CS 314 Lecture 6

Lambda calculus

February 7, 2019

Lambda calculus

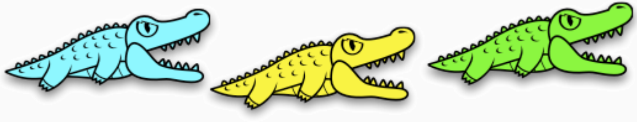
Alonzo Church

But first, let’s talk about alligators.

(Shamelessly stolen from

http://worrydream.com/AlligatorEggs/)

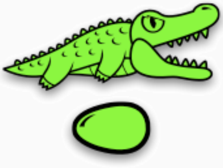
These are hungry alligators:



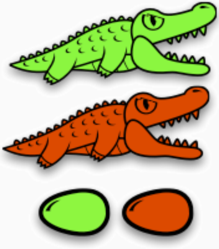
These are old alligators (who aren’t hungry anymore):

These are eggs:

Here an alligator is guarding her egg:

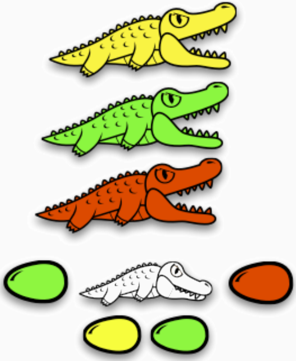


An alligator can also guard another alligator:

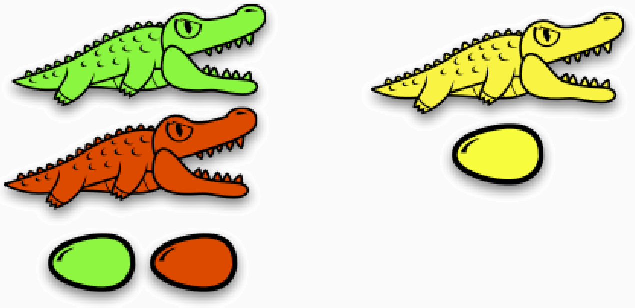


A more complex situation.

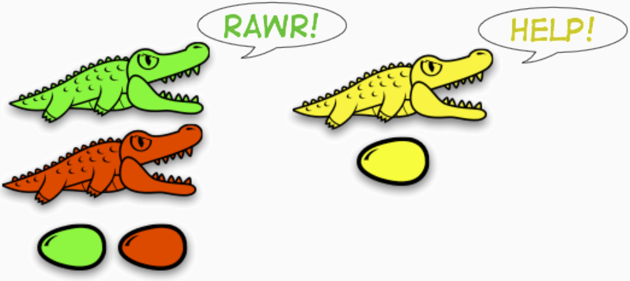
Note that the red alligator is guarding *three* things:



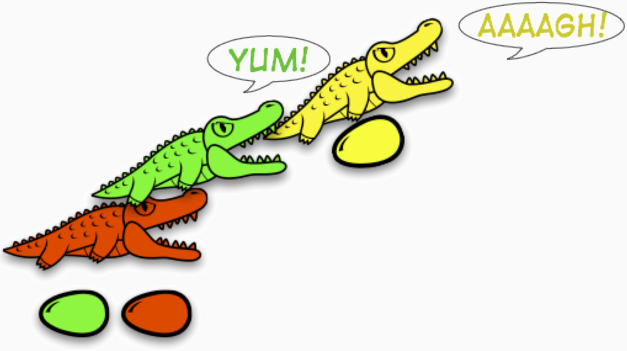
The green alligator is hungry, and a yellow alligator appears right in front of him!



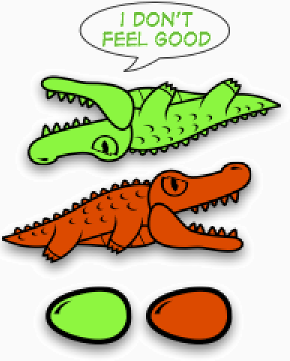
Uh oh...



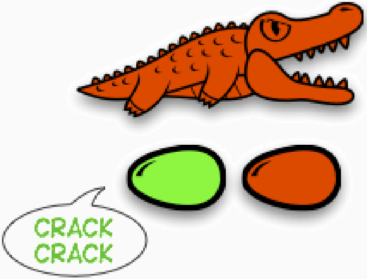
Oh dear...



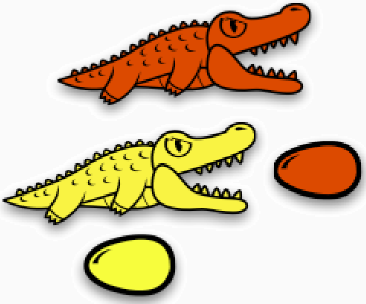
But someone ate too much.



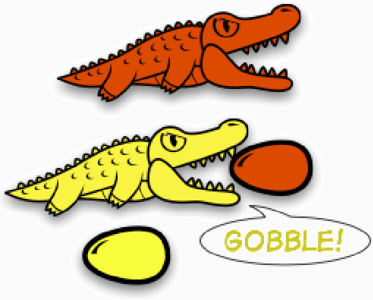
Now the green alligator's egg hatches!



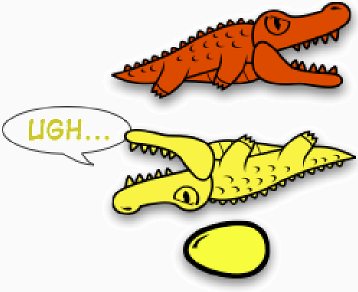
Miraculously, what emerges is just what the green alligator ate!



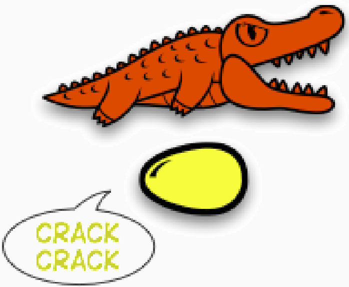
The yellow alligator is hungry too.



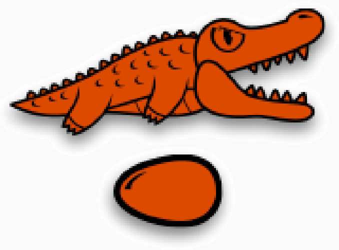
Food coma...



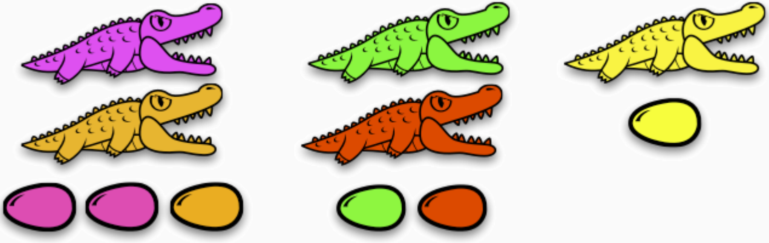
And now the yellow egg hatches!



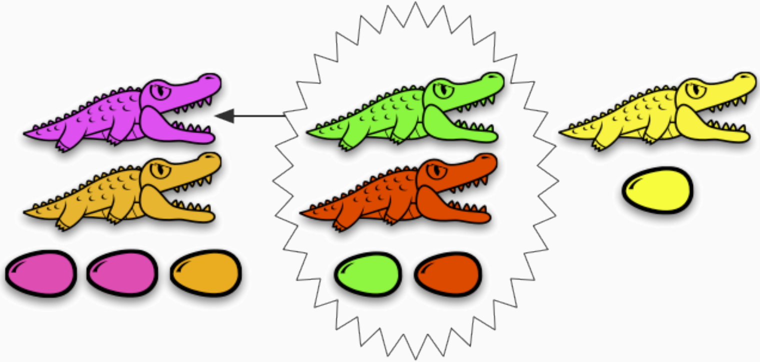
Producing the red egg the yellow alligator ate!



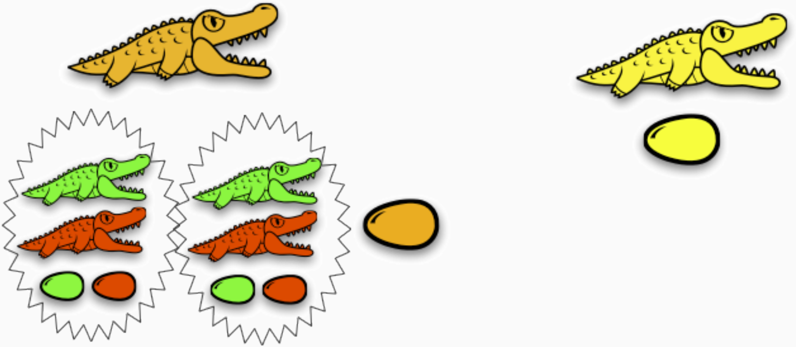
What if there are lots of alligator families?



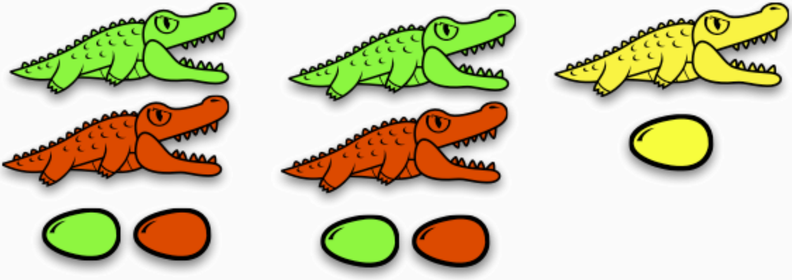
The one on the left eats first:



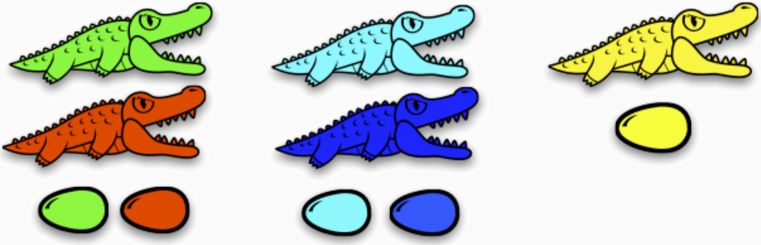
Both the purple eggs hatch to produce what the purple alligator ate.



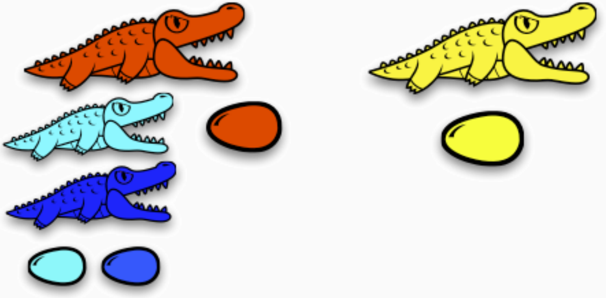
One rule: if two families have colors in common, they can't eat yet.



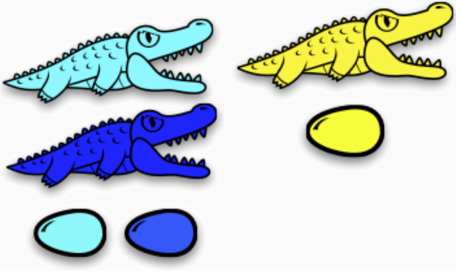
We have to change the colors first:



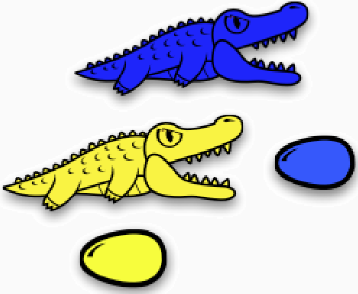
Then they can eat:



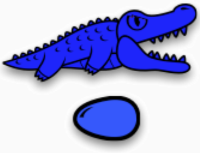
And eat:



And eat:



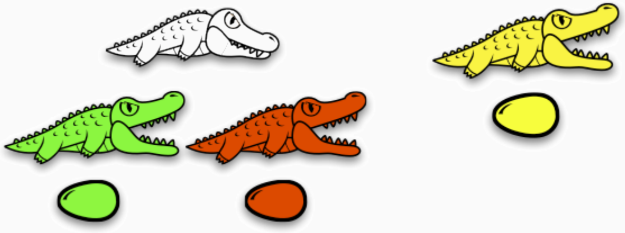
Until finally there’s no more eating to do.



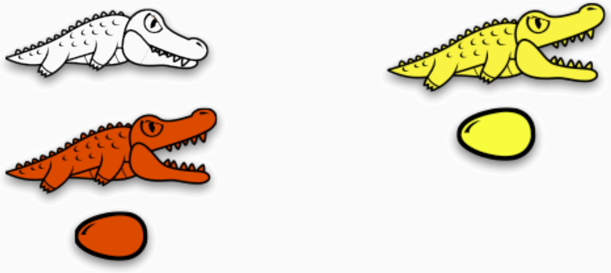
What happens with old alligators?

They're not hungry, so the yellow alligator is safe for now.

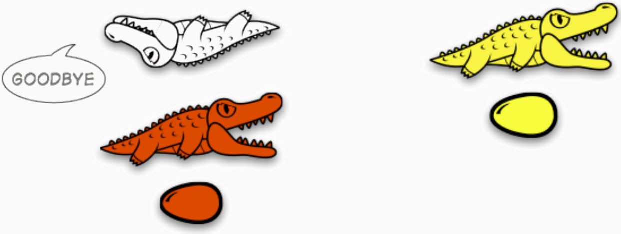
But the green alligator is hungry...



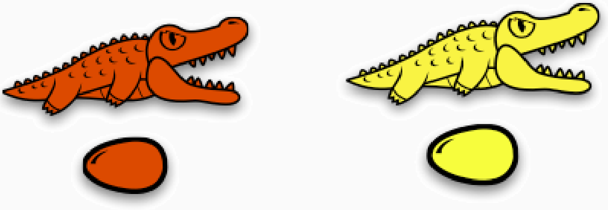
An old alligator lives only to protect her family.



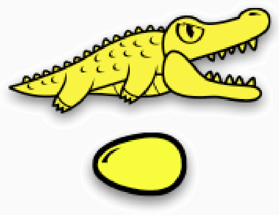
When she is only protecting one thing, she dies (the red alligator can protect herself):



Lunch time!



Finally, the feast is complete:



Lambda calculus, take 2

A mathematical model of computation, based on mathematical functions.

Consider the following:

- variables
- functions
- functions applied to some argument

Also known as:

- variables
- abstraction
- application

- variables
    - $x, y, z, \ldots$
- abstraction
    - $\lambda x.x$
- application
    - $(\lambda x.x)y$

Functions (abstractions)

We usually write

$$f(x) = x + 5$$

Lambda calculus functions aren't named, so it's more like

$$x \mapsto x + 5$$

Functions (abstractions)

$\lambda x.x$ is a function that “takes” x and “returns” x.

Function application

We apply functions by providing an argument:

$$(\lambda x.x)y$$

Evaluation

What does $(\lambda x.x)y$ mean?

Evaluation

Consider a (simplified) view of calling a function in some usual language:

```
int f(int x)
{
    int y = (x + 2) * (x + 3);
    return x * y;
}

f(5);
```

Evaluation

We can substitute the actual argument for the formal parameter *x*:

```
int f(5)
{
    int y = (5 + 2) * (5 + 3);
    return 5 * y;
}

f(5);
```

Evaluation

What does $(\lambda x.x)y$ mean?

- $\lambda x.x$ is a function with parameter x and body x.
- Replace every occurance of the parameter in the body with the actual argument (y).
- Replace every x in x with y.
- y

Evaluation

What does $(\lambda x.yxyzxz)w$ mean?

- $\lambda x.yxyzxz$ is a function with parameter x and body $yxyzxz$.
- Replace every occurance of the parameter in the body with the actual argument (w).
- Replace every x in $yxyzxz$ with w.
- $ywyzwz$

Evaluation

What does $(\lambda x.z)y$ mean?

- $\lambda x.z$ is a function with parameter x and body z.
- Replace every occurance of the parameter in the body with the actual argument (y).
- Replace every x in z with y.
- z

Function application

Note that function arguments can be other functions!

$$(\lambda x.x)(\lambda x.y)$$

Evaluation

What does $(\lambda x.x)(\lambda x.y)$ mean?

- $\lambda x.x$ is a function with parameter x and body x.
- Replace every occurance of the parameter in the body with the actual argument $(\lambda x.y)$.
- Replace every x in x with $\lambda x.y$.
- $\lambda x.y$

Evaluation

What does $(\lambda x.xz)(\lambda x.y)$ mean?

- $\lambda x.xz$ is a function with parameter x and body xz.
- Replace every occurance of the parameter in the body with the actual argument $(\lambda x.y)$.
- Replace every x in xz with $\lambda x.y$.
- $(\lambda x.y)z$
- Can apply again!
- y

Evaluation

Note: $(\lambda x.y)z$ is not the same as $\lambda x.yz$!

- $(\lambda x.y)z$ reduces to y
- $\lambda x.yz$ is a function with parameter x and body yz
- It has no argument yet, so it can't be reduced!

Functions

Note that function bodies can be other functions!

$$(\lambda x.(\lambda y.x))wv$$

Evaluation

What does $(\lambda x.(\lambda y.x))wv$ mean?

- $\lambda x.(\lambda y.x)$ is a function with parameter x and body $\lambda y.x$.
- Replace every occurance of the parameter in the body with the actual argument (w, not wv!).
- Replace every x in $\lambda y.x$ with w.
- $\lambda y.w$
- But we still have v, so we can apply this: $(\lambda y.w)v$
- w

Functions

And both function bodies and arguments can be other functions!

$$(\lambda x.(\lambda y.x))(\lambda z.w)$$

Evaluation

What does $(\lambda x.(\lambda y.x))(\lambda z.w)$ mean?

- $\lambda x.(\lambda y.x)$ is a function with parameter x and body $\lambda y.x$.
- Replace every occurance of the parameter in the body with the actual argument $(\lambda z.w)$.
- Replace every x in $\lambda y.x$ with $\lambda z.w$.
- $\lambda y.(\lambda z.w)$

Multi-argument functions

Note that functions that “return” functions are often abbreviated for convenience:

$$\begin{aligned}\lambda x.(\lambda y.z) &= \lambda x.\lambda y.z \\ &= \lambda xy.z\end{aligned}$$

Alligators again

- eggs = variables
- alligators = lambda abstractions (functions)
- alligators eating = function application
- old alligators = parentheses
- color changing = ?