# 15-150 Assignment 6


Karan Sikka
ksikka@andrew.cmu.edu

G
2/28/12

---

## 1: Task 2.2

For all $cs$ : `charlist` and $k$ : `charlist` $\rightarrow$ `bool`, if `matchWild` $cs\ k \cong$ `true` then $\exists p, s$ such that $p@s \cong cs$ and $p \in L(\_)$ and $k\ s \cong$ `true`.

**Proof:**
Assume `match Wild` $cs\ k \cong$ `true`

Then

```
case Wild of
  ... | Wild => case cs of
                    [] => false
                  | c'::cs' => k cs'
      | ...
```

must be true, so by inversion,

```
cs \cong c'::cs', and k cs' = true
```

Let $p = c'$ Let $s = cs'$

Then we know

```
cs = c'::cs' = p@s
```

$p \in L(Wild)$ because $p = c'$ and c' is a character since it is an elem of a char list.

`k cs'` $\cong$ `k s` $\cong$ `true` as shown above.

---

## 2: Task 2.4

Assume `match Both(r1,r2)` $cs\ k \cong$ `true` Then

```
case Both(r1,r2) of
  ... | Both (r1,r2) => match r1 cs (fn cs' => match r2 cs (fn cs'' =>
          charlisteq(cs',cs'') andalso (k cs'') ))
      | ...
```

must be true, so by inversion,

```
match r1 cs (fn cs' => match r2 cs (fn cs'' => charlisteq(cs',cs'') andalso (k cs'') ))
```

must be true.

Invoke the IH for `r1` where `k` $\cong$ `(fn cs' => match r2 cs (fn cs'' => charlisteq(cs',cs'') andalso (k cs'') ))`

Therefore, there exists a p1 and s1 such that p $\in L(r1)$ and `k s1` $\cong$ `true`, for the k above. From the fact that `k s1` is true:

`k s1` $\cong$ `true`
$\cong$ `(fn cs' => match r2 cs (fn cs'' => charlisteq(cs',cs'') andalso (k cs'') )) s1`
$\cong$ `match r2 cs (fn cs'' => charlisteq(s1,cs'') andalso (k cs'') )`

Now we can use the IH for r2 to say that there exists a p2 $\in L(r2)$ and an s2 such that k s2 $\cong$ true for the k above.

`k s2` $\cong$ `true`
$\cong$`(fn cs'' => charlisteq(s1,cs'') andalso (k cs'') ) s2`
$\cong$`charlisteq(s1,s2) andalso (k s2)`

Since this is true, we know that s1 and s2 are equal We can say that `s1 = s2 = s`.

Also, `k s2` is true, so `k s` is true.

All that's left to show is that `p@s` $\cong$ `cs`.

We know that `p1@s1` $\cong$ `cs` and `p2@s2` $\cong$ `cs`

This means that `p1@s` $\cong$ `p2@s` $\cong$ `cs`, and therefore `p1` $\cong$ `p2` $\cong$ `p`.

We have proven that `p@s` $\cong$ `cs` and `k s` $\cong$ `true`.

---

## 3: Task 2.6

**Claim:** For all `cs : char list`, if `matchany cs` $\cong$ `true` then $\exists$ `p, s` such that `p@s` $\cong$ `cs` with `k s` $\cong$ `true`.

We proceed by structural induction on cs.

Case []:
Assume `matchany []` $\cong$ `true`

`matchany []` $\cong$ `true`
$\cong$ `(case [] of [] => k cs | x::xs => k cs orelse match r xs matchany)`
$\cong$ `k cs` $\cong$ `true`
This works when we let `p = []` and `s = cs`. Therefore the base case holds.

Case `x::xs`:
Inductive hypothesis: The claim holds for xs.
Assume `matchany x::xs` $\cong$ `true`

`(case x::xs of [] => k cs | x::xs => k cs orelse match r xs matchany)`
$\cong$ `k cs orelse match r xs matchany`

By inversion on orelse, either `k cs` is true or `match r xs matchany` is true. If `k cs` is true, then let `p = [], s = cs`, and you're done.

If `match r xs matchany` is true, invoke the IH. Let p = x and s = xs. Clearly, `p@s` $\cong$ `cs`, and `k` `s` by the IH. QED.