1. Consider the set of indices
DEFINED = { f | ∃x ϕf(x)↓ }.

Use Rice’s Theorem to show that DEFINED is not decidable.

Hint: There are two properties that must be demonstrated.

Defined is not trivial as the index of S(x) = x+1 is in, but ↑(x) = μy [ y == y+1] is not.

Let f and g be indices of two arbitrary effective procedures such that the dom(f) = dom(g).

f ∈ DEFINED

⇔ ∃x **ϕ**f (x) ↓ by definition of DEFINED

⇔ dom(f) ≠ φ since f converges somewhere

⇔ dom(g) ≠ φ since dom(g) = dom(f)

⇔ ∃x **ϕ**g (x) ↓ since domain is not empty

⇔

g ∈ DEFINED by definition of DEFINED

1. Let P = { f | ∃x ϕf(x) converges in at most x steps }. Why does Rice’s theorem not tell us anything about the undecidability of P?

Because P is not an I/O behavior; it is a performance behavior.

To see this, consider the two functions F(x) = 0 and G(x) = μy [y>x] – μy [y>x] .

∀x F(x) = G(x), but F is in P and G is not.

1. Show that DEFINED is not decidable by reducing K0 to this set.

Let f be the index of an arbitrary function, F, and x be an arbitrary input.

Define Gfx(y) = F(x)-F(x).

Gfx(y) is defined everywhere and thus in DEFINED if <f,x> is in K0.

Gfx(y) is undefined everywhere and thus not in DEFINED if <f,x> is not in K0.

1. Is DEFINED re? Support your conclusion.

It is. We can semi-decide DEFINED by using the STP predicate as follows: f ∈ DEFINED iff ∃<x,t> [ STP(f,x,t) ]

1. Let Incr = { f | ∀x ϕf(x+1)>ϕf(x) }.
Let TOT = { f | ∀x ϕf(x) converges }.
Prove that Incr ≡m TOT.

Let f be an index of an arbitrary function, F.

Define Gf(x) = F(x)-F(x) + x.

Gf is the Identity function and thus is in INCR if f is in TOTAL.

Gf diverges on at least one input and thus is in not INCR if f is not in TOTAL.

Let f be an index of an arbitrary function, F.

Define Gf(x) = μy [F(x+1) > F(x)].

Gf(x) is the constant 0 and hence is TOTAL if f is in INCR.

Gf(x) diverges on at least one input and thus is not TOTAL if f is not in INCR.

1. Let sets A and B each be re non-recursive.
Consider C = A ∩ B. For (a)-(c), either show sets A and B with the specified property or demonstrate that this property cannot hold.
	1. Can C be recursive?
	2. Can C be re non-recursive
	3. Can C be non-re?

Consider A = { 2x | x ∈ K0 }, B = { 2x+1 | x ∈ K0 }.

A and B are each 1-1 equivalent to K0 and hence re, non-recursive.

A ∩ B = ϕ and is hence recursive, so (a) can hold.

A = K0, B = K0

A ∩ B = K0 and is hence re, non-recursive, so (b) can hold.

C can be semi-decided since we can just take the semi-decision procedures for A and B, say fA and fB and provide a semi-decision procedure for C via fC(x) = fA(x) \* fB(x). Here fC diverges iff either fA or fB diverge. Also, if either returns 0 (false), then fC either returns 0 or diverges, if the other diverges. Thus, (c) cannot hold.