<i>emputational</i>	Week 11 - Laziness Overview
Functional Programming WS 2007/08 Christian Sternagel ¹ (VO + PS) Friedrich Neurauter ² (PS) Harald Zankl ³ (PS)	Week 11 - Laziness Summary of Week 10 Lazyness
Computational Logic Institute of Computer Science University of Innsbruck 18 January 2008	
¹ christian.sternagelQuibk.ac.at ² friedrich.neurauterQuibk.ac.at ³ harald.zanklQuibk.ac.at ICSQUIBK) FP OCaml Bash	CS (ICS@UIBK) FP
ek 11 - Laziness Summary of Week 10	Week 11 - Laziness
Overview	Type Checking
Week 11 - Laziness	
Summary of Week 10	prove that some expression really has a given type w.r.

Summary of Week 10

- environment Formally: $E \vdash e : \tau$
- use the inference rules of C to do so

FP

OCaml Bash

Summary of Week 10

Week 11 - Laziness

Lazyness

Summary of Week 10

- ▶ get the most general type for an expression w.r.t. an environment
- formally: $E \triangleright e : \tau$
- ► task is split into two parts:
 - 1. transform given type inference problem into a unification problem
 - 2. solve the unification problem (result is substitution)

CS (ICS@UIBK)	FP	OCaml Bash
Week 11 - Laziness		Lazyness
Lazyness in OCaml		

Keyword lazy

used to transform arbitrary expression into lazy expression

Example

- ▶ let e0 = lazy (Format.printf "test\n");;
- let e1 = lazy (let rec $f x = print_i x; f(x + 1)$ in f 0)

FP

Function Lazy.force

used to evaluate lazy expressions

Example

- ► Lazy.force e0;;
- ► Lazy.force e1;;

OCaml Bash CS (ICS@UIBK) Week 11 - Laziness Example - Lazy Lists

Live Demonstration

FP