CS 145 – Foundations of Computer Science

Professor Eric Aaron

Lecture – T Th 9:00am
Lab – F 10:30am

Lecture Meeting Location: SP 309
Lab Meeting Location: SP 309

Business

• HW4 due already
• HW5-Lookahead out now; full HW5 out today
  – HW5 due Nov. 11 / Nov. 12 (Extend those due dates?)
• HW5 note:
  – HW5 Programming exercises build upon lab4
  – I will post lab4 solutions Friday morning
  – Until then, feel free to use your own lab4 solutions (and asmt-helper file)
    for HW5…
  – … but before submitting your HW5, please make sure your code works
    with the posted solution files

• Coaching Hours Change! From now until the end of the semester:
  – Coaching Hours on Thursdays 7-9pm
  – No Coaching Hours on Tuesdays
Business, pt. 2

- Exam grading update

- Reading: Makinson, Ch.4.1-4.6
  - Our coverage of the material will be different from that in the textbook, but it’s good to see the textbook’s presentation, as well

- Reading: Prof. Hunsberger’s document “The Natural Numbers, Induction, and Numeric Recursion”
  - Posted on the Additional Notes / Readings page of the CS145 website

Multiplication

- Multiplication on the naturals can be defined recursively, similarly to addition on the naturals

- Definition of addition:
  1. Case z=0 -- For all n in N, n + 0 = n
  2. Case z=S(m) -- For all n in N, z = S(m) for some m in N; n + S(m) = S(n + m)

- Definition of multiplication:
  3. Base case: x * 0 = 0
  4. Inductive case: x * Sy = (x*y) + x

- Some notes about the phrasing of that definition
  - Abbreviations and condensed forms are used, but it really means what we would expect, based on the definition of addition
  - Sy is a shorthand for S(y)
  - Variables are implicitly universally quantified over the naturals. E.g., the base case is “For all x in N, x * 0 = 0”. (cf. definition of addition)
Multiplication

• Multiplication on the naturals can be defined recursively, similarly to addition on the naturals
• Definition of addition:
  1. Case z=0 -- For all n in N, n + 0 = n
  2. Case z=S(m) -- For all n in N, z = S(m) for some m in N: n + S(m) = S(n + m)
• Definition of multiplication:
  3. Base case: x * 0 = 0
  4. Inductive case: x * Sy = (x*y) + x
• Theorem: 0 * x = 0. Prove (for all x in N) by induction
  – Let P(n) be the proposition 0 * n = 0
  – Base case: P(0). To prove...
  – Inductive case: Assume P(k), prove P(Sk). To prove...

This proof might be subtle in places... if there are some small details that need proof, be sure to prove them!
Multiplication (Theorem: Jove)

- Multiplication on the naturals can be defined recursively, similarly to addition on the naturals
- Definition of addition:
  1. Case $z=0$ -- For all $n$ in $N$, $n + 0 = n$
  2. Case $z=S(m)$ -- For all $n$ in $N$, $z = S(m)$ for some $m$ in $N$; $n + S(m) = S(n + m)$
- Definition of multiplication:
  3. Base case: $x * 0 = 0$
  4. Inductive case: $x * Sy = (x^y) + x$
- Theorem: Jove—$x = 1 * x$ (We use 1 as a common shorthand for S0)
  - What proposition should we use for $P(n)$?
  - Base case: $P(0)$. To prove…
  - Inductive case: Assume $P(k)$, prove $P(Sk)$. To prove…

Exponentiation

- Exponentiation on the naturals can be defined recursively, similarly to multiplication on the naturals
- Definition of multiplication:
  3. Base case: $x * 0 = 0$
  4. Inductive case: $x * Sy = (x^y) + x$
- Definition of exponentiation:
  5. $x^0 = 1$
  6. $x^{(Sk)} = (x^k) * x$
- Theorem: $0^n = 0$
  - What proposition should we use for $P(n)$?
  - Base case: $P(0)$. To prove…
  - Inductive case: Assume $P(k)$, prove $P(Sk)$. To prove…
Exponentiation

- Exponentiation on the naturals can be defined recursively, similarly to multiplication on the naturals
- Definition of multiplication:
  3. Base case: $x \cdot 0 = 0$
  4. Inductive case: $x \cdot Sy = (x \cdot y) + x$
- Definition of exponentiation:
  5. $x^0 = 1$
  6. $x^{Sk} = (x^k) \cdot x$
- Theorem: $0^n = 0$
  - Be careful! This “Theorem” is actually false!