Language Engineering - Problem Sheet 1

Dom Hutchinson

October 14, 2018

My Solution 1.1

```
data Robot = Forward Int Robot
           | LeftTurn Robot
           | RightTurn Robot
           | Stop
```

My Solution 1.2

```
distTrav (Forward n r) = n + distTrav r
distTrav (LeftTurn r)  = distTrav r
distTrav (RightTurn r) = distTrav r
distTrav (Stop)        = 0
```

My Solution 1.3

```
distTravForward :: Int → Robot → Int
distTravForward 0 (Forward n r) = n + distTravForward 0 r
distTravForward d (Forward n r) = distTravForward d r
distTravForward d (LeftTurn r) = distTravForward d r
distTravForward d (RightTurn r) = distTravForward d r
distTravForward _ (Stop r) = 0

distTrav :: Robot → Int
disTrav r = distTravForward 0 r
```

My Solution 1.4

```
distTrav :: Robot → Float
distTrav r = sqrt(fromIntegral(x*x) + fromIntegral(y*y))
  where
    y = distTravForward 0 r
    x = distTravForward 1 r
```

My Solution 2.1

```
potato = (0, 3, False, "potato")
```

My Solution 2.2

```
peel (t, w, c, s)                = (t+(2*w), w, c, "peeled " ++ s)
roast (t, w, c, s)               = (t+70, w, True, "roasted " ++ s)
boil em (t, w, c, s)             = (t+25, w, True, "boiled " ++ s)
mash em (t, w, c, s)             = (t+w, w, c, "mashed " ++ s)
stick em in a stew (t, w, c, s) = (t+120, w, True, "stewed " ++ s)
```

My Solution 2.3

```
mix (t1, w1, False, s1) (t2, w2, _, s2) = (t1+t2, w1+w2, False, s1 ++ " " ++
 s2)
mix (t1, w1, _, s1) (t2, w2, False, s2) = (t1+t2, w1+w2, False, s1 ++ " " ++
 s2)
mix (t1, w1, _, s1) (t2, w2, _, s2)     = (t1+t2, w1+w2, True, s1 ++ " " ++
 s2)
```

My Solution 2.4

```
carrot  = (0, 1, False, "carrot")
parsnip = (0, 1, False, "parsnip")

addcarrot (t, w, c, s)  = mix (t, w, c, s) carrot
addparsnip (t, w, c, s) = mix (t, w, c, s) parsnip
```

My Solution 3

A general purpose language is Turing Complete so can be used to develop programs in any domain, whereas a domain specific language is not. This means GPLs require their own support systems, while DSLs can make usr of these support systems and so don't need to fully develop their own.

My Solution 3

When using multiple semantics in shallow embeddings you provide all the semantics as a tuple and providing their interpretations with them. In deep embeddings you define a new function for each semantics.