CS3110 Spring 2016 Lecture 14: Specifications, Computational Evidence, and Cantor's Theorem from Bishop

Topics

1. Discuss prelim and Piazza question on logical specifications.
2. Cantor's Theorem.

Prelim

This will be discussed in recitations as necessary.

Syntax and evaluation – Notice that the notion of a *canonical form* allows us to express the problems very clearly and succinctly. How else could we state quetsions V1 and V2?

Notice how subtle it is to talk about finite sets (types of reals) because we cannot decide equality on reals.

One student posed this approach to finding the maximum in a list of reals. Use Bishop's max operation, folding over the list.

We will also look at the "computational meaning" of one of Bishop's simple results:

$$x, y \in \mathbb{R}.(x > 0 \;\&\; y > 0 \Rightarrow x + y > 0).$$

Additional facts about the reals

- Reals are uncountable, Cantor's theorem.
- The rationals are countable, and from an enumeration, we can construct an irrational.

Piazza logic question

Is $(L\alpha|R\beta) \to \gamma$ the same as $(L\alpha \to \gamma)|(R\beta \to \gamma)$?

In ordinary logical notation this is the same as asking if $(\alpha \vee \beta) \to \gamma$ is "the same as" $(\alpha \to \gamma) \vee (\beta \to \gamma)$?

What does "the same as" mean?

- Does not mean *syntactically* the same.
- Does the same program solve both?
- Can we "convert" a program for one into a program for the other?
- Is there a program that will transform any solution (program) for one into a solution for the other?

Suppose we have a function $f \in (\alpha \vee \beta) \to \gamma$. Can we use it to build a function in either $(\alpha \to \gamma)$ or $(\beta \to \gamma)$?

What would the function look like? fun $f \to$ ...

If we give f an La in $\alpha \vee \beta$, it will produce a value in γ, in $\alpha \to \beta$ in fact. Hence fun $f \to L(\text{fun } a \to f(La))$ where $f \in (\alpha \vee \beta) \to \gamma$.

Also fun $f \to R(\text{fun } b \to f(Rb))$.

Hence $((\alpha \vee \beta) \to \gamma) \to (\alpha \vee \gamma) \vee (\beta \to \gamma)$.

What about the "other direction"? $((\alpha \to \gamma) \vee (\beta \to \gamma)) \to ((\alpha \vee \beta) \to \gamma)$.

This seems harder and perhaps impossible, e.g. suppose we have $\alpha \to \gamma$, how to compute $(\alpha \vee \beta) \to \gamma$?

How could we show that this is impossible?

If we looked at Boolean logic, we could ask whether the logical expression is always true under all assignments of T, F to α, β, γ. This is not a large "*truth table.*"

If we take $\alpha = \mathrm{T}$, $\beta = \mathrm{F}$, $\gamma = \mathrm{F}$ then

$$\begin{array}{cccccccc} (\alpha & \to & \gamma) & \vee & (\beta & \to & \gamma) & \text{has value T} \\ \mathrm{T} & & \mathrm{F} & & \mathrm{F} & & \mathrm{F} & \\ & \mathrm{F} & & & & \mathrm{T} & & \\ & & & \mathrm{T} & & & & \end{array}$$

but

$$\begin{array}{cccccc} (\alpha & \vee & \beta) & \to & \gamma & \text{has value F} \\ \mathrm{T} & & \mathrm{F} & & \mathrm{F} & \\ & \mathrm{T} & & \to & \mathrm{F} & \end{array}$$

Now we use the fact ("deepish") that if a logical formula is programmable, then its truth value is T. *So this formula is <u>not</u> programmable.* It is <u>not</u> a *tautology.*

While we are close to logical issues, let's see what we mean when we say that constructive logic is about finding computational evidence that causes us to know a logical relationship.

How do we know that for x and y real numbers, $(x > 0 \ \& \ y > 0)$ implies $x + y > 0$?

$$\forall x, y : \mathbb{R}.((x > 0 \ \& \ y > 0) \Rightarrow x + y > 0).$$

To say $x > 0$ means that we have an n such that $x_n > \frac{1}{n}$ for some n.

To say $y > 0$ means that we have an m such that $y_m > \frac{1}{m}$ for some m.

So why is $x + y > 0$?

Recall $x + y = (x_{2n} + y_{2n})$, $n = 1, 2, 3,...$

Knowing this is a matter of simple arithmetic, computing the right evidence bound.

In the realm of analysis, the *evidence is numerical.* When we examine other topics, such as properties of geometric figures, we can reduce the evidence to numbers. In other realms, such as Euclidean geometry, the evidence might be more "abstract," i.e. geometric constructions as in Euclid creating an equilateral triangle using ruler and compass.

Cantor's Theorem

Cantor's Theorem (2.19) Bishop page 27.

Let (a_n) be a sequence of real numbers, and let x_0 and y_0 be reals with $x_0 < y_0$. Then we can construct a real number x such that

- $x_0 \leq x \leq y_0$ and
- $x \neq a_n$ for all $n \in \mathbb{N}^+$.

This says that we cannot computationally enumerate all the real numbers in an interval $[x_0, y_0]$. There are "uncountably many of them."

What do we get if the sequence of reals are the rational numbers in this interval? The construction will provide an irrational number.

Let (a_n) be a sequence of real numbers, e.g. it could be an attempted enumeration of $\mathbb{R}$. Note, (a_n) is computable.

Let $x_0 < y_0$ define an interval on the real line.

Then we can find a real number x such that:

1. $x \neq a_n$ for all $n \in \mathbb{N}$ and
2. $x_0 \leq x \leq y_0$.

Proof

Assume that $x_0, ..., x_{n-1}$ and $y_0, ...y_{n-1}$ have been constructed.

Either

(a) $a_n > x_{n-1}$ or

(b) $a_n < y_{n-1}$.

In case (a) let x_n be any rational with $x_{n-1} < x_n < \min\{a - N, y_{n-1}\}$ and y_n such that $x_n < y_n < \min\{a_n, y_{n-1}, x_n + \frac{1}{n}\}$.

We can show that $|x_m - x_n| = x_m - x_n < y_n - x_n < \frac{1}{n}$, $m \geq n$. See Bishop for case (b).

QED