(Defn)

composite : n is composite if $n = k \cdot \ell$ for some numbers $k \geq 2$ and $\ell \geq 2$

prime : n is prime if it is not composite

quot(a,b) : q is the quotient of a over b

rem(a,b) : r is the remainder of a over b

Weak induction :

- Base $P(0)$
- Assume $P(n)$, WTS $P(n+1)$
- Assume $P(n-1)$, WTS $P(n)$

Strong Induction :

- Base $P(0)$
- Assume $P(k)$, $\forall k < n$
- WTS $P(n)$

Euclidean Division :

- Base $P(0)$: let $q = r = 0$
- Assume $P(a-1)$, WTS $P(a)$
  - if $r' = b-1$ let $q = q'+1$ and $r = 0$
  - otherwise, let $q = q'$ and $r = r'+1$

$(d_i)_b = \sum_i d_i b^i = d_s b^s + d_{s-1} b^{s-1} + \cdots d_2 b^2 + d_1 b + d_0$

Base b representation : If (d_i) is a sequence of base b digits, and if $a = (d_i)_b$, then (d_i) is the Base b representation of a

- base b digit d is a natural number with $0 \leq d < b$
- base-b interpretation of (d_i) is $(d_i)_b = \sum_i d_i b^i$
- base-b representation of $n \in \mathbb{N}$ is a sequence of digits (d_i) with $n = (d_i)_b$

divides: a divides b $(a|b)$ if $\exists$ an integer $c \in \mathbb{Z}$ satisfying $ac=b$ Integers

greatest common divisor: if $a, b \in \mathbb{Z}$, then g of a and b satisfies
① $g|a$ and $g|b$
② $\forall c$ that divides a and b, $\exists c \le g$, $c|g$

Equivalence class: If R is equivalence relation on set A, then equivalence class of a by R ($[\![a]\!]_R$) is set of elements $b \in A$ with aRb

$[\![a]\!]_R := \{b \in R \mid aRb\}$

Mod: If R is equivalence relation on set A, then A mod R (A/R) is set of all equivalence classes of A by R $A/R := \{[\![a]\!] \mid a \in A\}$

Representative: If $c \in A/R$ is an equivalence class of A by R, and $a \in A$, we say a is representative of c if $a \in c$.

$\mathbb{Z}_m^*$ is the set of units of $\mathbb{Z}_m$

Modular Numbers: $[a]_m$ is an equivalence class under $\equiv_m$

$\mathbb{Z}_m$ is the set of modular numbers mod m. $\{[a]_m \mid a \in \mathbb{Z}\}$ same as $\mathbb{Z}/\equiv_m$

Equivalent mod: $a \equiv_m b$ if $\text{rem}(a,m) = \text{rem}(b,m)$ same as
- $[a]_m = [b]_m$
- $m \mid a-b$
  - $\exists c,\ a-b = mc$
  - $\exists c,\ a = b+mc$

Modular addition: $[a]_m + [b]_m := [a+b]_m$ is well-defined

Modular multiplication: $[a]_m \cdot [b]_m = [a \cdot b]_m$ is well-defined

Modular Exponentiation: $[a]^{[b]}$ as $[a^b]$ but NOT well-defined

- $[a]^{-1}$ (if $[a]$ is a unit) is well-defined
- $[a]_m^{[b]_{\varphi(m)}} := [a^b]_m$ is well-defined

Unit: A unit is an element with a multiplicative inverse

Multiplicative Inverse: A multiplicative inverse of x is an element y with $xy=1$

Totient: $\varphi(m)$ is the number of units of $\mathbb{Z}_m$ i.e size of $\mathbb{Z}_m^*$

Bézout Coefficient: $\exists$ Bézout Coefficient s and t satisfying $\gcd(a,b) = sa+tb$

If p is prime, $\varphi(p) = p-1$ because $\mathbb{Z}_p^* = \{\cancel{[0]}, [1], [2] \cdots [p-1]\}$

Euler's Theorem: If $[a]_m$ is a unit (need $[a]^{\varphi(m)-1} = [a]^{-1}$), $[a]_m^{[b]_{\varphi(m)}} := [a^b]_m$ is well defined

fast exponentiation: $[3^{23}]_{10} = [3^{16} \cdot 3^4 \cdot 3^2 \cdot 3^1]_{10} = [3^{16}]_{10} \cdot [3^4]_{10} \cdot [3^2]_{10} \cdot [3]_{10}$
$= [1] \cdot [1] \cdot [9] \cdot [3]$
$= [27]_{10}$
$= [7]$

BNF (Backus-Naur Form): $var_1 \in Set_1 ::= \text{rule } 1 \mid \text{rule } 2 \mid \cdots$
$var_2 \in Set_2 ::= \text{rule } 1 \mid \text{rule } 2 \mid \cdots$
$\vdots$

String: finite sequence of characters
- Σ^* refers to set of all strings with alphabet
  - the empty string " ", ε
  - the string xa where x is a string, a is a char

Alphabet Σ is a finite set of characters
- Character is element

Structure Induction: $\forall x \in X$, $P(x)$ ← Inductively defined set

ex① to prove $\forall x \in \Sigma^*$, $P(x)$
- prove $p(\varepsilon)$
- prove $p(xa)$ assuming $p(x)$

ex② to prove $\forall t \in T$, $p(t)$
- prove $P(\otimes)$
- prove $p\left(\begin{smallmatrix} & a & \\ t_1 & & t_2 \end{smallmatrix}\right)$ assuming $P(t_1)$ and $p(t_2)$

(proof)

table of proof techniques :

case 1: P is true
R is true?
case 2: Q is true
R is true?

Proposition	Symbol	To prove it	To use it	Logical negation
P and Q	(P ∧ Q)	prove both P and Q	you may use either P or Q	(¬P) ∨ (¬Q)
P or Q	(P ∨ Q)	You may either prove P or prove Q	case analysis	(¬P) ∧ (¬Q)
P is false (or "not P")	¬ P	disprove P	contradiction	P
if P then Q (or "P implies Q")	P ⇒ Q	assume P, then prove Q	if you know P, conclude Q	P ∧ ¬ Q
for all x, P	∀x, P	choose an arbitrary value x	apply to a specific x	∃ x, ¬ P
there exists x such that P	∃x, P	give a specific x	use an arbitrary x satisfying P	∀ x, ¬ P

weak induction

proof : $\forall n \in N, \exists k \in N$ s.t either $n = 2k$ or $n = 2k+1$

- Base: $p(0)$
  - $0 = 2 \cdot 0$ so $k = 0$
  - so $p(0)$ holds
- Assume $p(n)$, WTS $p(n+1)$
  - i.e assume $\exists k \in N$ s.t $n = 2k$ or $n = 2k+1$
  - i.e WTS $\exists k' \in N$ s.t $n+1 = 2k'$ or $n+1 = 2k'+1$

case 1: assume $n = 2k$

- $n+1 = 2k+1$
- let $k = k'$, then $n+1 = 2k'+1$
- so $p(n+1)$ holds

case 2: assume $n = 2k+1$

- $n+1 = 2k+1+1$
  $= 2k+2$
  $= 2(k+1)$
- let $k' = k+1$, then $n+1 = 2k'$
- so $p(n+1)$ holds

weak induction

proof: Every non-empty set with n elements has 2^n subsets

- WTS $p(n)$: $\forall n > 0$, $n \in \mathbb{N}$
- Base : $p(1)$
  - choose arb set $X = \{x_0\}$
  - $2^X = \{\phi, \{x_0\}\}$
  - By defn of power set, X has 2 elements, $2^1 = 2$
  - so $p(1)$ holds
- Assume $p(n)$, WTS $p(n+1)$
  - i.e assume for some arb $X = \{x_1, x_2 \cdots x_n\}$, X has 2^n subsets
  - i.e WTS $X' = \{x_1, x_2, \cdots x_{n+1}\}$, X' has 2^{n+1} subsets
  - $2^{X'}$ is consists of all the subsets of X and all the subsets of $X \cup \{x_{n+1}\}$
  - so $2^{X'} = 2^n + 2^n$
    $= 2^{n+1}$
  - so $p(n+1)$ holds

weak induction

proof: Every natural number $n \geq 2$ can be written as a product of one or more primes

- WTS $p(n)$: n can be written as a prod. of primes
- Base: $p(2)$
  - let $p_1 = 2$
  - 2 is prime so $2=2$
  - so $p(2)$ holds
- Assume $p(n)$, WTS $p(n+1)$
  - case 1: $n+1$ is prime
    - let $p_1 = n+1$
    - so $p(n+1)$ holds
  - case 2: $n+1$ is not prime
    - i.e $n+1 = k \cdot \ell$ for some $k \geq 2$ and $\ell \geq 2$
    - $p(k)$: $k = p_1 \cdot p_2 \cdots p_i$ ← all primes
    - $p(\ell) = \ell = q_1\, q_2 \cdots q_j$ ← all primes

      because we assumed $p(n)$
    - so $k \cdot \ell = p_1 \cdot p_2 \cdots p_i \cdot q_1 \cdot q_2 \cdots q_j$
    - so $p(n+1)$ holds

Euclidean Division Algorithm

proof: $\forall a, b > 0, \exists q$ and r satisfying ① $a = qb + r$ ② $0 \le r < b$

- let $p(a) := \forall b > 0, \exists q, r$ satisfying ① and ②
- Base $p(0)$:
  - WTS $\forall b > 0, \exists q, r$ s.t $0 = qb + r$ and $0 \le r < b$
  - let $q = r = 0$, then $0 = 0b + 0$ and $0 \le 0 < b$
  - so $p(0)$ holds
- Assume $p(a-1)$, WTS $p(a)$
  - i.e assume $\exists q', r'$ s.t $a - 1 = q'b + r'$, $0 \le r' < b$
  - i.e WTS $\exists q, r$ s.t $a = qb + r$, $0 \le r < b$
  - let $q = q'$ and $r = r' + 1$

case 1:

- $a - 1 = q'b + r'$

  $a = q'b + r' + 1$

  $= qb + r$

case 2:

- $0 \le r' + 1 < b$ is not true if $r' = b - 1$
- $a - 1 = q'b + r'$

  $a = q'b + r' + 1$

  $a = q'b + b$

  $a = (q' + 1)b + 0$

- if $r' = b - 1$, then let $q = q' + 1$ and $r = 0$

Uniqueness of Euclidean Division

proof: If $a = qb + r$ and $a = q'b + r'$ then $q = q'$ and $r = r'$

- Assume $a = qb + r = q'b + r'$
  - Then, $(r' - r) = (q - q')b$ $\leftarrow$ $(r' - r)$ is multiple of b
  - since $0 \le r < b$ and $0 \le r' < b$,

    $0 - b < r' - r < 0 + b$

    $-b < r' - r < b$

    - so $r' - r = 0$, $r = r'$
  - $r' - r = (q - q')b$

    $0 = (q - q')b$

    - so $q - q' = 0$, $q = q'$

proof: $\forall a \in \mathbb{N}$, and $b > 1$, $\exists$ a sequence (d_i) with $(d_i)_b = a$

- Base : $p(0)$
  - $d_0 = 0$
  - $0 = \Sigma d_i b^i = 0 \cdot 1$
  - so $p(0)$ holds
- Assume $p(a-1)$, WTS $p(a)$
  - let $d_0 := \text{rem}(a.b)$ and $d_{i+1} = d_i'$
  - $(d_i)_b = \sum_{i=0}^{k+1} d_i b^i$ by defn

    $= \sum_{i=1}^{k+1} d_i b^i + d_0 b^0$ algebra

    $= \sum_{j=0}^{k} d_{j+1} b^{j+1} + d_0 b^0$ let $j+1 = i$

    $= b \sum_{j=0}^{k} d_j' b^j + d_0 b^0$ by defn of d_{j+1}

    $= b(d_j')_b + d_0 b^0$ by defn

    $= qb + r$

    $= a$

Uniqueness of base-b representation

Proof: If $\sum_{i=0}^{n} d_i b^i = \sum_{i=0}^{n} d_i' b^i$ with $0 \le d_i < b$, then for all i, $d_i = d_i'$

- $p(n)$: if $\sum_{i=0}^{n} d_i b^i = \sum_{i=0}^{n} d_i' b^i$ with $0 \le d_i < b$ then for all i, $d_i = d_i'$
- $p(0)$: $d_0 = \sum_{i=0}^{0} d_i b^i = \sum_{i=0}^{0} d_i' b^i = d_0'$
- $p(n+1)$, assuming $p(n)$
  - $d_{n+1} = \sum_{i=0}^{n+1} d_i b^i = \sum_{i=0}^{n+1} d_i' b^i$

    $\sum_{i=1}^{n+1} d_i b^i + d_0 = \sum_{i=1}^{n+1} d_i' b^i + d_0'$

    $\left(\sum_{i=1}^{n+1} d_i b^{i-1}\right) b + d_0 = \sum_{i=1}^{n+1} \left(d_i' b^{i-1}\right) b + d_0'$

    $\sum_{i=1}^{n+1} d_i b^{i-1} = \sum_{i=1}^{n+1} d_i' b^{i-1}$ by base case $d_0 = d_0'$

    $\sum_{j=0}^{n} d_{j+1} b^j = \sum_{j=0}^{n} d_{j+1}' b^j$ $i = j+1$ or $i-1 = j$

  - so $d_{j+1} = d_{j+1}'$
  - so $d_i = d_i'$

Strong Induction

proof : let $g: \mathbb{N} \times \mathbb{N} \to \mathbb{N}$, $g(a,b) := \begin{cases} a & \text{if } b = 0 \\ g(b,r) \text{ where } r = rem(a,b) & \text{else} \end{cases}$

then $g(a,b)$ is gcd of a and b

(common divisor)

- WTS $\forall a, b \in \mathbb{N}$, $g(a,b) \mid a$ and $g(a,b) \mid b$
- let $p(b)$ be $\forall a$, $g(a,b) \mid a$ and $g(a,b) \mid b$
- Base : $p(0)$
    - WTS $\forall a$, $g(a,0) \mid a$ and $g(a,0) \mid 0$
    - choose arb a — proved by $p(r)$
    - $a \mid a$ since $a \cdot 1 = a$, $\exists c = 1$
    - $a \mid 0$ since $a \cdot 0 = a$, $\exists c = 0$
    - so $p(0)$ holds
- WTS $p(b)$ assuming $p(b-1)$, $p(b-2) \cdots p(0)$
    - choose arb a
    - $g(a,b) = g(b,r)$ where $r = rem(a,b)$
    - $p(r)$: $g(b,r) \mid b$ and $g(b,r) \mid r$ — $*$ $r < b$
        - $\exists c$, $b = cg(b,r)$
        - $\exists d$, $r = dg(b,r)$
    - $a = qb + r$

      $= qcg(b,r) + dg(b,r)$

      $= (qc + d)g(b,r)$
    - so a is multiple of $g(b,r)$

      (b is multiple of $g(b,r)$ by assumption)
    - so $g(b,r) \mid a$

(greatest)

- WTS $\forall c$, If $c|a$ and $c|b$ then $c|g$
- Base: $p(0)$
  - assume $c|a$ and $c|0$
  - WTS $c|g(a,0)$
  - $g(a,0) = a$, so $c|a$ by assumption
- WTS $p(b)$ assuming $p(b-1), p(b-2) \cdots p(0)$
  - choose arb a and c
  - assume $c|a$ and $c|b$
    - $\exists n_a$, $a = n_a c$
    - $\exists n_b$, $b = n_b c$

proved by assumption

  - we know $g(a,b) = g(b,r)$, WTS $c|b$ and $c|r$
    - $a = qb + r$

$n_a c = q n_b c + r$

$r = (n_a - q n_a) c$

  - since $c|r$, $c|g(a,b)$

* if $[a]=[a']$ then $[a+b]=[a'+b]$

proof : if $[a]=[a']$ and $[b]=[b']$ then $[a+b] = [a'+b']$

- assume $[a] = [a']$ and $[b] = [b']$ WTS $[a]+[b] = [a']+[b']$
- 방법① $[a]+[b] = [a+b]$

$$
\begin{aligned}
&= [a'+b] \\
&= [b+a'] \\
&= [b'+a'] \\
&= [a'+b'] \\
&= [a']+[b']
\end{aligned}
$$

- 방법② $[a] = [a']$ means $a \equiv_m a'$
  - so $a = a'+mc$ for some c
  - WTS $a+b = a'+b+md$ for some d
    - $a = a'+mc$

$a+b = a'+b+mc$ let $c=d$

$= a'+b+md$

* if $[a]=[a']$ then $[a\cdot b]=[a'\cdot b]$

proof : if $[a]=[a']$ and $[b]=[b']$ then $[a\cdot b] := [a'\cdot b']$

- assume $[a] = [a']$ so $a = a'+mc$
- WTS $[a\cdot b] = [a'\cdot b]$
  - i.e $ab = a'b+md$ for some d
- $ab = (a'+mc)b$

$= a'b+mcb$ let $cb=d$

$= a'b+md$

i.e a and m have no factors in common

proof: $[a]_m$ is a unit iff $\gcd(a,m) = 1$

- assume $\gcd(a,m)=1$
- then, $\exists\, s,t$ with $1 = sa + tm$
- then, $[1] = [sa+tm]$

$$= [s][a] + [t]\underbrace{[m]}_{[0]}$$

$$= [s][a] + [0]$$

$$= [s][a]$$

- so $[s]$ is an inverse of $[a]$
- so $[a]$ is a unit

Exercise: If p and q are distinct primes, find $\varphi(pq)$ by drawing Z_{pq} and crossing off non-units

- pq total elements $- 1 - (q-1) - (p-1)$

$$= pq - p - q + 1$$

$$= (p-1)(q-1)$$

Euler's Theorem V1

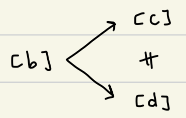
proof : If $[a]_m$ is a unit, $[a]_m^{[b]_{\varphi(m)}} := [a^b]_m$ is well-defined

- use Euler V2 : If $[a]_m$ is a unit then $[a]_m^{\varphi(m)} = [1]_m$
- WTS If $[a]_m = [a']_m$ and $[b]_{\varphi(m)} = [b']_{\varphi(m)}$, then $[a^b]_m = [a'^{b'}]_m$
  - WTS If $[a] = [a']$ then $[a^b] = [a'^b]$
    - $[a^b] = [\underbrace{a \cdot a \cdot a \cdots a}_{b \text{ times}}] = [a][a] \cdots [a]$
      $= [a'][a'] \cdots [a']$ b times
  - WTS If $[b]_{\varphi(m)} = [b']_{\varphi(m)}$ then $[a^b]_m = [a^{b'}]_m$
    - assume $[b]_{\varphi(m)} = [b']_{\varphi(m)}$
    - so $b \equiv_{\varphi(m)} b'$, so $b = b' + \varphi(m)K$
    - $[a^b] = [a^{b' + \varphi(m)K}]$
      $= [(a^{b'})(a^{\varphi(m)K})]$
      $= [a^{b'}][a^{\varphi(m)K}]$
      $= [a^{b'}]([a]^{\varphi(m)})^K$
    - by Euler V2, $[a]_m^{\varphi(m)} = [1]_m$
    - so $[a^{b'}] \cdot [1]_m^K = [a^{b'}]$

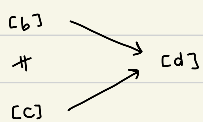
Euler's Theorem V2

proof: If $[a]_m$ is a unit, then $[a]_m^{\varphi(m)} = [1]_m$

- choose arb a and m with $[a]_m$ a unit
- Draw elements of $\mathbb{Z}_m^*$ with an arrow from each $[b]$ and $[a][b]$

① 

- If $[b] \to [c]$ and $[b] \to [d]$ then $[c] = [d]$
- $[b] \to [c]$ means $[a][b] = [c]$
- If $[b] \to [d]$ then $[a][b] = [d]$, so $[c] = [d]$

② 

- If $[a][b] = [d]$ and $[a][c] = [d]$ then $[b] = [c]$
- $[a][b] = [a][c]$ so $[a]^{-1}[a][b] = [a]^{-1}[a][c]$
- so $[b] = [c]$

③ $[b] \to$ non unit

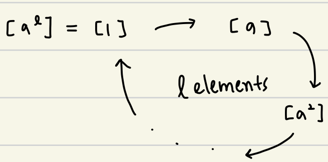
- $[b][a]$ is a unit because $([b][a])^{-1} = [b]^{-1}[a]^{-1}$

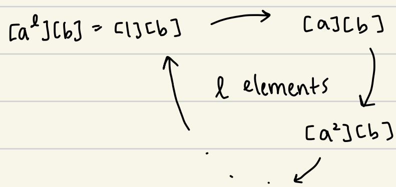
$([b]^{-1}[a]^{-1})([b][a]) = [b]^{-1}[b][a]^{-1}[a] = 1$

④ $[b] \longrightarrow \cdots$ (forever)

- there's fewer than m units
- must repeat eventually

- so units form cycles





...

- all loops have to be same length
    - n loops
    - $\varphi(m)$ total so $\varphi(m) = nl$
- To find $[a]^{\varphi(m)}$, start at $[1]$, $\underbrace{\text{take } \overset{nl}{\varphi(m)} \text{ steps}}_{\text{go around loop } n \text{ times}}$
- So end at $[1]$

Public Key cryptography (RSA)

- choose large primes p and q
  - choose a large odd number
  - check if it's prime
  - if not add two and try again
- multiply them to find m
- compute $\varphi(m)$
  - we know $m = pq$
  - $\varphi(m) = (p-1)(q-1)$
- choose a unit $[k]$ of $\mathbb{Z}_{\varphi(m)}$
  - $[k]_{\varphi(m)}$ is unit iff $\gcd(k, \varphi(m)) = 1$
  - try $k=7$, compute gcd
  - If not, try larger prime ($k=17$)
- compute $[k]^{-1}_{\varphi(m)}$
  - $1 = sk + t\varphi(m)$
  - $[1] = [s][k] + [t][0]$
    $= [s][k]$
  - $[s] = [k]^{-1}$
- compute $c = [msg]^{k}_{m}$
  - raise $c^{[k]^{-1}}$

example: public key $m = 392501 = 389 \cdot 1009$ and $k = 184049$.

Decrypt the message $c = 297627$

- $p = 389$, $q = 1009$
- $\varphi(m) = (389-1)(1009-1)$
  $= 391104$
- find gcd of:

a	b
391104	184049
184049	23006 ← rem(a,b)
23006	1

- divide to find $a = qb + r$:

q	r
2	23006
8	1

- find t' by $1 = sa + tb$: $1 = 1 \cdot 23006 + (-23005) \cdot 1$
  $1 = -23005 \cdot 184049 + (184041) \cdot 23006$
  $1 = 184041 \cdot 391104 + (-391087) \cdot 184049$ ← k
- use s' and t', compute $s = t'$ and $t = s' - t'q$:

s	t
184041	-391087
-23005	184041
1	-23005

↙ coefficient of k

- $[k]^{-1}_{\varphi(m)} = [-391087] = [17]$
- $c = [297627]_m$
  $c^2 = [242944]_m$
  $c^4 = [234263]_m$
  $c^8 = [55850]$
  $c^{16} = [17053]$
  $c^{17} = c \cdot c^{16} = [2800]$

Defn : String using BNF

- String is either string ε or ya where y is a string and a is a character
  - $x \in \Sigma^* ::= \varepsilon \mid xa \qquad a \in \Sigma$
  - to define $f : \Sigma^* \to X$: give defⁿ of $f(\varepsilon)$ and $f(xa)$ using $f(x)$

Defn : Natural Number using BNF

- $n \in N$ is either zero or successor of another natural number
  - $n \in N ::= Z \mid Sn$

    Inductively defined functions
  - to define $f : N \to X$: give $f(Z)$ and $f(Sn)$ using $f(n)$

Defn : Binary Tree using BNF

- $t \in T$ of integer is either empty tree or formed by combining two trees with an integer at the root

  * $\otimes$ is empty tree
  - $b \in T ::= \otimes \mid \overset{a}{t_1 \;\; t_2}$ where $a \in N$

    Inductively defined functions
  - to define $f : T \to X$: $f(\otimes)$ and $f\left(\overset{a}{t_1 \;\; t_2}\right)$ using $f(t_1)$ and $f(t_2)$

    Structural Induction
  - prove $P(\otimes)$, prove $P\left(\overset{a}{t_1 \;\; t_2}\right)$ assuming $P(t_1)$ and $P(t_2)$

Defn : Arithmetic expression using BNF

- arithmetic expression $e \in Expr$ is either a number or sum of two expressions or product of two expressions or negation of an expression
-
  - $e \in Expr ::= n \mid e_1 + e_2 \mid e_1 \cdot e_2 \mid -e$ where $n \in N$

Q1: Inductively define a function cat: $\Sigma^* \times \Sigma^* \to \Sigma^*$ that concatenates two strings : cat("abc", "de") should be "abcde"

- $cat(\varepsilon, \varepsilon) := \varepsilon$
- $cat(\varepsilon, yb) := yb$
- $cat(xa, \varepsilon) := xa$
- $cat(xa, yb) := xayb$

Q2: prove that for all $x, y \in \Sigma^*$, $len(cat(x,y)) = len(x) + len(y)$

- prove $p(\varepsilon)$
  - WTS $len(cat(x, \varepsilon)) = len(x) + len(\varepsilon)$

    $x = len(x) + 0$
- prove $p(yb)$ assuming $p(y)$
  - i.e assume $len(cat(x,y)) = len(x) + len(y)$
  - i.e WTS $len(cat(x, yb)) = len(x) + len(yb)$
    - $len(cat(x,y)b) = 1 + len(cat(x,y))$
    - $len(x) + len(yb) = len(x) + len(y) + 1$

      $= len(cat(x,y)) + 1$

same