$(\forall x\,\theta \rightarrow (\phi \,\&\, \neg\phi))$	CP, 11
2, 4	(13)	$\neg\forall x\,\theta$	Taut, 12
1, 2	(14)	$\neg\forall x\,\theta$	EH, 13

Therefore $\exists x\,(\neg\phi \,\&\, \psi),\ \forall x\,((\psi \,\&\, \theta) \rightarrow \phi) \vdash \neg\forall x\,\theta$.

[[The solution can be shortened by combining some of the tautologies.]]

(ii)(b)

1	(1)	$\forall x\,((\psi \,\&\, \theta) \rightarrow \phi)$	Ass
2	(2)	$\forall x\,\theta$	Ass
3	(3)	$\exists x\,(\neg\phi \,\&\, \psi)$	Ass
1, 3	(4)	$\neg\forall x\,\theta$	Using part (a)
1, 2, 3	(5)	$(\forall x\,\theta \,\&\, \neg\forall x\,\theta)$	Taut, 2, 4
1, 2	(6)	$(\exists x\,(\neg\phi \,\&\, \psi) \rightarrow (\forall x\,\theta \,\&\, \neg\forall x\,\theta))$	CP, 5
1, 2	(7)	$\neg\exists x\,(\neg\phi \,\&\, \psi)$	Taut, 6

Therefore $\forall x\,((\psi \,\&\, \theta) \rightarrow \phi),\ \forall x\,\theta \vdash \neg\exists x\,(\neg\phi \,\&\, \psi)$.

Question 15

(i) [[Looks as if both sides of the equation equal (x.y).]]

	(1)	$((x . y) + \mathbf{0}) = ((x . y) + \mathbf{0})$	II
1	(2)	$\forall x\ (x + \mathbf{0}) = x$	Ass. Q4
1	(3)	$((x . y) + \mathbf{0}) = (x . y)$	UE, 2
-	(4)	$((x + \mathbf{0}) . y) = ((x + \mathbf{0}) . y)$	II
1	(5)	$(x + \mathbf{0}) = x$	UE, 2
1	(6)	$(x . y) = ((x + \mathbf{0}) . y)$	Sub, 4, 5
1	(7)	$((x . y) + \mathbf{0}) = ((x + \mathbf{0}) . y)$	Sub, 3, 6
1	(8)	$\forall y\ ((x . y) + \mathbf{0}) = ((x + \mathbf{0}) . y)$	UI, 7
1	(9)	$\forall x\ \forall y\ ((x . y) + \mathbf{0}) = ((x + \mathbf{0}) . y)$	UI, 8

As the assumption is axiom Q4 of Q then the sentence is a theorem of Q.

(ii) [[(ii) ⇒ (iii). So this is probably false.]]

In $\mathcal{N}^{**}$ let $x = \alpha$ and $y = \beta$.

$(x . y) = (\alpha . \beta) = \beta$.
$(y . x) = (\beta . \alpha) = \alpha$.

Therefore $\forall x\ \forall y\ (x . y) = (y . x)$ is not true in $\mathcal{N}^{**}$.

All the axioms of Q hold in $\mathcal{N}^{**}$. As $\exists y\ \forall x\ (x.y) = (y.x)$ does not hold in the interpretation $\mathcal{N}^{**}$ then, it follows by the Correctness Theorem, the sentence is not a theorem of Q.

(iii) [[Same as 2007, 15(ii) and 2003, 15(iii)]]

-	(1)	$(x.x) = (x.x)$	II
-	(2)	$\exists y\ (x.y) = (y.x)$	EI, 1
-	(3)	$\forall x\ \exists y\ (x.y) = (y.x)$	UI, 2

As there are no assumptions then the sentence is a theorem of Q.

Question 16

(i) True

Let A be the set of the Gödel numbers of the infinite number of axioms of PA.

Given any number n then we can write an algorithm to determine whether n is the Gödel number of a formula. If it is not then return 0. If it is and equals the Gödel number of any of the axioms of Q then return 1. By processing the Gödel number we can determine whether it has the form of an induction axiom. If it has return 1 otherwise return 0.

Therefore, by Church's Thesis, the characteristic function of the set A is recursive, and so the set A is recursive. Therefore PA is recursively axiomatizable.

(ii) True

Assume that PA is not consistent. Therefore, by definition, there is a sentence Φ of the formal language such that both $\vdash_{PA} \Phi$ and $\vdash_{PA} \neg \Phi$.

The Correctness Theorem tells us that if a formula is derivable from PA that it is a logical consequence of PA providing that there is an interpretation of PA.

Since $\mathcal{N}$ is an interpretation of PA then both Φ and $\neg \Phi$ are logical consequences of PA. Since Φ cannot be both true and false in an interpretation then our assumption that PA is not consistent must be incorrect. So PA is consistent and the statement is true.

(iii) False

By Unit 7, Theorem 2.1(b) $\vdash_Q \mathbf{0} \neq \mathbf{1}$. So $\vdash_Q \neg\, \mathbf{0} = \mathbf{1}$. Since PA is an extension of Q then theorems of Q are also theorems of PA. So $\vdash_{PA} \neg\, \mathbf{0} = \mathbf{1}$.

Since PA is consistent, by part (ii), then both $\vdash_{PA} \mathbf{0} = \mathbf{1}$ and $\vdash_{PA} \neg\, \mathbf{0} = \mathbf{1}$ cannot be true.

Therefore $\vdash_{PA} \mathbf{0} = \mathbf{1}$ is not true.

OR

1	(1)	$\forall x \neg\, \mathbf{0} = x'$	Ass
1	(2)	$\neg\, \mathbf{0} = \mathbf{0}'$	UE, 1

As the assumption is an axiom of PA then $\vdash_{PA} \neg\, \mathbf{0} = \mathbf{1}$.

Since PA is consistent ….

(iv) False

Gödel's First Completeness theorem states that there is no complete and consistent recursively axiomatizable extension of Q.

Since PA is an extension of Q and, as we have shown in parts (i) and (ii), it is consistent and recursively axiomatizable this means PA cannot be complete.

END OF PART II SOLUTIONS