2

W PAUL G. ALLEN SCHOOL of computer science & engineering

CSE341: Programming Languages

Lecture 4 Records, Datatypes, Case Expressions

Brett Wortzman Summer 2019

Slides originally created by Dan Grossman

Five different things

- 1. Syntax: How do you write language constructs?
- 2. Semantics: What do programs mean? (Evaluation rules)
- 3. Idioms: What are typical patterns for using language features to express your computation?
- Libraries: What facilities does the language (or a well-known project) provide "standard"? (E.g., file access, data structures)
- 5. Tools: What do language implementations provide to make your job easier? (E.g., REPL, debugger, code formatter, ...)
 - Not actually part of the language

These are 5 separate issues

- In practice, all are essential for good programmers
- Many people confuse them, but shouldn't

Spring 2019 CSE341: Programming Languages

Our Focus

This course focuses on semantics and idioms

- Syntax is usually uninteresting
 - A fact to learn, like "The American Civil War ended in 1865"
 People obsess over subjective preferences
- Libraries and tools crucial, but often learn new ones "on the job"
 We are learning semantics and how to use that knowledge to understand all software and employ appropriate idioms
 - By avoiding most libraries/tools, our languages may look "silly" but so would any language used this way

Spring 2019

CSE341: Programming Languages

3

How to build bigger types

Already know:

Spring 2019

- Have various base types like int bool unit char
- Ways to build (nested) compound types: tuples, lists, options
- · Coming soon: more ways to build compound types
- First: 3 most important type building blocks in any language

 "Each of": A t value contains values of each of t1 t2 ... tn
 "One of": A t value contains values of one of t1 t2 ... tn
 "Self reference": A t value can refer to other t values Remarkable: A lot of data can be described with just these building blocks

 Note: These are not the common names for these concepts

Spring 2019 CSE341: Programming Languages

Examples Tuples build each-of types int * bool contains an int and a bool Options build one-of types int option contains an int or it contains no data Lists use all three building blocks int list contains an int and another int list or it

contains no data

Spring 2019

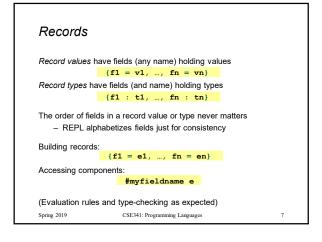
And of course we can nest compound types
 - ((int * int) option * (int list list)) option

CSE341: Programming Languages

Rest of this Lecture Another way to build each-of types in ML Records: have named fields Connection to tuples and idea of syntactic sugar A way to build and use our own one-of types in ML For example, a type that contains an int or a string Will lead to pattern-matching, one of ML's coolest and strangest-to-Java-programmers features Later in course: How OOP does one-of types Key contrast with procedural and functional programming

CSE341: Programming Languages

1



Example				
{nam	e = "Matai	", $id = 4$ ·	- 3}	
Evaluates to				
{i	d = 1, na	me = "Matai	."}	
And has type				
	{id : int	, name : st	ring}	
If some expression	on such as a	variable x has	this type,	then get
noido with.	TU X	Tidlie A		
	program cou	re any record ty ld also make a se} of type {a		,ego:bool}
Spring 2019	CSE341: P	rogramming Languages	5	8

