# Making an operation commutative is easy

Suppose we have a binary operation `g` and a strict total ordering `less` (e.g. lexicographical ordering of bit representations).

Then this operation is commutative:

```
def f(x: A, y: A) = if (less(y,x)) g(y,x) else g(x,y)
```

Indeed `f(x,y)==f(y,x)` because:

- if `x==y` then both sides equal `g(x,x)`
- if `less(y,x)` then left sides is `g(y,x)` and it is not `less(x,y)` so right side is also `g(y,x)`
- if `less(x,y)` then it is not `less(y,x)` so left sides is `g(x,y)` and right side is also `g(x,y)`

We know of no such efficient trick for associativity

# Associative operations on tuples

Suppose `f1: (A1,A1) => A1` and `f2: (A2,A2) => A2` are associative

Then `f: ((A1,A2), (A1,A2)) => (A1,A2)` defined by

```
f((x1,x2), (y1,y2)) = (f1(x1,y1), f2(x2,y2))
```

is also associative:

```
f(f((x1,x2), (y1,y2)), (z1,z2)) ==
f((f1(x1,y1), f2(x2,y2)), (z1,z2)) ==
(f1(f1(x1,y1), z1), f2(f2(x2,y2), z2)) == (because f1, f2 are associative)
(f1(x1, f1(y1,z1)), f2(x2, f2(y2,z2))) ==
f((x1 x2), (f1(y1,z1), f2(y2,z2))) ==
f((x1 x2), f((y1,y2), (z1, z2)))
```

We can similarly construct associative operations on for n-tuples

# Example: rational multiplication

Suppose we use 32-bit numbers to represent numerator and denominator of a rational number.

We can define multiplication working on pairs of numerator and denominator

```
times((x1,y1), (x2, y2)) = (x1*x2, y1*y2)
```

Because multiplication modulo $2^{32}$ is associative, so is `times`

# Example: average

Given a collection of integers, compute the average

```
val sum = reduce(collection, _ + _)
val length = reduce(map(collection, (x:Int) => 1), _ + _)
sum/length
```

This includes two reductions. Is there a solution using a single reduce?

# Example: average

Use pairs that compute sum and length at once

```
f((sum1,len1), (sum2, len2)) = (sum1 + sum1, len1 + len2)
```

Function `f` is associative because addition is associative.

Solution is then:

```
val (sum, length) = reduce(map(collection, (x:Int) => (x,1)), f)
sum/length
```

# Associativity through symmetry and commutativity

Although commutativity of `f` alone does not imply associativity, it implies it if we have an additional property. Define:

```
E(x,y,z) = f(f(x,y), z)
```

We say arguments of `E` can rotate if `E(x,y,z) = E(y,z,x)`, that is:

```
f(f(x,y), z) = f(f(y,z), x)
```

Claim: if `f` is commutative and arguments of `E` can rotate then `f` is also associative.

Proof:

```
f(f(x,y), z) = f(f(y,z), x) = f(x, f(y,z))
```

# Example: addition of modular fractions

Define

```
plus((x1,y1), (x2, y2)) = (x1*y2 + x2*y1, y1*y2)
```

where $*$ and $+$ are all modulo some base (e.g. $2^{32}$).

We can have overflows in both numerator and denominator

Is such `plus` associative?

# Example: addition of modular fractions

```
plus((x1,y1), (x2, y2)) = (x1*y2 + x2*y1, y1*y2)
```

Observe: `plus` is commutative. Moreover:

```
E((x1,y1), (x2,y2), (x3,y3)) ==
plus(plus((x1,y1), (x2,y2)), (x3,y3)) ==
plus((x1*y2 + x2*y1, y1*y2), (x3,y3)) ==
((x1*y2 + x2*y1)*y3 + x3*y1*y2, y1*y2*y3) ==
(x1*y2*y3 + x2*y1*y3 + x3*y1*y2, y1*y2*y3)
```

Therefore

```
E((x2,y2), (x3,y3), (x1,y1)) ==
(x2*y3*y1 + x3*y2*y1 + x1*y2*y3, y2*y3*y1)
```

which is the same. By previous claim, `plus` is associative.

# Example: relativistic velocity addition

Let $u, v$ range over rational numbers in the open interval $(-1, 1)$

Define $f$ to add velicities according to special relativity

$$f(u, v) = \frac{u + v}{1 + uv}$$

Clearly, $f$ is commutative: $f(u, v) = f(v, u)$.

$$f(f(u, v), w) = \frac{\frac{u+v}{1+uv} + w}{1 + \frac{u+v}{1+uv}w} = \frac{u + v + w + uvw}{1 + uv + uw + vw}$$

We can rotate arguments $u, v, w$

# Example: relativistic velocity addition

Let $u, v$ range over rational numbers in the open interval $(-1, 1)$

Define $f$ to add velicities according to special relativity

$$f(u, v) = \frac{u + v}{1 + uv}$$

Clearly, $f$ is commutative: $f(u, v) = f(v, u)$.

$$f(f(u, v), w) = \frac{\frac{u+v}{1+uv} + w}{1 + \frac{u+v}{1+uv}w} = \frac{u + v + w + uvw}{1 + uv + uw + vw}$$

We can rotate arguments $u, v, w$

$f$ is commutative and we can rotate, so $f$ is associative.

# Consequences of non-associativity on floating points

If we implement $f$ given by expression

$$f(u, v) = \frac{u + v}{1 + uv}$$

using floating point numbers, then the operation is not associative.

Even though the difference between $f(x, f(y, z))$ and $f(f(x, y), z)$ is small in one step, over many steps it accumulates, so the result of the `reduceLeft` and a `reduce` may differ substantially.

# A family of associative operations on sets

Define binary operation on sets $A, B$ by $f(A, B) = (A \cup B)^*$ where $*$ is any operator on sets (closure) with these properties:

- $A \subseteq A^*$ (expansion)
- if $A \subseteq B$ then $A^* \subseteq B^*$ (monotonicity)
- $(A^*)^* = A^*$ (idempotence)

Example of $*$: convex hull, Kleene star in regular expressions

Claim: every such $f$ is associative.

Proof: $f$ is commutative. It remains to show

$$f(f(A, B), C) = ((A \cup B)^* \cup C)^* = (A \cup B \cup C)^*$$

because from there it is easy to see that the arguments rotate.

# First subset inclusion

We need to prove: $((A \cup B)^* \cup C)^* \subseteq (A \cup B \cup C)^*$.

Since $A \cup B \subseteq A \cup B \cup C$, by monotonicity:

$$(A \cup B)^* \subseteq (A \cup B \cup C)^*$$

Similarly

$$C \subseteq A \cup B \cup C \subseteq (A \cup B \cup C)^*$$

Thus $(A \cup B)^* \cup C \subseteq (A \cup B \cup C)^*$. By monotonicity and idempotence

$$((A \cup B)^* \cup C)^* \subseteq ((A \cup B \cup C)^*)^* = (A \cup B \cup C)^*$$

# Second subset inclusion

We need to prove: $(A \cup B \cup C)^* \subseteq ((A \cup B)^* \cup C)^*$

From expansion we have $A \cup B \subseteq (A \cup B)^*$. Thus

$$A \cup B \cup C \subseteq (A \cup B)^* \cup C$$

The property then follows by monotonicity.

# End of Slide Deck