

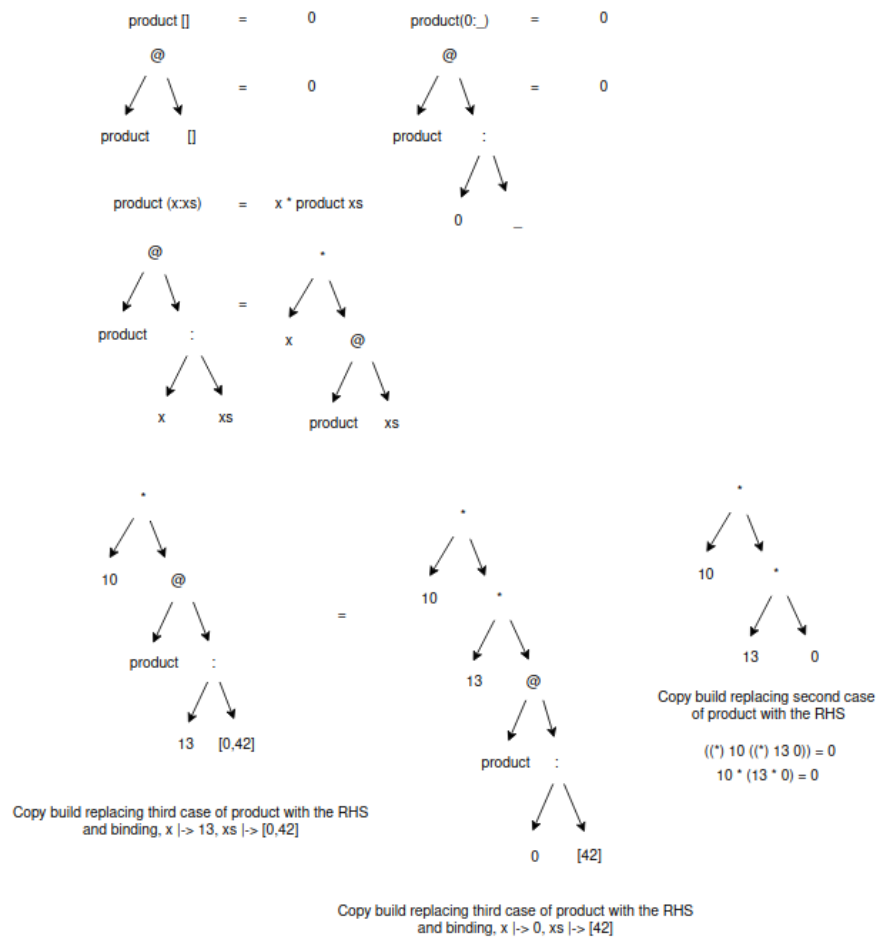
AST Exam Solutions

1 2018 Q3

1.1 Part (a)

```
product [] = 1
product (0:_) = 0
product (x:xs) = x * product xs
product [10,13,0,42]
```

```
product (10:[13,0,42]) = 10 * product [13,0,42] -- third case
10 * product (13:[0,42]) = 10 * (13 * product [0,42]) -- third case
10 * (13 * product (0:[42])) = 10 * (13 * 0) -- second case
```



2 2018 Q3

2.1 Part (a)

`length [] = 0`

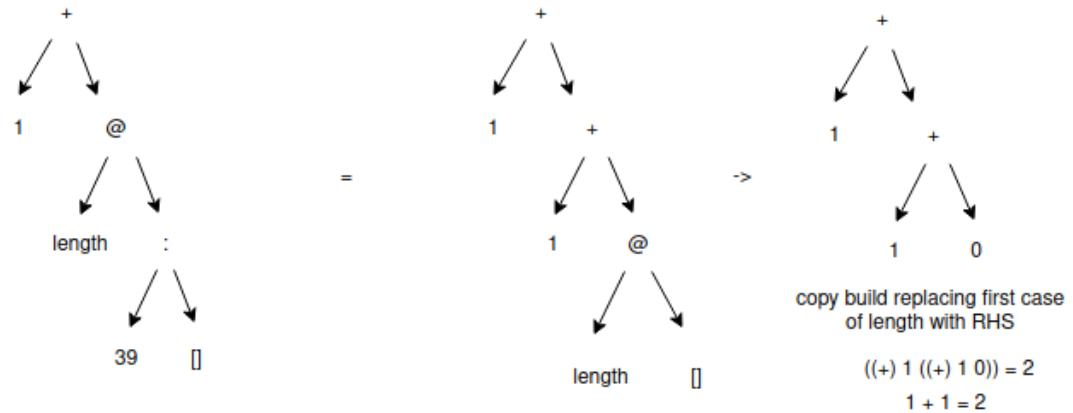
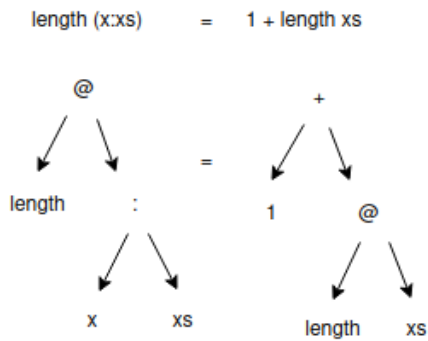
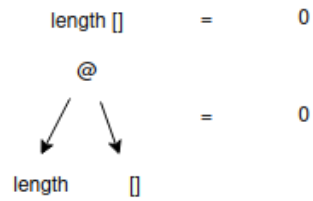
`length (x:xs) = 1 + length xs`

`length [3,39]`

`length (3:[39]) = 1 + length [39] -- second case`

`1 + length ([39]) = 1 + (1 + length []) -- second case`

`1 + (1 length []) = 1 + (1 + 0) -- first case`



Copy build replacing second case of length with the RHS and binding, xs \mapsto [39]

Copy build replacing second case of length with the RHS and binding, xs \mapsto []

3 2016 Q3

3.1 Part (b)

```

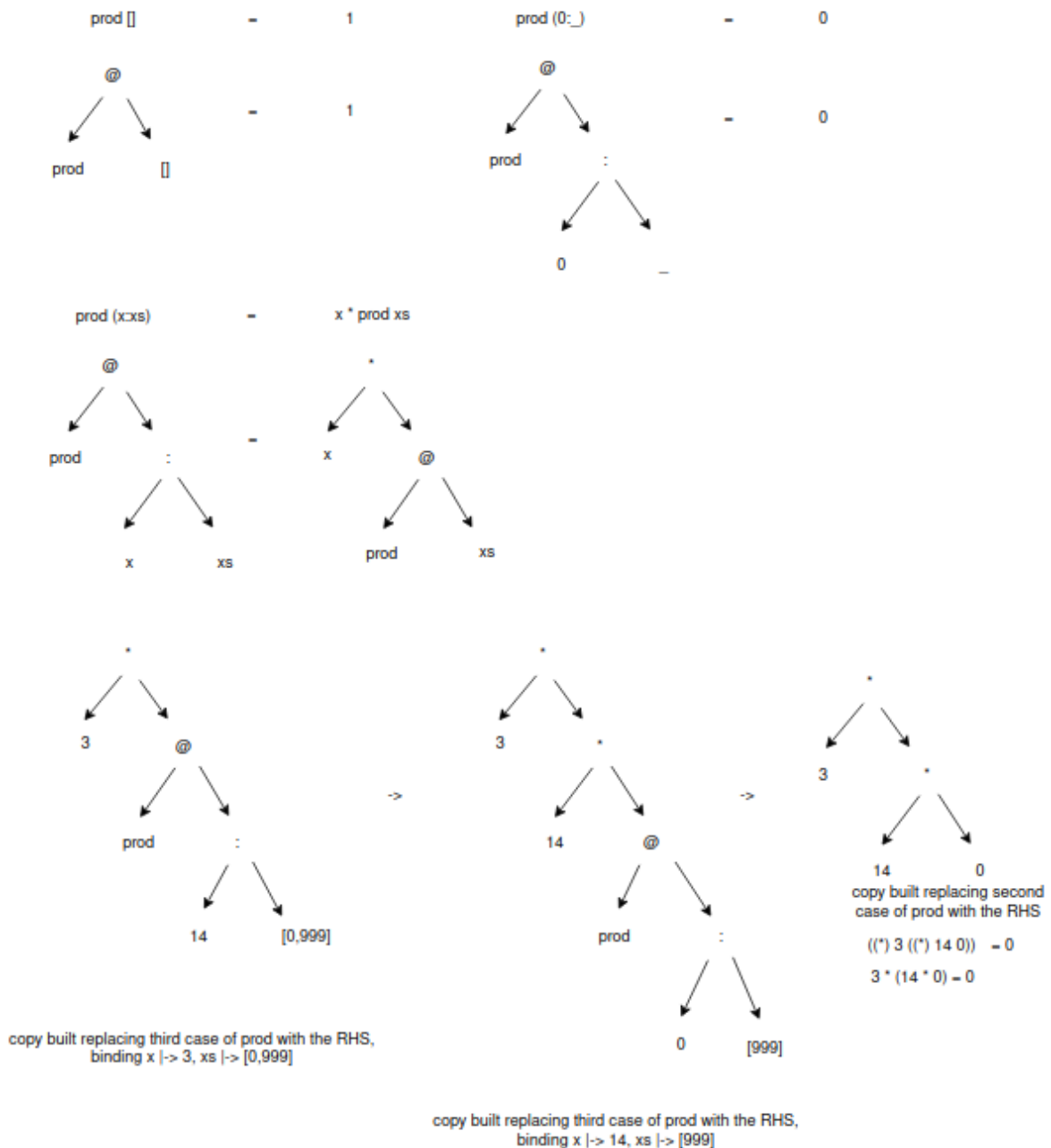
prod [] = 1
prod (0:_) = 0
prod (x:xs) = x * product xs
prod [3,14,0,999]

```

```

prod (3:[14,0,999]) = 3 * prod [14,0,999] -- third case
3 * product (13:[0,42]) = 3 * (14 * prod [0,42]) -- third case
3 * (13 * product (0:[42])) = 3 * (14 * 0) -- second case

```



4 2015 Q4

4.1 Part (a)

`diffsq [] = 0`

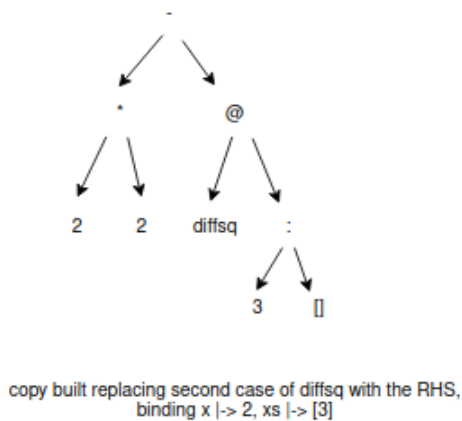
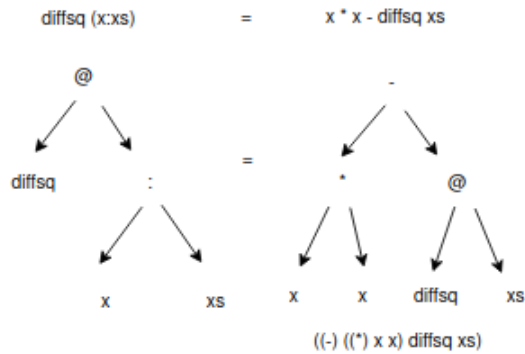
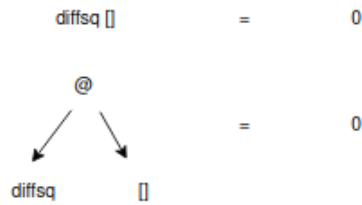
`diffsq (x:xs) = x * x - diffsq xs`

`diffsq [2,3]`

`diffsq (2:[3]) = (2 * 2) - diffsq [3] -- second case`

`(2 * 2) - diffsq (3:[]) = (2 * 2) - ((3 * 3) - diffsq []) -- second case`

`(2 * 2) - ((3 * 3) - diffsq []) = (2 * 2) - ((3 * 3) - 0) -- first case`



copy built replacing third case of diffsq with the RHS,
binding x |-> 3, xs |-> []

5 2014 Q4

5.1 Part (a)

`sum [] = 0`

`sum (x:xs) = x + sum xs`

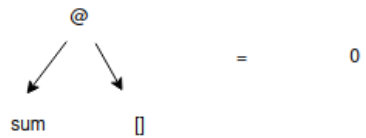
`sum [3,39]`

`sum (3:[39]) = 3 + sum [39] -- second case`

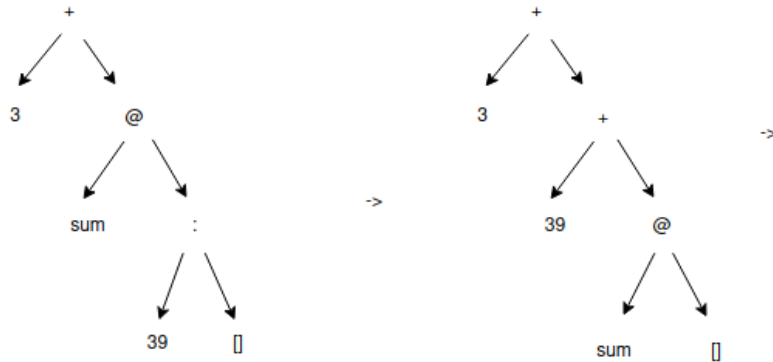
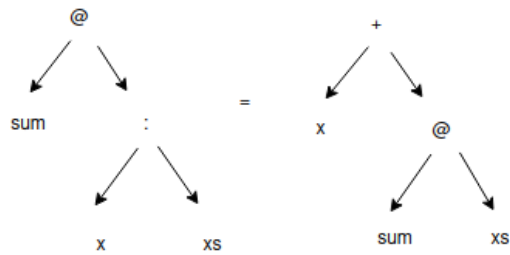
`3 + sum (39:[]) = 3 + (39 + sum []) -- second case`

`3 + (39 + sum []) = 3 + (39 + 0) -- first case`

`sum [] = 0`

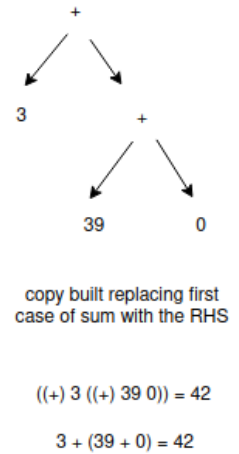


`sum (x:xs) = x + sum xs`



copy built replacing second case of sum with the RHS,
binding `x` to `3`, `xs` to `39:[]`

copy built replacing third case of sum with the RHS,
binding `x` to `3`, `xs` to `[]`



6 2013 Q4

6.1 Notes

- The 'sum(x:xs) = x * x + sumsq' example is not entirely correct the RHS of that part, should contain the x's and xs

```
+ | @
  | sumsq
  | xs
  | *
  | x
  | x
```

6.2 Part (a)

```
sumsq [] = 0
sumsq (x:xs) = x * x + diffsq xs
sumsq [2,3]
```

```
sumsq (2:[3]) = (2 * 2) + diffsq [3] -- second case
(2 * 2) + sumsq (3:[]) = (2 * 2) + ((3 * 3) + sumsq []) -- second case
(2 * 2) + ((3 * 3) + sumsq []) = (2 * 2) + ((3 * 3) + 0) -- first case
```

