CSC165 fall 2014

Mathematical expression

Danny Heap
heap@cs.toronto.edu
BA4270 (behind elevators)
Web page: http://www.cdf.toronto.edu/~heap/165/F14/
416-978-5899

Using Course notes, chapter 1, 2

Outline

what's CSC165?

a course about expression (communication):

- with and through programs
- with developers
- knowing what you mean
- understanding what others mean
- analyzing arguments, programs

why CSC165?

do you:

- **memorize** math?
- **have trouble** explaining what you're doing in technical work?
- **have trouble** understanding word problems?

don't you:

- **enjoy** reading math books for new material
- **like** talking about abstract x and y as much as particular examples of what x and y represent?

CS needs math:

- graphics
- cryptography
- artificial intelligence
- numerical analysis
- networking
- databases

⋮

doing well in CSC165

Doing well has two aspects: one being recognized as doing well by being awarded credit (grades), another being able to retain concepts and tools for use later on. Here's how to do both:

- Read the course web page, and emails, regularly. Understand the course information sheet.
- Spend enough time. We assume an average of 8 hours/week — three in lecture, two in tutorial, three preparing, reviewing, or working on assignments.
- Ask questions. Make your own annotations.

ambiguity

When you use a natural language (English, Chinese) you can make it as precise or ambiguous as you need. For some purposes (jokes, gossip) rich ambiguity is essential. For other purposes (getting instructions on heart surgery) precision is essential. We're all equipped to work in both modes. Work out the double meanings of these headlines:

- Prostitutes appeal to Pope
- Death may cause loneliness, feelings of isolation
- Two sisters reunite after 18 years at checkout counter
- Iraqi head seeks arms
- Police begin campaign to run down jaywalkers

precision

We achieve precision by restricting our language. For certain jobs, in certain communities, we use some words or symbols with restricted meanings. Becoming part of the "club" involves learning the definitions of these meanings — my kids don't mean the same thing as I do when they say some car is "sick."
Some words and symbols used in special ways by mathematicians:

?

- continuous
- field — different for mathematicians
- group — two different math thing
- for all (each) $\forall$
- there is (exists) $\exists$

in this course

balance

deterministic

- computers are precise — in identical environments they execute identical instructions identically
- humans are as precise as necessary, and different human audiences require different levels of precision
- The *really* difficult job is finding the right level of precision. Too much precision introduces unbearable tedium; too little introduces unfathomable ambiguity.
- Proofs are primarily works of literature: they communicate with humans, and the best proofs have suspense, pathos, humour and surprise. As a side-effect, proofs present a convincing argument for some fact.

building sets. . . in math

$5 \in S$? True

$6 \in S$? False

$\{1,3\} \subseteq S$? True

$\{1,4\} \subseteq S$? False

$$S = \{1, 3, 5, 7, 9, 11\}$$

Known set

$$\{x \mid x \in S \text{ and } x > 6\}$$

$\{7, 9, 11\}$

$$\{x + 2 \mid x \in S \text{ and } x > 6\}$$

$\{9, 11, 13\}$

building those sets...in python

```
S = {1, 3, 5, 7, 9, 11}

# python doesn't like this!
{x for x in S and x > 6}
```

∈

```
# ... so use this (not standard use of if)
{x for x in S if x > 6}
```

and.

```
{x + 2 for x in S if x > 6}
```

∈ and

any and all...

not True → False
not False → True

If all elements of a set S are not False, then `all(S)` produces True in python.

If at least one element of a set S is not False, then `any(S)` produces True in python.

comment these! 24/4

25/4 → 1
27/4 → 3

Functions q0 through q3 each say something different about the relationship between sets S1 and S2. Write a comment that explains what each function says about this relationship.

True
False

```
def q0(S1, S2):
    ''' (set, set) -> bool
    Return whether ...
    '''
    return not all({x in S2 for x in S1})
```

True
False

```
def q1(S1, S2):
    ''' (set, set) -> bool
    Return whether ...
    '''
    return any({x in S2 for x in S1})
```

0

more comments

```
def q2(S1, S2) :
    ''' (set, set) -> bool
    Return whether ...
    '''
    return all({x in S2 for x in S1})
```

$S1 \subseteq S2$

Every elt of S1 is also elt S2.

```
def q3(S1, S2):
    '''(set, set) -> bool
    Return whether ...
    '''
    return not any({x in S2 for x in S1})
```

verify

Check your comments for q0–q3 in various ways (checking isn't proving, but it increases our confidence or reveals flaws):

- Try out particular values for S1 and S2; see whether the results are consistent with your comments. Check "corner" values, e.g. when one or both lists are empty. Try reversing rôles of S1, S2.
- draw a venn diagram: interlocking circles representing S1 and S2, enclosed in a rectangle representing the "universe" from which list elements may be drawn. Try to make some of the functions q0–q3 false by having elements in some regions. Try to make some of the functions true in a similar way.
- Find sets that create patterns such as `[q0(S1,S2),q1(S1,S2),q2(S1,S2),q3(S1,S2)] = [True,True,False,False]`. Are some patterns impossible?

rigor without mortis

You need both rigor and intuition to solve problems you haven't seen a template for. In this course I'll present some open-ended problems, and recommend the following steps for *getting started* on them:

Understand the problem: Know what's given, what's required. Re-state the problem in your own words, perhaps draw some diagrams.

Plan solution(s): If you've seen something similar, you may be able to use its *result* or its *method*. Work backwards: assume you've solved the problem and think about the next-to-last step. Try solving simpler, smaller versions of the problem. Have more than one plan before you attack the problem (!).

. . . mortis, continued

Carry out your plan: Does it lead somewhere? If not, repeat earlier steps. Articulate *exactly* why and how you're stuck (if you are).

Review: Look back to savour breakthroughs and think about roadblocks. Verify your solution as much as possible. Convince a skeptical peer that you have a solution. Extend your solution to new problems. . .

streetcar drama

Do the first two recommended steps of problem-solving for the following puzzle:

A: Haven't seen you in a long time! How old are your three kids now?

B: The product of their ages (rounded down to nearest year) is 36.

A: That doesn't really answer my question.

B: Well, the sum of their ages is — [fire engine goes by]

A: That still doesn't tell me how old they are.

B: Well, the eldest plays piano.

A: Okay, I see: their ages are — [you have to get off the streetcar]

quantifiers

Quantifiers connect properties of elements to properties of sets. The statement "Every employee earns less than 70,000" isn't about one or even several employees — it's about an entire set of employees. How do you check whether it's true? What if 70,000 is replaced by 40,000?

Employee	Gender	Salary
Al	male	60,000
Betty	female	500
Carlos	male	40,000
Doug	male	30,000
Ellen	female	50,000
Flo	female	20,000

universal claims: $\forall$

How do you verify/disprove:

- All female employees earn less than 55,000.
- All employees earning less than 55,000 are female.
- All male employees earn less than 55,000.

Employee	Gender	Salary
Al	male	60,000
Betty	female	500
Carlos	male	40,000
Doug	male	30,000
Ellen	female	50,000
Flo	female	20,000

existential claims: $\exists$

Another sort of claim "Some employee earns less than 15,000" appear to be about some individual, un-named, employee. Other phrasings "There exists an employee who earns less than 15,000" or "At least one employee earns less than 15,000" also seem to be about the property of some individual employee. However the small modification "Some employee earns more than 15,000" makes it clear that we are talking about the non-emptiness of {Al, Carlos, Doug, Ellen, Flo}.

Employee	Gender	Salary
Al	male	60,000
Betty	female	500
Carlos	male	40,000
Doug	male	30,000
Ellen	female	50,000
Flo	female	20,000

Notes

monday's annotated slides