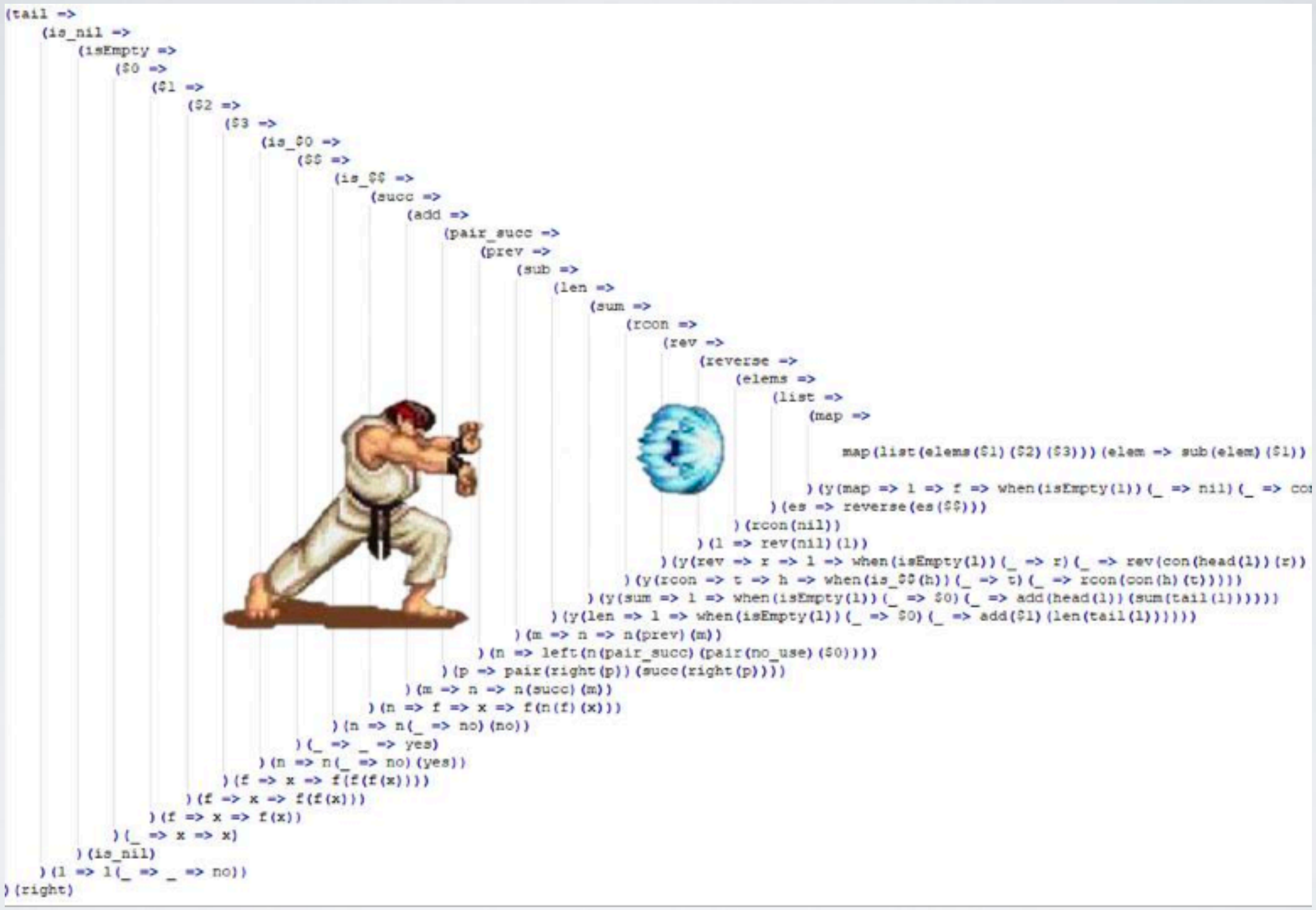

```
(tail =>
    (is_nil =>
        (isEmpty =>
            ($0 =>
                ($1 =>
                    ($2 =>
                        ($3 =>
                            (is_$0 =>
                                ($$ =>
                                    (is_$$ =>
                                        (succ =>
                                            (add =>
                                                (pair_succ =>
                                                    (prev =>
                                                        (sub =>
                                                            (len =>
                                                                (sum =>
                                                                    (rcon =>
                                                                        (rev =>
                                                                            (reverse =>
                                                                                (elems =>
                                                                                    (list =>
                                                                                        (map =>

                                                                                            map(list(elems($1)($2)($3)))(elem => sub(elem)($1))

                                                                                        )(y(map => l => f => when(isEmpty(l))(_ => nil)(_ => co
                                                                                    )(es => reverse(es($$)))
                                                                                )(rcon(nil))
                                                                            )(l => rev(nil)(l))
                                                                        )(y(rev => r => l => when(isEmpty(l))(_ => r)(_ => rev(con(head(l))(r))
                                                                    )(y(rcon => t => h => when(is_$$(h))(_ => t)(_ => rcon(con(h)(t)))))
                                                                )(y(sum => l => when(isEmpty(l))(_ => $0)(_ => add(head(l))(sum(tail(l))))))
                                                            )(y(len => l => when(isEmpty(l))(_ => $0)(_ => add($1)(len(tail(l))))))
                                                        )(m => n => n(prev)(m))
                                                    )(n => left(n(pair_succ)(pair(no_use)($0))))
                                                )(p => pair(right(p))(succ(right(p))))
                                            )(m => n => n(succ)(m))
                                        )(n => f => x => f(n(f)(x)))
                                    )(n => n(_ => no)(no))
                                )(_ => _ => yes)
                            )(n => n(_ => no)(yes))
                        )(f => x => f(f(f(x))))
                    )(f => x => f(f(x)))
                )(f => x => f(x))
            )(_ => x => x)
        )(is_nil)
    )(l => l(_ => _ => no))
)(right)
```

MOCKITO

```
@RunWith(MockitoJUnitRunner::class)
class PresenterTest {

    @Mock
    lateinit var view: View

    @Mock
    lateinit var dataProvider: DataProvider

    @Test
    fun `given non-empty list when presenter start then display elements on view`() {
        val elements = listOf(
                Element(1, "first"),
                Element(2, "second")
        )

        `when`(dataProvider.getAll()).thenReturn(elements)

        val presenter = Presenter(view, dataProvider)

        presenter.start()

        verify(view).displayItems(elements)

    }
}
```