Learn You a Physics for Great Good!

>>> WORK IN PROGRESS <<<

Calculus / Integral calculus

[src: Calculus/IntegralCalc.lhs] Previous: Differential calculus Table of contents Next: Plotting graphs

```
module Calculus.IntegralCalc (integrateApprox, integrate, eval) where

import Calculus.FunExpr
import Calculus.DifferentialCalc
```

Integrals - An integral part of calculus

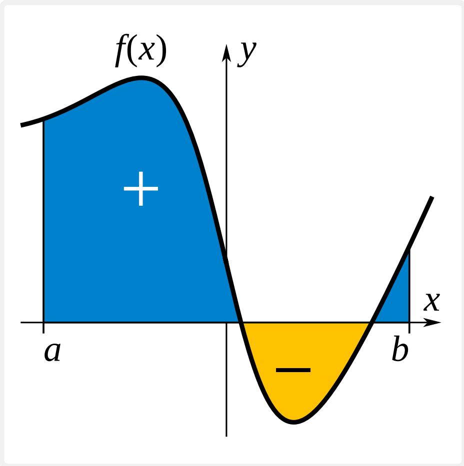
Integrals are functions used to describe area, volume, and accumulation in general. The operation of integration is the second fundamental operation of calculus, and the inverse of differentiation. Whereas derivatives are used to describe the rate of change in an instant, integrals are used to describe the accumulation of value over time.

Recall how we used derivatives before. If we know the distance traveled of a car and the time it took, we can use differentiation to calculate the velocity. Similarly but reversely, if we know the velocity of the car and the time it travels for, we can use integration to calculate the distance traveled.

$$x_{traveled} = \int_{t_0}^{t_1} v(t) dt$$

.

Ok, let's dive into this! We need to grok the syntax and find a rigorous, modelable definition of what *exactly* an integral is. We ask our kind friend Wikipedia for help. From the entry on *Integral*:

> Given a function f of a real variable x and an interval $[a, b]$ of the real line, the definite integral
>
> $$\int_a^b f(x)dx$$
>
> 
>
>
> A definite integral of a function can be represented as the signed area of the region bounded by its graph. (C) KSmrq
>
> is defined informally as the signed area of the region in the xy-plane that is bounded by the graph of f, the x-axis and the vertical lines $x = a$ and $x = b$. The area above the x-axis adds to the total and that below the x-axis subtracts from the total.
>
> Roughly speaking, the operation of integration is the inverse of differentiation. For this reason, the term integral may also refer to the related notion of the antiderivative, a function F whose derivative is the given function f. In this case, it is called an indefinite integral and is written:
>
> $$F(x) = \int f(x)dx$$

Ok, so first of all: confusion. Apparently there are two different kinds of integrals, *definite integrals* and *indefinite integrals*?

Let's start with defining *indefinite* integrals. *Wikipedia - Antiderivative* tells us that the *indefinite* integral, also known as the *antiderivative*, of a function f is equal to a differentiable function F such that $D(F) = f$. It further tells us that the process of finding the antiderivative is called *antidifferentiation* or *indefinite integration*.

The same article then brings further clarification

> Antiderivatives are related to definite integrals through the fundamental theorem of calculus: the definite integral of a function over an interval is equal to the difference between the values of an antiderivative evaluated at the endpoints of the interval.

So indefinite integrals are the inverse of derivatives, and definite integrals are just the application of an indefinite integral to an interval. If we look back at the syntax used, this makes sense. $\int f(x)dx$ is the indefinite integral. A function not applied to anything. $\int_a^b f(x)dx$ is the definite integral. The difference of the indefinite integral being applied to the endpoints of the interval $[a, b]$.

To simplify a bit, we see that just as with derivatives the x's everywhere are just there to confuse us, so we remove them.

$$\int f(x)dx$$

should really just be

$$\int f$$

.

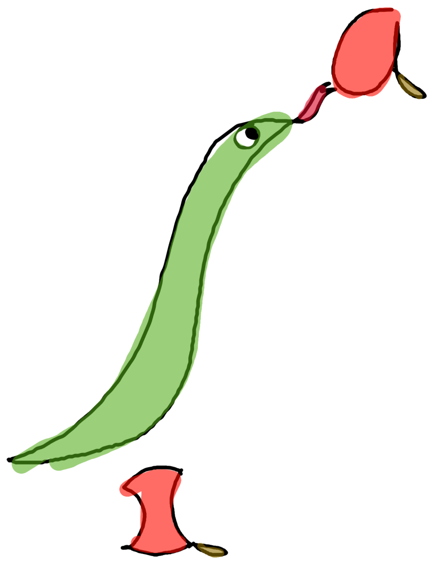
Next, the definition of definite integrals implies that we can write

$$\int_a^b f(x)dx$$

as

$$(\int f)[b] - (\int f)[a]$$

.



Only one of the two kinds of integral are fit to directly model in the syntax tree of our language. As `FunExpr` represents functions, it has to be the indefinite integral, which is a function unlike the definite integral which is a real value difference.

A thing to note is that while we may sometimes informally speak of the indefinite integral as a single unary function like any other, it's actually a set of functions, and the meaning of $F(x) = \int f(x)dx$ is really ambiguous. The reason for this is that for some function f, there is not just one function F such that $D(F) = f$. A simple counterexample is

$$D(x \mapsto x + 2) = 1 \text{ and } D(x \mapsto x + 3) = 1$$

The fact that adding a constant to a function does not change the the derivative, implies that the indefinite integral of a function is really a set of functions where the constant term differs.

$$\int f = \{F + const\ C \mid C \in \mathbb{R}\}$$

We don't want sets though. We want unary real functions (because that's our the type of our semantics!). So, we simply say that when integrating a function, the constant term will always be 0 in order to nail the result down to a single function! I.e. $(If)0 = 0$

```
| I FunExpr
```

Actually integrating with my man, Riemann

We've analyzed *what* an integral is, and we can tell if a function is the antiderivative of another. For example, x^2 is an antiderivative of $2x$ because $D(x^2) = 2x$. But *how* do we find integrals in the first place?

We start our journey with a familiar name, Leibniz. He, and also but independently Newton, discovered the heart of integrals and derivatives: The *fundamental theorem of calculus*. The definitions they made were all based on infinitesimals which, as said earlier, was considered too imprecise. Later, Riemann rigorously formalized integration using limits.

There exist many formal definitions of integrals, and they're not all equivalent. They each deal with different cases and classes of problems, and some remain in use mostly for pedagogical purposes. The most commonly used definitions are the Riemann integrals and the Lebesgue integrals.

The Riemann integral was the first rigorous definition of the integral, and for many practical applications it can be evaluated by the fundamental theorem of calculus or approximated by numerical integration. However, it is a deficient definition, and is therefore unsuitable for many theoretical purposes. For such purposes, the Lebesgue integral is a better fit.

All that considered, we will use Riemann integrals. While they may be lacking for many purposes, they are probably more familiar to most students (they are to me!), and will be sufficient for the level we're at.

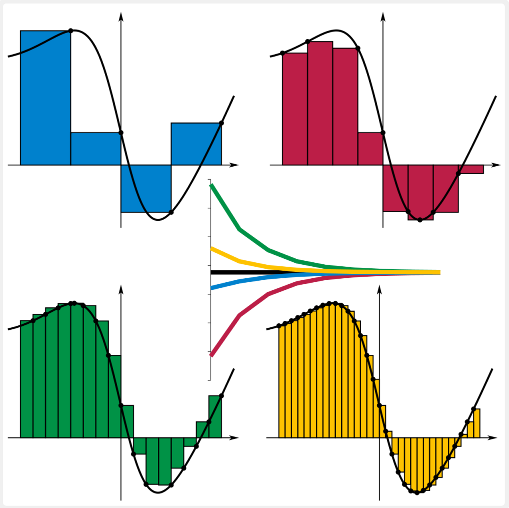
If we look back at the syntax of definite integrals

$$\int_a^b f(x) dx$$

the application of f and the dx part actually implies the definition of the Riemann integral. We can read it in english as "For every infinitesimal interval of x, starting at a and ending at b, take the value of f at that x (equiv. to taking the value at any point in the infinitesimal interval), and calculate the area of the rectangle with width dx and height $f(x)$, then sum all of these parts together.".

As we're dealing with an infinite sum of infinitesimal parts: a limit must be involved. a and b are the lower and upper limits of the sum. Our iteration variable should increase with infinitesimal dx each step. Each step we add the area of the rectangle with height $f(x')$, where x' is any point in $[x, x+dx]$. As $x+dx$ approaches x when dx approaches zero, $x' = lim_{dx \to 0} x + dx = x$.

$$\int_a^b f = \int_a^b f(x)dx = lim_{dx \to 0} \sum_{x=a,a+dx,a+2dx,\ldots}^{b} A(x, dx) \text{ where } A(x, dx) = f(x) * dx$$



Smaller dx result in better approximations. (C) KSmrq

Based on this definition, we could implement a function in Haskell to compute the numerical approximation of the integral by letting dx be a very small, but finite, number instead of being infinitesimal. The smaller our dx, the better the approximation

```
integrateApprox :: RealFunc -> RealNum -> RealNum ->
RealNum -> RealNum
integrateApprox f dx a b =
```

b must be greater than a for a definite integral to make sense, but if that's not the case, we can just flip the order of a and b and flip the sign of the area.

```
  let area x = f x * dx
  in if b >= a
     then sum (fmap area (takeWhile (<b) [a + 0*dx, a + 1*dx ..]))
     else -sum (fmap area (takeWhile (>b) [a - 0*dx, a - 1*dx ..]))
```

For example, let's calculate the area of the right-angled triangle under $y = x$ between $x = 0$ and $x = 10$. As the area of a right-angled triangle is calculated as $A = \frac{b*h}{2}$, we expect the result of to approach $A = \frac{b*h}{2} = \frac{10*10}{2} = 50$ as dx gets smaller

```
λ integrateApprox (\x -> x) 5   0 10
25
λ integrateApprox (\x -> x) 1   0 10
45
λ integrateApprox (\x -> x) 0.5 0 10
47.5
λ integrateApprox (\x -> x) 0.1 0 10
49.50000000000013
```

```
λ integrateApprox (\x -> x) 0.01 0 10
50.04999999999996
```

Great, it works for numeric approximations! This can be useful at times, but not so much in our case. We want closed expressions to use when solving physics problems, regardless of whether there are computations or not!

To find some integrals, making simple use of the fundamental theorem of calculus, i.e. $D(\int f) = f$, is enough. That is, we "think backwards". For example, we can use this method to find the integral of cos.

Which function derives to cos? Think, think, think ... I got it! It's sin, isn't it?

$$D(sin) = cos \implies \int cos = sin + constC$$

So simple! The same method can be used to find the integral of polynomials and some other simple functions. Coupled with some integration rules for products and exponents, this can get us quite far! But what if we're not superhumans and haven't memorized all the tables? What if we have to do integration without a cheat sheet for, like, an exam? In situations like these we make use of the definition of the Riemann integral, like we make use of the definition of differentiation in a previous chapter. As an example, let us again integrate cos, but now with this second method. Keep in mind that due to the technical limitations of Riemann integrals, not all integrals may be found this way.

Using the trigonometric identity of $\lim_{x \to 0} \frac{sinx}{x} = 1$ we find

$$\int_a^b cos$$

$$\begin{aligned}
&\quad \{ \text{ Riemann integral } \} \\
&= \lim_{dx \to 0} \sum_{x=a,a+dx,a+2*dx,\ldots}^{b} cos(x) * dx \\
&= \lim_{dx \to 0} dx * \sum_{x=a,a+dx,a+2*dx,\ldots}^{b} cos(x) \\
&= \lim_{dx \to 0} dx * (cos(a) + cos(a + dx) + cos(a + 2 * dx) + \ldots + cos(a + \frac{b-a}{dx} * dx)) \\
&\quad \{ \text{ Sums of cosines with arguments in arithmetic progression } \} \\
&= \lim_{dx \to 0} dx * \frac{sin(\frac{(\frac{b-a}{dx}+1)dx}{2}) * cos(a + \frac{\frac{b-a}{dx}dx}{2})}{sin(dx/2)} \\
&= \lim_{dx \to 0} dx * \frac{sin(\frac{b-a+dx}{2}) * cos(\frac{a+b}{2})}{sin(dx/2)} \\
&\quad \{ \text{ Trig. product-to-sum ident. } \} \\
&= \lim_{dx \to 0} dx * \frac{sin(\frac{b-a+dx}{2} + \frac{a+b}{2}) + sin(\frac{b-a+dx}{2} - \frac{a+b}{2})}{2sin(dx/2)} \\
&= \lim_{dx \to 0} dx * \frac{sin(b + dx/2) + sin(-a + dx/2)}{2sin(dx/2)} \\
&= \lim_{dx \to 0} \frac{sin(b + dx/2) + sin(-a + dx/2)}{\frac{sin(dx/2)}{dx/2}} \\
&\quad \{dx \to 0\} \\
&= \frac{sin(b + 0/2) + sin(-a + 0/2)}{1} \\
&= sin(b) + sin(-a) \\
&= sin(b) - sin(a)
\end{aligned}$$

The definition of definite integrals then give us that

$$\int_a^b cos = sin(b) - sin(a) \wedge \int_a^b f = F(b) - F(a) \implies F = sin$$

The antiderivative of cos is sin (again, as expected)!

Let's implement these rules as a function for symbolic (indefinite) integration of functions. We'll start with the nicer cases, and progress to the not so nice ones.

`integrate` takes a function to symbolically integrate, and returns the antiderivative where $F(0) = 0$.

Important to note is that not all functions are integrable. Unlike derivatives, some antiderivatives of elementary functions simply cannot be expressed as elementary functions themselves, according to [Liouville's theorem](). Some examples include e^{-x^2}, $\frac{sin(x)}{x}$, and x^x.

```
integrate :: FunExpr -> FunExpr
```

First, our elementary functions. You can prove them using the methods described above, but the easiest way to find them is to just look them up in some table of integrals (dust off that old calculus cheat sheet) or on WolframAlpha (or Wikipedia, or whatever. Up to you).

```
integrate Exp = Exp :- Const 1
```

Note that $log(x)$ is not defined in $x = 0$, so we don't have to add any corrective constant as $F(0)$ wouldn't make sense anyways.

```
integrate Log  = Id :* (Log :- Const 1)
integrate Sin  = Const 1 :- Cos
integrate Cos  = Sin
integrate Asin =    (Const 1 :- Id:^(Const 2)):^(Const 0.5)
                 :+ Id :* Asin
                 :- Const 0
integrate Acos =    Id :* Acos
                 :- (Const 1 :- Id:^(Const 2)):^(Const 0.5)
                 :+ Const 1
```

These two good boys. Very simple as well.

```
integrate Id        = Id:^Const 2 :/ Const 2
integrate (Const d) = Const d :* Id
```

Addition and subtraction is trivial. Just use the backwards method and compare to how sums and differences are differentiated.

```
integrate (f :+ g) = integrate f :+ integrate g
integrate (f :- g) = integrate f :- integrate g
```

Delta is easy. Just expand it to the difference that it is, and integrate.

```
integrate (Delta h f) = integrate (f :. (Id :+ Const h) :- f)
```

A derivative? That's trivial, the integration and differentiation cancel each other, right? Nope, not so simple! To conform to our specification of `integrate` that $F(0) = 0$, we have to make sure that the constant coefficient is equal to 0, which it might not be if we just cancel the operations. The simplest way to solve this is to evaluate the function at $x = 0$, and check the value. We then subtract a term that corrects the function such that $I(D(f))[0] = 0$. We'll write a simple function `center` for this later.

```
integrate (D f) = center f
```

Integrating an integral? Just integrate the integral!

```
integrate (I f) = integrate (integrate f)
```

Aaaaaand now it starts to get complicated.

There exists a great product rule in the case of differentiation, but not for integration. There just isn't any nice way to integrate a product that always works! The integration rule that's most analogous to the product rule for differentiation, is integration by parts:

$$\int f(x)g(x)dx = f(x)G(x) - \int f'(x)G(x)dx$$

Hmm, this doesn't look quite as helpful as the differentiation product rule, does it? We want this rule to give us an expression of simpler and/or fewer integrals, and it may indeed do so. For example, the integration of the product $x * e^x$ is a great examples of a case where it works well:

$$\int xe^x dx = xe^x - \int 1e^x dx = xe^x - e^x = e^x(x-1)$$

Now THAT is a simplification. However, just by flipping the order of the expressions, we get a case where the integration by parts rule only makes things worse:

$$\begin{aligned}
\int e^x x dx =& e^x \frac{x^2}{2} - \int e^x \frac{x^2}{2} dx \\
=& e^x \frac{x^2}{2} - (e^x \frac{x^3}{3!} - \int e^x \frac{x^3}{3} dx) \\
=& e^x \frac{x^2}{2} - (e^x \frac{x^3}{3!} - (e^x \frac{x^4}{4!} - \int e^x \frac{x^4}{4!} dx)) \\
=& \ldots
\end{aligned}$$

Oh no, it's an infinite recursion with successive increase in complexity! Sadly, there's no good way around it. By using heuristics, we could construct a complicated algorithm that guesses the best order of factors in a product when integrating, but that's *way* out of scope for this book.

Further, as a consequence of Liouville's theorem, the integration by parts rule is simply not defined in the case of $g(x)$ not being integrable to $G(x)$. And so, as there exists no definitely good way to do it in ALL cases, we're forced to settle for a conservative approach.

If we rewrite the formula for integration by parts to use g' instead of g

$$\int f(x)g'(x)dx = f(x)g(x) - \int f'(x)g(x)dx$$

we see that there are two cases where the integral is well defined:

$$\int f(x)g'(x)dx = f(x)g(x) - \int f'(x)g(x)dx$$

and

$$\int f'(x)g(x)dx = f(x)g(x) - \int f(x)g'(x)dx$$

I.e., if we already know the integral of one factor, we can integrate the product.

```
integrate (D f :* g) = center (f :* g :- integrate (f :* derive g))
integrate (f :* D g) = center (f :* g :- integrate (derive f :* g))
```

Also, we can add a few cases for integrals we know, like multiplication with a constant

```
integrate (Const c :* f) = Const c :* integrate f
```

The rule for quotients is very similar

```
integrate (D f :/ g) =
  center (f :/ g :+ integrate (f :* (D g :/ (g:^Const 2))))
```

There is no good rule for exponentials in general. Only for certain combinations of functions in the base and exponent is symbolic integration well defined. We'll only treat the special case of x^c, which at least implies that we can use polynomials.

```
integrate (Id :^ Const c) = Id:^(Const (c+1)) :/ Const (c+1)
```

Exercise. Find more rules of integrating exponentials and add to the implementation.

▼ **Solution**

Wikipedia has [a nice list of integrals of exponentials]

```
/details>
```

Integration of function composition is, simply said, somewhat complicated. The technique to use is called "integration by substitution", and is something like a reverse of the chain-rule of differentiation. This method is tricky to implement, as the way humans execute this method is highly dependent on intuition and a mind for patterns. A brute-force solution would be possible to implement, but is out of scope for this book, and not really relevant to what we want to learn here. We'll leave symbolic integration of composition undefined.

And if we couldn't integrate the expression as is, first try simplifying it and see if we know how to integrate the new expression. If that fails, just wrap the expression in the `I` constructor, unchanged. This will signify that we don't know how to symbolically integrate the expression. During evaluation, we can use `integrateApprox` to compute the integral numerically.

```
integrate f = let fsim = simplify f
              in if f == fsim
                 then I f
                 else integrate fsim
```

Finally, the helper function `center` to center functions such that $f(0) = 0$.

```
center f = f :- Const ((eval f) 0)
```

The value of evaluation

What comes after construction of function expressions? Well, using them of course!

One way of using a function expression is to evaluate it, and use it just as you would a normal Haskell function. To do this, we need to write an evaluator.

An evaluator simply takes a syntactic representation and returns the semantic value, i.e. `eval :: SYNTAX -> SEMANTICS`.

In the case of our calculus language:

```
eval :: FunExpr -> (RealFunc)
```

To then evaluate a `FunExpr` is not very complicated. The elementary functions and the `Id` function are simply substituted for their Haskell counterparts.

```
eval Exp = exp
eval Log = log
eval Sin = sin
eval Cos = cos
eval Asin = asin
eval Acos = acos
eval Id = id
```

`Const` is evaluated according to the definition $const(c) = x \mapsto c$

```
eval (Const c) = \x -> c
```

How to evaluate arithmetic operations on functions may not be as obvious, but we just implement them as they were defined earlier in the chapter.

```
eval (f :+ g) = \x -> (eval f x + eval g x)
eval (f :- g) = \x -> (eval f x - eval g x)
eval (f :* g) = \x -> (eval f x * eval g x)
eval (f :/ g) = \x -> (eval f x / eval g x)
eval (f :^ g) = \x -> (eval f x ** eval g x)
```

Function composition is similarly evaluated according to the earlier definition

```
eval (f :. g) = \x -> eval f (eval g x)
```

Delta is just expanded to the difference that it really is

```
eval (Delta h f) = eval (f :. (Id :+ Const h) :- f)
```

For derivatives, we just apply the symbolic operation we wrote, and then evaluate the result.

```
eval (D f) = eval (derive f)
```

With integrals, we first symbolically integrate the expression as far as possible, then apply the numerical `integrateApprox` if we can't figure out the integral further.

```haskell
eval (I f) = let _F = simplify (integrate f)
             in if _F == (I f)
                then integrateApprox (eval f) 0.001 0
                else eval _F
```

[src: Calculus/IntegralCalc.lhs]   Previous: Differential calculus   Table of contents   Next: Plotting graphs