A Universal Language

One-Slide Summary

- The **lambda calculus** is a universal, fundamental model of computation. You can view it as "the essence of Scheme". It contains terms and rules describing variables, function abstraction, and function application.
- There are two key reduction rules in the lambda calculus. Alpha reduction allows you to rename variables uniformly. **Beta reduction** is the essence of computation: in beta reduction, a function evaluation is equivalent to replacing all instances of the formal parameter in the function body with the actual argument.
- It is possible to **encode** programming concepts, such as true, false, if, numbers, plus, etc., in the lambda calculus.

Final Project Presentations

- December 6th, 3:30-4:45
  - Optional: Game Theory, OLS 011
- December 6th, 5:00pm+
  - Optional: OLS 009
- Attending is worth extra credit.
  - And you'll see the fun projects of your fellow students.
- You must request to give a presentation.
- Requests are due Dec 04.

λ-calculus

Alonzo Church, 1940

(LISP was developed from λ-calculus, not the other way round.)

$$\begin{aligned} term = \; & variable \\ & \mid term\ term \\ & \mid \lambda\ variable\ .\ term \end{aligned}$$

What is Calculus?

- In High School:

  $d/dx\ x^n = nx^{n-1}$ [Power Rule]

  $d/dx\ (f + g) = d/dx\ f + d/dx\ g$ [Sum Rule]

  Calculus is a branch of mathematics that deals with limits and the differentiation and integration of functions of one or more variables...

Surprise Liberal Arts Trivia

- This branch of mathematics involving symbolic expressions manipulated according to fixed rules takes its name from the diminutive form of calx/calcis, the latin word for rock or limestone. The diminutive word thus means "pebble": in ancient times pebbles were placed in sand and used for counting using techniques akin to those of the abacus.

Real Definition

- A **calculus** is just a bunch of rules for manipulating symbols.
- People can give meaning to those symbols, but that's not part of the calculus.
- Differential calculus is a bunch of rules for manipulating symbols. There is an interpretation of those symbols corresponds with physics, slopes, etc.

Lambda Calculus

- Rules for manipulating strings of symbols in the language:

term = *variable*
| *term term*
| λ *variable* . *term*

- Humans can give meaning to those symbols in a way that corresponds to computations.

Why?

- Once we have precise and formal rules for manipulating symbols, we can reason with those symbols and rules.
- Since we can interpret the symbols as representing computations, we can use this system to reason about programs.
- (It will provide additional evidence that Scheme and Turing machines have equivalent computational power.)

Evaluation Rules

α-reduction (renaming)

$\lambda y.\, M \Rightarrow_{\alpha} \lambda v.\, (M\,[\text{each } y \text{ replaced by } v])$

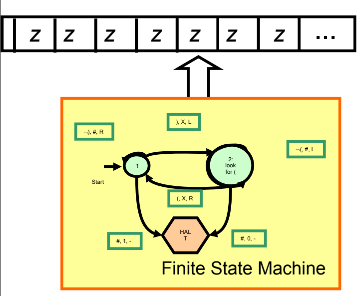
where v does not occur in M.

β-reduction (substitution)

$(\lambda x.\, M)N \Rightarrow_{\beta} M\,[\text{each } x \text{ replaced by } N\,]$

We'll see examples in a bit!

Equivalent Computers?



Turing Machine

≡

term = *variable*
| *term term*
| (*term*)
| λ *variable* . *term*

$\lambda y.\, M \Rightarrow_{\alpha} \lambda v.\, (M\,[y \rightarrow v])$

where v does not occur in M.

$(\lambda x.\, M)N \Rightarrow_{\beta} M\,[\,x \rightarrow N\,]$

Lambda Calculus

Liberal Arts Trivia: Music

- This music genre originated in Jamaica in the 1950s and was the precursor to reggae. It combines elements of Caribbean mento and calypso with American jazz and rhythm and blues. It is characterized by a walking bass line accented with rhythms on the offbeat. In the 1980s it experience a third wave revival and is often associated with punk and brass instruments.

Liberal Arts Trivia: Geography

- This baltic country borders Romania, Serbia, Macedonia, Greece, Turkey and the Black Sea. It was at one point ruled by the Ottomans, but is now a member of the EU and NATO. Sofia, the capital and largest city, is one of the oldest cities in Europe and can be traced back some 7000 years. The traditional cuisine of this country features rich salads at every meal, as well as native pastries such as the *banitsa*.

Lambda Examples

- Identity Function
  - Identity = lambda x : x
  - **identity** = λ x. x
- Square Function
  - Square = lambda x : x * x
  - **square** = λ x. (x * x)
- Add Function
  - add = lambda x, y : x + y
  - add = lambda x : lambda y : x + y
  - **add** = λ x. λ y. (x + y)

β-Reduction (the source of all computation)

$$(\lambda x.\ M)N \Rightarrow_\beta M\,[\,x \rightarrow N\,]$$

Replace all x's in M with N's

Note the syntax is different from Python:
$(\lambda x.M)N$ === (lambda x: M)(N)

β-Reduction Examples

Recall: $(\lambda x.\ M)N \Rightarrow_\beta M\,[\,x \rightarrow N\,]$

- Square Function
  - **square** = λ x. (x * x)
  - (λ x. (x * x)) 5
  - (λ x. (x * x)) 5 $\Rightarrow_\beta$ (x * x)[x→5]
  - (λ x. (x * x)) 5 $\Rightarrow_\beta$ (x * x)[x→5] $\Rightarrow_\beta$ (5 * 5)
- Add Function
  - **add** = λ x. λ y. (x + y)
  - (λ x. λ y. (x + y)) 3 $\Rightarrow_\beta$???
  - ((λ x. λ y. (x + y)) 2) 6 $\Rightarrow_\beta$???

Get out some paper!

β-Reduction Examples

Recall: $(\lambda x.\ M)N \Rightarrow_\beta M\,[\,x \rightarrow N\,]$

- Square Function
  - **square** = λ x. (x * x)
  - (λ x. (x * x)) 5
  - (λ x. (x * x)) 5 $\Rightarrow_\beta$ (x * x)[x→5]
  - (λ x. (x * x)) 5 $\Rightarrow_\beta$ (x * x)[x→5] $\Rightarrow_\beta$ (5 * 5)
- Add Function
  - **add** = λ x. λ y. (x + y)
  - (λ x. λ y. (x + y)) 3 $\Rightarrow_\beta$ **λ y. (3 + y)**
  - ((λ x. λ y. (x + y)) 2) 6 $\Rightarrow_\beta$ (λ y. (2 + y)) 6 $\Rightarrow_\beta$ **(2 + 6)**

Evaluating Lambda Expressions

- ***redex***: Term of the form $(\lambda x.\ M)N$

  Something that can be β-reduced
- An expression is in ***normal form*** if it contains no redexes (*redices*).
- To evaluate a lambda expression, keep doing reductions until you get to *normal form*.

Example

$$\lambda f. ((\lambda x.f (xx)) (\lambda x. f (xx)))$$

Do it on paper!

Possible Answer

$(\lambda f. ((\lambda x.f (xx)) (\lambda x. f (xx)))) (\lambda z.z)$

$\rightarrow_\beta (\lambda x.(\lambda z.z)(xx)) (\lambda x. (\lambda z.z)(xx))$

$\rightarrow_\beta (\lambda z.z) (\lambda x.(\lambda z.z)(xx)) (\lambda x.(\lambda z.z)(xx))$

$\rightarrow_\beta (\lambda x.(\lambda z.z)(xx)) (\lambda x.(\lambda z.z)(xx))$

$\rightarrow_\beta (\lambda z.z) (\lambda x.(\lambda z.z)(xx)) (\lambda x.(\lambda z.z)(xx))$

$\rightarrow_\beta (\lambda x.(\lambda z.z)(xx)) (\lambda x.(\lambda z.z)(xx))$

$\rightarrow_\beta \ldots$

Alternate Answer

$(\lambda f. ((\lambda x.f (xx)) (\lambda x. f (xx)))) (\lambda z.z)$

$\rightarrow_\beta (\lambda x.(\lambda z.z)(xx)) (\lambda x. (\lambda z.z)(xx))$

$\rightarrow_\beta (\lambda x.xx) (\lambda x.(\lambda z.z)(xx))$

$\rightarrow_\beta (\lambda x.xx) (\lambda x.xx)$

$\rightarrow_\beta (\lambda x.xx) (\lambda x.xx)$

$\rightarrow_\beta \ldots$

Be Very Afraid!

- Some λ-calculus terms can be β-reduced forever!
  - Just like some computer programs, which can evaluate forever
- The order in which you choose to do the reductions might change the result!
  - Just like lazy evaluation vs. eager evaluation

Liberal Arts Trivia: Classics

- The Temple of Artemis at Ephesus, the Statue of Zeus at Olympus, and the Tomb of Maussollos are three of the **Seven Wonders of the Ancient World**. Name the other four.

Liberal Arts Trivia: Biology

- These even-toed ungulate have one or two distinctive fatty deposits on their backs. They are native to the dry desert areas of Asia. They are domesticated to provide meat and milk, as well as to serve as beasts of burden. The US Army had an active cavalry corps based on these beasts in California in the 19th century, and they have been used in wars throughout Africa.

Liberal Arts Trivia: British Lit

- This 1883 coming-of-age tale of “pirates and buried gold” by Robert Louis Stevenson had a vast influence on the popular perception of pirates. Its legacies include treasure maps with an “X”, the Black Spot, tropical islands, and one-legged seamen with parrots on their shoulders.
  - Name the book.
  - Name the morally gray, parrot-holding mutineer.

Universal Language

- Is Lambda Calculus a ***universal language***?
  - Can we compute any computable algorithm using Lambda Calculus?
- To prove it is **not**:
  - Find *some* Turing Machine that *cannot* be simulated with Lambda Calculus
- To prove it **is**:
  - Show you can simulate *every* Turing Machine using Lambda Calculus

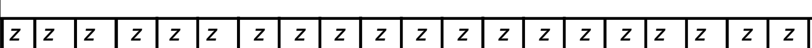
Universal Language

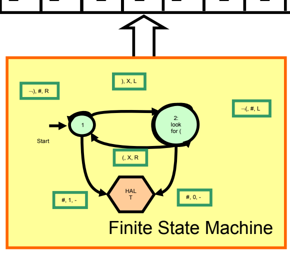
- Is Lambda Calculus a ***universal language***?
  - Can we compute any computable algorithm using Lambda Calculus?
- To prove it is **not**:
  - Find *some* Turing Machine that *cannot* be simulated with Lambda Calculus
- To prove it **is**:
  - Show you can simulate *every* Turing Machine using Lambda Calculus

Simulating *Every* Turing Machine

- A **Universal** Turing Machine can simulate every Turing Machine
- So, to show Lambda Calculus can simulate every Turing Machine, all we need to do is show it can simulate a Universal Turing Machine!

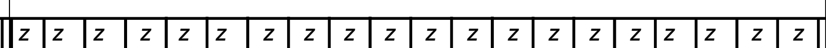
Simulating Computation

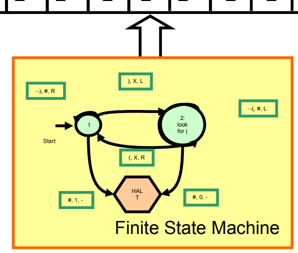




- Lambda expression corresponds to a computation: input on the tape is transformed into a lambda expression
- Normal form is that value of that computation: output is the normal form
- How do we simulate the FSM?

Simulating Computation





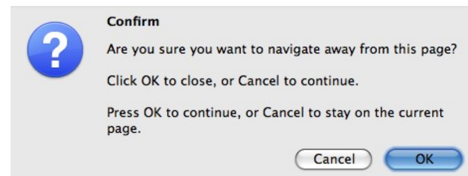
Read/Write Infinite Tape
Mutable Lists
Finite State Machine
Numbers
Processing
Way to make decisions (if)
Way to keep going

Making "Primitives" from Only Glue (λ)

In search of *the truth*?

- What does **true** mean?
- **True** is something that when used as the first operand of **if**, makes the value of the **if** the value of its second operand:

  if T M $N \rightarrow M$



Don't search for T, search for if

$$\mathbf{T} \equiv \lambda x\ (\lambda y.\ x)$$

$$\equiv \lambda xy.\ x$$

$$\mathbf{F} \equiv \lambda x\ (\lambda\ y.\ y))$$

$$\mathbf{if} \equiv \lambda pca\ .\ pca$$

The Truth Is Out There

$\mathbf{T} \equiv \lambda x\ .\ (\lambda y.\ x)$

$\mathbf{F} \equiv \lambda x\ .\ (\lambda y.\ y)$

$\mathbf{if} \equiv \lambda p\ .\ (\lambda c\ .\ (\lambda a\ .\ pca)))$

$\mathbf{if\ T}\ \boldsymbol{M}\ \boldsymbol{N}$

$((\lambda pca\ .\ pca)\ (\lambda xy.\ x))\ \boldsymbol{M}\ \boldsymbol{N}$

$\rightarrow_\beta$???

Finding the Truth

$\mathbf{T} \equiv \lambda x\ .\ (\lambda y.\ x)$

$\mathbf{F} \equiv \lambda x\ .\ (\lambda y.\ y)$

$\mathbf{if} \equiv \lambda p\ .\ (\lambda c\ .\ (\lambda a\ .\ pca)))$

$\mathbf{if\ T}\ \boldsymbol{M}\ \boldsymbol{N}$

$((\lambda pca\ .\ pca)\ (\lambda xy.\ x))\ \boldsymbol{M}\ \boldsymbol{N}$

$\rightarrow_\beta (\lambda ca\ .\ (\lambda x.(\lambda y.\ x))\ ca))\ \boldsymbol{M}\ \boldsymbol{N}$

$\rightarrow_\beta \rightarrow_\beta (\lambda x.(\lambda y.\ x))\ \boldsymbol{M}\ \boldsymbol{N}$

$\rightarrow_\beta (\lambda y.\ \boldsymbol{M}\ ))\ \boldsymbol{N} \rightarrow_\beta \boldsymbol{M}$

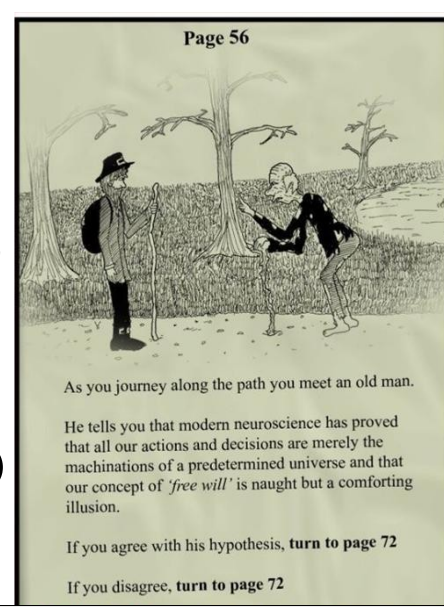
and and or?

$\mathbf{and} \equiv$

$\lambda x\ (\lambda y.\ \mathbf{if}\ x\ y\ \mathbf{F}))$

$\mathbf{or} \equiv$

$\lambda x\ (\lambda y.\ \mathbf{if}\ x\ \mathbf{T}\ y))$


Page 56

As you journey along the path you meet an old man.

He tells you that modern neuroscience has proved that all our actions and decisions are merely the machinations of a predetermined universe and that our concept of *'free will'* is naught but a comforting illusion.

If you agree with his hypothesis, **turn to page 72**

If you disagree, **turn to page 72**

What is 42?

42

forty-two

XLII

cuarenta y dos

#37

Meaning of Numbers

- "42-ness" is something who's **successor** is "43-ness"
- "42-ness" is something who's **predecessor** is "41-ness"
- "Zero" is special. It has a **successor** "one-ness", but no **predecessor**.

#38

Meaning of Numbers

pred (succ N) $\rightarrow N$

succ (pred N) $\rightarrow N$

succ (pred (succ N)) $\rightarrow$ succ N

zero? **zero** $\rightarrow$ **T**

zero? (succ **zero**) $\rightarrow$ **F**

#39

Is this enough?

Can we define **add** with **pred**, **succ**, **zero?** and **zero**?

$$\textbf{add} \equiv \lambda xy.\textbf{if}\ (\textbf{zero?}\ x)\ y\ (\textbf{add}\ (\textbf{pred}\ x)\ (\textbf{succ}\ y))$$

#40

Can we define lambda terms that behave like **zero**, **zero?**, **pred** and **succ**?

Hint: what if we had **cons**, **car** and **cdr**?

cons(x,y) = x + [y]

car(x) = x[0]

cdr(x) = x[1:]

#41

Numbers are Lists...

$$\textbf{zero?} \equiv \textbf{null?}$$

$$\textbf{pred} \equiv \textbf{cdr}$$

$$\textbf{succ} \equiv \lambda\ x\ .\ \textbf{cons}\ \textbf{F}\ x$$

The *length* of the list corresponds to the number value.

#42

Liberal Arts Trivia: Religious Studies

- In Sunni Islam, the Five Pillars of Islam are five duties incumbent on Muslims. They include the Profession of Faith, Formal Prayers, and Giving Alms. Name the remaining two pillars.

Making Pairs

```
def make-pair(x,y):
  return lambda selector: \
    x if selector else y

def car-of-pair(p): return p(True)
def cdr-of-pair(p): return p(False)
```

A pair is just an if statement that chooses between the car (then) and the cdr (else).

cons and car

cons $\equiv \lambda x.\lambda y.\lambda z.zxy$

Example: cons M N = $(\lambda x.\lambda y.\lambda z.zxy)$ M N

$\rightarrow_\beta (\lambda y.\lambda z.z\mathrm{M}y)$ N

$\rightarrow_\beta \lambda z.z\mathrm{MN}$

car $\equiv \lambda p.p$ **T**

T $\equiv \lambda xy.\ x$

Example: car (cons M N) $\equiv$ car $(\lambda z.z\mathrm{MN}) \equiv (\lambda p.p$ **T**$)$ $(\lambda z.z\mathrm{MN}) \rightarrow_\beta (\lambda z.z\mathrm{MN})$ **T** $\rightarrow_\beta$ TMN

$\rightarrow_\beta (\lambda xy.\ x)$ MN

$\rightarrow_\beta (\lambda y.\ \mathrm{M})\mathrm{N}$

$\rightarrow_\beta$ M

cdr too!

cons $\equiv \lambda xyz.zxy$

car $\equiv \lambda p.p$ T

cdr $\equiv \lambda p.p$ F

Example: cdr (cons *M N)*

cdr $\lambda z.z\mathrm{MN}$ = $(\lambda p.p\ \mathrm{F})\ \lambda z.z\mathrm{MN}$

$\rightarrow_\beta (\lambda z.z\mathrm{MN})$ F

$\rightarrow_\beta$ FMN

$\rightarrow_\beta$ N

Null and null?

null $\equiv \lambda x.$**T**

null? $\equiv \lambda x.(x\ \lambda y.\lambda z.$**F**$)$

Example:

null? null $\rightarrow \lambda x.(x\ \lambda y.\lambda z.$**F**$)$ $(\lambda x.$ **T**$)$

$\rightarrow_\beta (\lambda x.$ **T**$)(\lambda y.\lambda z.$**F**$)$

$\rightarrow_\beta$ **T**

Null and null?

null $\equiv \lambda x.$**T**

null? $\equiv \lambda x.(x\ \lambda y.\lambda z.$**F**$)$

Example:

null? (cons M N) $\rightarrow \lambda x.(x\ \lambda y.\lambda z.$**F**$)$ $\lambda z.z\mathrm{MN}$

$\rightarrow_\beta (\lambda z.z\ \mathrm{MN})(\lambda y.\lambda z.$**F**$)$

$\rightarrow_\beta (\lambda y.\lambda z.$**F**$)$ MN

$\rightarrow_\beta$ **F**

Counting

$\mathbf{0} \equiv$ null

$\mathbf{1} \equiv \mathbf{cons\ F\ 0}$

$\mathbf{2} \equiv \mathbf{cons\ F\ 1}$

$\mathbf{3} \equiv \mathbf{cons\ F\ 2}$

…

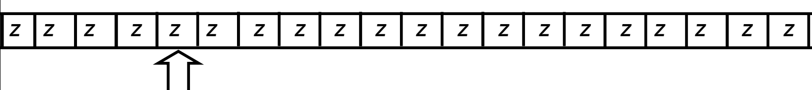
$\mathbf{succ} \equiv \lambda x.\mathbf{cons\ F}\ x$

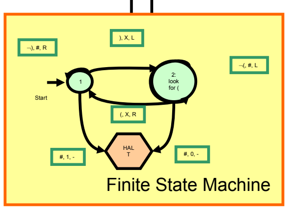
$\mathbf{pred} \equiv \lambda x.\mathbf{cdr}\ x$

$42 = \lambda xy.(\lambda z.z\ xy)\ \lambda xy.\ y\ \lambda xy.(\lambda z.z\ xy)\ \lambda xy.\ y$
$\lambda xy.(\lambda z.z\ xy)\ \lambda xy.\ y\ \lambda xy.(\lambda z.z\ xy)\ \lambda xy.\ y\ \lambda xy.(\lambda z.z\ xy)\ \lambda xy.\ y\ \lambda xy.$
$(\lambda z.z\ xy)\ \lambda xy.\ y\ \lambda xy.(\lambda z.z\ xy)\ \lambda xy.\ y\ \lambda xy.(\lambda z.z\ xy)\ \lambda xy.\ y\ \lambda xy.(\lambda z.z\ xy)$
$\lambda xy.\ y\ \lambda xy.(\lambda z.z\ xy)\ \lambda xy.\ y\ \lambda xy.(\lambda z.z\ xy)\ \lambda xy.\ y\ \lambda xy.(\lambda z.z\ xy)\ \lambda xy.\ y$
$\lambda xy.(\lambda z.z\ xy)\ \lambda xy.\ y\ \lambda xy.(\lambda z.z\ xy)\ \lambda xy.\ y\ \lambda xy.(\lambda z.z\ xy)\ \lambda xy.\ y\ \lambda xy.$
$(\lambda z.z\ xy)\ \lambda xy.\ y\ \lambda xy.(\lambda z.z\ xy)\ \lambda xy.\ y\ \lambda xy.(\lambda z.z\ xy)\ \lambda xy.\ y\ \lambda xy.(\lambda z.z\ xy)$
$\lambda xy.\ y\ \lambda xy.(\lambda z.z\ xy)\ \lambda xy.\ y\ \lambda xy.(\lambda z.z\ xy)\ \lambda xy.\ y\ \lambda xy.(\lambda z.z\ xy)\ \lambda xy.\ y$
$\lambda xy.(\lambda z.z\ xy)\ \lambda xy.\ y\ \lambda xy.(\lambda z.z\ xy)\ \lambda xy.\ y\ \lambda xy.(\lambda z.z\ xy)\ \lambda xy.\ y\ \lambda xy.$
$(\lambda z.z\ xy)\ \lambda xy.\ y\ \lambda xy.(\lambda z.z\ xy)\ \lambda xy.\ y\ \lambda xy.(\lambda z.z\ xy)\ \lambda xy.\ y\ \lambda xy.(\lambda z.z\ xy)$
$\lambda xy.\ y\ \lambda xy.(\lambda z.z\ xy)\ \lambda xy.\ y\ \lambda xy.(\lambda z.z\ xy)\ \lambda xy.\ y\ \lambda xy.(\lambda z.z\ xy)\ \lambda xy.\ y$
$\lambda xy.(\lambda z.z\ xy)\ \lambda xy.\ y\ \lambda xy.(\lambda z.z\ xy)\ \lambda xy.\ y\ \lambda xy.(\lambda z.z\ xy)\ \lambda xy.\ y\ \lambda xy.$
$(\lambda z.z\ xy)\ \lambda xy.\ y\ \lambda xy.(\lambda z.z\ xy)\ \lambda xy.\ y\ \lambda xy.(\lambda z.z\ xy)\ \lambda xy.\ y\ \lambda xy.(\lambda z.z\ xy)$
$\lambda xy.\ y\ \lambda xy.(\lambda z.z\ xy)\ \lambda xy.\ y\ \lambda xy.(\lambda z.z\ xy)\ \lambda xy.\ y\ \lambda xy.(\lambda z.z\ xy)$

$\lambda xy.\ y\ \lambda x.\mathrm{T}$

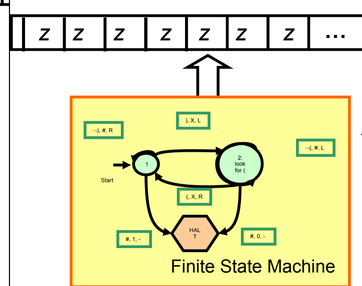
Lambda Calculus is a Universal Computer

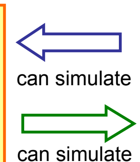




- Read/Write Infinite Tape
  - ✓ Mutable Lists
- Finite State Machine
  - ✓ Numbers to keep track of state
- Processing
  - ✓ Way of making decisions (if)
  - ✓ Way to keep going

Equivalent Computers!





Turing Machine

term = *variable*
| *term term*
| (*term*)
| λ *variable* . *term*

$\lambda y.\ M \Rightarrow_{\alpha} \lambda v.\ (M\ [y \rightarrow v])$
where v does not occur in M.

$(\lambda x.\ M)N \Rightarrow_{\beta} M\ [\ x \rightarrow N\ ]$

Lambda Calculus

Liberal Arts Trivia: Greek Myths

- The Sphinx is said to have guarded the entrance to the city of Thebes and to have asked a riddle of would-be travelers. The Sphinx, originally from Ethiopia, is said to have been sent by Hera or Ares. Oedipus solved her riddle, and is thus seen as a threshold figure, helping to transition between the old religious practices and the new Olympian gods.
- State the Riddle of the Sphinx and its answer.

Universal Computer

- Lambda Calculus can simulate a Turing Machine
  - Everything a Turing Machine can compute, Lambda Calculus can compute also
- Turing Machine can simulate Lambda Calculus (we didn't prove this)
  - Everything Lambda Calculus can compute, a Turing Machine can compute also
- **Church-Turing Thesis**: this is true for ***any*** other mechanical computer also

CS 1120

- Language: Formal Systems, Rules of Eval
- Recursive Definitions
- Programming with Lists
- Programming with Mutation and Objects
- Interpreters, Lazy Eval, Type Checking
- Programming for the Internet
- Measuring Complexity
- Computability
- Models of Computation

Homework

- PS 9 Presentation Requests

Liberal Arts Trivia: Bias

- Weimer recommends that you take classes on philosophy until you've covered epistemology, free will, logic, the philosophy of science, and “what it is like to be a bat”. Take cognitive psychology classes until you've covered perception and the Flynn effect. Take speech or rhetoric classes until you've covered persuasion. Take anthropology as well as gender studies classes until you've covered Mead and Freeman and you have a better feel for which behaviors are socially constructed and which may be essential. Take classes in statistics until you can avoid being fooled. Take classes in religion or ethics until you've covered the relationship between unhappiness and unrealized desires. Take classes in physics until you can explain how a microphone, radio and speaker all work. Take classes on government until you have an opinion about the feasibility of legislating morality. Take classes on history until you are not condemned to repeat the mistakes of the past.