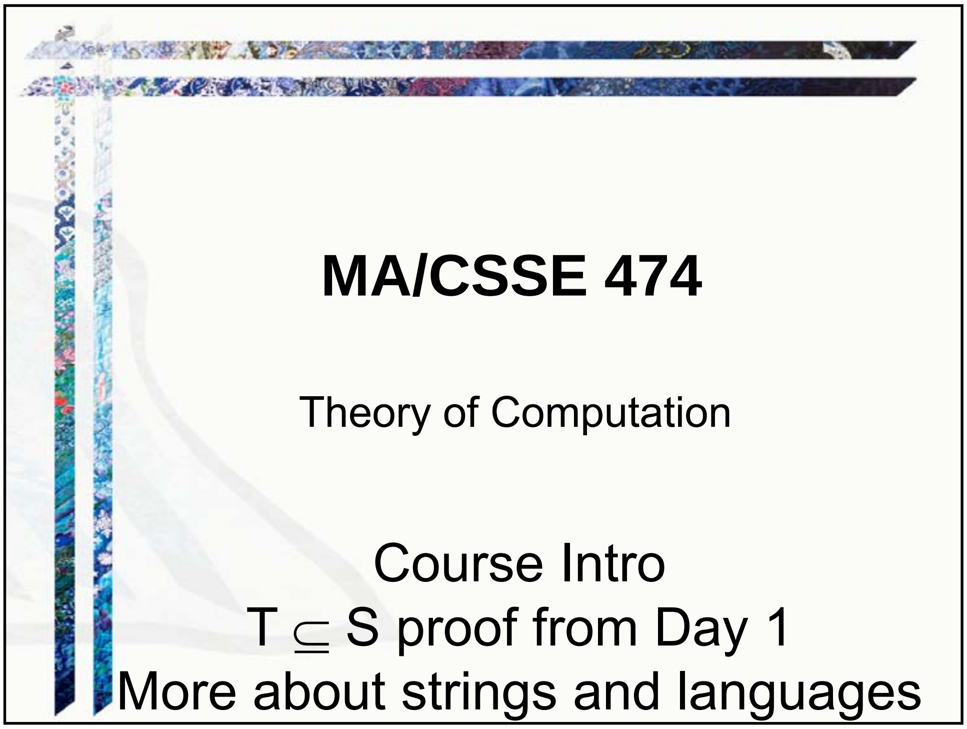
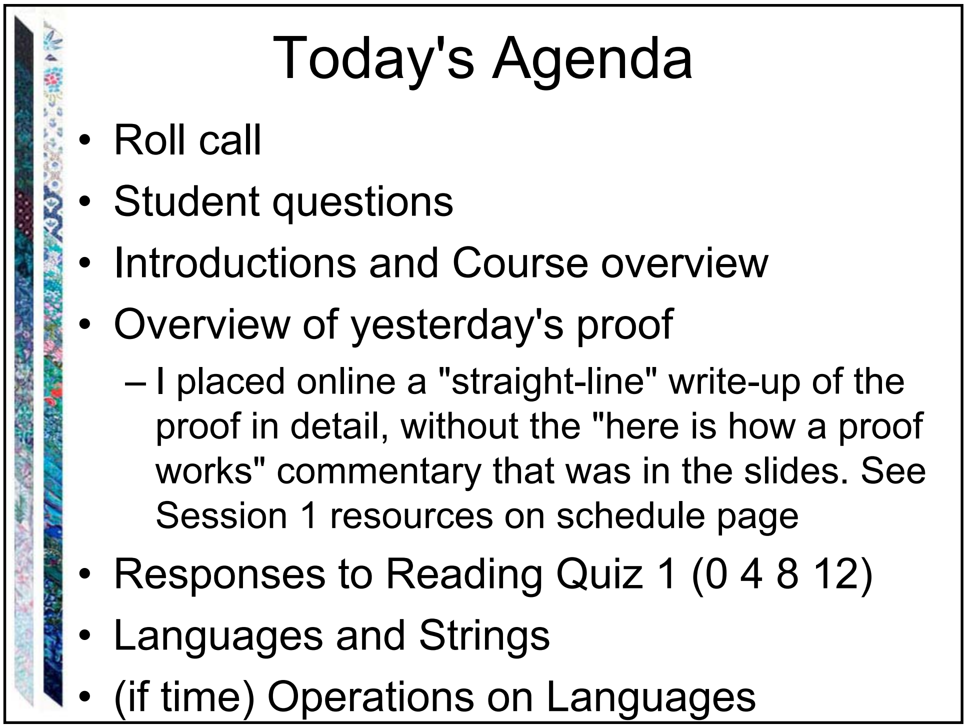


MA/CSSE 474

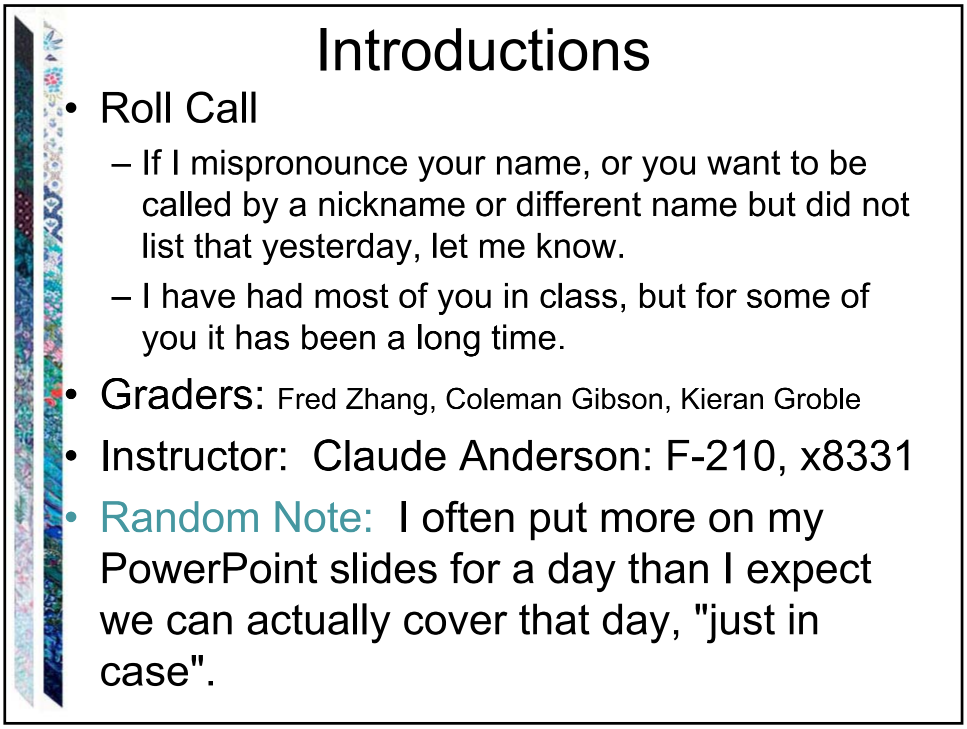
Theory of Computation

Course Intro
T $\subseteq$ S proof from Day 1
More about strings and languages



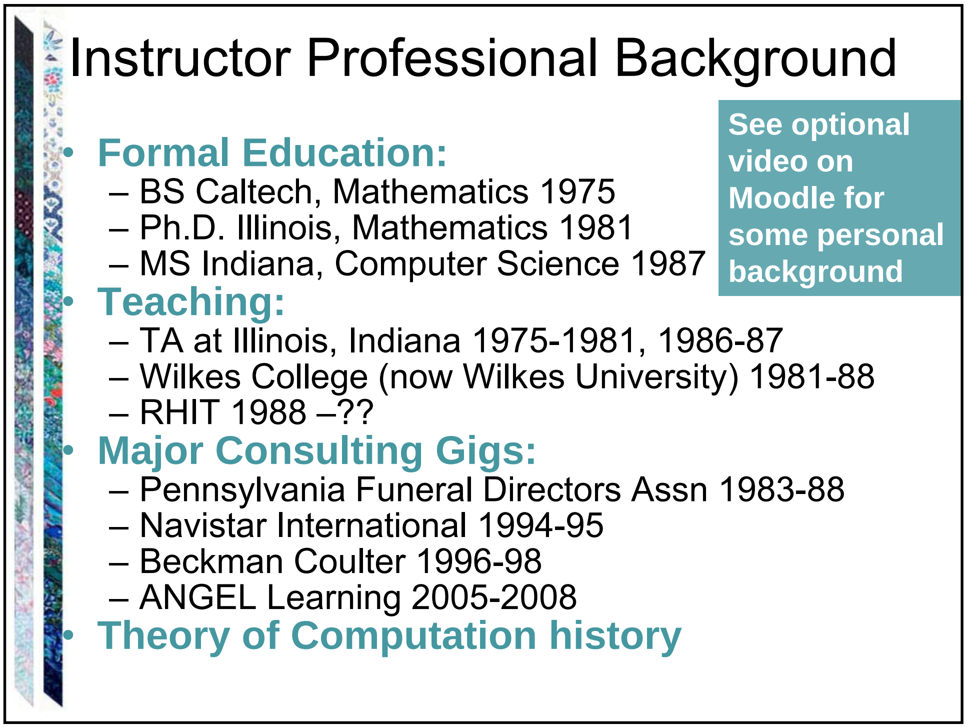
Today's Agenda

- Roll call
- Student questions
- Introductions and Course overview
- Overview of yesterday's proof
  - I placed online a "straight-line" write-up of the proof in detail, without the "here is how a proof works" commentary that was in the slides. See Session 1 resources on schedule page
- Responses to Reading Quiz 1 (0 4 8 12)
- Languages and Strings
- (if time) Operations on Languages



Introductions

- Roll Call
  - If I mispronounce your name, or you want to be called by a nickname or different name but did not list that yesterday, let me know.
  - I have had most of you in class, but for some of you it has been a long time.
- Graders: Fred Zhang, Coleman Gibson, Kieran Groble
- Instructor: Claude Anderson: F-210, x8331
- Random Note: I often put more on my PowerPoint slides for a day than I expect we can actually cover that day, "just in case".



Instructor Professional Background

See optional video on Moodle for some personal background

- **Formal Education:**
  - BS Caltech, Mathematics 1975
  - Ph.D. Illinois, Mathematics 1981
  - MS Indiana, Computer Science 1987
- **Teaching:**
  - TA at Illinois, Indiana 1975-1981, 1986-87
  - Wilkes College (now Wilkes University) 1981-88
  - RHIT 1988 –??
- **Major Consulting Gigs:**
  - Pennsylvania Funeral Directors Assn 1983-88
  - Navistar International 1994-95
  - Beckman Coulter 1996-98
  - ANGEL Learning 2005-2008
- **Theory of Computation history**

What do we Study in Theory of Computation?

- Larger issues, such as
  - What can be computed, and what cannot?
  - What problems are tractable?
  - What are reasonable mathematical models of computation?

Applications of the Theory

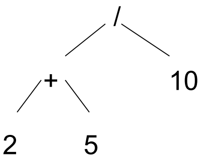
- Finite State Machines (FSMs) for parity checkers, vending machines, communication protocols, and building security devices.
- Interactive games as nondeterministic FSMs.
- Programming languages, compilers, and context-free grammars.
- Natural languages are mostly context-free. Speech understanding systems use probabilistic FSMs.
- Computational biology: DNA and proteins are strings.
- The undecidability of a simple security model.
- Artificial intelligence: the undecidability of first-order logic.

Some Language-related Problems

```
int alpha, beta;
alpha = 3;
beta = (2 + 5) / 10;
```

(1) **Lexical analysis**: Scan the program and break it up into variable names, numbers, operators, punctuation, etc.

(2) **Parsing**: Create a tree that corresponds to the sequence of operations that should be executed, e.g.,



(3) **Optimization**: Realize that we can skip the first assignment since the value is never used, and that we can pre-compute the arithmetic expression, since it contains only constants.

(4) **Termination**: Decide whether the program is guaranteed to halt.

(5) **Interpretation**: Figure out what (if anything) useful it does.

A Framework for Analyzing Problems

We need a single framework in which we can analyze a very diverse set of problems.

The framework we will use is

Language Recognition

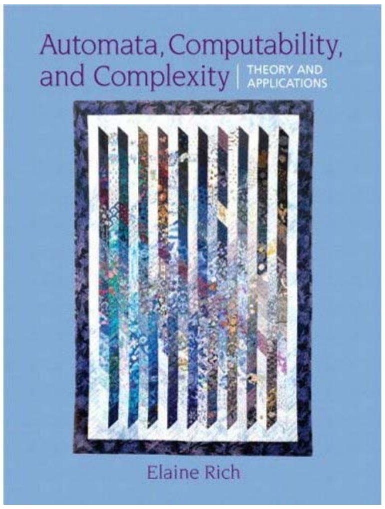
Most interesting problems can be restated as language recognition problems.

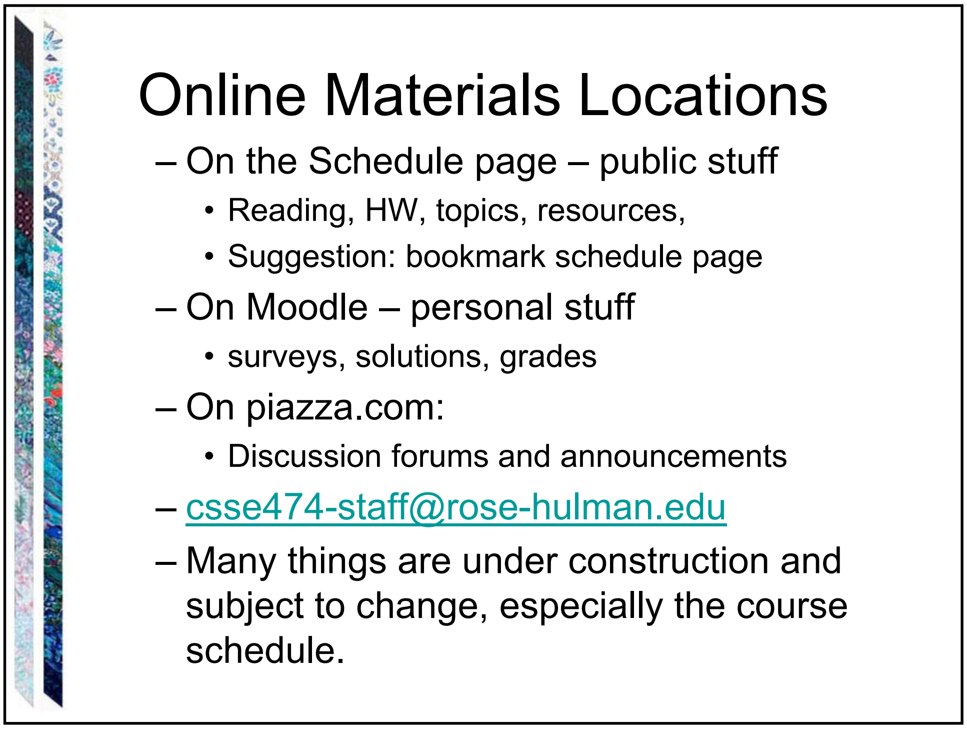
What we will focus on in 474

- Definitions
- Theorems
- Examples
- Proofs
- A few applications, but mostly theory

Textbook

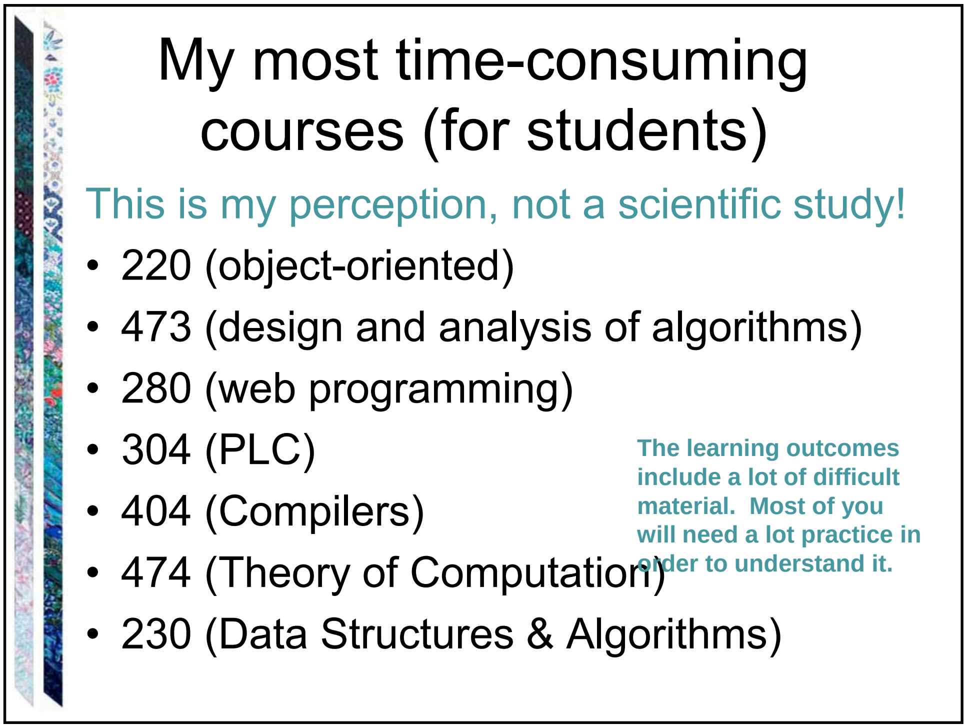
- Thorough
- Literate
- Large (and larger!)
- Theory and Applications
- We'll focus more on theory; applications are there for you to see





Online Materials Locations

- On the Schedule page – public stuff
  - Reading, HW, topics, resources,
  - Suggestion: bookmark schedule page
- On Moodle – personal stuff
  - surveys, solutions, grades
- On piazza.com:
  - Discussion forums and announcements
- csse474-staff@rose-hulman.edu
- Many things are under construction and subject to change, especially the course schedule.



My most time-consuming courses (for students)

This is my perception, not a scientific study!

- 220 (object-oriented)
- 473 (design and analysis of algorithms)
- 280 (web programming)
- 304 (PLC)
- 404 (Compilers)
- 474 (Theory of Computation)
- 230 (Data Structures & Algorithms)

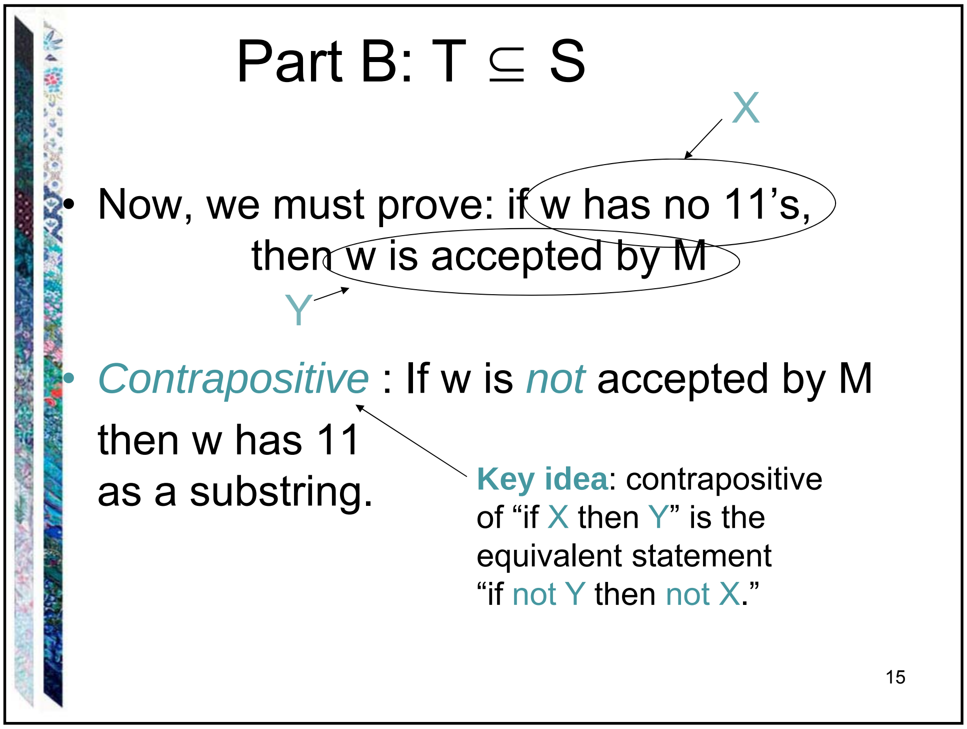
The learning outcomes include a lot of difficult material. Most of you will need a lot practice in order to understand it.

Questions about course policies and procedures?

- From Syllabus?
- Schedule page?
- Things said in class yesterday?
- Anything else?
- Attendance?
- Late Days?
- How to find my office hours for a given day?
- Anything else?

Overview of yesterday's proof

- $S = L(M)$, language accepted by M
- $T = \{w \in \{0,1\}^* :$ w does not have 11 as substring$\}$
- Show that $S = T$. i.e., $S \subseteq T$ and $T \subseteq S$
- $S \subseteq T$: i.e. if $w \in S$, then $w \in T$
  - By induction on $|w|$, showed
    - If $\delta(q0, w) = q0$, w has no 11 and does not end in 1.
    - If $\delta(q0, w) = q1$, w has no 11 and ends in 1.
- $T \subseteq S$: i.e. if $w \in T$, then $w \in S$.
  - We show the contrapositive: if $w \notin S$, then $w \notin T$.
    - If $w \notin S$ then $\delta(q0, w) = q2$ (the only non-accepting state).
    - Show by induction that if $\delta(q0, w) = q2$, then w contains 11.
    - This uses the property of q1 that was proved by induction as part of $S \subseteq T$.



Part B: T ⊆ S

X

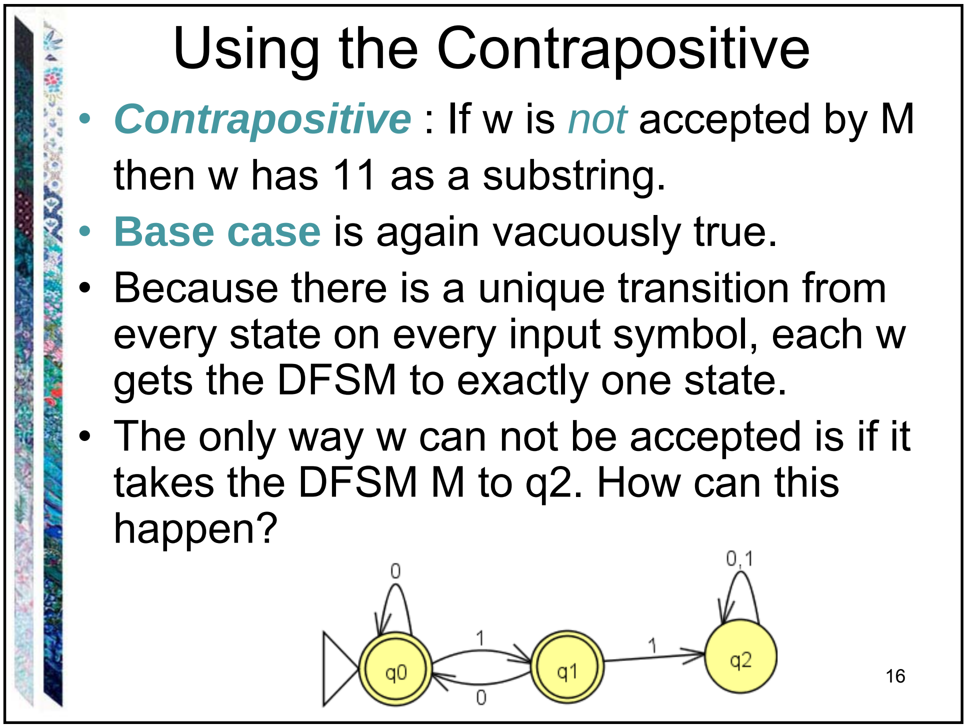
- Now, we must prove: if w has no 11's, then w is accepted by M

Y

- *Contrapositive* : If w is *not* accepted by M then w has 11 as a substring.

Key idea: contrapositive of "if X then Y" is the equivalent statement "if not Y then not X."

15



Using the Contrapositive

- ***Contrapositive*** : If w is *not* accepted by M then w has 11 as a substring.
- **Base case** is again vacuously true.
- Because there is a unique transition from every state on every input symbol, each w gets the DFSM to exactly one state.
- The only way w can not be accepted is if it takes the DFSM M to q2. How can this happen?

0
0,1
1
1
q0
q1
q2
0

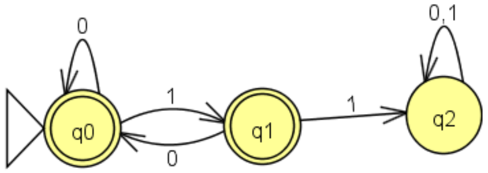
16

Using the Contrapositive – (2)

Looking at the DFSM, there are two possibilities: (recall that w=ua)

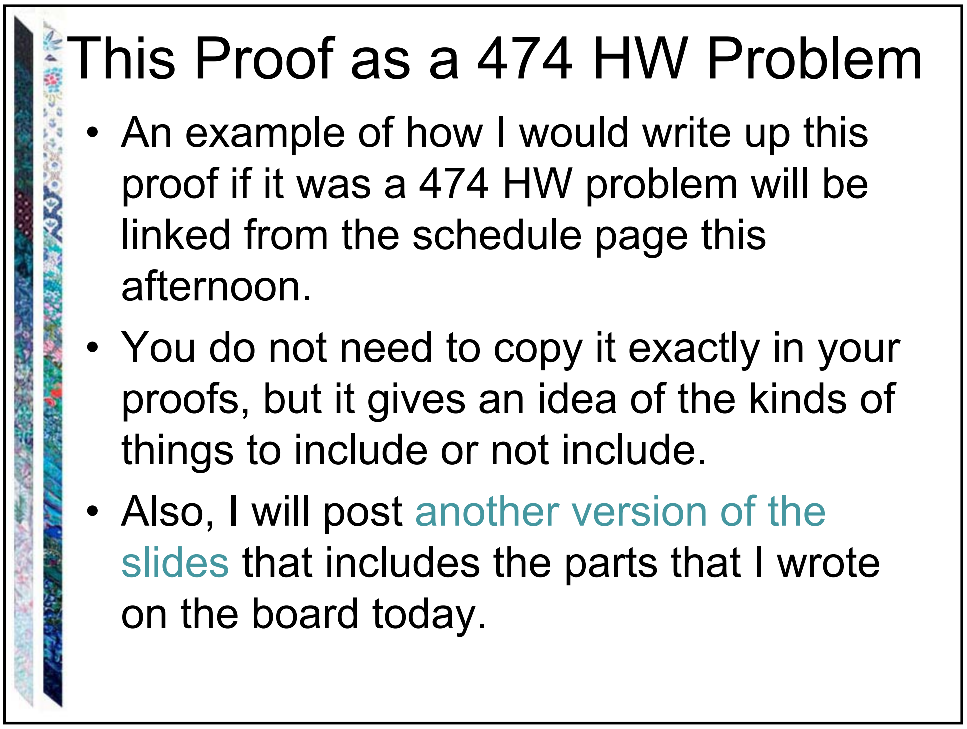
1. δ(q0,u) = q1 and a is 1. We proved earlier that if δ(q0,u) = q1, then u ends in 1. Thus w ends in 11.

2. δ(q0,u) = q2. In this case, the IH says that u contains 11 as a substring. So does w=ua.



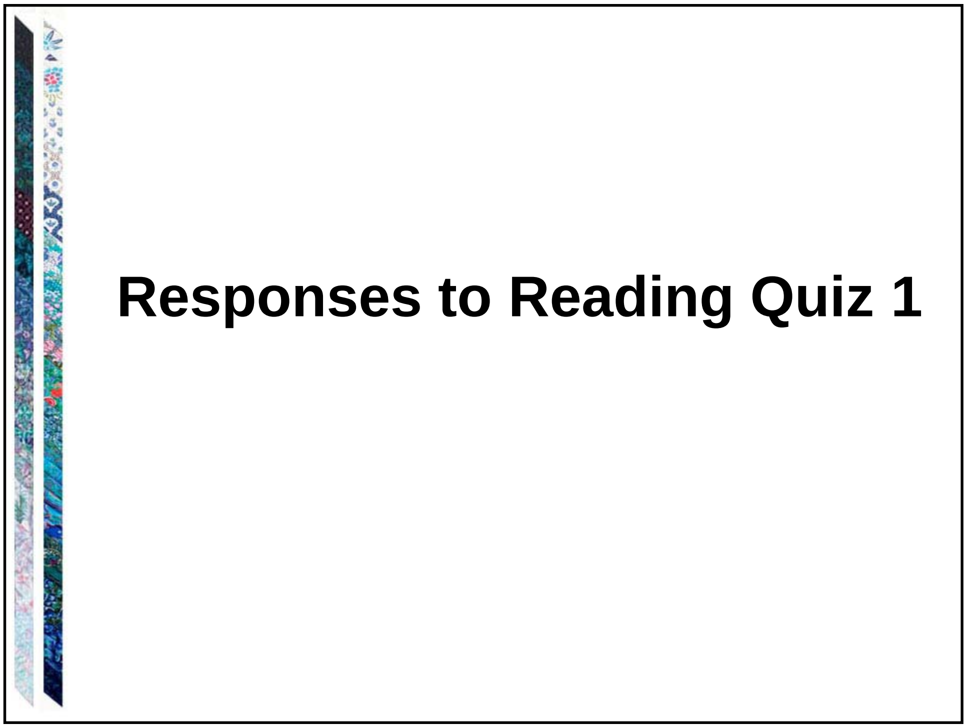
Your 474 HW induction proofs

- Can be slightly less detailed
  - Many of the details above were about how the proof process works in general, rather than about the proof itself.
  - *You* can assume that the reader knows the proof *techniques*.
- You must always make it clear what the IH is, and where you apply it.
  - When in doubt about whether to include a detail, include it!
- Well-constructed proofs often contain more words than symbols.



This Proof as a 474 HW Problem

- An example of how I would write up this proof if it was a 474 HW problem will be linked from the schedule page this afternoon.
- You do not need to copy it exactly in your proofs, but it gives an idea of the kinds of things to include or not include.
- Also, I will post another version of the slides that includes the parts that I wrote on the board today.



Responses to Reading Quiz 1

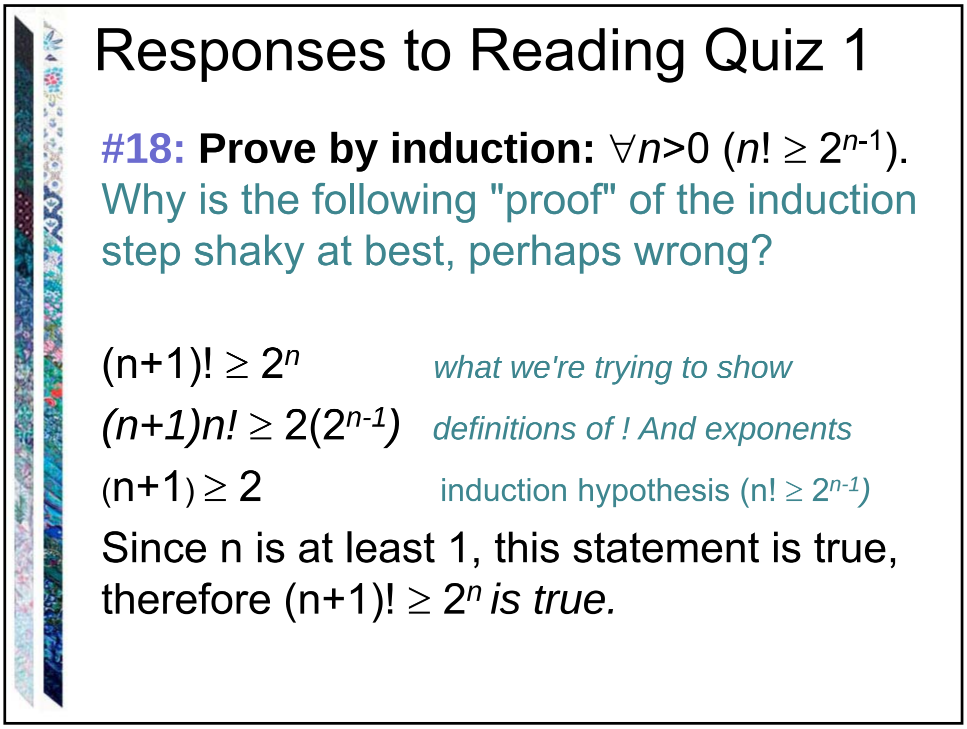
Responses to Reading Quiz 1

- **From #4:** $\wp(\emptyset) = \{ \emptyset \}$ (not $\wp(\emptyset) = \emptyset$)
  What is $\wp(\wp(\emptyset))$?
- **From #4:** $\{a, b\} \mathrm{X} \{1, 2, 3\} \mathrm{X} \emptyset = \emptyset$
- **#10:** (representing $\{1, 4, 9, 16, 25, 36, \ldots\}$
  in the form: $\{x \in A : P(x)\}$
  $\{x \in \mathbb{N} : x>0 \land \exists y \in \mathbb{N} \, (y*y = x)\}$
  Why not $\{x \in \mathbb{N} : x>0 \land \mathrm{sqrt}(x) \in \mathbb{N}\}$?
- **From #15:** $\forall x \in \mathbb{N} \, (\exists y \in \mathbb{N} \, (y < x))$.
  Why is this **not** satisfiable? (e, g, by x=3, y=2)

Responses to Reading Quiz 1

#16: Let $\mathbb{N}$ be the set of nonnegative integers. Let A be the set of nonnegative integers x such that $x \equiv_3 0$.
Show that $|\mathbb{N}| = |A|$.
Define a function $f : \mathbb{N} \to A$ by $f(n) = 3n$.

f is one-to-one: if $f(n) = f(m)$, then $3n = 3m$, so $m=n$.
f is onto: Let $k \in A$. Then $k = 3m$ for some $m \in \mathbb{N}$. So $k = f(m)$.



Responses to Reading Quiz 1

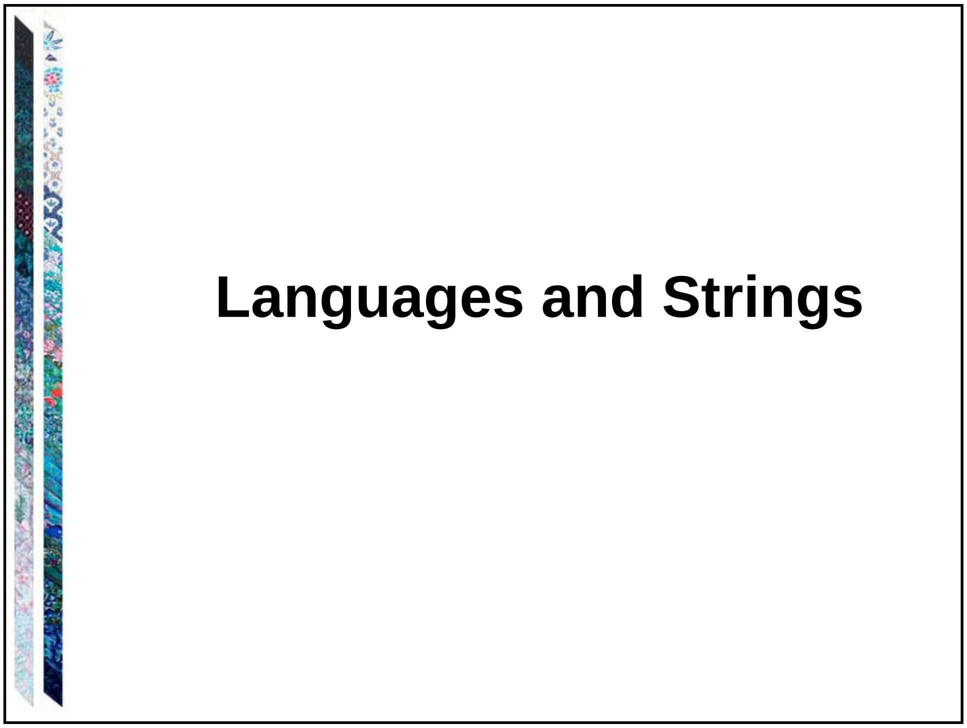
#18: Prove by induction: $\forall n>0\ (n! \geq 2^{n-1})$. Why is the following "proof" of the induction step shaky at best, perhaps wrong?

$(n+1)! \geq 2^n$ *what we're trying to show*

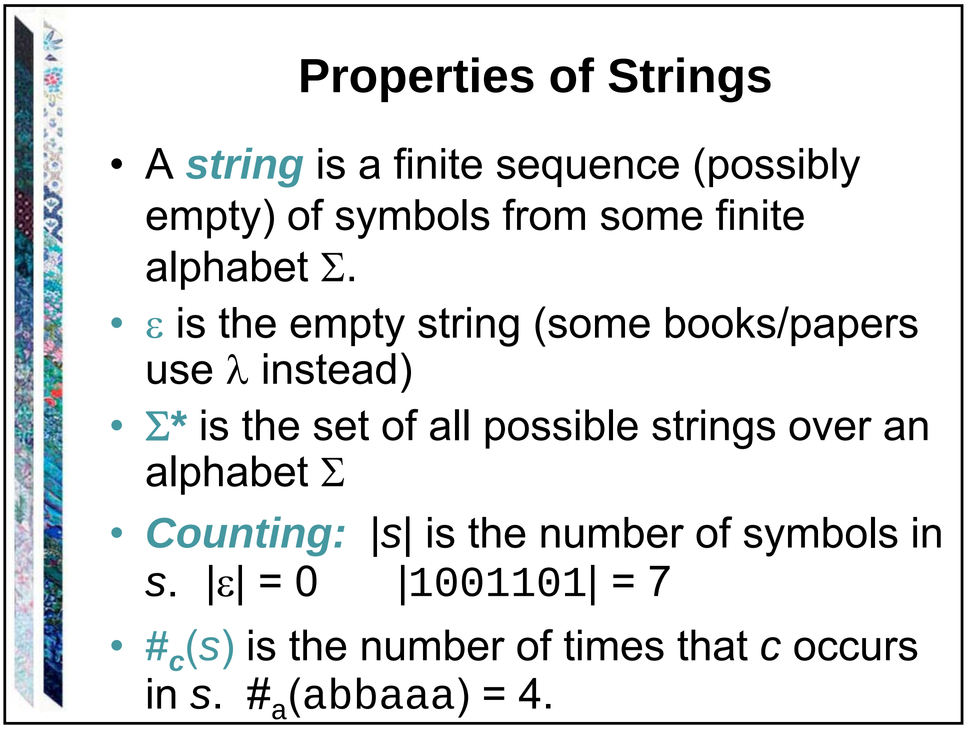
$(n+1)n! \geq 2(2^{n-1})$ *definitions of ! And exponents*

$(n+1) \geq 2$ induction hypothesis $(n! \geq 2^{n-1})$

Since n is at least 1, this statement is true, therefore $(n+1)! \geq 2^n$ *is true.*

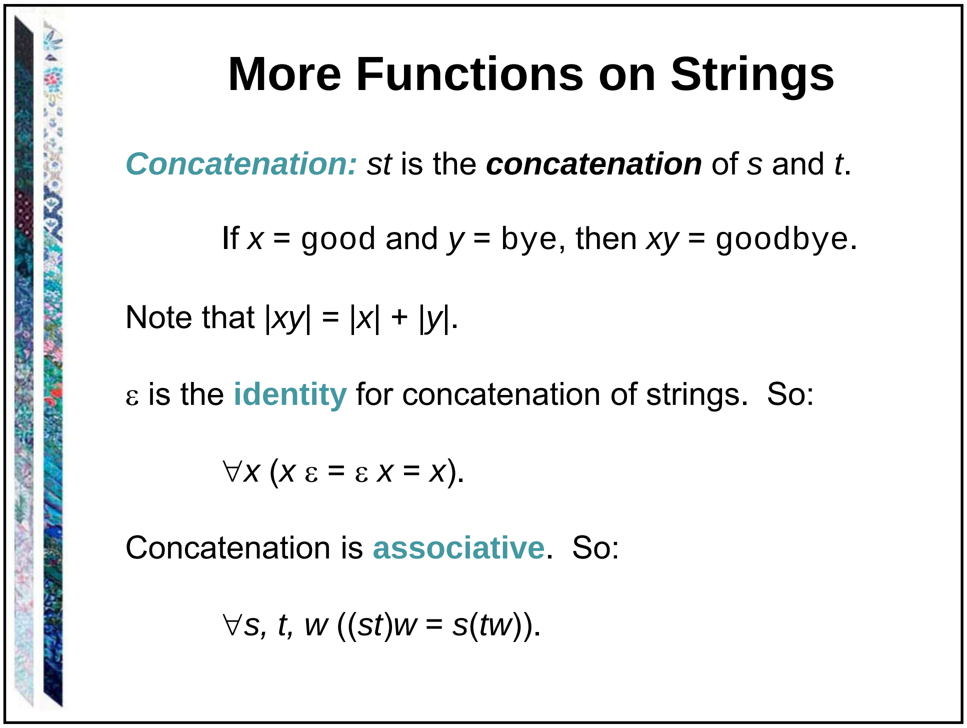


Languages and Strings



Properties of Strings

- A ***string*** is a finite sequence (possibly empty) of symbols from some finite alphabet Σ.
- ε is the empty string (some books/papers use λ instead)
- Σ^* is the set of all possible strings over an alphabet Σ
- ***Counting:*** $|s|$ is the number of symbols in s. $|\varepsilon| = 0$ $|\texttt{1001101}| = 7$
- $\#_c(s)$ is the number of times that c occurs in s. $\#_a(\texttt{abbaaa}) = 4$.



More Functions on Strings

Concatenation: *st* is the ***concatenation*** of *s* and *t*.

If x = good and y = bye, then xy = goodbye.

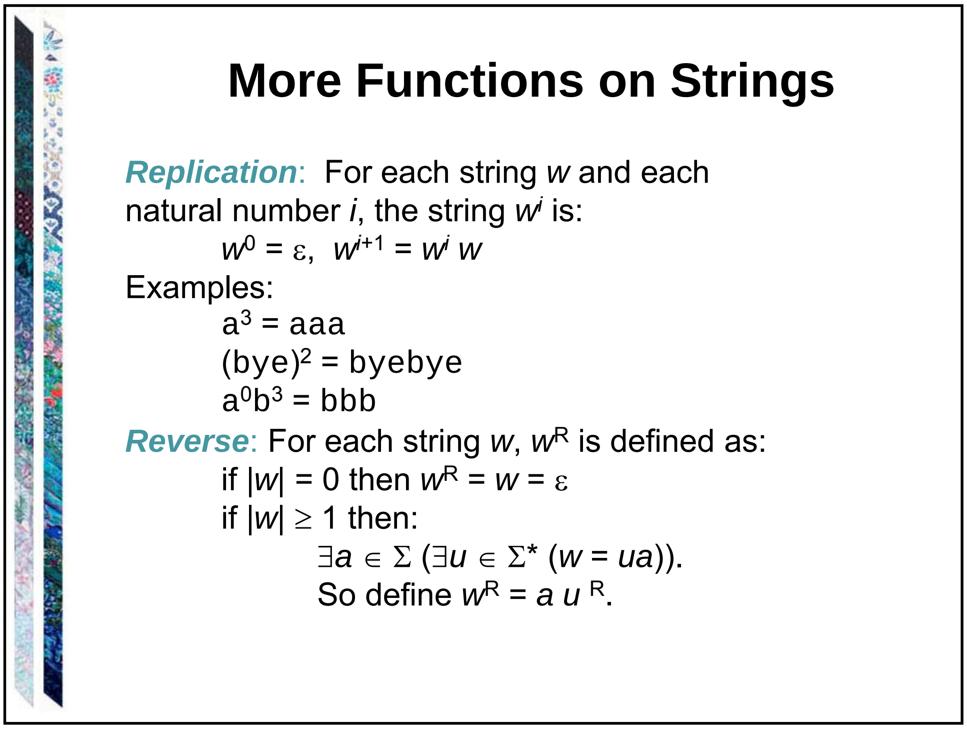
Note that $|xy| = |x| + |y|$.

ε is the **identity** for concatenation of strings. So:

$\forall x\ (x\,\varepsilon = \varepsilon\, x = x)$.

Concatenation is **associative**. So:

$\forall s, t, w\ ((st)w = s(tw))$.



More Functions on Strings

Replication: For each string w and each natural number i, the string w^i is:

$w^0 = \varepsilon$, $w^{i+1} = w^i\, w$

Examples:

$\texttt{a}^3 = \texttt{aaa}$

$(\texttt{bye})^2 = \texttt{byebye}$

$\texttt{a}^0\texttt{b}^3 = \texttt{bbb}$

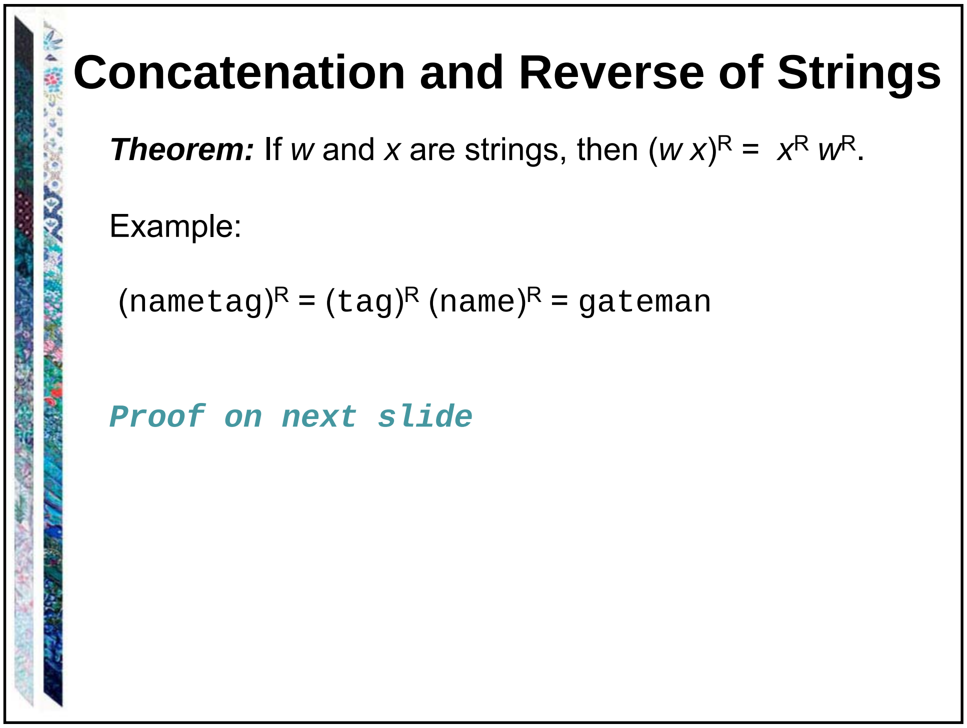
Reverse: For each string w, w^R is defined as:

if $|w| = 0$ then $w^R = w = \varepsilon$

if $|w| \geq 1$ then:

$\exists a \in \Sigma\ (\exists u \in \Sigma^*\ (w = ua))$.

So define $w^R = a\, u^R$.



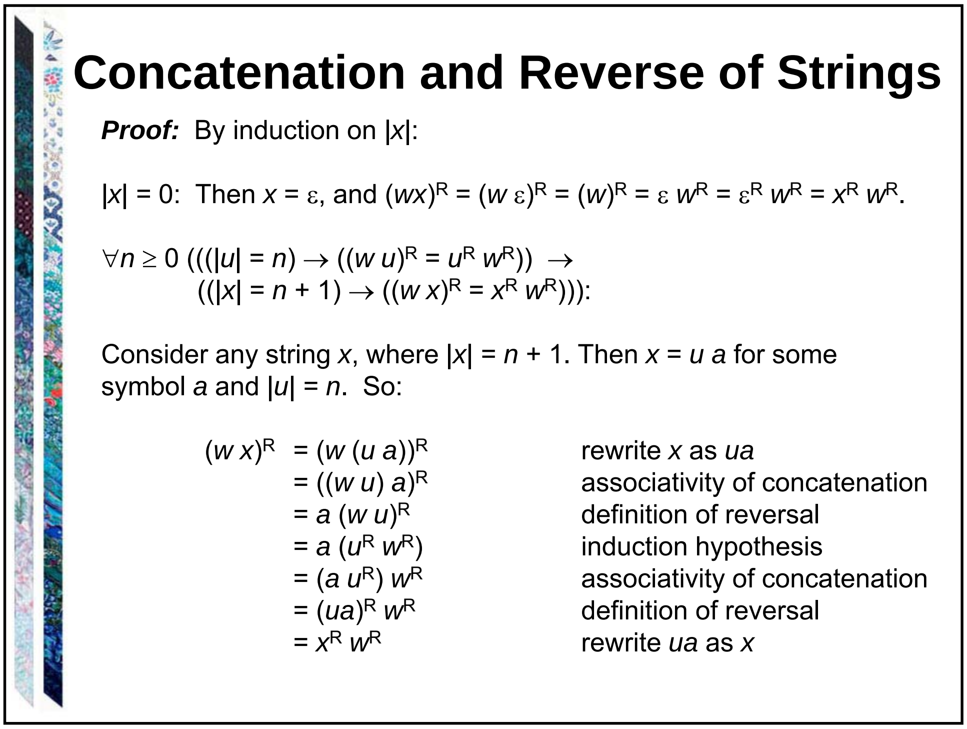
Concatenation and Reverse of Strings

Theorem: If w and x are strings, then $(w\,x)^R = x^R\, w^R$.

Example:

$(\texttt{nametag})^R = (\texttt{tag})^R\, (\texttt{name})^R = \texttt{gateman}$

Proof on next slide



Concatenation and Reverse of Strings

Proof: By induction on $|x|$:

$|x| = 0$: Then $x = \varepsilon$, and $(wx)^R = (w\,\varepsilon)^R = (w)^R = \varepsilon\, w^R = \varepsilon^R\, w^R = x^R\, w^R$.

$\forall n \geq 0\ (((|u| = n) \rightarrow ((w\,u)^R = u^R\, w^R)) \rightarrow$
$((|x| = n + 1) \rightarrow ((w\,x)^R = x^R\, w^R)))$:

Consider any string x, where $|x| = n + 1$. Then $x = u\,a$ for some symbol a and $|u| = n$. So:

$(w\,x)^R$	$= (w\,(u\,a))^R$	rewrite x as ua
	$= ((w\,u)\,a)^R$	associativity of concatenation
	$= a\,(w\,u)^R$	definition of reversal
	$= a\,(u^R\,w^R)$	induction hypothesis
	$= (a\,u^R)\,w^R$	associativity of concatenation
	$= (ua)^R\,w^R$	definition of reversal
	$= x^R\,w^R$	rewrite ua as x