

Chapter 2

[Qn. 8a]

split-add(3) = g where $g(y) = \text{add}(3,y)$
i.e. $\text{split-add}(3) : \mathbb{N} \rightarrow \mathbb{N}$
thus, $\text{graph}(\text{split-add}(3)) = \{(0,3), (1,4), (2,5), (3,6), \dots\}$

[Qn. 8b]

$\text{split-add}(2)(1) = g(1) = \text{add}(2,1) = 3$
i.e. $\text{split-add}(\text{split-add}(2)(1)) = \text{split-add}(3)$
thus, $\text{graph}(\text{split-add}(\text{split-add}(2)(1))) = \text{graph}(\text{split-add}(3))$
 $= \{(0,3), (1,4), (2,5), (3,6), \dots\}$

[Qn. 9a]

For $\text{split-sub} : \mathbb{Z} \rightarrow \mathbb{Z} \rightarrow \mathbb{Z}$
 $\text{graph}(\text{split-sub}) = \{ \dots, (-2, \{\dots, (-1,-1), (0,-2), (1,-3), \dots\}),$
 $(-1, \{\dots, (-1, 0), (0,-1), (1,-2), \dots\}),$
 $(0, \{\dots, (-1, 1), (0, 0), (1,-1), \dots\}),$
 $(1, \{\dots, (-1, 2), (0, 1), (1, 0), \dots\}),$
 $(2, \{\dots, (-1, 3), (0, 2), (1, 1), \dots\}), \dots \}$

[Qn. 9b]

For $\text{split-sub} : \mathbb{N} \rightarrow \mathbb{N} \rightarrow \mathbb{Z}$
 $\text{graph}(\text{split-sub}) = \{ (0, \{(0,0), (1,-1), (2,-2), \dots\}),$
 $(1, \{(0,1), (1, 0), (2,-1), \dots\}),$
 $(2, \{(0,2), (1, 1), (2, 0), \dots\}), \dots \}$

[Qn. 10]

Let $B = \{T,F\}$
So, $B \times B : (T,T), (T,F), (F,T), (F,F)$
and $(B \times B) \rightarrow B :$
 $\{((T,T),T), ((T,F),T), ((F,T),T), ((F,F),F)\}$
is just one of $(2^4 = 16)$ possible function sets
(and in this case it's the OR function)

$B \rightarrow B : \{(T,F), (F,T)\}$ is just one of 4 possible function sets
(this is the NOT function... the 4 possible functions are
 $f(x) = T, f(x) = F, f(x) = x$ and $f(x) = \text{NOT } x$)

$B \rightarrow (B \rightarrow B) :$
 $\{(T, \{(T,T), (F,F)\}), (F, \{(T,F), (F,T)\})\}$
is just one of $(4^2 = 16)$ possible function sets

Finally, for $\text{curryB} : ((B \times B) \rightarrow B) \rightarrow (B \rightarrow (B \rightarrow B))$

$\text{graph}(\text{curryB}) = \{ \dots \dots ,$
 $(\{((T,T),T), ((T,F),T), ((F,T),T), ((F,F),F)\} ,$
 $\{(T, \{(T,T), (F,T)\}), (F, \{(T,T), (F,F)\})\}) ,$
 $(\{((T,T),T), ((T,F),F), ((F,T),F), ((F,F),F)\} ,$
 $\{(T, \{(T,T), (F,F)\}), (F, \{(T,F), (F,F)\})\}) ,$

... ... }

are just 2 of the total 16 elements.

The set $((B \times B) \rightarrow B) \rightarrow (B \rightarrow (B \rightarrow B))$ contains 16^{16} elements.

[Qn. 11]

- a. $B \rightarrow N$
- b. $(B \times N) \rightarrow (B \times B)$
- c. $(B \rightarrow B) \rightarrow B$

=====

Chapter 3

[Qn. 4]

- b. $\{(zero, ()), (one, (tt, tt)), (one, (tt, ff)), (one, (ff, tt)), (one, (ff, ff)), _|_ \}$
- e. $\{ \{((), tt)\}, \{((), ff)\}, \{((), _|_)\} \}$
- f. $\{ \{((), tt)\}, \{((), ff)\}, _|_ \}$

[Qn. 7]

```
Domain Set-of-A = A -> Tr
empty-set = lambda_n.false
make-singleton = lambda_k.lambda_n.(n=k) => true [] false
member-of = lambda_k.lambda_S.S(k)
union = lambda_S1.lambda_S2.lambda_n.S1(n) => true [] S2(n)
```

[Qn. 8]

Note: absent_error and acc_error are special elements in A.

```
Domain Array = (Nat -> A) x Nat x Nat
newarray = lambda_x.lambda_y.(lambda_n.absent_error, x, y)
  (assume x is the lower bound and y is the upper bound,
   and r.2 means r"down-arrow"2 which is the disassembler of product)
access = lambda_n.lambda_r.(n < r.2) => acc_error []
  (n > r.3) => acc_error [] r.1(n)
update = lambda_n.lambda_v.lambda_r.(n < r.2) => r.1 []
  (n > r.3) => r.1 []
  ([n |-> v]r.1, r.2, r.3)
```