# 15-150 Assignment 2

Karan Sikka
ksikka@andrew.cmu.edu
Section G but im not sure
January 25, 2012

---

### 1: Task 2.1

No, this does not type check. When smlnj tries to evaluate the expression `square 7.0`, it fails to do so, because 7.0 is not an int but `square` is a function of type `int -> int`.

---

### 2: Task 2.2

a. The value 3 : int is put in the place of the x in line 5. This is because of the val declaration in line 4 takes precedence over the x defined in the arguments of the function, due to the scope properties of the let clause.

b. The value of the expression `5.2 * (real x)` is substituted for p, where x is 3, as said in part a. This evaluates to `5.2 * (real 3)`, then `5.2 * 3.0`, then `15.6`. So ultimately the value which replaces p is 15.6. This is because p is within the `in` block so it takes the value prescribed to it in the `let` block, specifically line 5. Also, SML is pass-by-value, so the value is passed to the `in` block, not the expression.

c. The value of the first argument passed to the fucntion replaces x, because the declaration of the function names the first argument x. The x in line 4 or 7 do not substitute for the x in line 11, because those are within the scope of the inner let statement, and they disappear after the end on line 9.

d. assemble (x, 2.0) evaluates to 58.

---

### 3: Task 2.3

```
let val x : real = real (square 6)
in 3 + (trunc x)
end
```

$\mapsto$

```
let val x : real = real (6 * 6)
in 3 + (trunc x)
end
```

$\mapsto$

```
let val x : real = real (36)
in 3 + (trunc x)
end
```

↦

```
let val x : real = 36.0
in 3 + (trunc x)
end
```

↦ `3 + (trunc 36.0)`

↦ `3 + 36`

↦ `39`

---

## 4: Task 2.4

They are equivalent. The fact function steps to a case statement, which eventually steps to `~1*fact(~2)`. This repeats for quite a long time, and assuming integers do not overflow, and that our heap is infinitely large, this function will infinite loop.

In the case of f 10, this function steps to f 10. It steps to the same thing consistently, using no additional memory. This will also infinite loop.

Therefore, the expressions are equivalent.

---

## 5: Task 2.5

Consider the expression `fact 3`.

```
fact(3)
↦ case 3 of 0 => 1 | _ => 3 * fact(3-1)
↦ 3 * fact(3-1)
↦ 3 * fact(2)
↦ 3 * case 2 of 0 => 1 | _ 2 * fact(2-1)
↦ 3 * 2 * fact(2-1)
↦ 3 * 2 * fact(1)
↦ 6 * fact(1)
↦ 6 * case 1 of 0 => 1 | _ 1 * fact(1-1)
↦ 6 * 1 * fact(1-1)
↦ 6 * 1 * fact(0)
↦ 6 * fact(0)
↦ 6 * case 0 of 0 => 1 | _ 0 * fact(0-1)
↦ 6 * 1
↦ 6
```

Thus, fact(3) evaluates to 6.

Now consider:

| | | |
|---|---|---|
| | `fact(fact(3))` | |
| $\cong$ | `fact(6)` | equivalence proven above |
| $\cong$ | `case 6 of 0 => 1 \| _ 6 * fact(6-1)` | step |
| $\cong$ | `6 * fact(6-1)` | step |
| $\cong$ | `fact(3) * fact(fact(3) - 1)` | step, equiv. is symmetric, fact(3)=6 as shown above. |

Therefore:
`fact(fact(3))` $\cong$ `fact(3) * fact(fact(3) - 1)`

---

## 6: Task 3.2

$H_0$ is 0. $H_1$ is 1. $H_2$ is $\frac{3}{2}$. $H_3$ is $\frac{11}{6}$. $H_4$ is $\frac{25}{12}$.

---

## 7: Task 3.4

$I_0$ is 0. $I_1$ is 1. $I_2$ is $\frac{1}{2}$. $I_3$ is $\frac{5}{6}$. $I_4$ is $\frac{7}{12}$.

---

## 8: Task 4.1

Claim: For all natural numbers n, `2(n+1)` $\cong$ `double(n+1)` .
We proceed by induction on $n$. Base Case:

| | | |
|---|---|---|
| | `double n` where n = 0 | |
| $\cong$ | `double 0` | substituting |
| $\cong$ | `case 0 of 0 => 0 \| _ 2 + double(n-1)` | step |
| $\cong$ | `0` | step |
| $\cong$ | `2*0` | math, also step is symmetric |
| $\cong$ | `2n where n = 0` | step |

Inductive Hypothesis: Assume `double k` $\cong 2 * k$ for some $k \in \mathbb{N}$
Inductive Step: Consider:

| | | |
|---|---|---|
| | `2(k+1)` | |
| $\cong$ | `2k + 2` | math, distributive property |
| $\cong$ | `2 + double(k)` | IH |
| $\cong$ | `2 + double(k + 1 - 1)` | math, step, equiv. is sym. |
| $\cong$ | `case 0 of 0 => 0 \| 2 + double(k + 1 - 1)` | step, equiv. is symmetric |
| $\cong$ | `double(k+1)` | step, equiv. is symmetric |

Therefore `2(k+1)` $\cong$ `double(k+1)`. By induction, the claim holds.

---

## 9: Task 4.2

Claim: For all natural numbers $n$, `summ n` $\cong$ `(n*(n+1)) div 2` .
We proceed by induction on n.

Base Case:

| | | |
|---|---|---|
| | `summm n` where n = 0 | |
| $\cong$ | `summ 0` | substitution |
| $\cong$ | `case 0 of 0 => 0 \| _ 0 + (summ(0-1))` | step |
| $\cong$ | `0` | step |
| $\cong$ | `0*1` | math |
| $\cong$ | `0*(0+1)` | math |
| $\cong$ | `0*(0+1) div 2` | math |
| $\cong$ | `n*(n+1) div 2 where n = 0` | substitution |

Inductive Hypothesis:
Assume `summ n` $\cong$ `n*(n+1) div 2` for some $n \in \mathbb{N}$.

Inductive Step:
First, we make the assumption that n+1 evaluates to a value that we will call n+1. Consider:

| | | |
|---|---|---|
| $\cong$ | `summ(n+1)` | |
| $\cong$ | `case n+1 of 0 => 0 \| _ n+1 + summ((n+1)-1)` | step |
| $\cong$ | `n+1 + summ((n+1)-1)` | step |
| $\cong$ | `n+1 + summ(n)` | math, addition is associative |
| $\cong$ | `n+1 + n*(n+1) div 2` | IH |
| $\cong$ | `((n+1)(n+2)) div 2` | algebra shown below |

The algebra:

$$\frac{n^2+n}{2} + \frac{2(n+1)}{2}$$

Distribute the right fraction:

$$\frac{n^2+n}{2} + \frac{2n+2}{2}$$

Add the fractions:

$$\frac{n^2+3n+2}{2}$$

Factor:

$$\frac{(n+1)(n+2)}{2}$$

By induction, the claim holds.