DEPARTMENT OF COMPUTER SCIENCES

#### **Distributed Execution**

William Cook and Jayadev Misra

Department of Computer Science University of Texas at Austin

Email: {cook,misra}@cs.utexas.edu
web: http://www.cs.utexas.edu/users/psp

DEPARTMENT OF COMPUTER SCIENCES

# **Centralized Execution Model**

• semantics (and execution) described for one machine.

• Then,

- immediate sites respond instantaneously.
- Rtimer is exact.
- We develop a theory to permit distributed execution.

# **Distributed Execution Model**

- Assign subexpressions to different machines.
- Execution starts on a goal machine.
- A machine requests another machine to start evaluation.
- The calling machine supplies the context and values of variables as they become defined.
- The calling machine orders all tokens to be killed, and asks for ack.
- All messages among these machines are delivered after arbitrary but finite delay.

#### When can we distribute?

•  $Rtimer(1) \gg let(0) \mid Rtimer(2) \gg let(1)$ 

Neither branch can be executed on another machine.

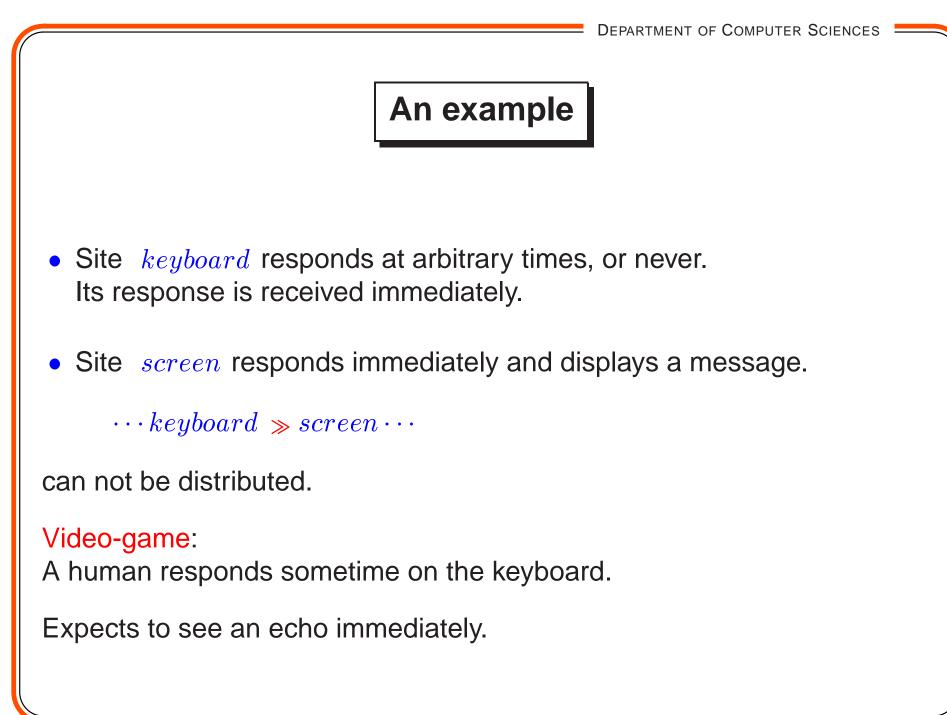
*c.get* and *e.put* are on different machines. P(c, e) and P(d, e) can be evaluated on different machines.

•  $Rtimer(1) \gg M \mid Rtimer(2) \gg N$ 

The two branches can be executed on different machines.

The theory identifies subexpressions which can be distributed.





## Punctual Site

- A site is punctual if communication delay with it is zero.
- A site is unpunctual if communication delay with it is arbitrary.
- Simplification: Assume all sites are either punctual or unpunctual.
- A punctual site has to be implemented on the caller's machine.
- An unpunctual site may be implemented on a remote machine.

Punctual: *Rtimer*, *let*, *if*, ..., all immediate sites, ..., *keyboard*, *screen* 

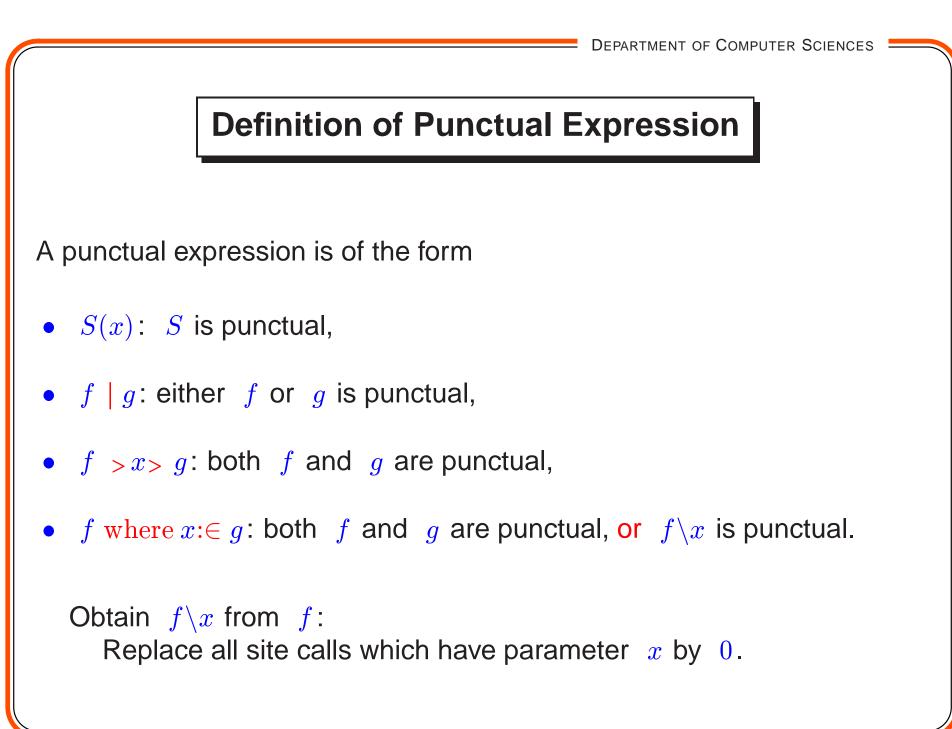
**Unpunctual:** 0, CNN, PostWeb,  $\cdots$  Generic symbol M

DEPARTMENT OF COMPUTER SCIENCES

# **Punctual Expression**

- Any delay in starting it or processing its response can be detected.
- Some timing information for site calls or publications is known.
- A punctual expression has to be implemented on the caller's machine.
- An unpunctual expression may be implemented on another machine.
- A punctual expression may have unpunctual subexpressions.

The subexpression may be implemented on another machine.



### **Examples: Punctual Expressions**

- let(x)let(x) > x > Rtimer(x)
- let(x) where  $x \in Rtimer(1)$
- let(x) > x > Rtimer(1)where  $x \in Rtimer(1)$

*let* is punctual (immediate) both sites are punctual

Rtimer(1) where  $x \in g$   $f \setminus x = Rtimer(1)$  is punctual

let(x) and Rtimer(1) are both punctual

 $Rtimer(1) \mid h \text{ where } x \in g$   $f \setminus x = Rtimer(1) \mid h \setminus x \text{ is punctual}$ 

let(x) > x > Rtimer(1) and Rtimer(1)are both punctual



### **Examples: Unpunctual Expressions**

 $M \gg Rtimer(1)$ 

 $Rtimer(1) \gg M$ 

N(x) > x > Rtimer(x)

Rtimer(x) where  $x \in M$ 

N(x) where  $x \in Rtimer(1)$ 

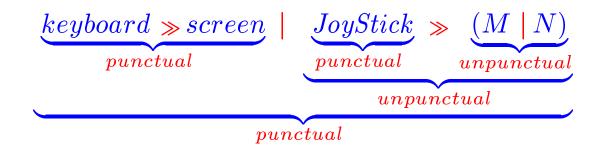
let(x) where  $x \in M$ 

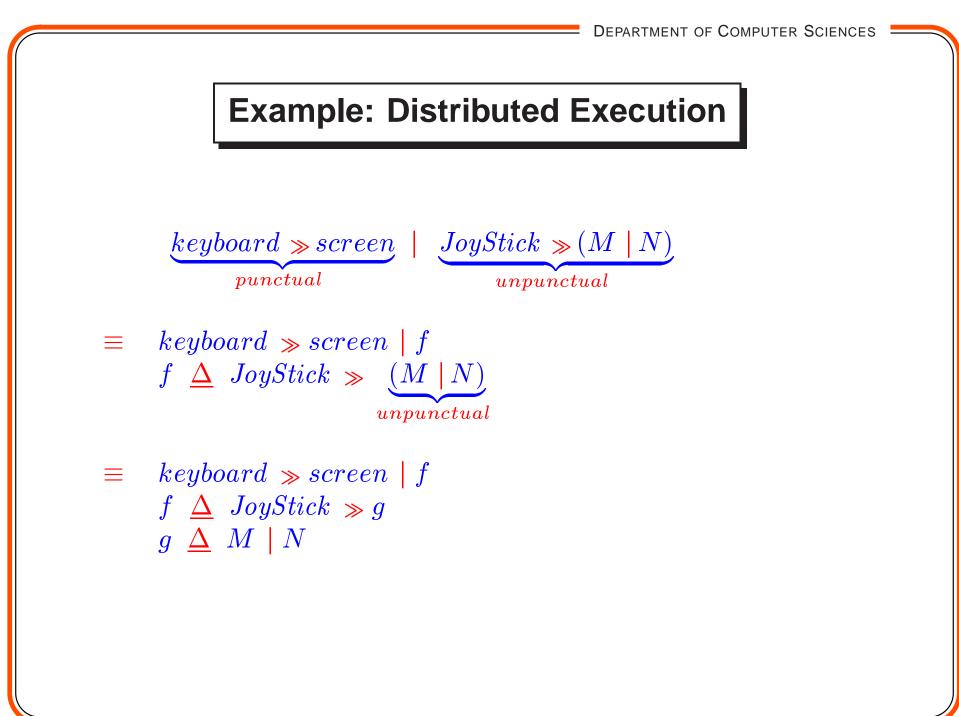




```
keyboard \gg screen | JoyStick \gg (M \mid N), where
```

*keyboard*, *screen*, *JoyStick* are punctual M and N are unpunctual.





### Some properties

- If all sites in an expression are punctual, the expression is punctual.
- If all sites in an expression are unpunctual, the expression is unpunctual.
- Monotonicity: An unpunctual expression remains unpunctual if you replace any site by an unpunctual site.

M punctual, f(M) unpunctual, N unpunctual  $\Rightarrow f(N)$  unpunctual.

 $M \sqsubseteq N \Rightarrow f(M) \sqsubseteq f(N)$ , where punctual  $\Box$  unpunctual

Proofs by structural induction.

## Punctuality is conservative

 $>b> \qquad \begin{array}{l} let(\textit{true}) \\ (if(b) \gg g \\ |if(\neg b) \gg Rtimer(1) \end{array} \equiv \qquad g \\ \end{array}$ 

Though the execution can be distributed, the expression is punctual.

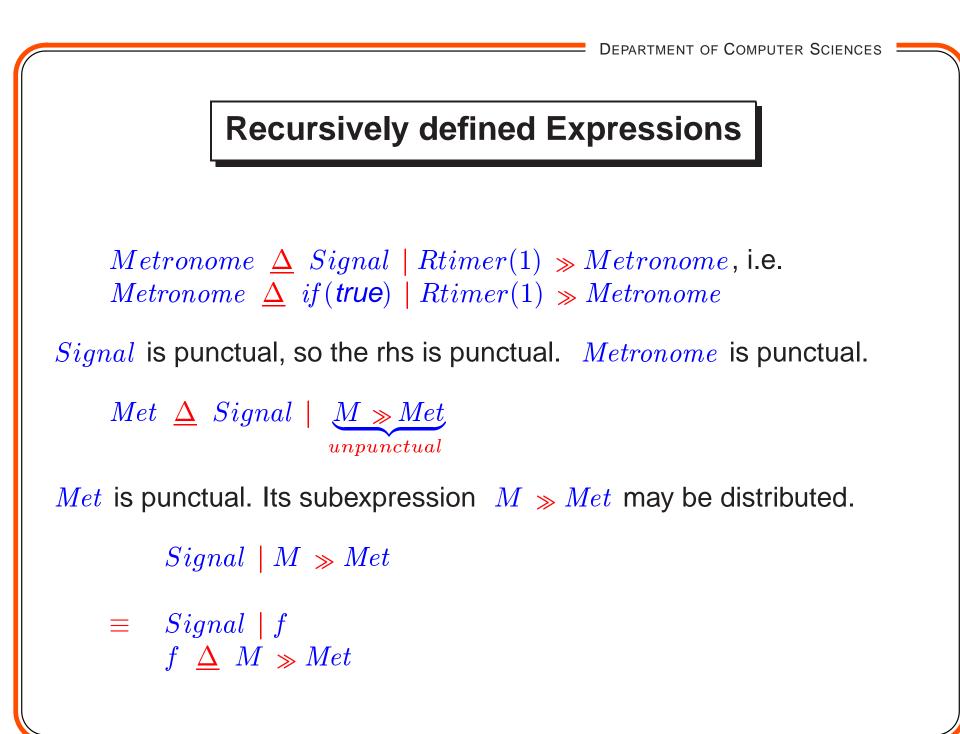
#### • Random > x > Rtimer(x)

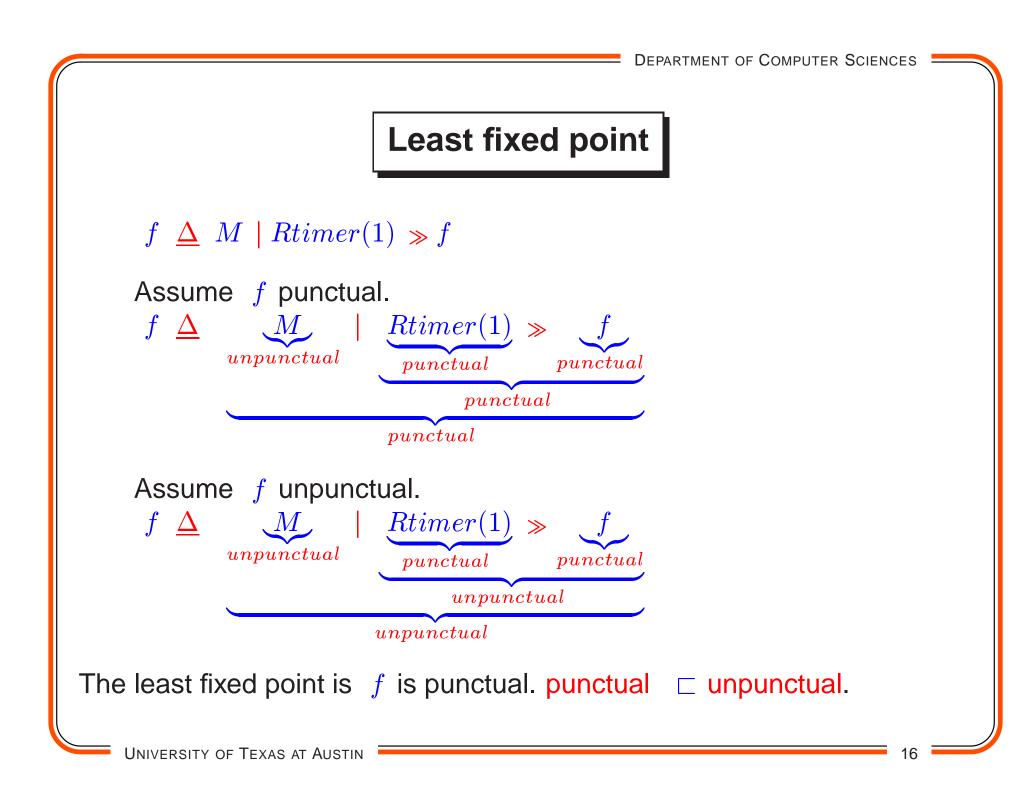
*Random* publishes a random natural number within some bounds. The expression is punctual, and it can not be distributed.

• Nat >x> Rtimer(x)

*Nat* publishes any natural number.

Though the execution can be distributed, the expression is punctual.







#### Existence of least fixed point

Least fixed point exists for any defined expression f.

Assume f is punctual.

- definition of f is punctual: f is punctual.
- definition of f is unpunctual: f is unpunctual.

Monotonicity: assuming f unpunctual, definition of f is unpunctual.