MATH230: Tutorial Eight

Peano Arithmetic & Recursive Functions

Key ideas and learning outcomes

- Further understanding of the expressibility of PA,
- Practice writing natural deductions in PA,
- Recognise when procedures are primitive recursive,
- Informally describe procedures with primitive recursive steps.

Relevant notes: Peano and Primitive Recursion Slides
Relevant reading: *Computability Theory* Enderton, Chapter 2.

Hand in exercises: 2b, 3a, 3f, 3g, 7
Due following Friday @ 5pm to the tutor, or to lecturer.

Discussion Questions

1. The first-order language of Peano Arithmetic is often presented with an extra binary relation symbol $<$ where $x < y$ is given the usual interpretation: x is strictly less than y. In fact it is not necessary to add anything extra, for this relation can be defined using a sentence in PA as stated.

   Write down a wff in PA which, when interpreted in the standard model, defines the binary relation $<$ of being "strictly less than". Use this to write down formulae that represent: less than or equal to, strictly greater than, and greater than or equal to.

2. If $P(\mathbf{x})$ and $Q(\mathbf{x})$ are two primitive recursive n-ary predicates, then show that the predicate $P(\mathbf{x}) \to Q(\mathbf{x})$ is primitive recursive.

3. Discuss whether the search for roots to a given polynomial equation is a primitive recursive procedure.

Tutorial Exercises

Peano Arithmetic

1. Write down well-formed formulae in the first-order language of PA corresponding to the following statements.

   (a) Each natural number is either equal to 0 or greater than 0.

   (b) If x is not less than y, then x equals y or y is less than x.

   (c) If x is less than or equal to y and y is less than or equal to x, then $x = y$.

2. Provide natural deductions of the following theorems of Peano Arithmetic.

   (a) PA, $0 < a \ \vdash \ 0 < s(a)$

   (b) PA, $a < b \ \vdash \ s(a) < s(b)$

   (c) PA, $(a < b) \wedge (b < c) \ \vdash \ a < c$

   (d) PA $\vdash \ \forall x \ [(x = 0) \vee (0 < x)]$ (Challenge!)

   (e) PA $\vdash \ \forall x \ \forall y \ [\neg(x < y) \rightarrow ((x = y) \vee (y < x))]$ (Challenge!!!)

   (f) PA $\vdash \ \forall x \ \forall y \ [(x \leq y) \wedge (y \leq x)] \rightarrow x = y$ (Challenge!!!)

Primitive Recursive Functions

For each of the procedures below it is sufficient to provide an informal explanation of how they can be computed from primitive recursive functions using composition and recursion. For an extra challenge, you can also provide the formal definition from the base primitive recursive functions.

3. Give an informal description of a primitive recursive procedure that computes the following predicates and functions. You may refer to any primitive recursive procedure from class or earlier in the tutorial.

   (a) Less than relation $<?(x, y) = 1$ if $x < y$ otherwise $<?(x, y) = 0$.

   (b) Strictly less than relation $\leq?(x, y) = 1$ if $x \leq y$ otherwise $\leq?(x, y) = 0$.

   (c) Greater than relation $>?(x, y) = 1$ if $x > y$ otherwise $>?(x, y) = 0$.

   (d) Strictly greater than relation $\geq?(x, y) = 1$ if $x \geq y$ otherwise $\geq?(x, y) = 0$.

   (e) Identity relation $=?(x, y) = 1$ if $x = y$ otherwise $=?(x, y) = 0$.

   (f) Absolute difference $|x - y|$

   (g) Min(x,y)

   (h) Max(x,y)

4. Addition is iterated succession. Multiplication is iterated addition. Exponentiation is iterated multiplication. Use recursion and exponentiation to define a new primitive recursive function $f(a, b + 1) = exp(a, f(a, b))$, such that $f(a, 0) = 1$. What does this function do? Compute $f(5, 3)$.

5. Give an informal description of the primitive recursive definitions of the function which returns the maximum (resp. minimum) of three inputs.

6. Use bounded minimisation (along with other functions already shown to be primitive recursive) to show that bounded maximisation is primitive recursive.

7. Give informal descriptions of how the following functions could be defined using functions and predicates that were shown to be primitive recursive in class. Hence conclude that each of these procedures are primitive recursive.

   (a) Remainder(x, y): remainder when x divided by y.

   (b) Divides?(x, y): 0 or 1 according to whether x divides y.

   (c) Prime?(x): 0 or 1 according to whether x is prime.

   (d) nextPrime(x): returns the smallest prime greater than[1] x.

   (e) $p : \mathbb{N} \to \mathbb{N}$ such that $p(n) =$ nth prime.

8. Propositional logic is primitive recursive. Suppose $P(\mathbf{x})$ and $Q(\mathbf{x})$ are two primitive n-ary predicates i.e. the characteristic functions χ_P and χ_Q are primitive recursive. Show that the (characteristic functions of the) following predicates are primitive recursive.

   (a) $\neg P(\mathbf{x})$

   (b) $P(\mathbf{x}) \vee Q(\mathbf{x})$

   (c) $P(\mathbf{x}) \wedge Q(\mathbf{x})$

   (d) $P(\mathbf{x}) \to Q(\mathbf{x})$

   (e) NAND$(P(\mathbf{x}), Q(\mathbf{x}))$

[1]Hint 1: Prove that there must be a prime, p, between $x < p \leq x! + 1$.
Hint 2: Chebyshev said it, so I'll say it again, there's always a prime, between N and $2N$.