

---

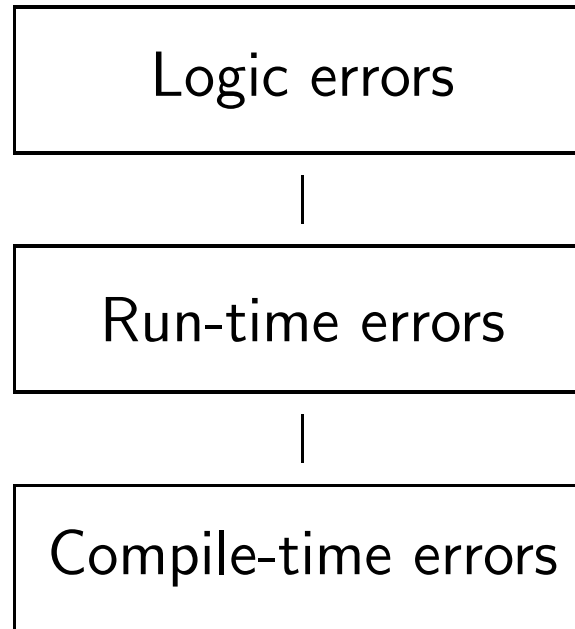
# **Ada for Software Engineers** **(Second Edition with Ada 2005)**

**Mordechai (Moti) Ben-Ari**

**Copyright Springer-Verlag London Limited, 2009**

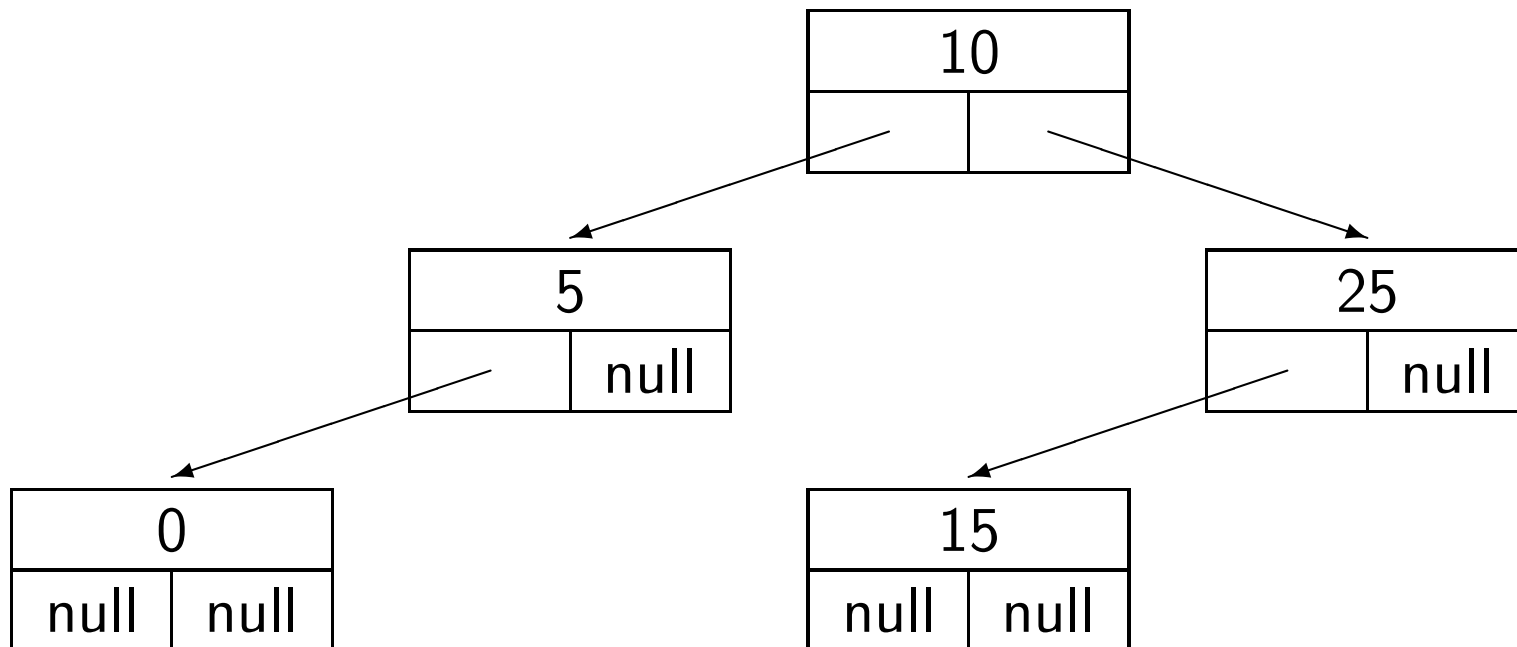
# Errors

---

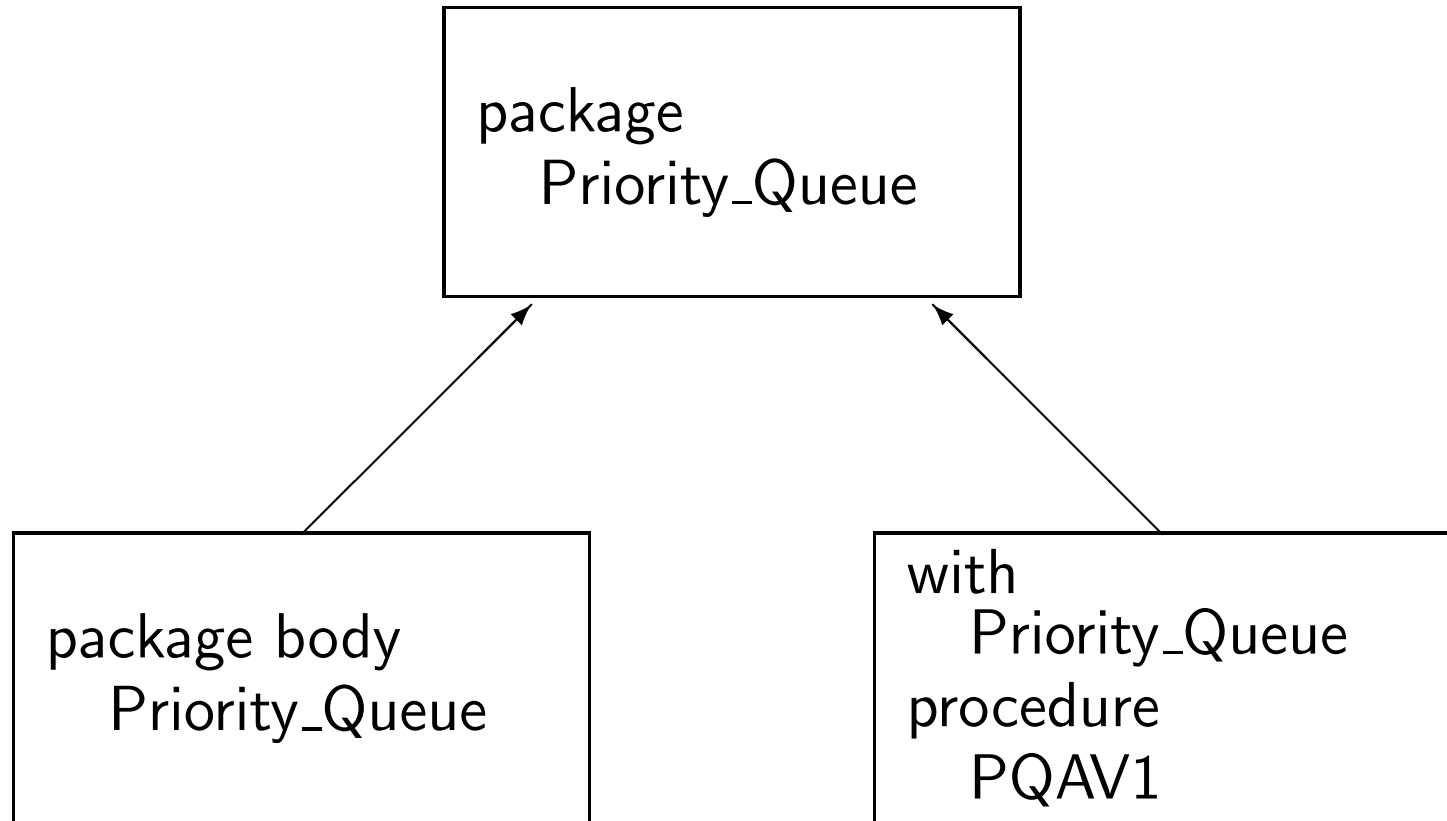


# Priority Queues

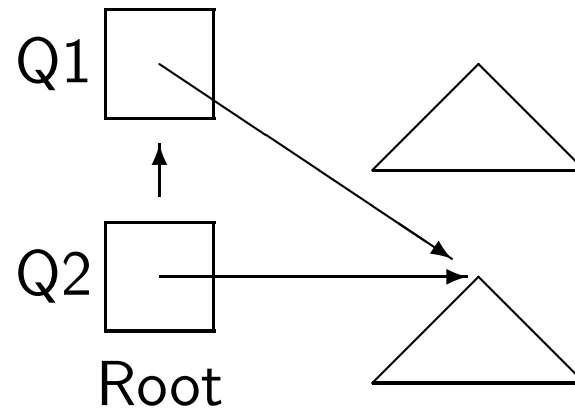
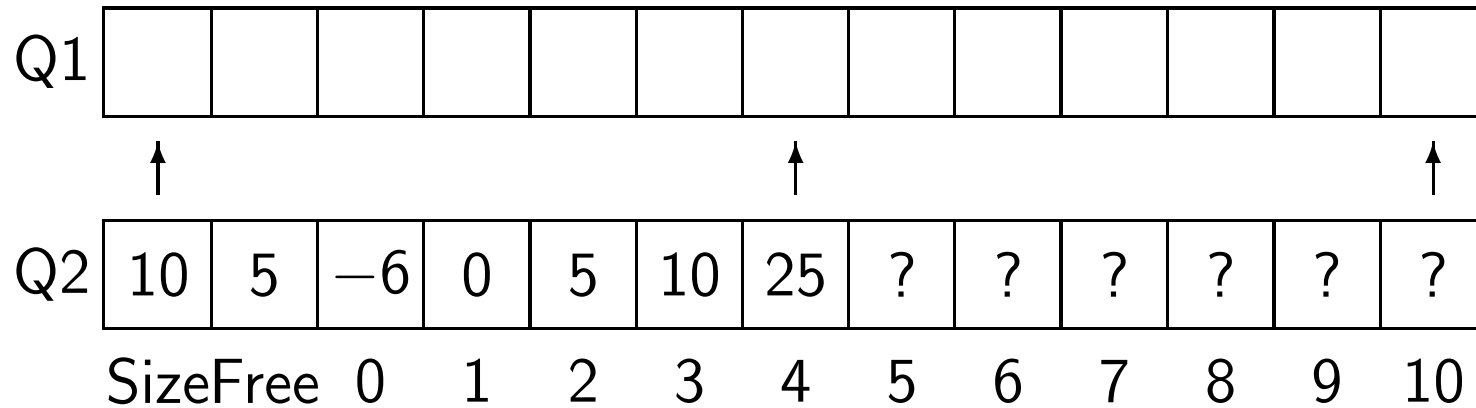
10	5	-6	0	5	10	25	?	?	?	?	?	?
Size	Free	0	1	2	3	4	5	6	7	8	9	10



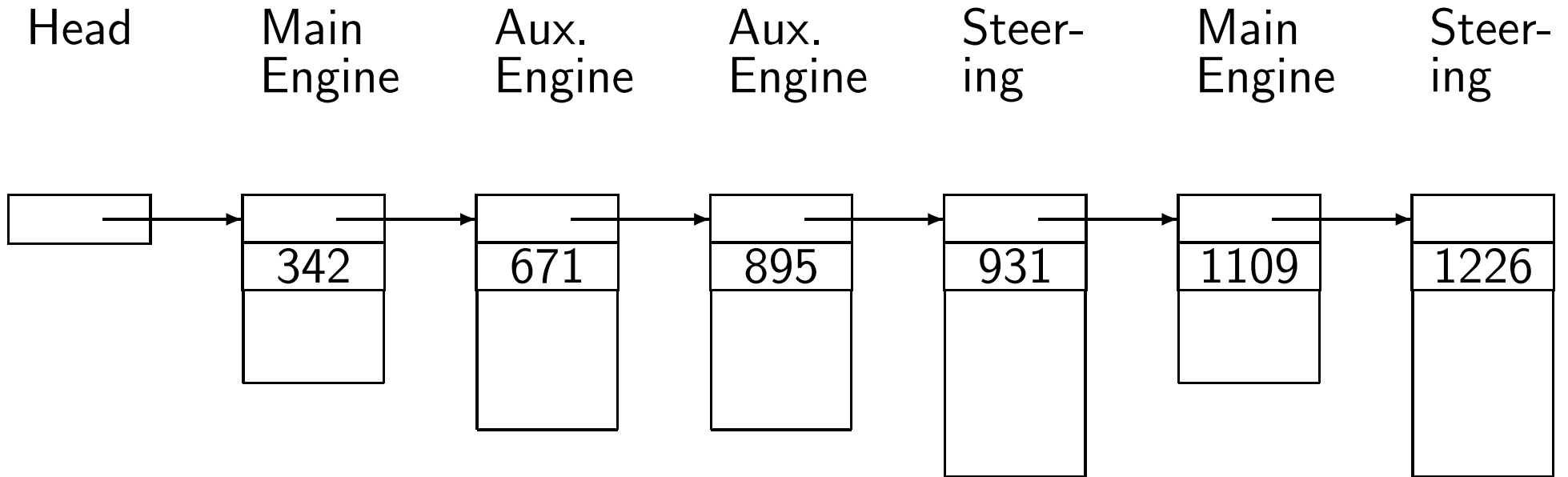
# Semantic Dependences



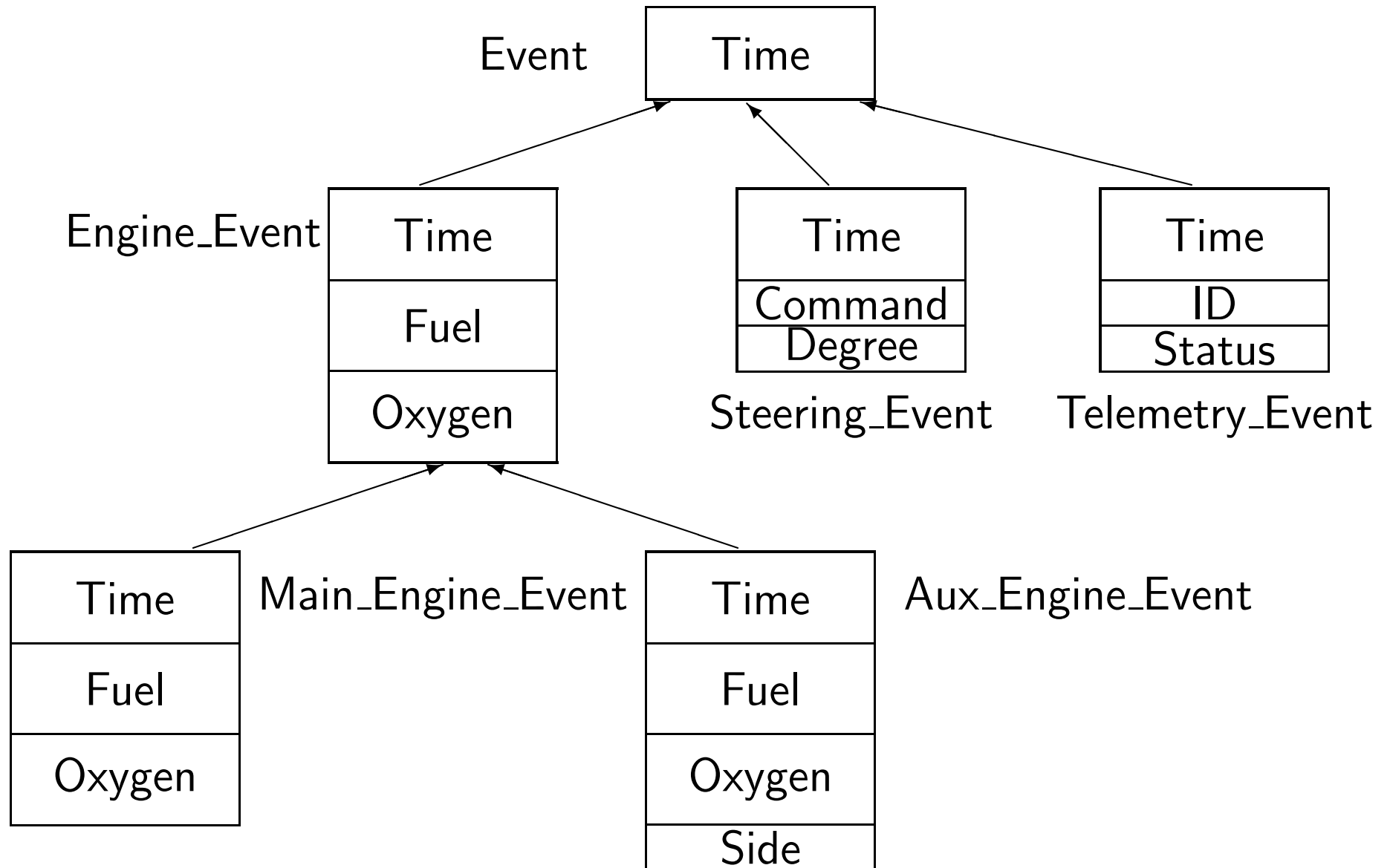
# Copying a Data Structure



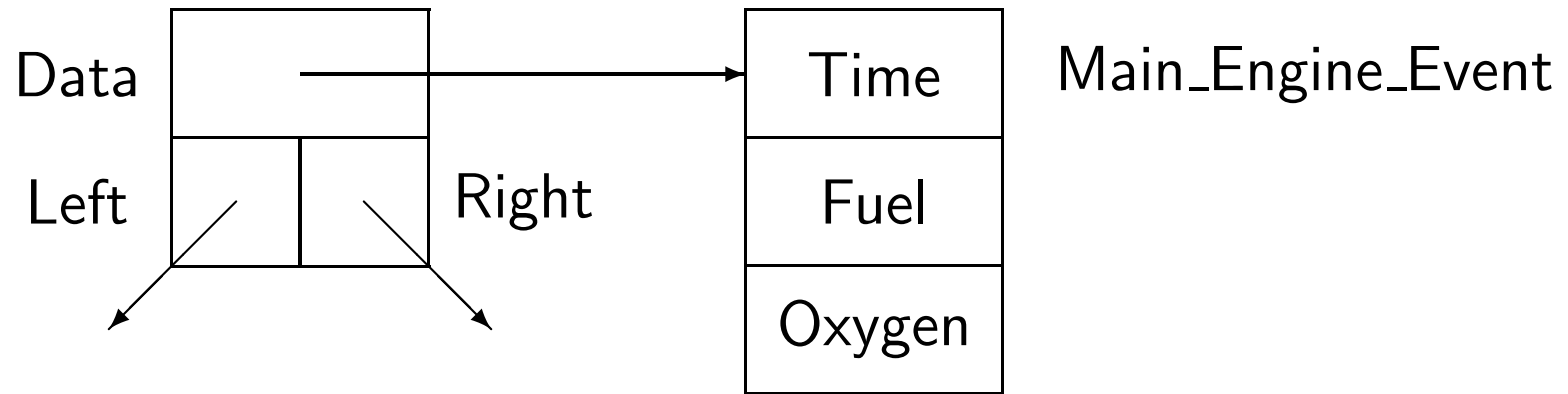
# Heterogeneous Data Structure



# Derivation Class of Types



# Indirect Data Structure

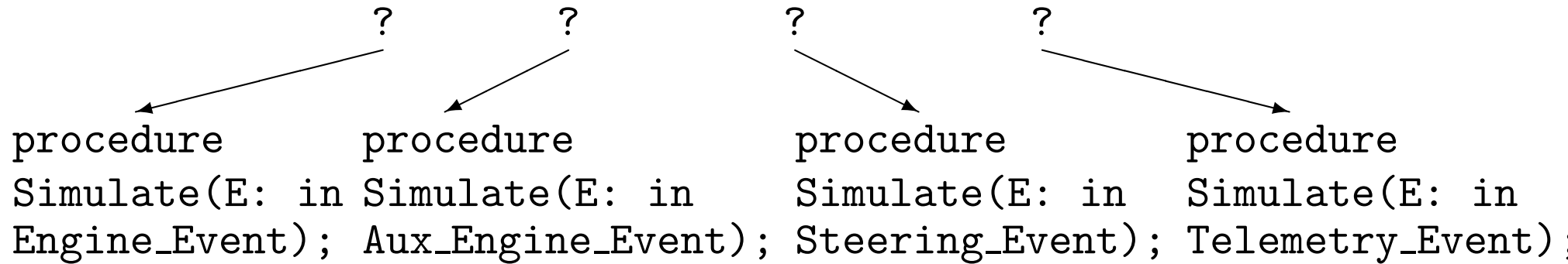




# Dispatching

---

Root\_Event.Simulate(*actual parameter of type Event*'Class);



# Child Packages

---

```
package  
Root_Event
```

```
package  
Root_Event.  
Engine
```

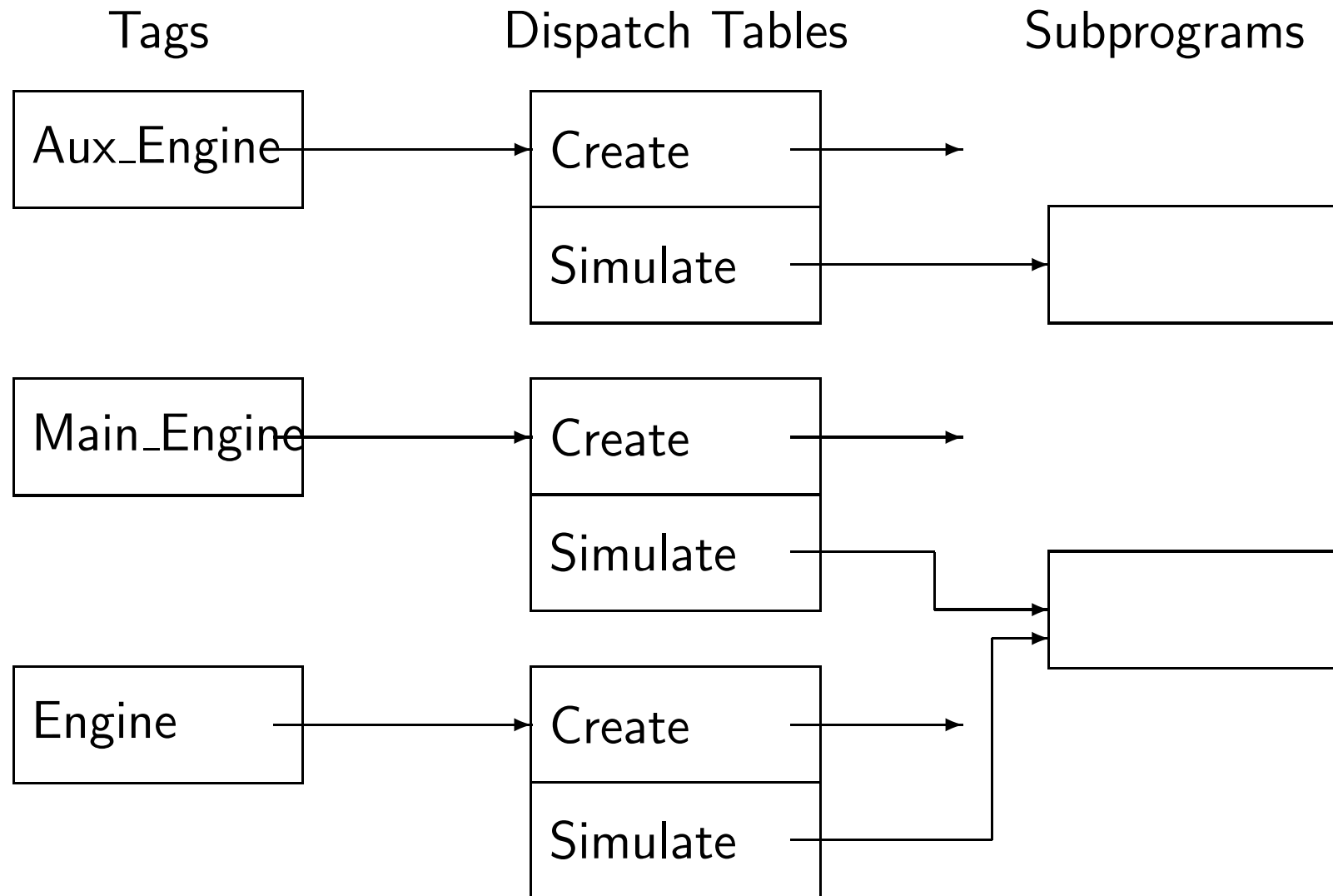
```
package  
Root_Event.  
Steering
```

```
package  
Root_Event.  
Telemetry
```

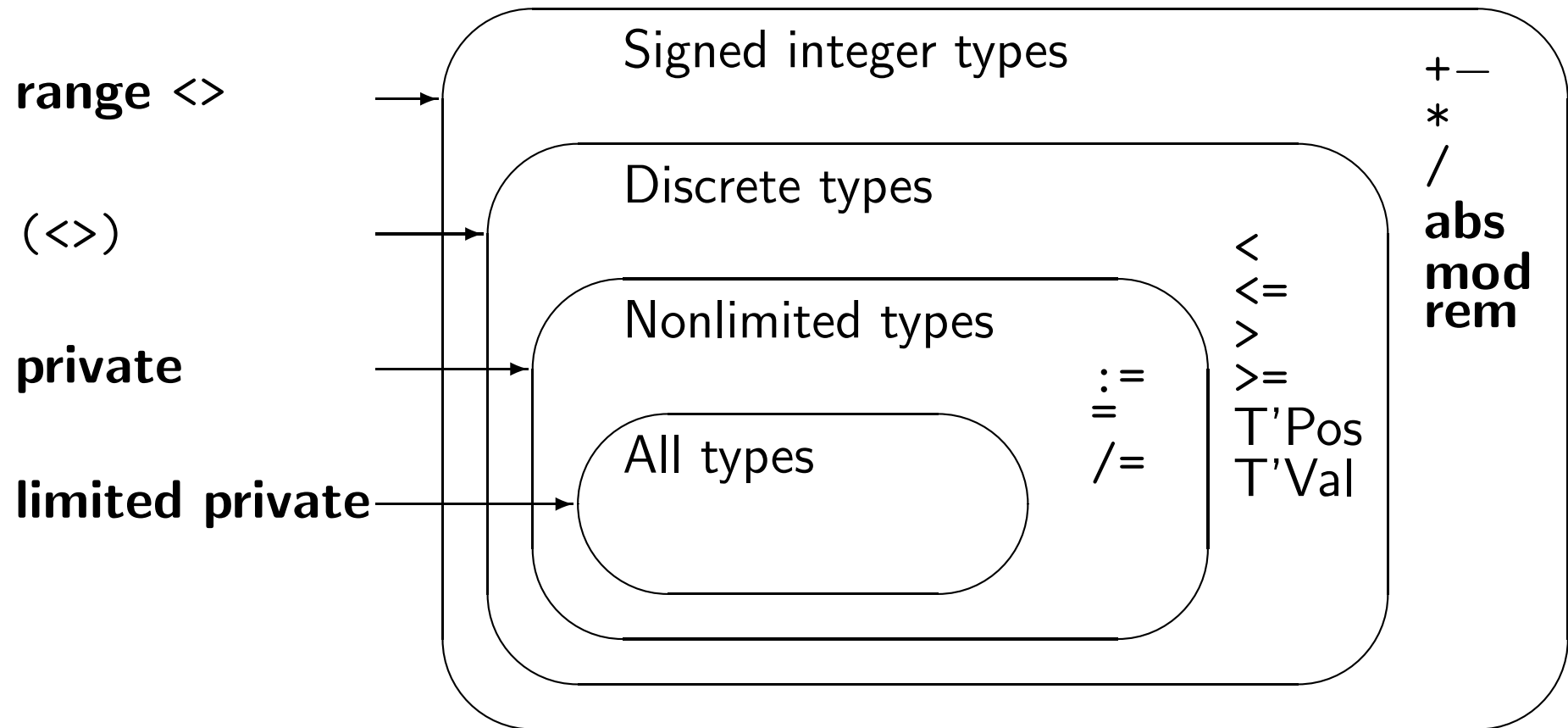
```
private package  
Root_Event.  
Random_Time
```

```
package body  
Root_Event
```

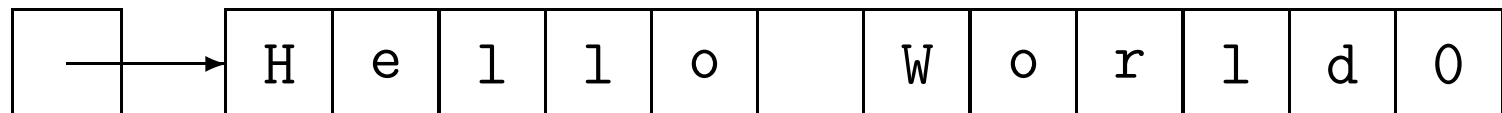
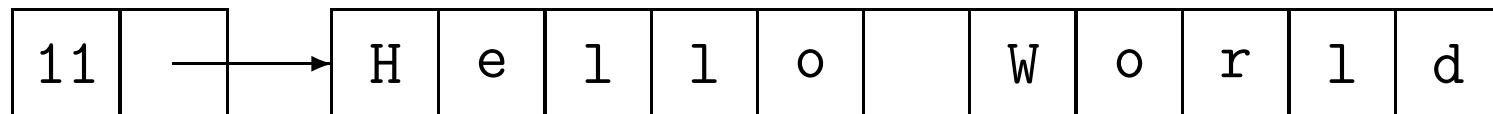
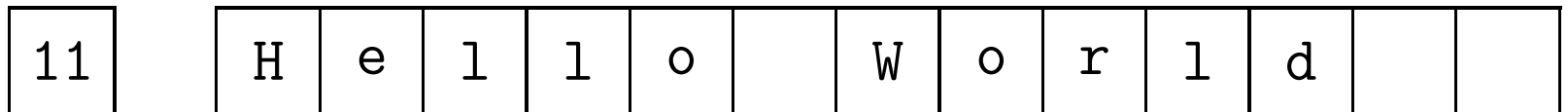
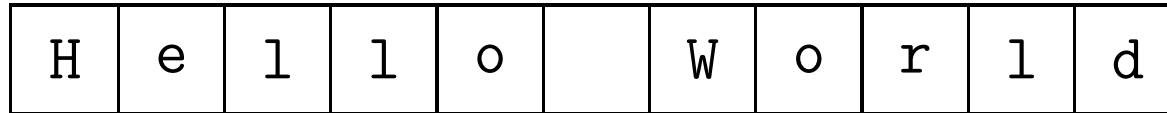
# Implementation of Dispatching



# Generic Contracts



# String Representations



# Cursor

---



First



Position



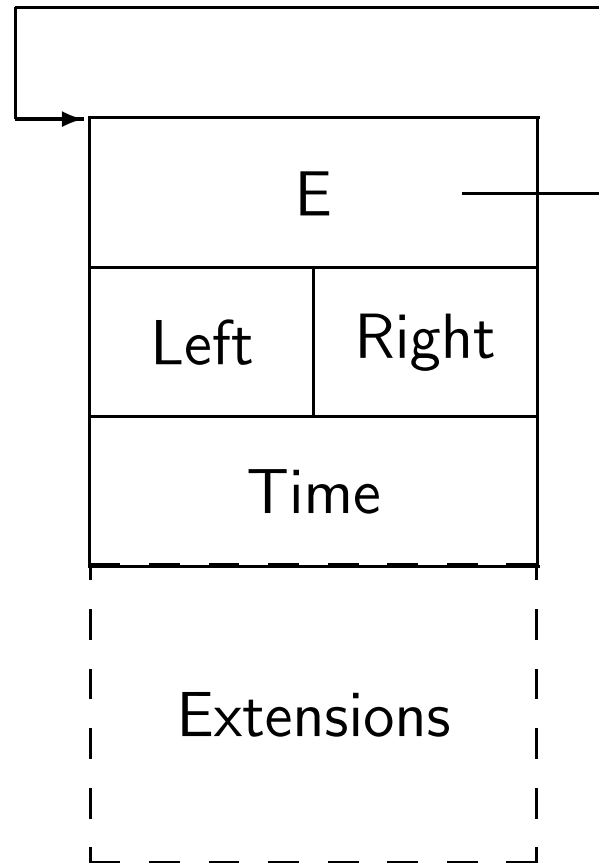
Last

# Discriminant

---

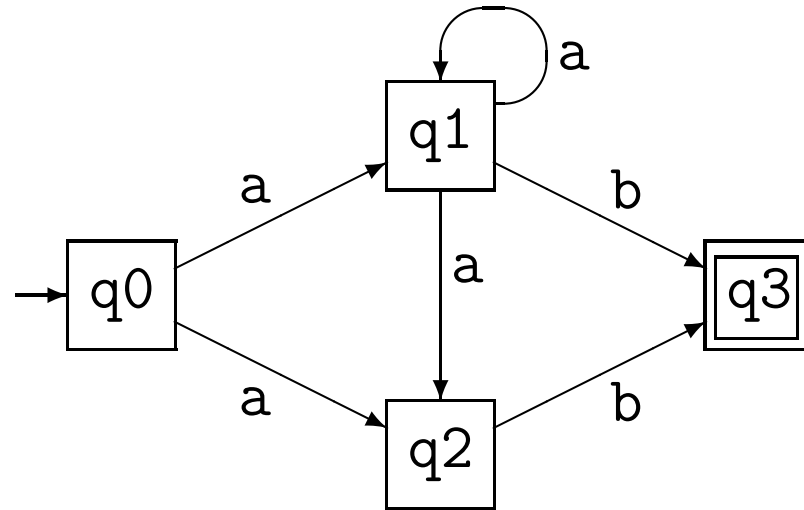
80	A	X	W	...	R	?	...	?
Length	1	2	3		80	81		255

# Access Discriminant

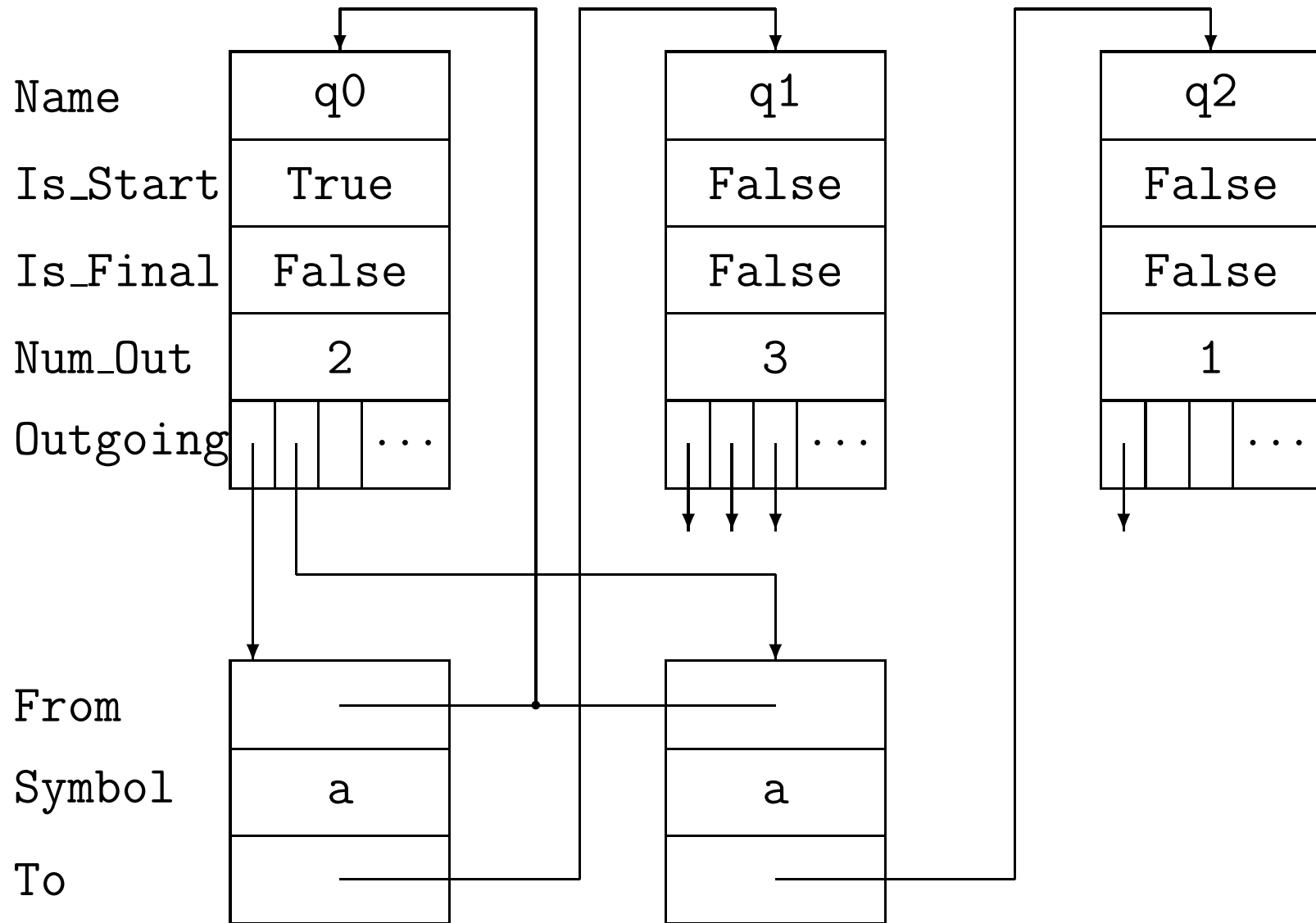




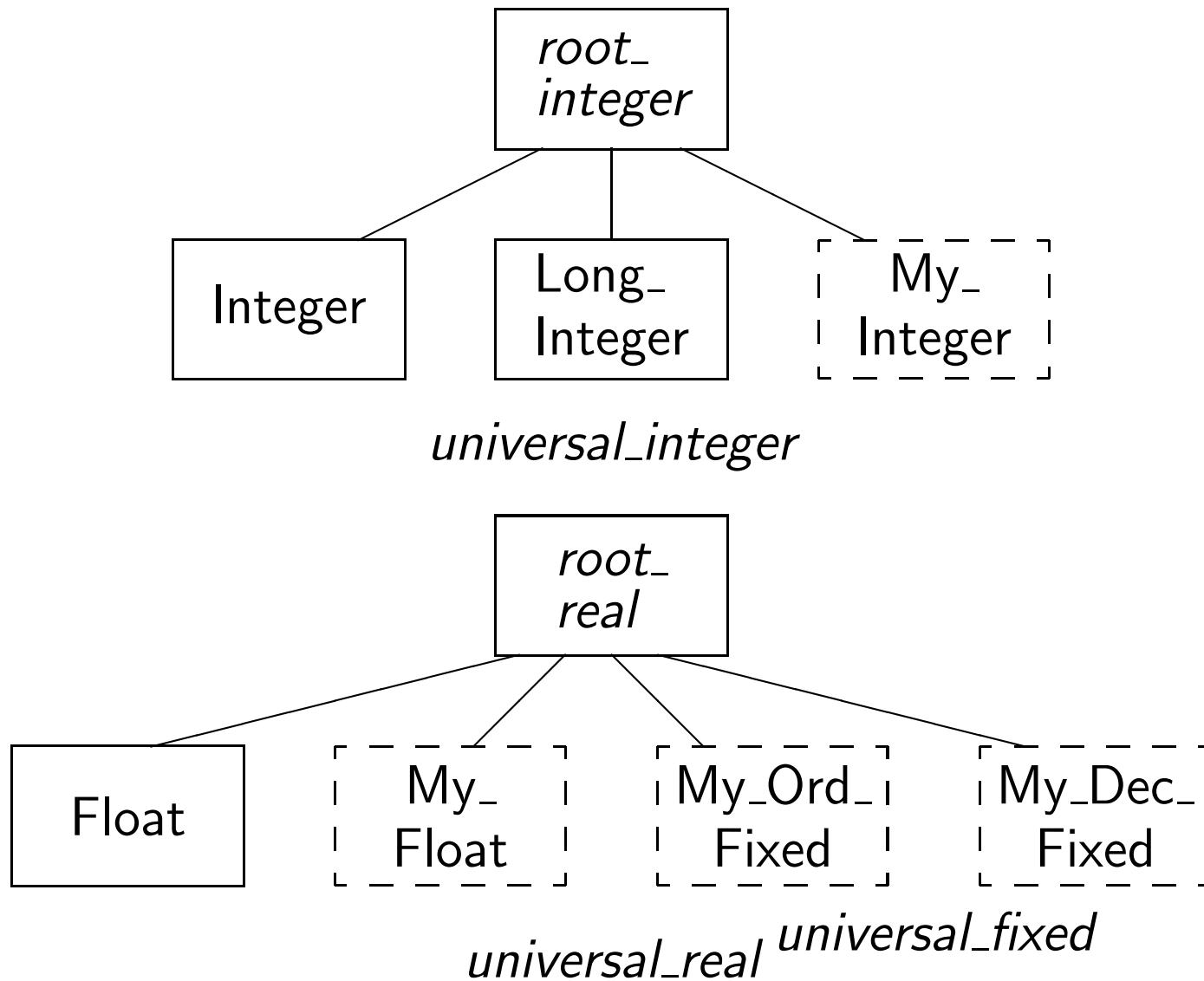
# Nondeterministic Finite Automaton



# Data Structure for NDFA

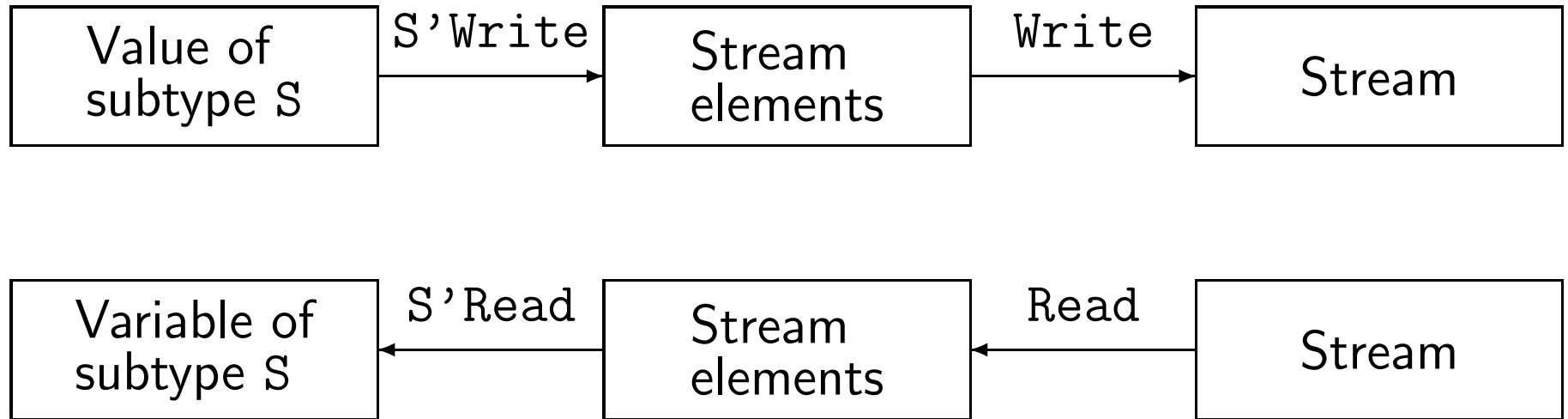


# Derivation Classes of Numeric Types

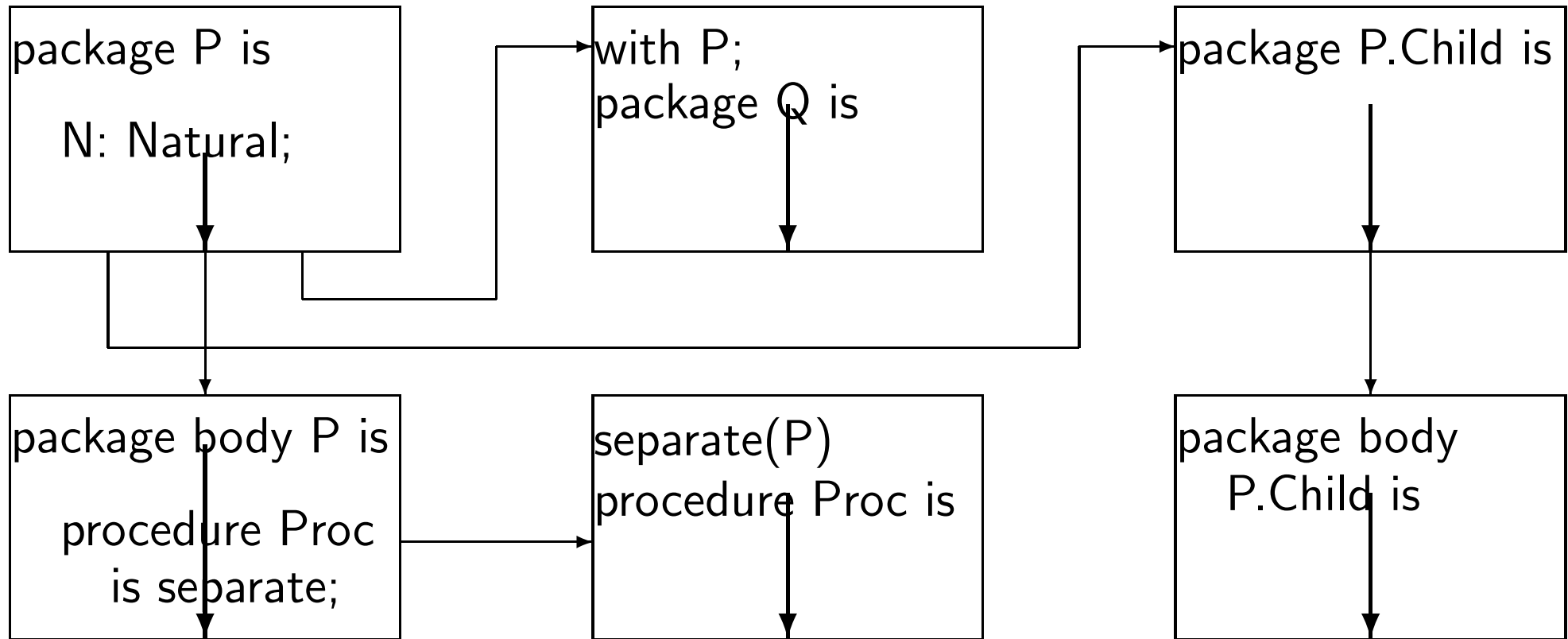


# Streams

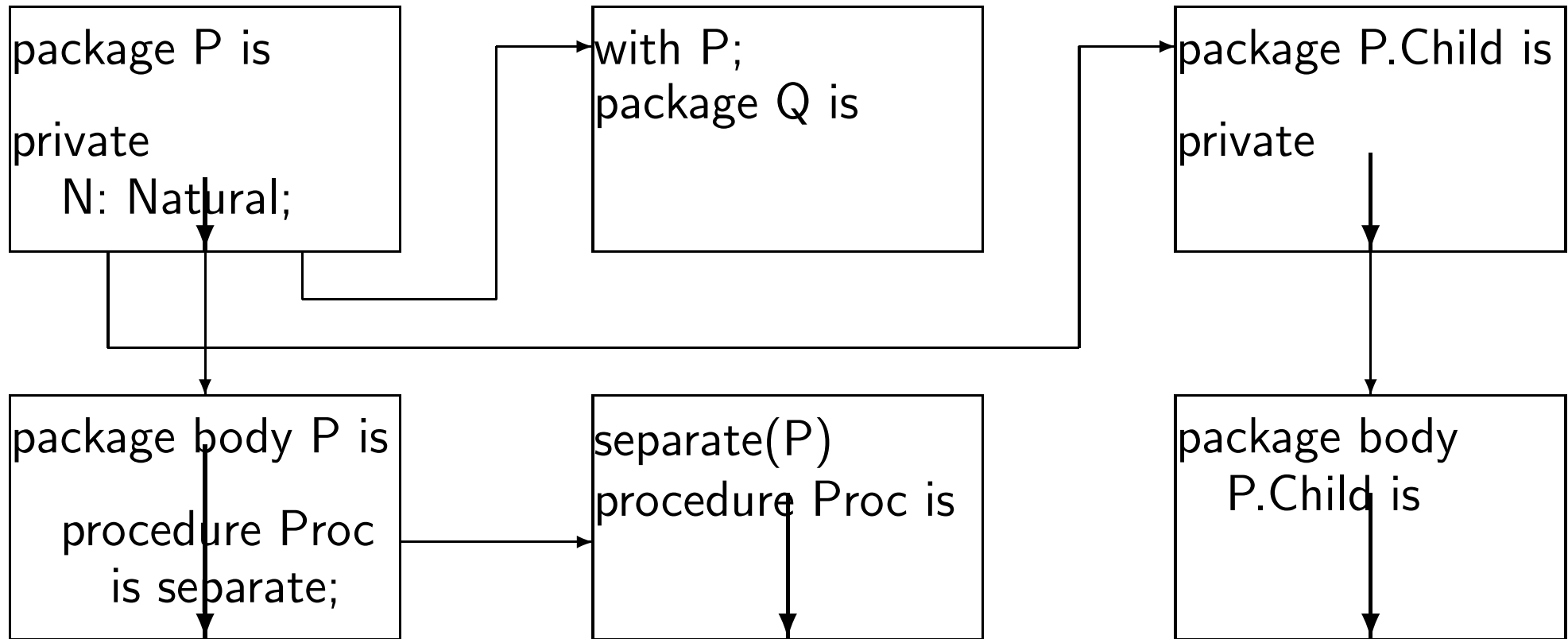
---



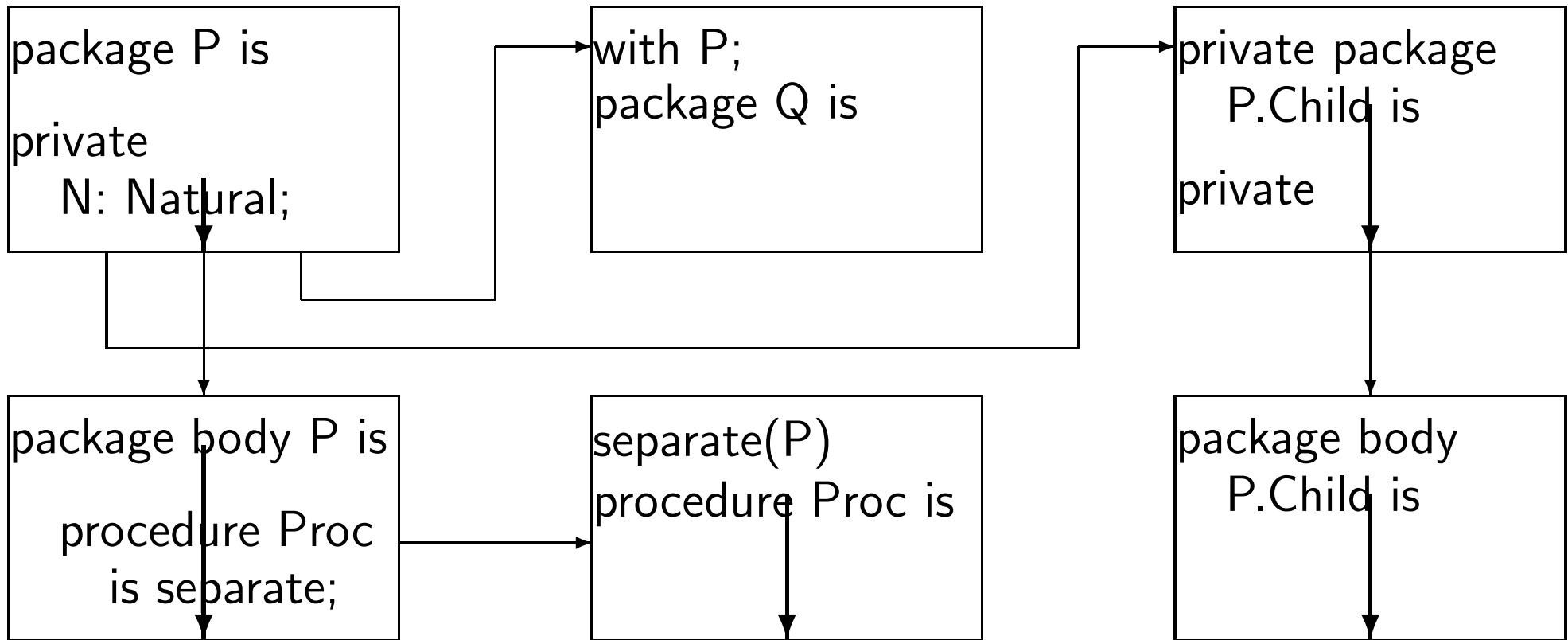
# Declaration in the Visible Part of a Package



# Declaration in the Private Part of a Package

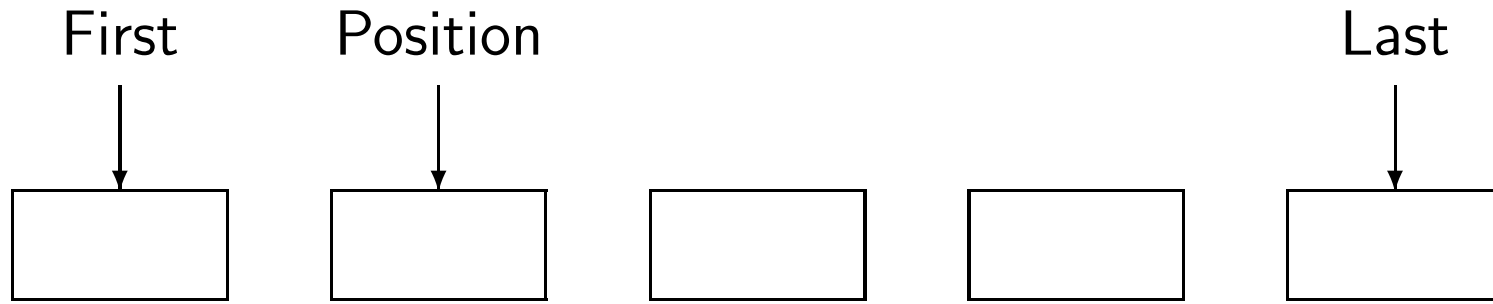


# Private Packages



# Cursor

---



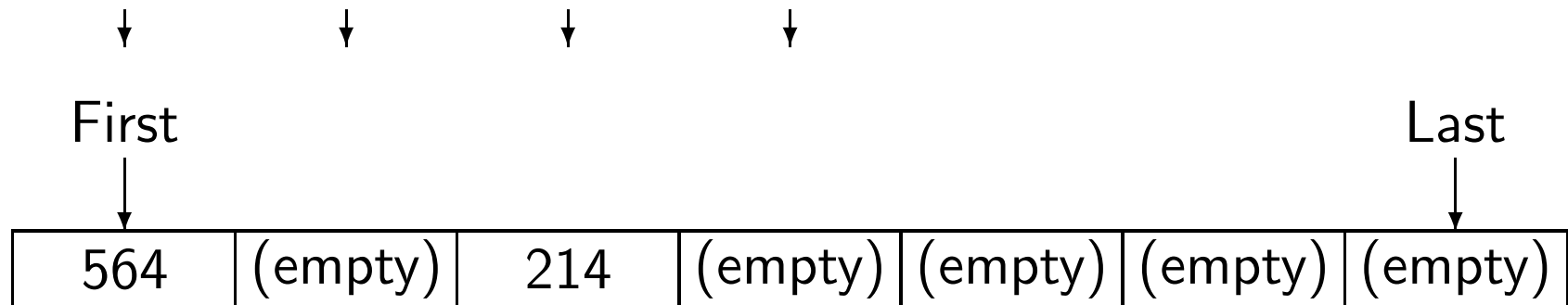
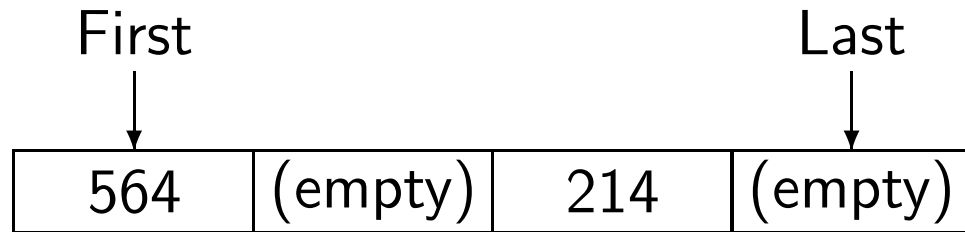


# Vector

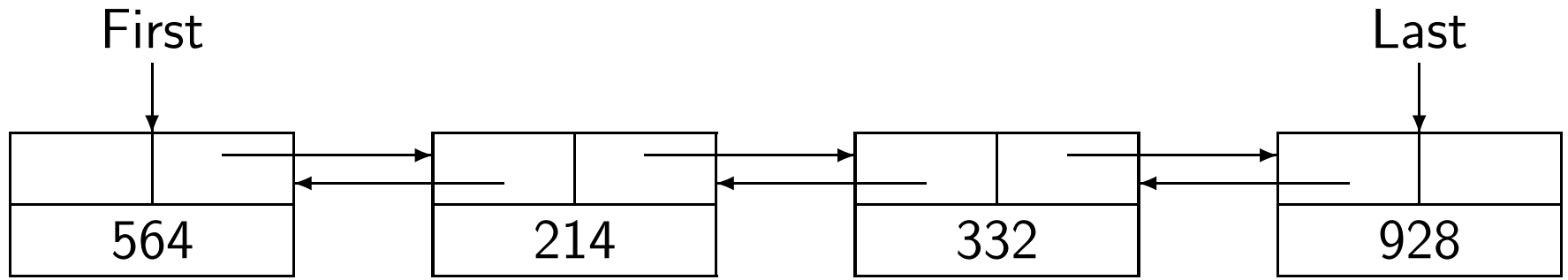
---



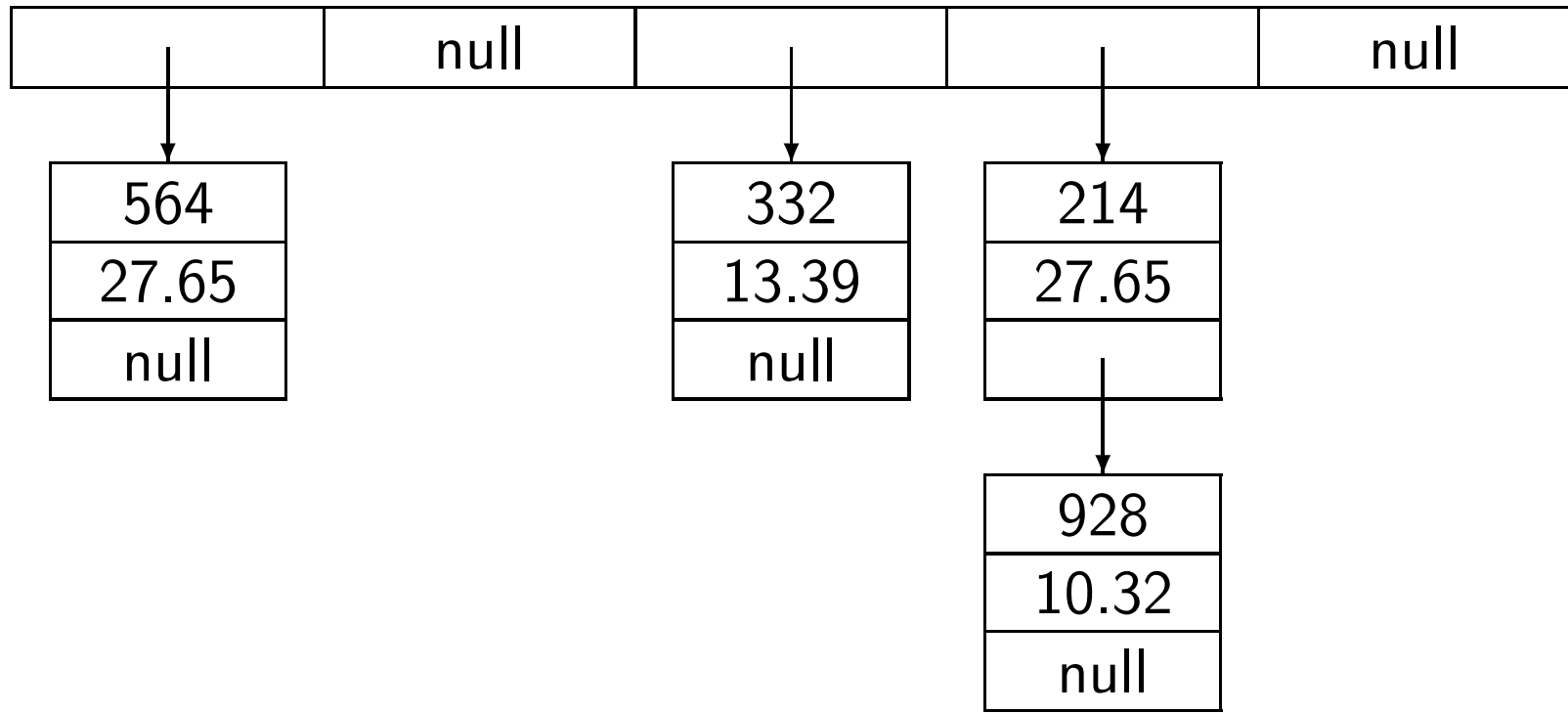
# Expanding a Vector



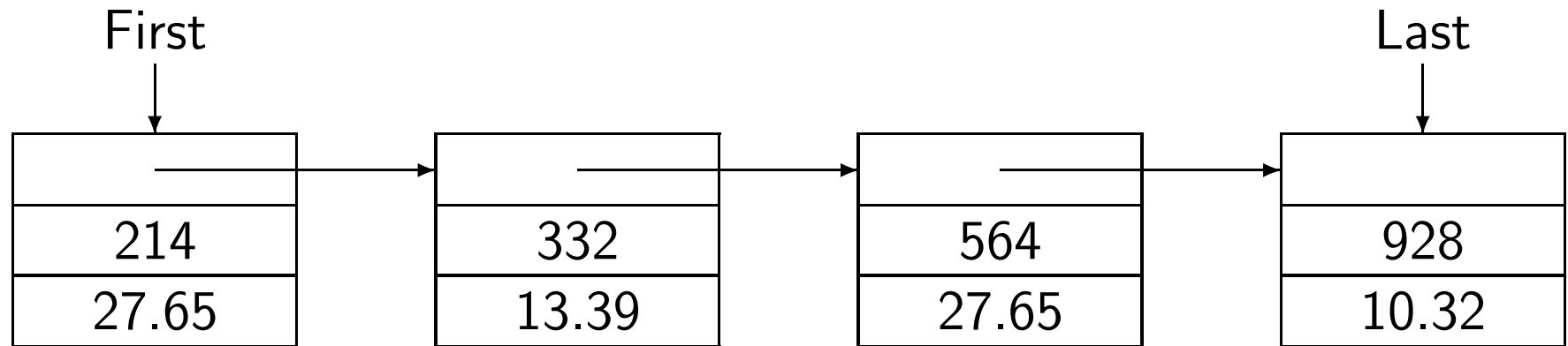
# Doubly-Linked List



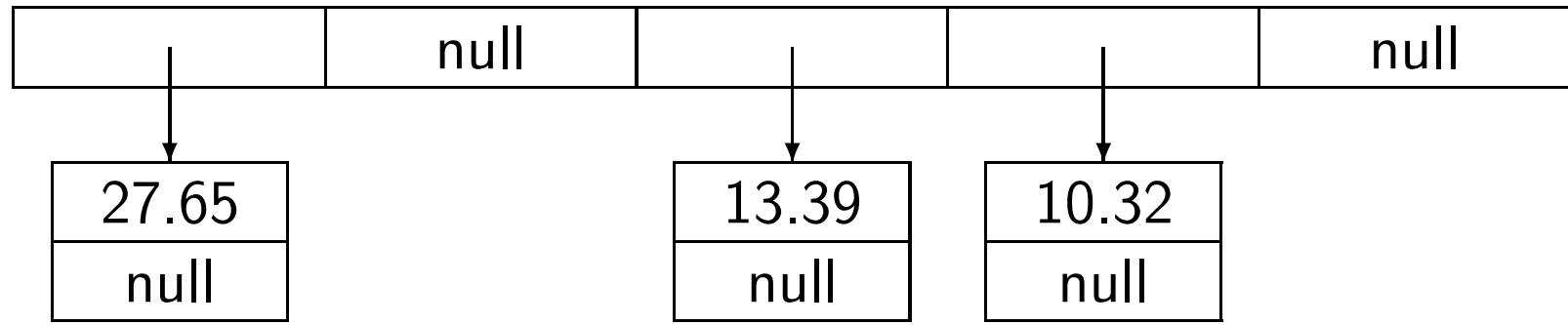
# Hashed Map



# Ordered Map

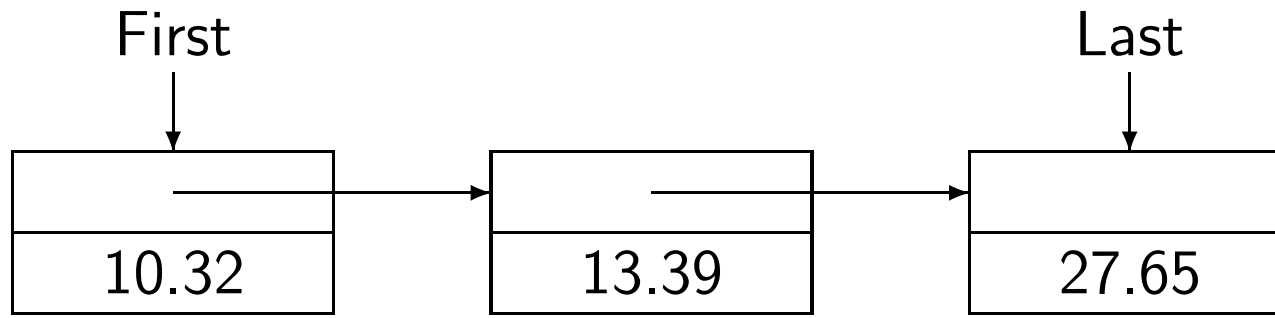


# Hashed Set

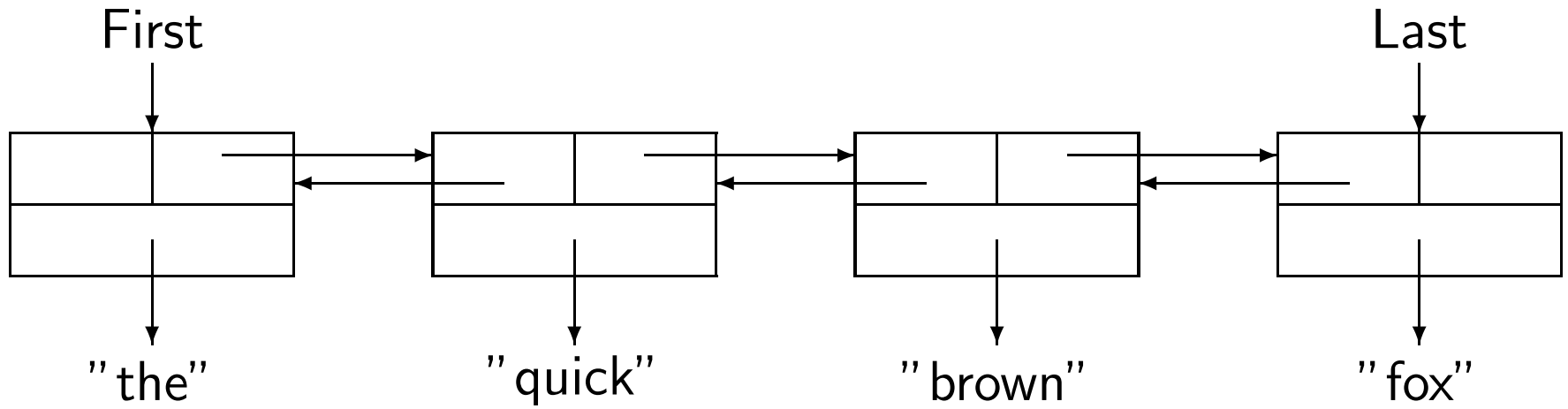


# Ordered Set

---

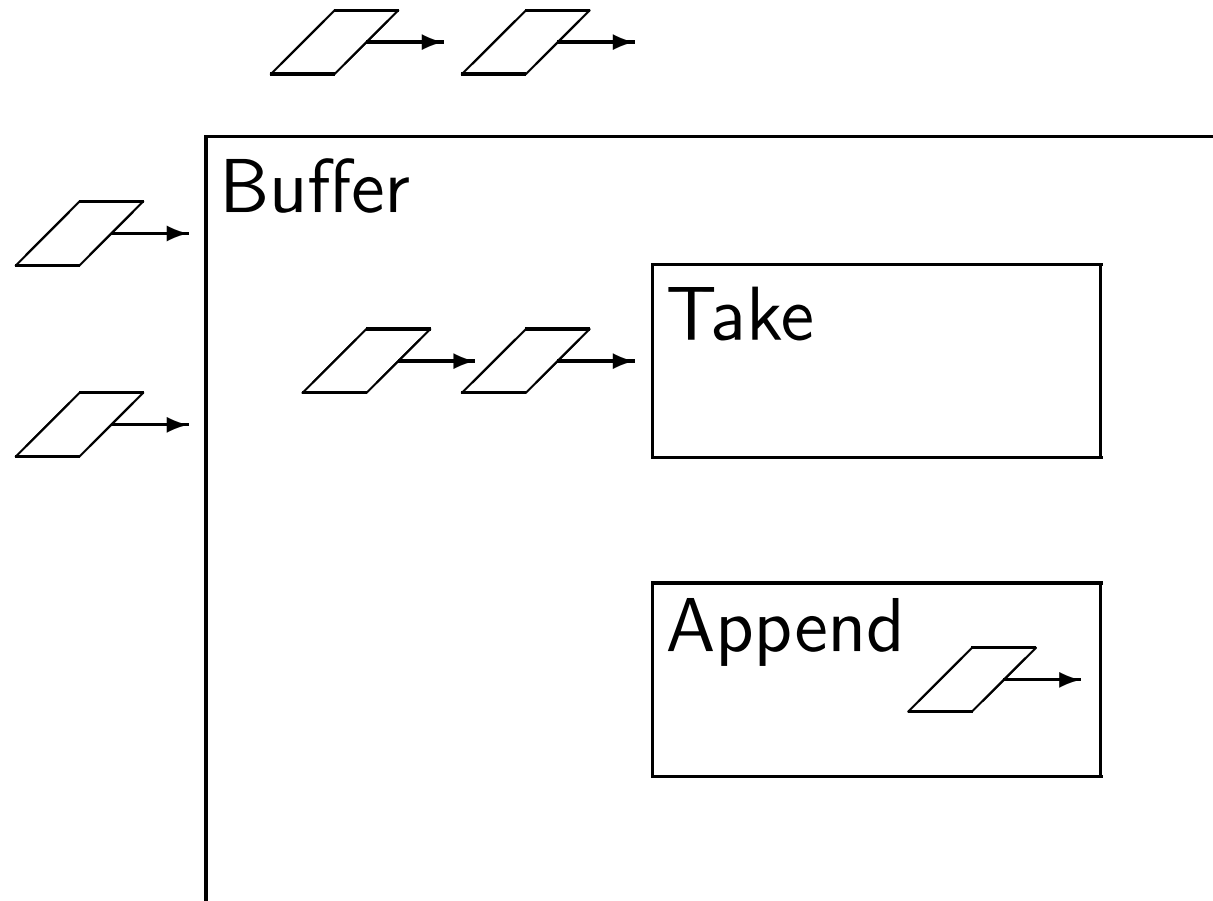


# Indefinite Container

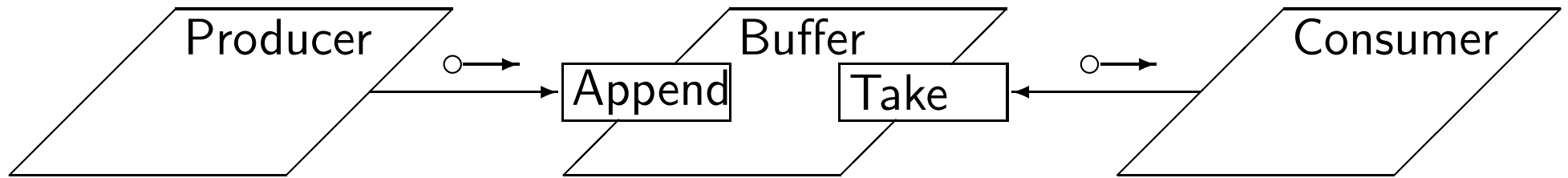




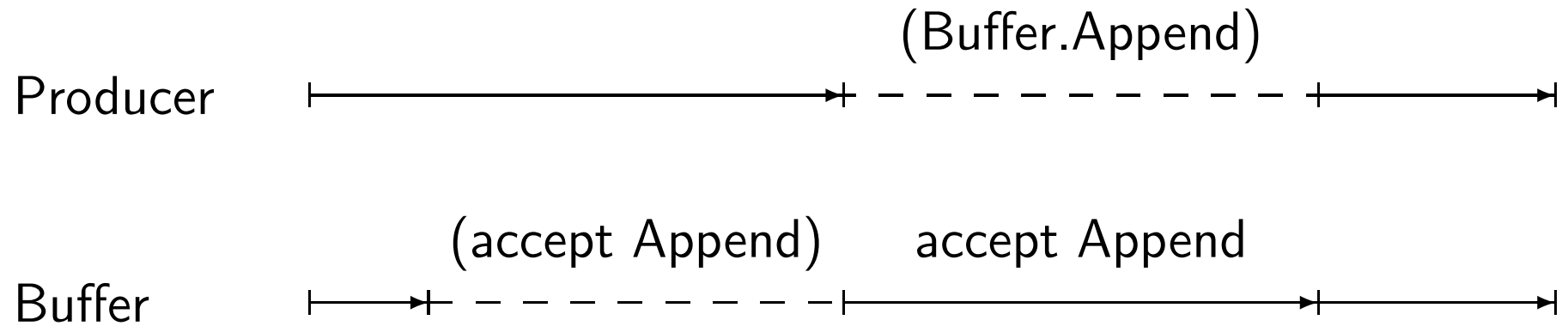
# Protected Object



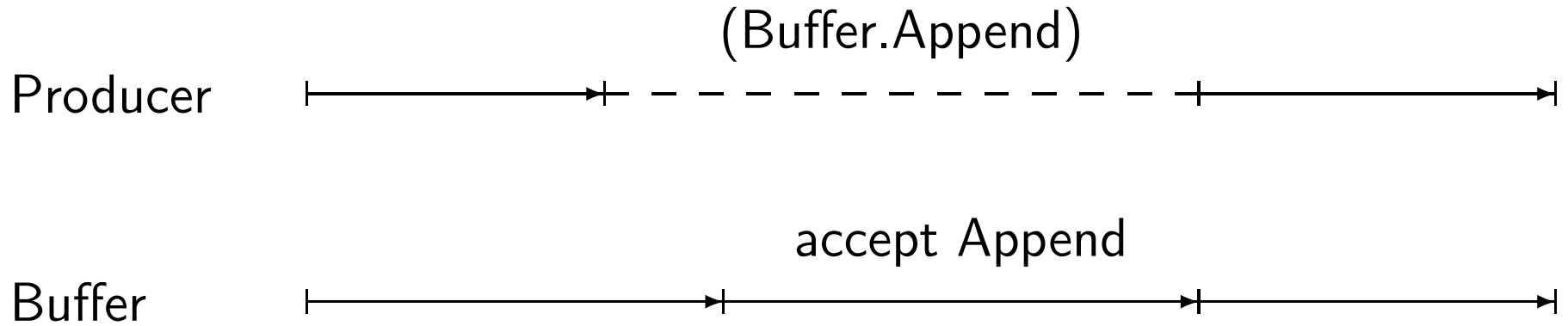
# Rendezvous



# Accept Waits for Entry Call

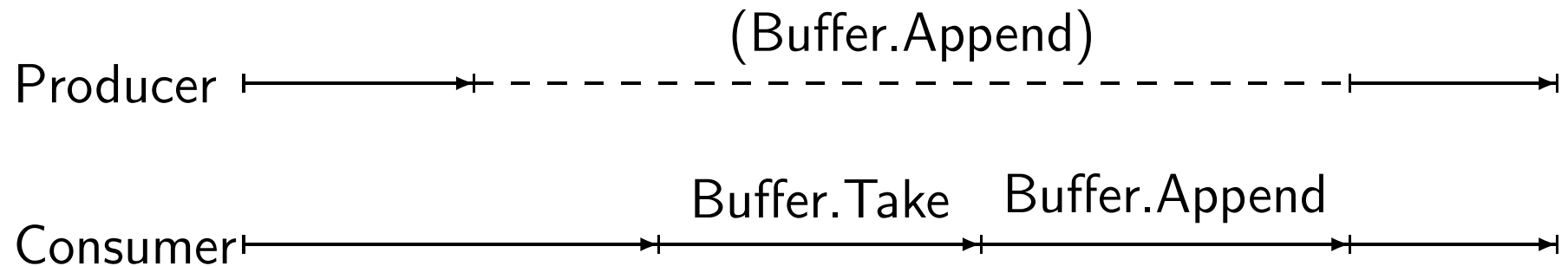


# Entry Call Waits for Accept

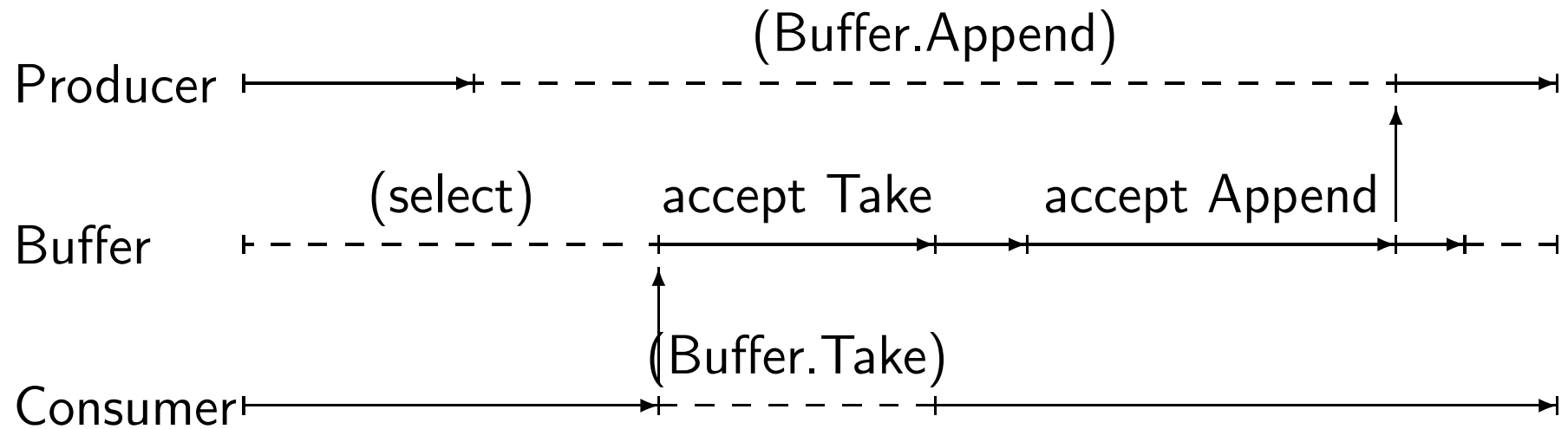


# Servicing of an Entry Queue by a Different Task

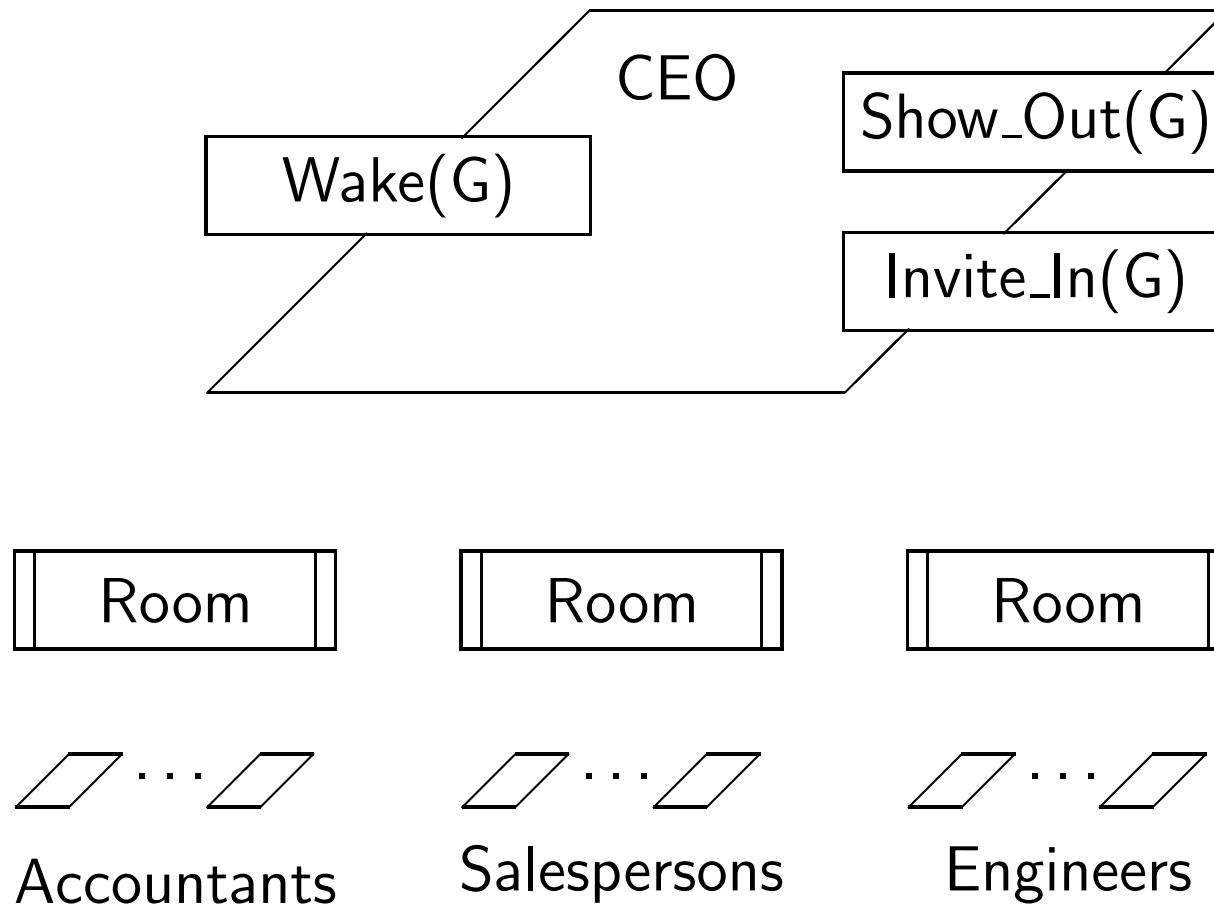
---



# Context Switches for Rendezvous

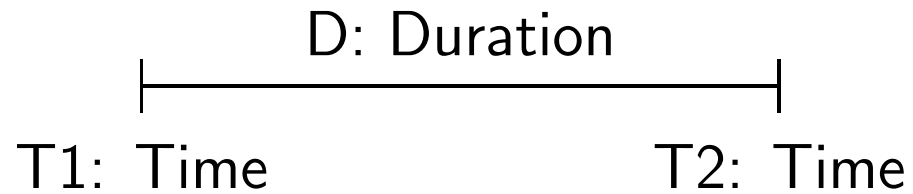


# The CEO Program



# Time and Duration

---



$T2 := T1 + D;$

$D := T2 - T1;$



# Concurrent Simulation Framework

Root\_Event

```
type Event
procedure Simulate
```

Root\_Event.  
Simulation

```
type Producer
  Queue
  Lock
  Generator
  Latest
procedure Put
function Get
```

Root\_Event.  
Simulation.X

```
type X_Event
procedure Simulate
```

Root\_Event.  
Simulation.X.Creator

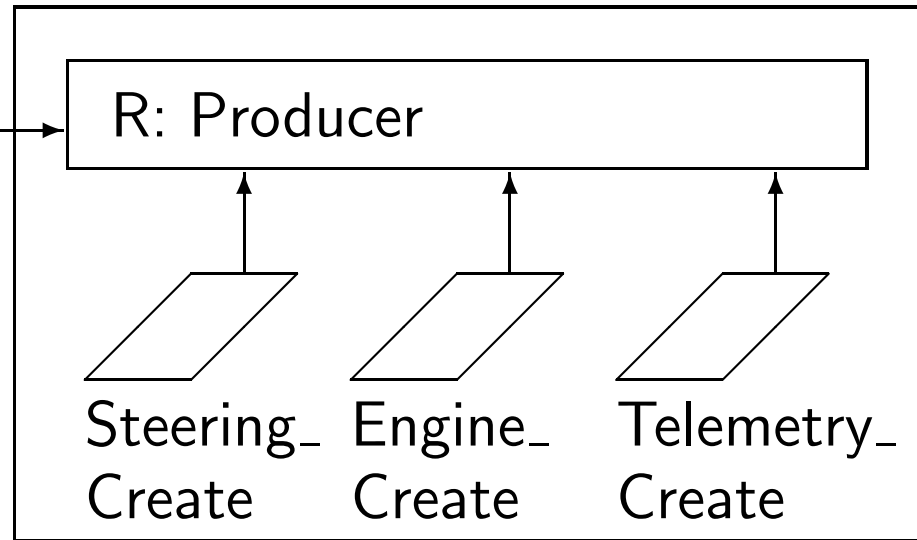
```
task type Create
(P: access Producer)
```

# Concurrent Rocket Simulation

Rocket

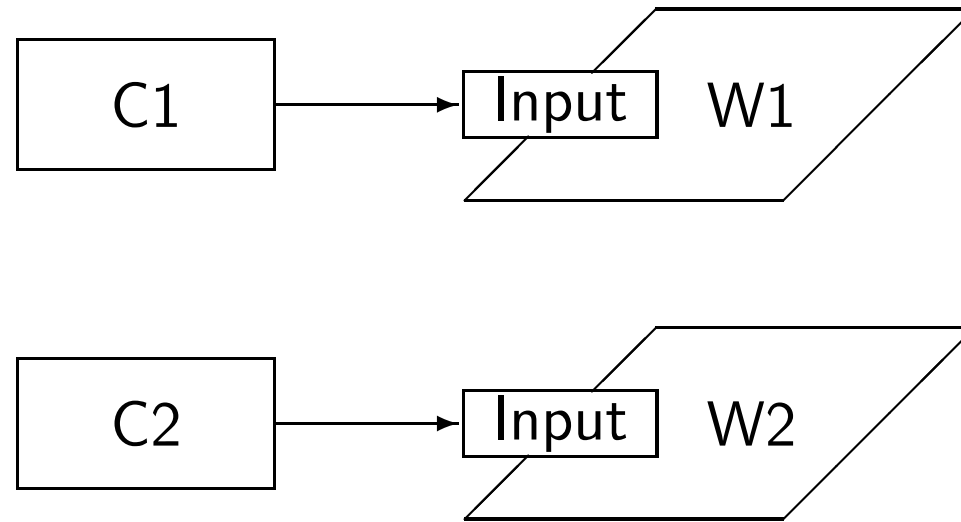
```
loop  
  Simulate(Get(R'Access));  
end loop;
```

Rocket\_Simulation



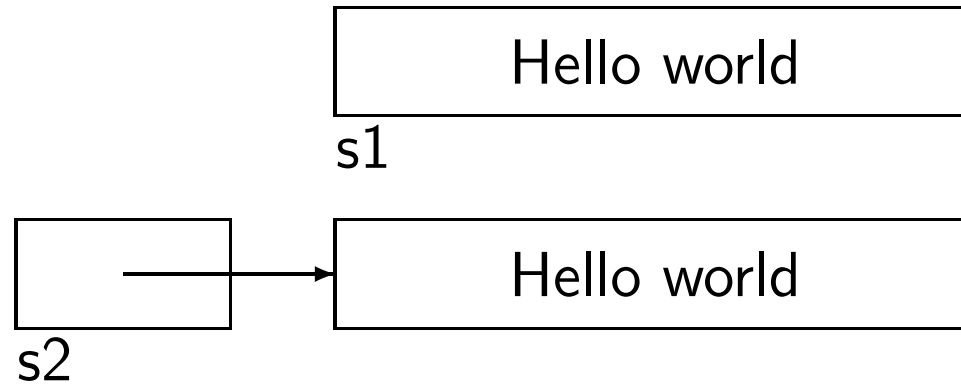
# Simulating Dispatching on Entries

---



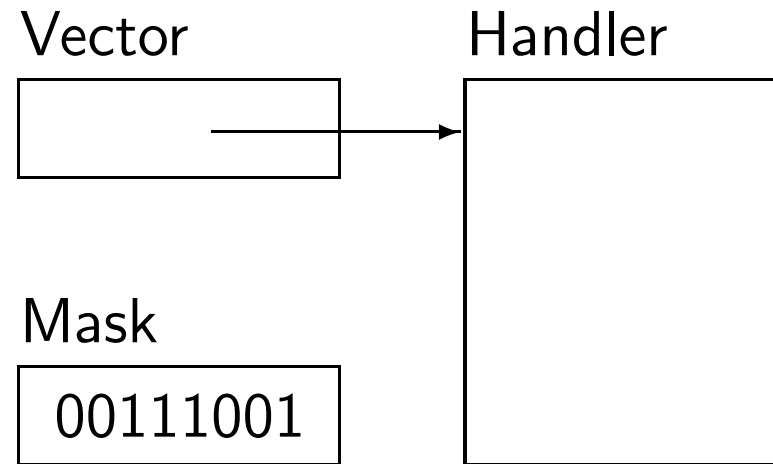
# Interfacing with C

---



# Interrupts

---



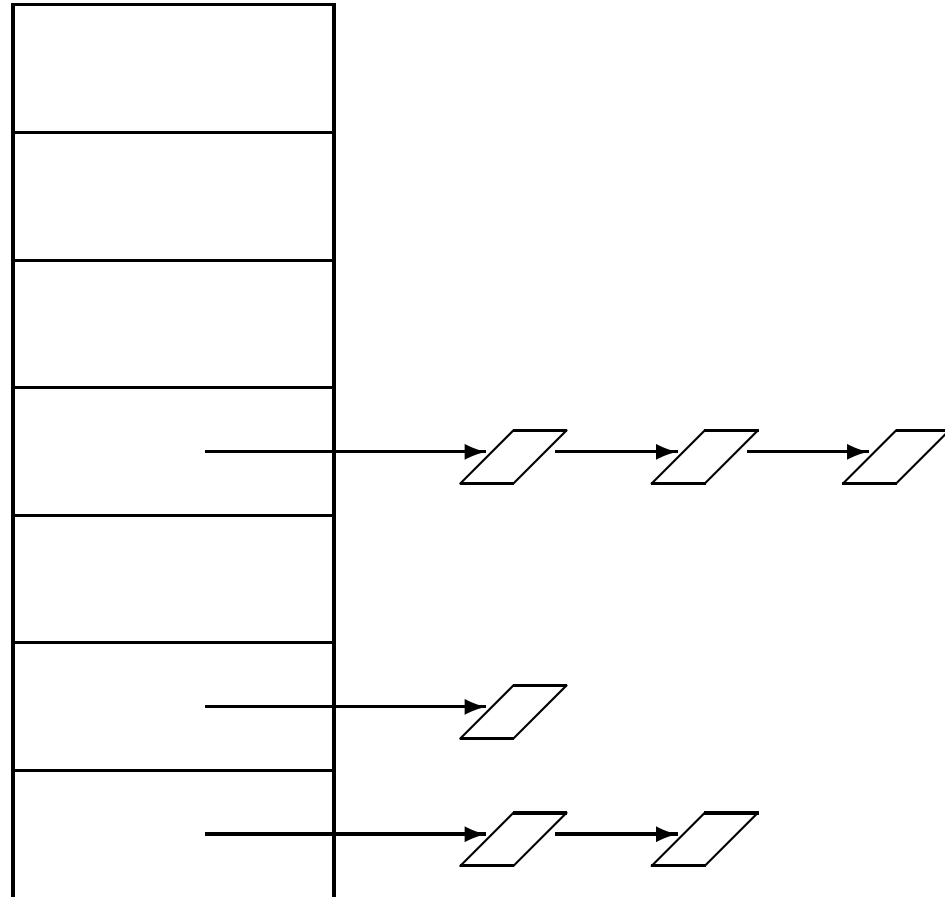
# Ready Queues

Interrupt\_Priority'Last

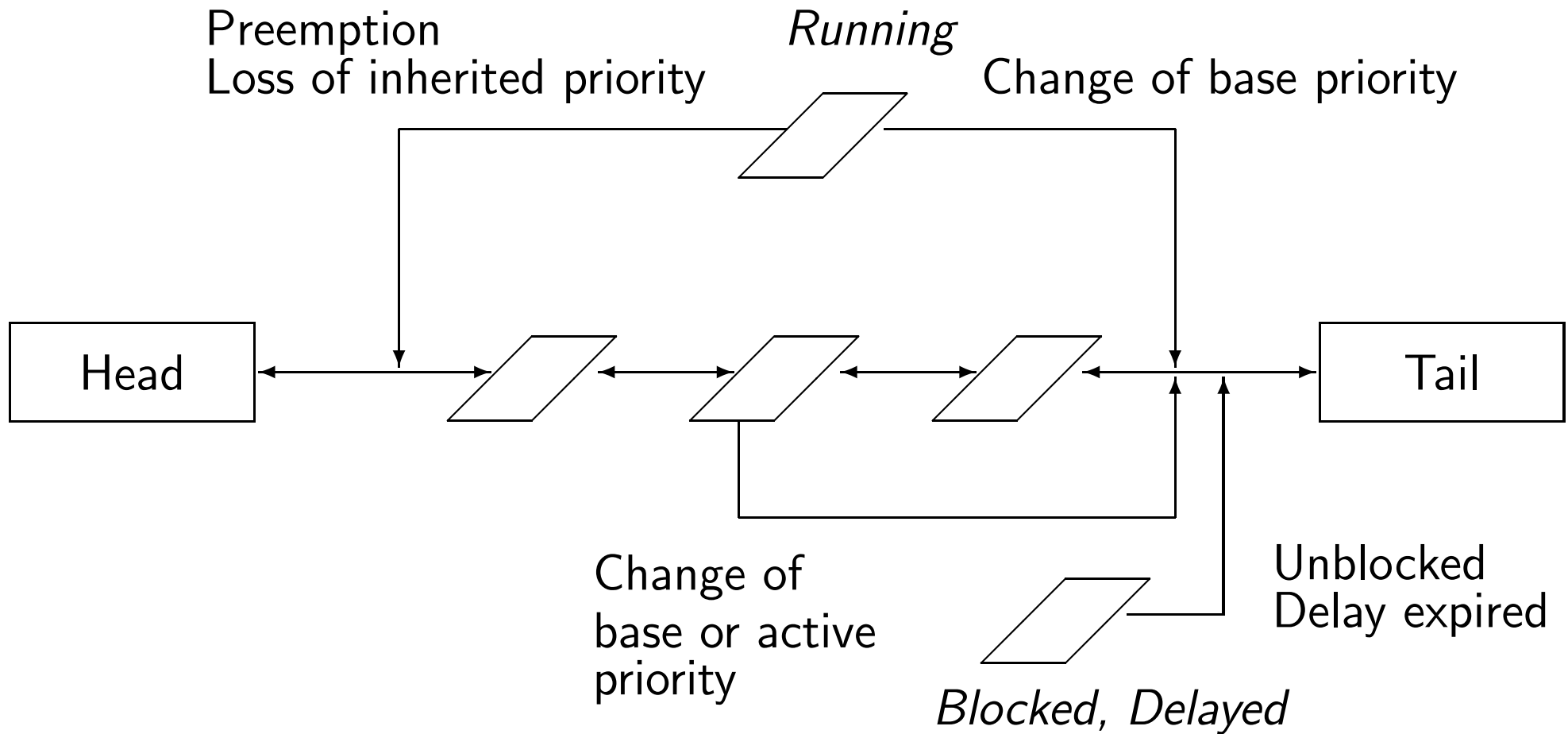
Interrupt\_Priority'First

Priority'Last

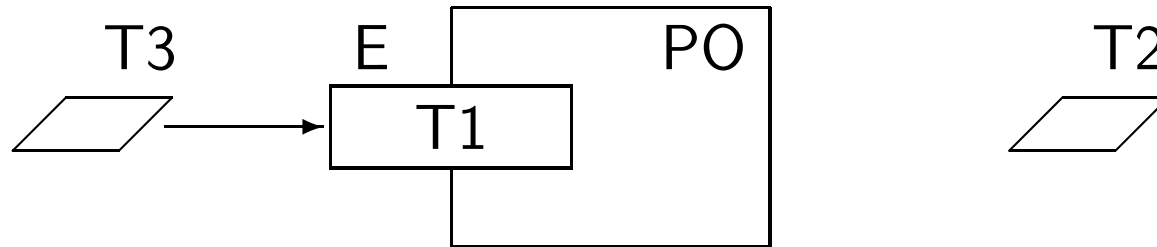
Priority'First



# FIFO\_Within\_Priorities Policy



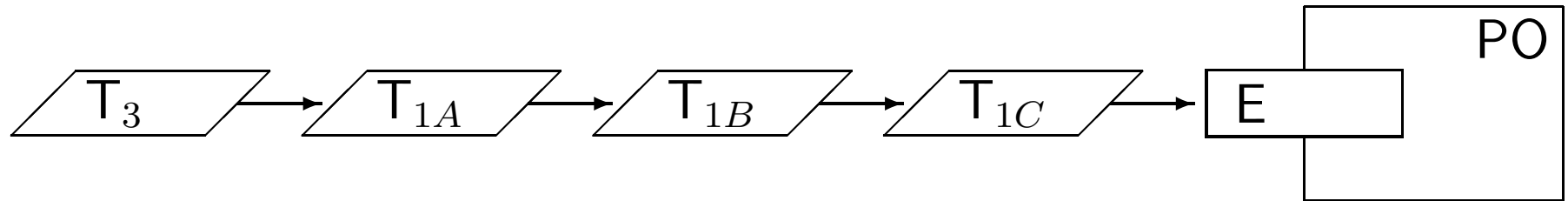
# Priority Inversion



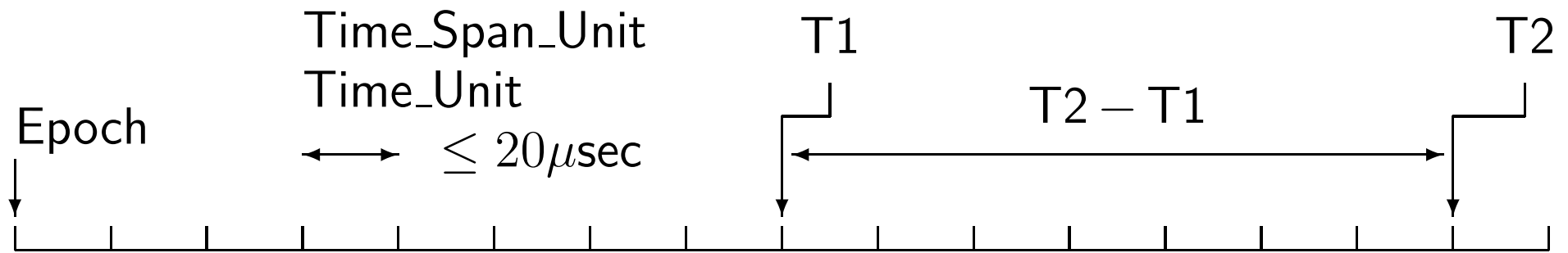


# FIFO\_Queueing Policy

---

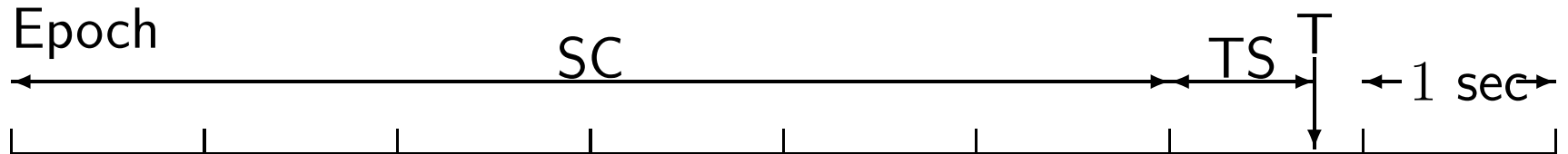


# Monotonic Time

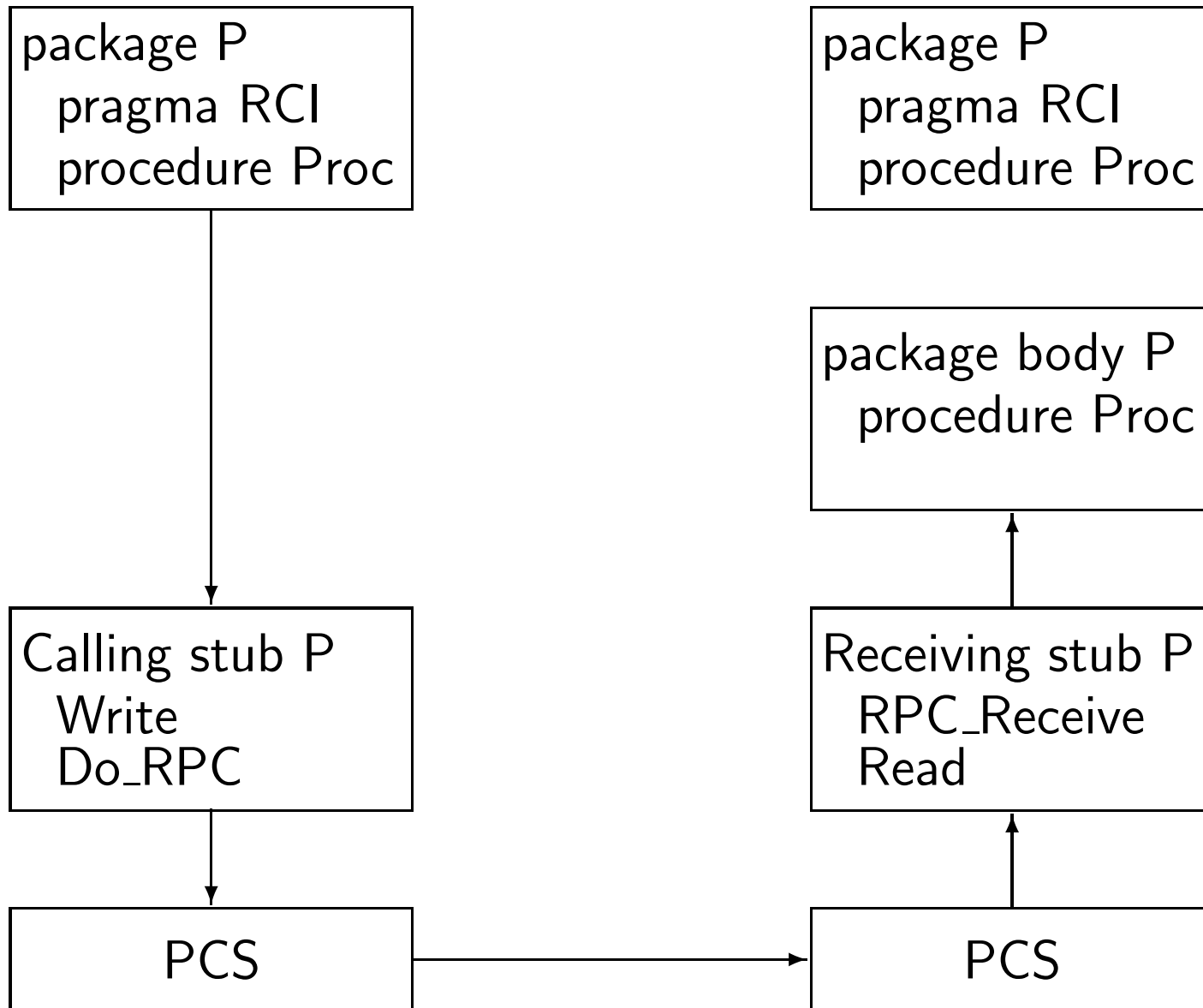


# Splitting Ada.Real\_Time.Time

---



# Remote Procedure Call



# Distributed Simulation

---

