On the Commutability of Subtraction

by Mike Harris

subtraction does not commute, the talk is really about testing

Commutative

An operator $\oplus$ on a set S is commutative iff

$x \oplus y = y \oplus x$

for all $x, y \in S$

```
let assert = require('assert');

describe('Subtraction', () => {
  describe('commutes', () => {
    it('0 - 0 === 0 - 0', () => {
      assert.deepEqual(0 - 0, 0 - 0);
    });
  });
});
```

JS

Subtraction
commutes
✓ 0 - 0 === 0 - 0

1 passing (2ms)

JS

```
Example test_subtraction_commutes :
0 - 0 = 0 - 0.
Proof.
  simpl.
  reflexivity.
Qed.
```

QED

```
Example test_subtraction_commutes' :
1 - 0 = 0 - 1.
Proof.
```

```
Goals
1 - 0 = 0 - 1
```

```
Example test_subtraction_commutes' :
1 - 0 = 0 - 1.
Proof.
  simpl.
```

```
Goals
1 = 0
```

```
Example test_subtraction_commutes' :
1 - 0 = 0 - 1.
Proof.
  simpl.
Abort.
```

```
Goals
1 = 0
```

Commutative

An operator $\oplus$ on a set S is commutative iff

$x \oplus y = y \oplus x$

for all $x, y \in S$

```
Lemma subtraction_does_not_commute :
forall a b : nat, a <> b -> a - b <> b - a.
Proof.
```

```
Goals
forall a b : nat, a <> b -> a - b <> b - a
```

```
Lemma subtraction_does_not_commute :
forall a b : nat, a <> b -> a - b <> b - a.
Proof.
induction a. intros b.
```

```
Sub Goal
0 <> b -> 0 - b <> b - 0
```

```
Lemma subtraction_does_not_commute :
forall a b : nat, a <> b -> a - b <> b - a.
Proof.
induction a. intros b.
- now rewrite Nat.sub_0_r.
```

```
Sub Goal
forall b : nat,
  S a <> b -> S a - b <> b - S a
```

```
Lemma subtraction_does_not_commute :
forall a b : nat, a <> b -> a - b <> b - a.
Proof.
induction a. intros b.
- now rewrite Nat.sub_0_r.
- destruct b.
```

```
Sub Goal
S a <> 0 -> S a - 0 <> 0 - S a
```

```
Lemma subtraction_does_not_commute :
forall a b : nat, a <> b -> a - b <> b - a.
Proof.
induction a. intros b.
- now rewrite Nat.sub_0_r.
- destruct b.
  + trivial.
```

```
Sub Goal
S a <> S b -> S a - S b <> S b - S a
```

```
Lemma subtraction_does_not_commute :
forall a b : nat, a <> b -> a - b <> b - a.
Proof.
induction a. intros b.
- now rewrite Nat.sub_0_r.
- destruct b.
  + trivial.
  + repeat rewrite Nat.sub_succ; auto.
```

Sub Goal
done

```
Lemma subtraction_does_not_commute :
forall a b : nat, a <> b -> a - b <> b - a.
Proof.
induction a. intros b.
- now rewrite Nat.sub_0_r.
- destruct b.
  + trivial.
  + repeat rewrite Nat.sub_succ; auto.
Qed.
```

Goals
done

QED

```
Lemma subtraction_does_not_commute' :
forall a b : nat, a <> b -> a - b <> b - a.
Proof.
intros; omega.
Qed.
```

Goals
done

QED

npm install --save jsverify

```
let jsc = require('jsverify');

describe('Subtraction', () => {
  describe('commutes', () => {
    jsc.property(
      'forall a, b: int, a - b = b - a',
      jsc.integer, jsc.integer,
      (a, b) => a - b === b - a
    );
  });
});
```

JS

Subtraction
  commutes
    1) forall a, b: int, a - b = b - a

0 passing (2ms)
1 failing

1) Subtraction commutes forall a, b: int, a - b = b - a:
    Error: Failed after 1 tests and 3 shrinks. rngState: 8dd4cf91b02e8a5376; Counterexample: 0; 1;

JS

```
let jsc = require('jsverify');

describe('Subtraction', () => {
  describe('does not commute', () => {
    jsc.property(
      'forall a, b: int, a <> b -> a - b <> b - a',
      jsc.integer, jsc.integer,
      (a, b) =>
        (a !== b) ? a - b !== b - a : true
    );
  });
});
```

JS

Subtraction
  does not commute
    ✓ forall a, b: int, a <> b -> a - b <> b - a

1 passing (2ms)

JS

QED

Thank you!

Mike Harris
@MikeMKH

http://bit.ly/2Acgflt

Next Steps

- JSVerify
  - Try jsverify, http://jsverify.github.io/
  - Read Functional Programming in JavaScript, https://www.manning.com/books/functional-programming-in-javascript
- Coq
  - Read Coq in a Hurry, https://cel.archives-ouvertes.fr/inria-00001173v6/document
  - Read My Unusual Hobby, https://www.stephanboyer.com/post/134/my-unusual-hobby
  - Work through Software Foundations, https://softwarefoundations.cis.upenn.edu/