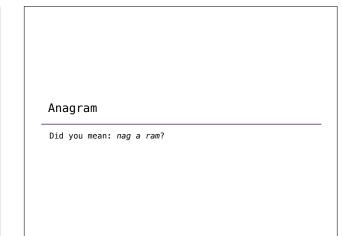
Lecture 24: Logic II

Brian Hou August 2, 2016

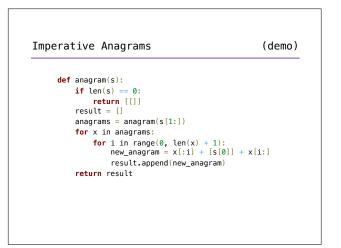
Announcements

- Project 4 is due Friday (8/5)
- Finish through Part II today for 1 EC point
- Homework 9 is due Wednesday (8/3)
- Quiz 9 on Thursday (8/4) at the beginning of lecture
 Will cover Logic
- Final Review on Friday (8/5) from 11-12:30pm in 2050 VLSB
 Final Exam on Friday (8/12) from 5-8pm in 155 Dwinelle
- Ants composition revisions due Saturday (8/6)
- Scheme Recursive Art Contest is open! Submissions due 8/9
- Potluck II on 8/10! 5-8pm (or later) in Wozniak Lounge
 Bring food and board games!

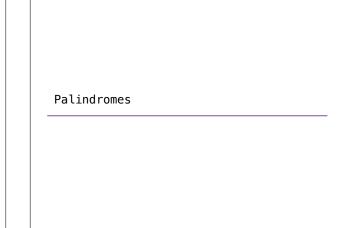
Introduction Functions Data Mutability Objects Interpretation	 This week (Paradigms), the goals are To study examples of paradigms that are very different from what we have seen so far To expand our definition of what counts as programming
Paradigms	

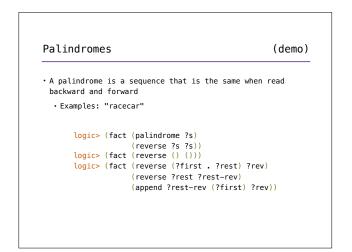


Anagrams			
Anayrallis			
			c at
		at	a <mark>c</mark> t
			at <mark>c</mark>
cat	at		
			<mark>c</mark> ta
		ta	t <mark>c</mark> a
			ta <mark>c</mark>



Declarative Anagrams	(demo)
<pre>logic> (fact (insert ?a ?r (?a . ?r)))</pre>	
logic> (fact (insert ?a (?b . ?r) (?b .	?s))
(insert ?a ?r ?s))	
<pre>logic> (fact (anagram () ()))</pre>	
<pre>logic> (fact (anagram (?a . ?r) ?b)</pre>	
(anagram ?r ?s)	
(insert ?a ?s ?b))	
<pre>logic> (query (anagram ?s (s t a r)))</pre>	

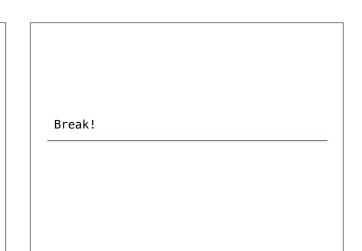




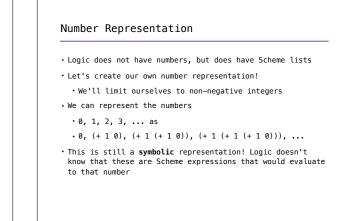
Declarative Programming

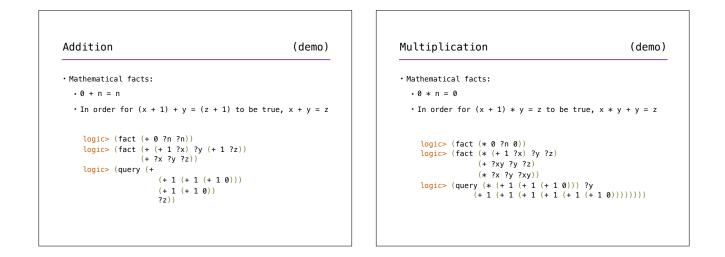
- In declarative programming, we tell the computer what a solution looks like, rather than how to get the solution
- If we describe a solution in two different ways, will the computer take the same amount of time to compute a solution?
 - Probably not...

verse			(demo
logic>	(fact	(reverse () ()))	
logic>	(fact	<pre>(reverse (?first . ?rest) ?rev)</pre>	
		(reverse ?rest ?rest-rev)	
		(append ?rest-rev (?first) ?rev)	,
logic>	(fact	(accrev (?first . ?rest) ?acc ?re	
logics	(fact	<pre>(accrev ?rest (?first . ?acc) ?re (accrev () ?acc ?acc))</pre>	ev))
		(accrev ?s ?rev)	
cogre.	((accrev ?s () ?rev))	



Arithmetic





Subtraction and Division (c	emo) Arithmetic	(demo)	
• Mathematical facts:	• We've implemented the	he four basic arithmetic operations!	
 Subtraction is the inverse of addition 	• We can now ask Logi	• We can now ask Logic about all the different ways to	
• In order for $x - y = z$, $y + z = x$	compute the number (6	
• Division is the inverse of multiplication			
 In order for x / y = z, y * z = x (assuming x is divisible by y) 			
<pre>logic> (fact (- ?x ?y ?z)</pre>		<pre>logic> (query (?op ?arg1 ?arg2</pre>	

Summary

- Some problems can be solved more easily or concisely with declarative programming than imperative programming
- However, just because the computer is the one solving the problem doesn't mean that we can write any declarative program and it will "just work"
- As declarative programmers, we (eventually) should understand how the underlying problem solver works
- This semester, just focus on writing declarative programs; no need to worry about the underlying solver yet!