60–100 LECTURE 19

More examples of induction

Mathematical induction

```
Prove that
  0 + 1 + 2 + ..n = (nˆ2 + n)/2
                   for all nats 0,1,2,3,4..
BASE CASE let n = 0
0 + .. 0 => 0 and (0ˆ2 + 0)/2 => 0
INDUCTIVE STEP
ASSUME TRUE FOR k
0 + 1 + 2 + .. k = (kˆ2 + k)/2
SHOW TRUE FOR k + 1
0 + 1 ..+ k + (k + 1)
          = ((k + 1)ˆ2 + (k + 1))/2
```

HOW to prove the following
using the assumption:

0 + 1 ..+ k + (k + 1)
= ((k + 1)ˆ2 + (k + 1))/2

Cut the assumption out:
remove lhs of assumption from lhs:
gives (k + 1)

Now expand rhs gives
(kˆ2 + 2k + 1 + k + 1)/2
remove rhs of assumption gives
(2k + 2)/2 => k + 1
which equals the lhs, proof complete.

PROVE that

$1 + 2n =< 3\hat{}n$ (for all n 0,1,2,3,4..)

BASE CASE let n = 0

$1 + 0 =< 1$ True

INDUCTIVE STEP

ASSUME TRUE FOR k

$1 + 2k =< 3\hat{}k$

SHOW TRUE FOR k + 1

$1 + 2(k + 1) =< 3\hat{}(k + 1)$

HOW to prove the following
using the assumption:
1 + 2(k + 1) =< 3^(k + 1)

Expand to find the assumption
=> 1 + 2k + 2 =< 3 * 3^k
=> 1 + 2k + 2 =< 3^k + (2 * 3^k)

remove assumption
=> 2 =< 2 * 3^k
True because k must be at least 0
Finished

Consider the reverse program

```
rev [] = []
rev (x:xs) = rev xs ++ [x]
```

Prove rev (m ++ n) = rev n ++ rev m

Can use "Structural induction" on the length of the list but need some identities on lists:

```
   [] ++ k    = k
    k ++ []   = k
  (y:ys) ++ k = y:(ys ++ k)
```

```
Proof of rev (m ++ n) = rev n ++ rev m
BASE CASE m = []
rev ([] ++ n)
   => rev n        (from ++)
   => rev n ++ [] (from ++)
   => rev n ++ rev [] (from defn of rev)

INDUCTIVE STEP
Assume rev (a ++ n)     = rev n ++ rev a
Show   rev ((x:a) ++ n) = rev n ++ rev (x:a)
How
rev ((x:a) ++ n)
 => rev (x:(a ++ n))       (from ++)
 => rev (a ++ n) ++ [x]    (from defn of rev)
 => rev n ++ rev a ++ [x]  (from assumption)
 => rev n ++ rev (x:a)     (from defn of rev)
```