# *Lecture 10: Non-Computable Functions*

Dr John Levine

CS103 Machines, Languages and Computation
October 26th 2015

# Preparing for the Class Test

- The first class test is on Friday 30th October at 10am
- What do you need to do to prepare for this?
- Have a look at “Class Test 1: Syllabus”
- Revise the material from the lecture slides
- Do (or redo) all five assignments
- Make sure you attend this week’s tutorial – I will be going over all of the assignments

# Class Test Format

- The class test will consist of five questions, and you do all five of them.
- There will be one question on each assignment.
- The questions in the test will be closely related to the five assignments.
- Each question is marked out of 20, you need 40% to pass the test, 60% average for exemption.

# Today's Lecture

- How there are more dogs than collars on Planet Zog (where the dogs' names are real numbers)
- Types of sets: finite, countable, uncountable.
- How big is the set of all computer programs?
- How big is the set of all functions from $N \longrightarrow N$ ?
- Are there more functions than there are programs?

# Dogs and Collars

- Consider an infinite set of dogs.
- Each dog has a name, which is a finite string of letters (all in uppercase), such as “FIDO” or “GROMIT”.
- Consider an infinite set of collars, where every collar is labelled with an integer $n \in \{0, 1, 2, \ldots\}$.
- Are there enough collars for all the dogs?
- In other words, can I find a computable mapping from the dog’s name to an integer, so that every dog gets a unique collar?
- And does every collar have a unique dog?

# Numbering the Dogs

- The strings have a natural order, so we can use that.
- First, assign the number 0 to the empty string “” (The Dog With No Name)
- Next, line up all the strings of length 1 in alphabetical order: “A”, “B”, … “Z”. These are numbered 1 up to 26.
- Next, line up all the strings of length 2: “AA”, “AB”, …, “AZ”, “BA”, …, “ZZ”. There are 676 of these, so they get numbered from 27 up to 702.
- Length 3: “AAA” $\rightarrow$ 703, …, “ZZZ” $\rightarrow$ 18278, and so on

# Reversing the Mapping

- Decoding the collars (i.e. mapping from collar numbers back to strings) is a bit tricky, but can be done:

```
> (map-dog-to-collar  “AAA”)
703
> (map-collar-to-dog  703)
“AAA”
> (map-dog-to-collar  “GROMIT”)
91667882
> (map-collar-to-dog  91667882)
“GROMIT”
>
```

# Assignment 4

1. Which collar is given to the dog called “ODIE”?

2. What is that name of the dog that gets collar number 106503?

# Real Numbered Dogs?

- On Planet Zog, they name their dogs not with strings of letters, but with real numbers.
- The real numbers are the decimal numbers, including recurring numbers like $0.33333333333\ldots$ and irrational numbers like $\sqrt{2}$ and $\pi$ which never repeat themselves.
- So the Zogians have dogs called things like: $3.58$, $15.6$, $0.3333333333\ldots$, $\sqrt{2}$, $\sqrt{3}$, $\pi$, $e$, $\ldots$
- If the collars are still numbered with integers from $0$ upwards, are there enough collars for all the dogs?

# A Possible Mapping?

- Let's try a possible mapping from $N$ to real numbers in the range 0.0 → 1.0 (i.e. a subset of $R$):

$$
\begin{array}{l}
0 \leftrightarrow 0\ .\ 3\ 3\ 3\ 3\ 3\ \ldots \\
1 \leftrightarrow 0\ .\ 3\ 1\ 4\ 1\ 5\ \ldots \\
2 \leftrightarrow 0\ .\ 5\ 0\ 0\ 0\ 0\ \ldots \\
3 \leftrightarrow 0\ .\ 0\ 1\ 4\ 1\ 5\ \ldots \\
4 \leftrightarrow 0\ .\ 0\ 1\ 3\ 1\ 5\ \ldots \\
\ldots
\end{array}
$$

- Have we managed to get a collar for every Zogian dog who has a name in the range 0.0 → 1.0?

# The Diagonalisation Proof

- Let's try a possible mapping from $N$ to real numbers in the range 0.0 → 1.0 (i.e. a subset of $R$):

$$
\begin{array}{l}
0 \leftrightarrow 0\ .\ 3\ 3\ 3\ 3\ 3\ \ldots \\
1 \leftrightarrow 0\ .\ 3\ 1\ 4\ 1\ 5\ \ldots \\
2 \leftrightarrow 0\ .\ 5\ 0\ 0\ 0\ 0\ \ldots \\
3 \leftrightarrow 0\ .\ 0\ 1\ 4\ 1\ 5\ \ldots \\
4 \leftrightarrow 0\ .\ 0\ 1\ 3\ 1\ 5\ \ldots \\
\ldots
\end{array}
$$

- Form a new number by adding 1 to the digits shown:

$? \leftrightarrow 0\ .\ 4\ 2\ 1\ 2\ 6\ \ldots$ no collar for this dog!

# Zog’s Dogs are Uncountable

- The diagonalisation proof shows that it is impossible to align the real numbers with the integers.
- No matter what mapping we try, there will clearly be a large number of Zogian dogs who will get no collar!
- This shows that the set of all real numbers is a “larger” infinite set than the set of all integers.
- The set of integers is called a *countable* infinite set.
- The set of real numbers is called an *uncountable* infinite set.

# Finite and Infinite Sets

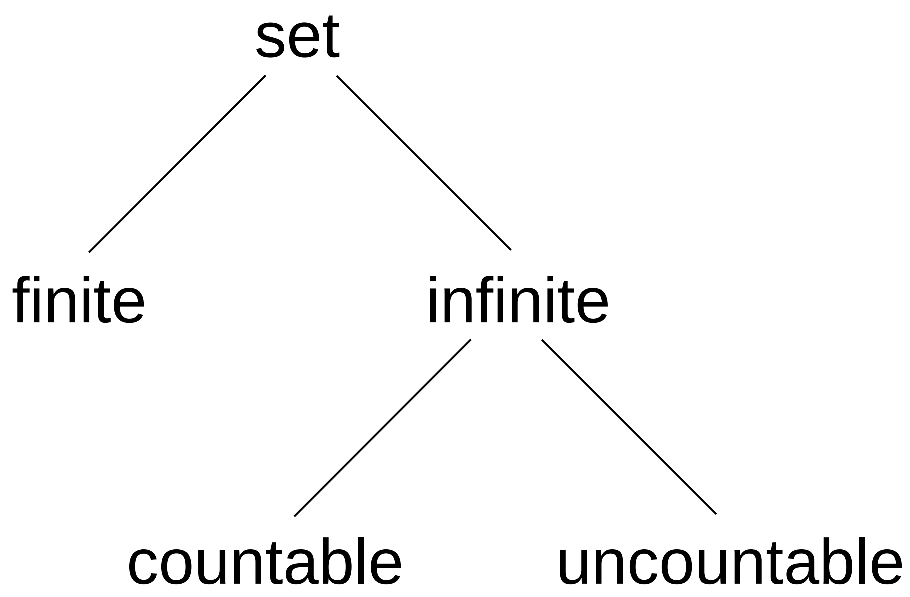
- A *finite* set is one which obeys the following property: if I list all the members of the set, I will eventually stop.
- Example: $S$ = the set of all odd numbers < 1000.
- $S = \{1, 3, 5, 7, 9, 11, 13, 15, \ldots, 993, 995, 997, 999\}$.
- An *infinite* set is one where, if I list all the members of the set, I will never stop:
- Example: $T$ = the set of all odd numbers > 1000.
- $T = \{1001, 1003, 1005, 1007, 1009, 1011, 1013, \ldots$

# Countable and Uncountable Sets

- There are two types of infinite sets: *countable* sets and *uncountable* sets.
- Countable infinite sets can be made to align with the integers, like the dogs with strings for names.
- Example: the set of all strings of the MIU system.
- Uncountable infinite sets cannot be made to align with the integers, no matter what mapping you try.
- Example: the set of all real numbers.
- Uncountable infinite sets are “bigger” than countable infinite sets.

# Countable and Uncountable Sets

- Summary of different types of sets:



# Non-Computable Functions

- The first 5 weeks of the course will build up towards the proof that non-computable functions exist:

  If $P$ is the set of all computer programs and $F$ is the set of all functions $f: n \rightarrow m$ such that $n,m$ are members of the set of natural numbers $N$, then $|F| > |P|$ and therefore there are some functions in $F$ for which no computer program can exist.

# How Big is the Set of all Programs?

- Computer programs, written in languages like Python, are just strings of characters (chosen from a finite set of characters).
- This means we can map any program to an integer using the same sort of numbering scheme that we used for “FIDO” and “GROMIT”.
- All computer programs get turned into binary anyway, and any binary string is clearly an integer.
- Since all programs can be transformed into integers, the set of all programs is *infinite and countable.*

# How Big is the Set of all Functions?

- The set of all possible functions from $N = \{0, 1, 2, \ldots\}$ to $N = \{0, 1, 2, \ldots\}$ defines the set of all programs that we might want to write.
- Example: $f(n) = \{(0, 0), (1, 1), (2, 4), (3, 9), (4, 16), \ldots\}$
- How many different possible functions are there?
- Is it a countable set, like the set of all programs?
- Can we align all possible functions with the integers?

# A Possible Mapping?

- As with the dogs, let's try to align all the functions:

$$
\begin{aligned}
0 &\leftrightarrow f_0(n) = \{(0, 0), (1, 0), (2, 0), (3, 0), \ldots\} \\
1 &\leftrightarrow f_1(n) = \{(0, 0), (1, 1), (2, 2), (3, 3), \ldots\} \\
2 &\leftrightarrow f_2(n) = \{(0, 0), (1, 1), (2, 0), (3, 1), \ldots\} \\
3 &\leftrightarrow f_3(n) = \{(0, 1), (1, 0), (2, 0), (3, 0), \ldots\} \\
&\vdots
\end{aligned}
$$

- Two functions are not equal if they differ on the output value for any input value (e.g. $f_0$ and $f_3$ are different functions even though they only differ on one value)

# A Possible Mapping?

- As with the dogs, let's try to align all the functions:
  $0 \leftrightarrow f_0(n) = \{(0, 0), (1, 0), (2, 0), (3, 0), \ldots\}$
  $1 \leftrightarrow f_1(n) = \{(0, 0), (1, 1), (2, 2), (3, 3), \ldots\}$
  $2 \leftrightarrow f_2(n) = \{(0, 0), (1, 1), (2, 0), (3, 1), \ldots\}$
  $3 \leftrightarrow f_3(n) = \{(0, 1), (1, 0), (2, 0), (3, 0), \ldots\}$
  $\vdots$
- Now consider the function $f'(n) = f_n(n) + 1$
  $f'(0) = f_0(0) + 1 = 1$, so $f'$ is not the same as $f_0$
  $f'(1) = f_1(1) + 1 = 2$, so $f'$ is not the same as $f_1$
  $f'(2) = f_2(2) + 1 = 1$, so $f'$ is not the same as $f_2$
  $f'(3) = f_3(3) + 1 = 1$, so $f'$ is not the same as $f_3$
  $\vdots$

# The Functions are Uncountable

- What did we just do?
- We took all the integers, and aligned each integer with a function, so that every integer had a function
- We then created a new function, $f'(n) = f_n(n) + 1$, which is a *different* function from all of the functions already associated with an integer
- This must mean that the number of functions from integers to integers is uncountable…
- And therefore there are more functions than there are programs. ■

# Assignment 5

All sets are either finite or infinite, and all infinite sets are either countable or uncountable.

(a) Define what is mean by a *finite set.*

(b) Define what is mean by a *countable infinite set.*

(c) Define what is meant by an *uncountable infinite set*.

# Assignment 5

(d) Label each of the following sets as finite or infinite, then label the infinite sets as countable or uncountable:

(i) the set of all students taking CS103
(ii) the set of all theorems of the MIU system
(iii) the set of all the prime numbers
(iv) the set $\{n \mid n \in N \text{ and } n < 5\}$
(v) the set of all the real numbers
(vi) the set of all dogs with four letter names
(vii) the set of all possible subsets of $N$
(viii) the set of all computer programs
(ix) the set of all functions from $N \rightarrow N$