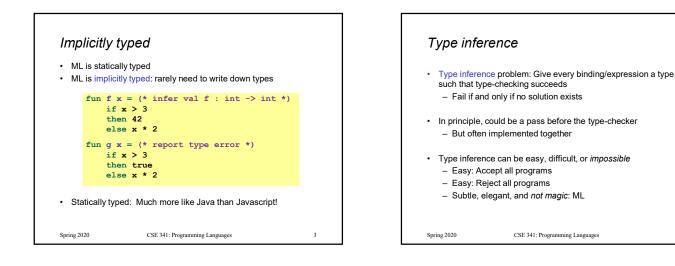
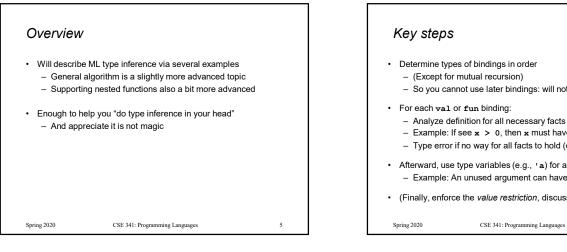
## PAUL G. ALLEN SCHOOL of computer science & engineering Type-checking (Static) type-checking can reject a program before it runs to prevent the possibility of some errors A feature of statically typed languages CSE341: Programming Languages Dynamically typed languages do little (none?) such checking - So might try to treat a number as a function at run-time Lecture 11 • Will study relative advantages after some Racket Type Inference - Racket, Ruby (and Python, Javascript, ...) dynamically typed • ML (and Java, C#, Scala, C, C++) is statically typed Brett Wortzman - Every binding has one type, determined "at compile-time" Spring 2020 Spring 2020 CSE 341: Programming Languages 2





Key steps
<ul> <li>Determine types of bindings in order</li> <li>(Except for mutual recursion)</li> <li>So you cannot use later bindings: will not type-check</li> </ul>
<ul> <li>For each val or fun binding:</li> <li>Analyze definition for all necessary facts (constraints)</li> <li>Example: If see x &gt; 0, then x must have type int</li> <li>Type error if no way for all facts to hold (over-constrained)</li> </ul>
<ul> <li>Afterward, use type variables (e.g., 'a) for any unconstrained types</li> <li>Example: An unused argument can have any type</li> </ul>
• (Finally, enforce the <i>value restriction</i> , discussed later)

