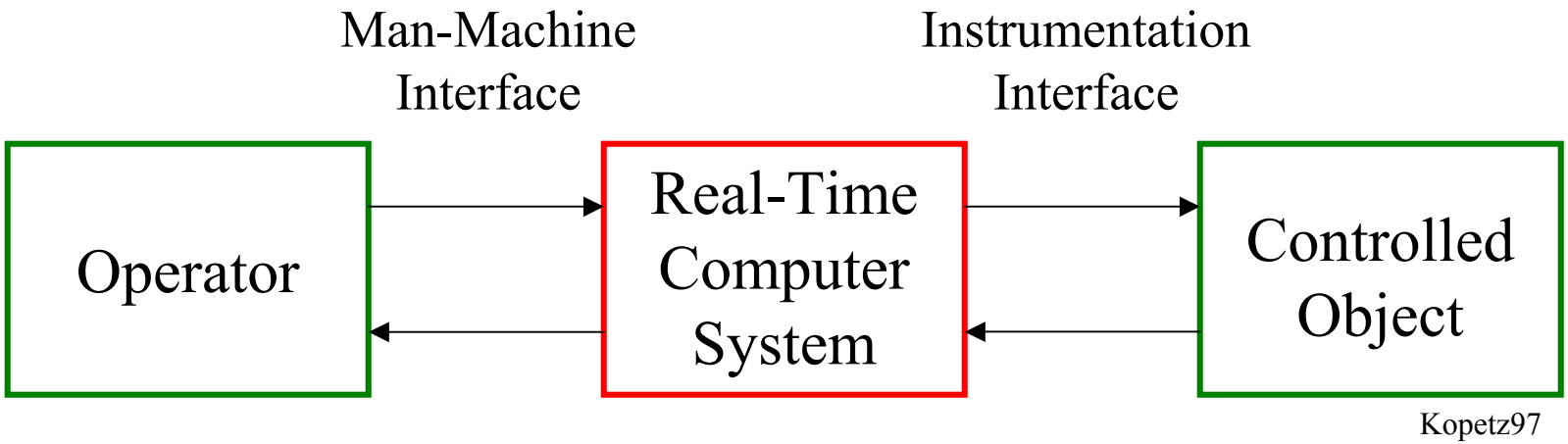
Giotto

Thomas A. Henzinger, Benjamin Horowitz, Christoph Kirsch

UC Berkeley

www.eecs.berkeley.edu/~fresco/giotto

The Problem



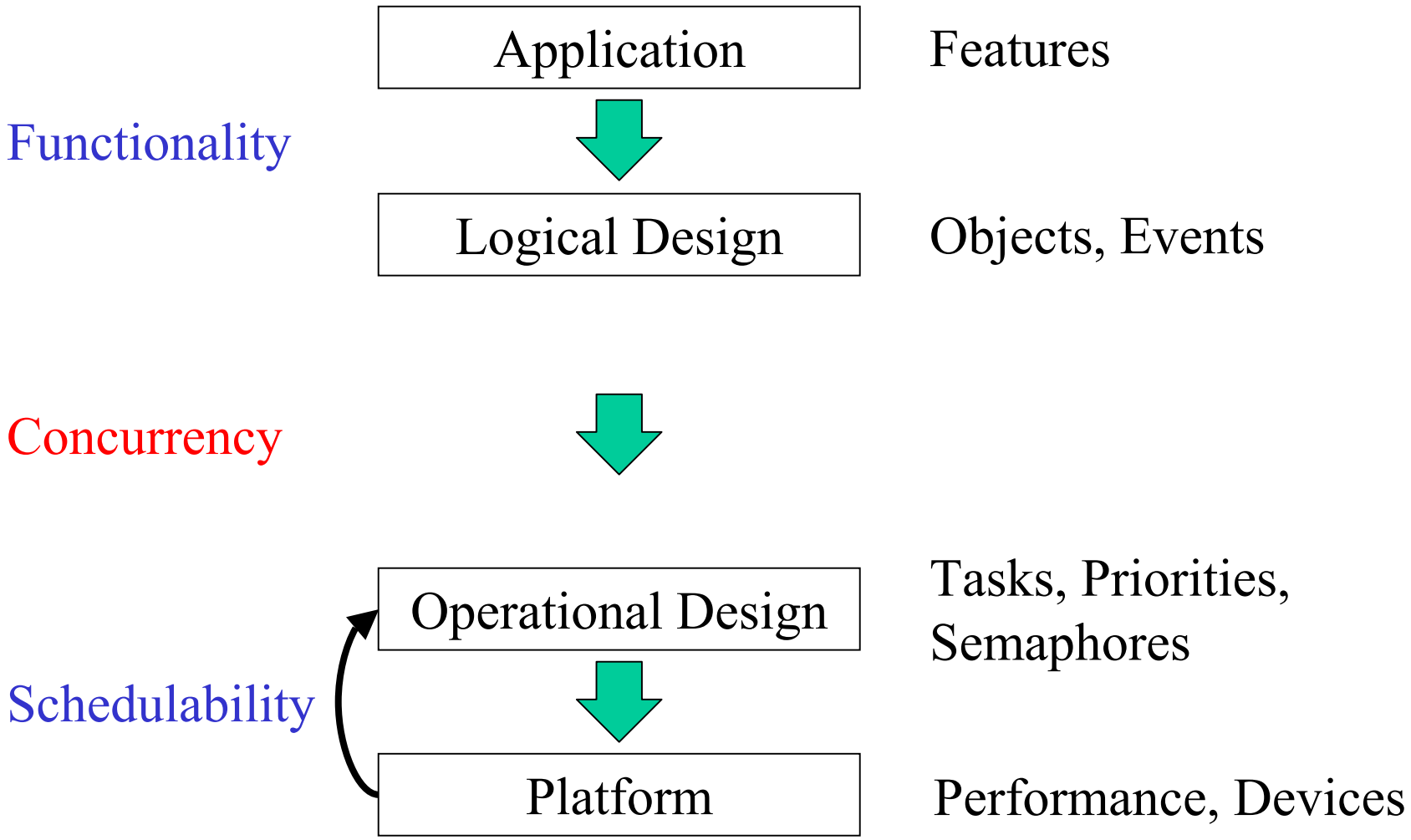
Methodologies for the implementation of embedded real-time applications on distributed platforms

- Methodology: tool-supported, logical, compositional
- Implementation: compositional, scalable, dependable

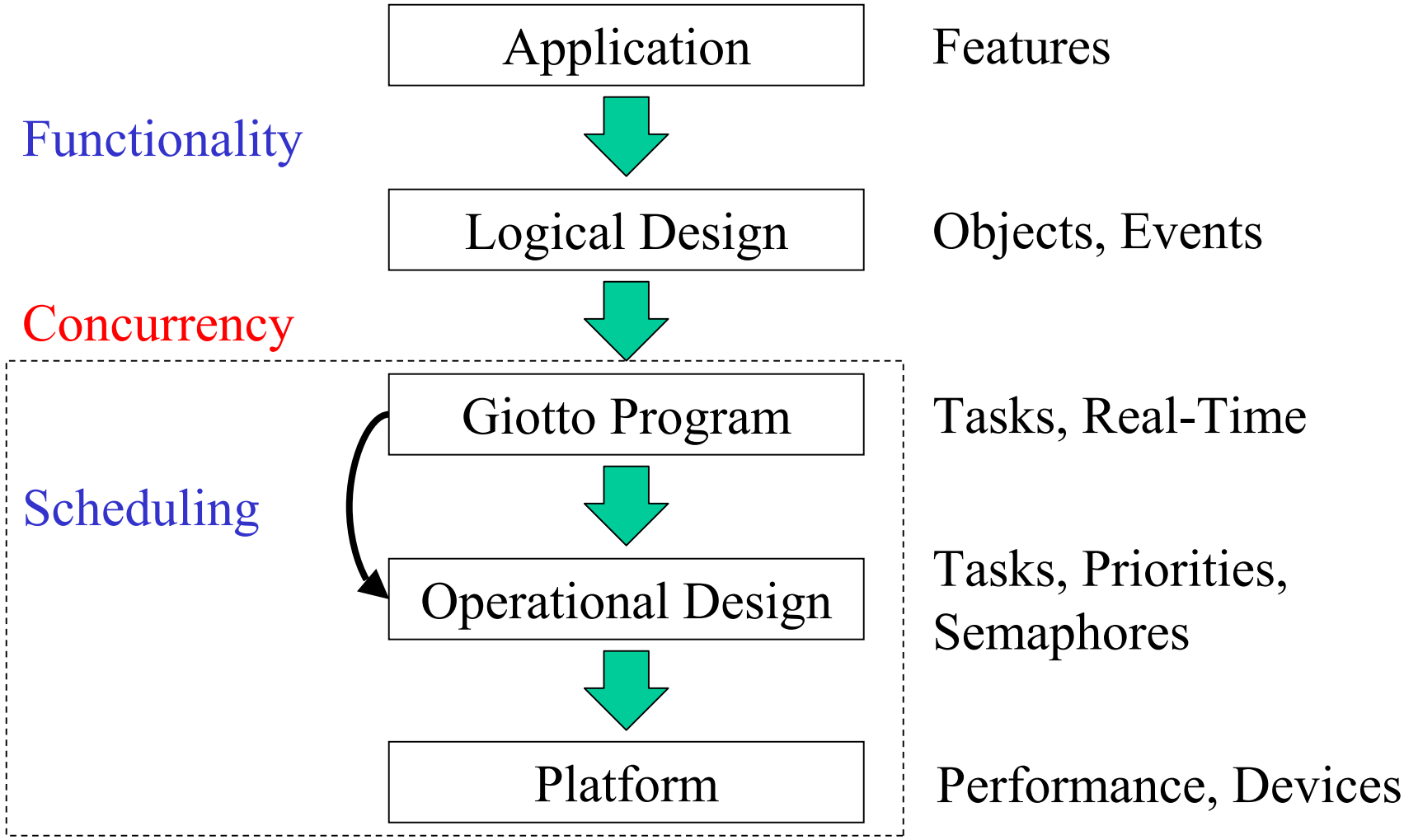
Giotto

- Giotto is a tool-supported methodology for embedded real-time systems design
- Giotto consists of:
  - A time-triggered and platform independent programming language
  - A compiler
  - A runtime library (or virtual scheduling machine)
- Giotto provides an abstract programming model

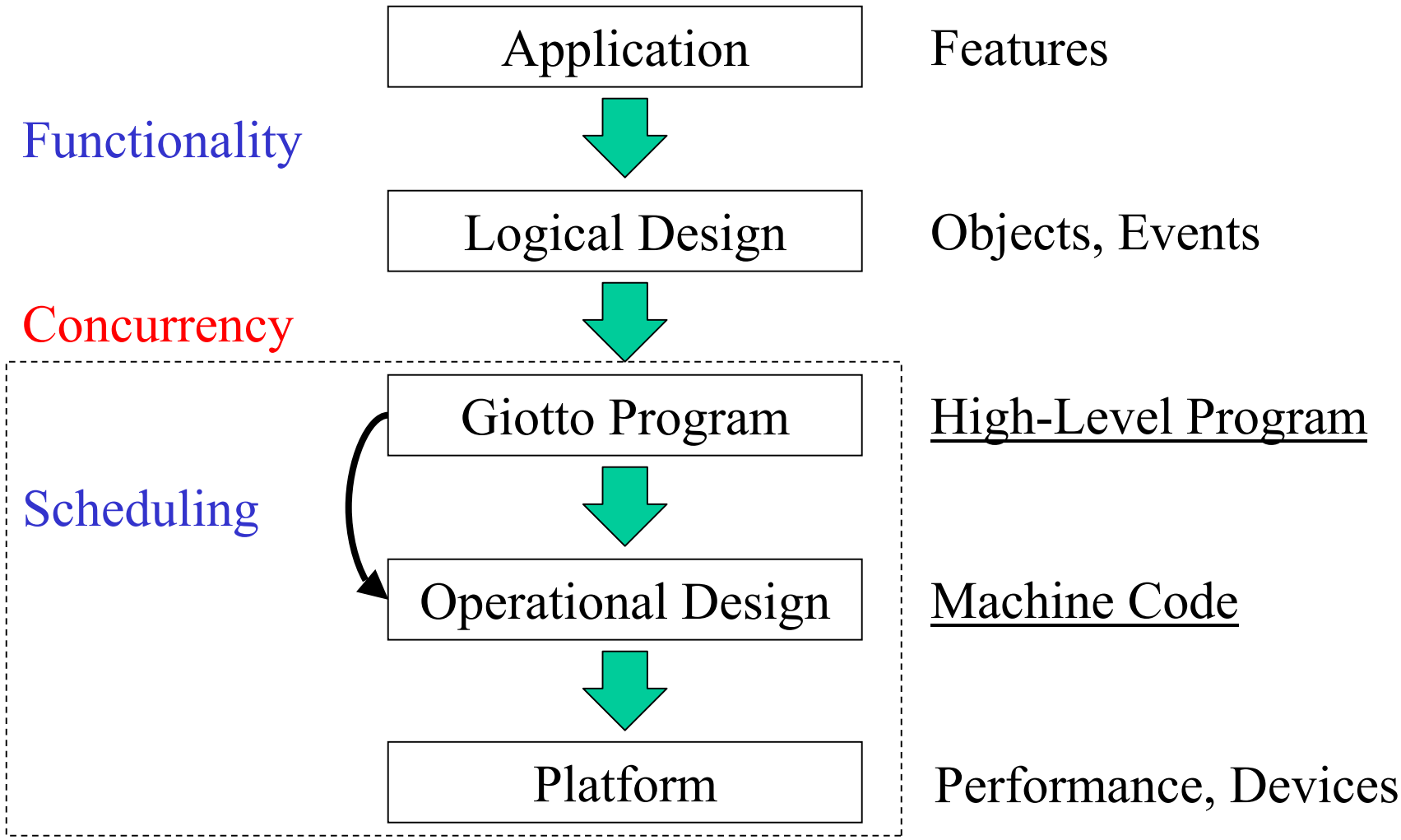
Typical Development Process



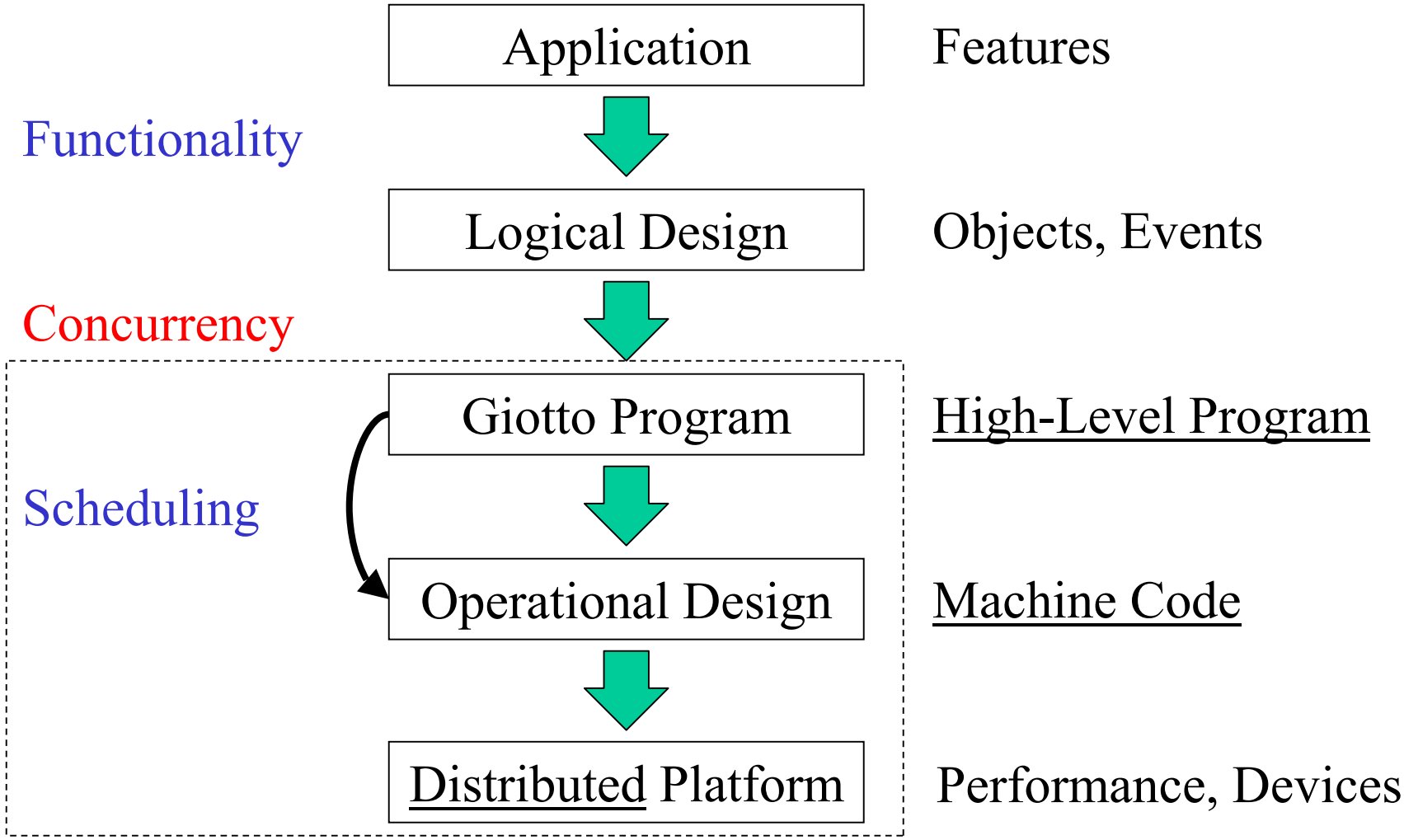
Paradigm Shift: Giotto



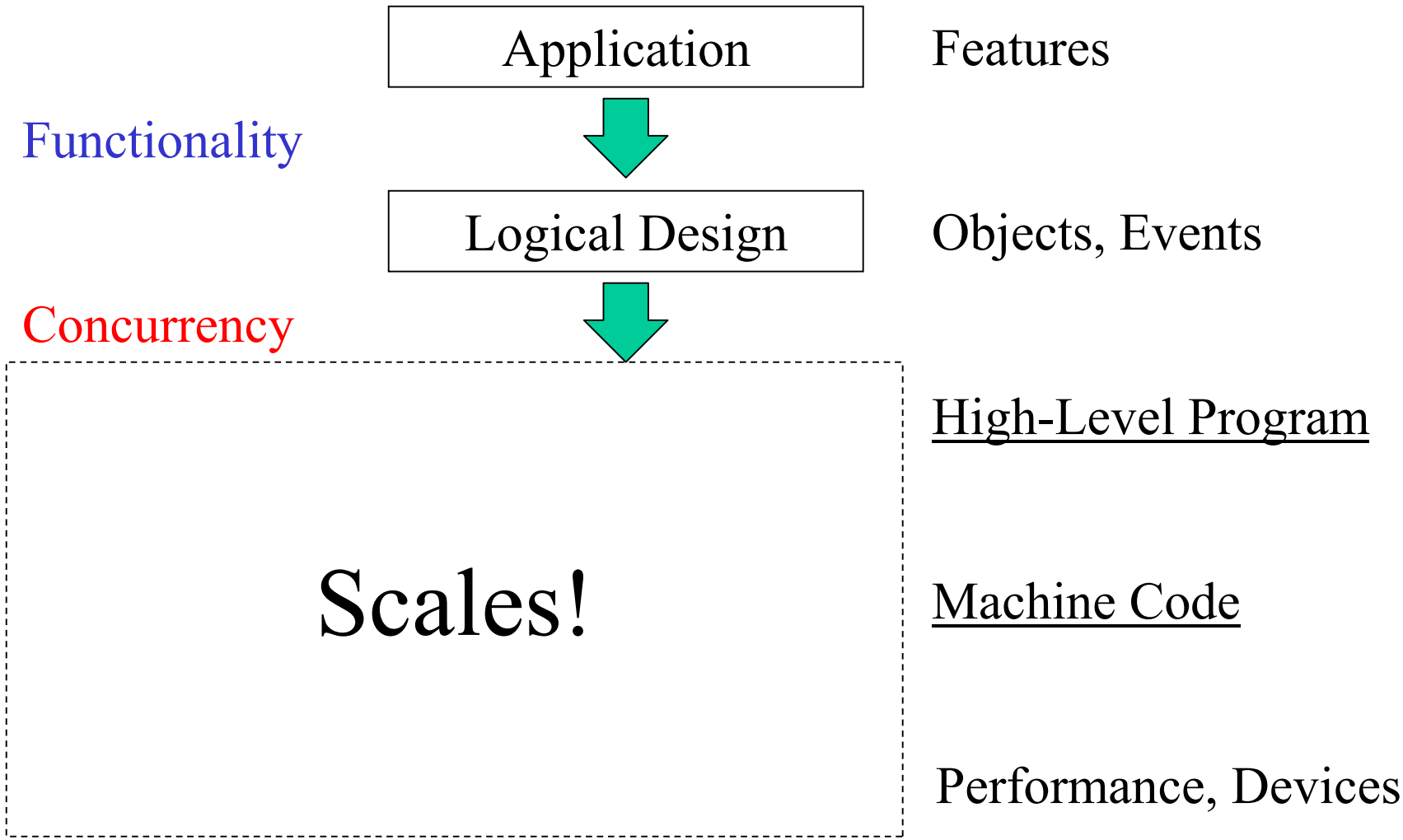
Compilation



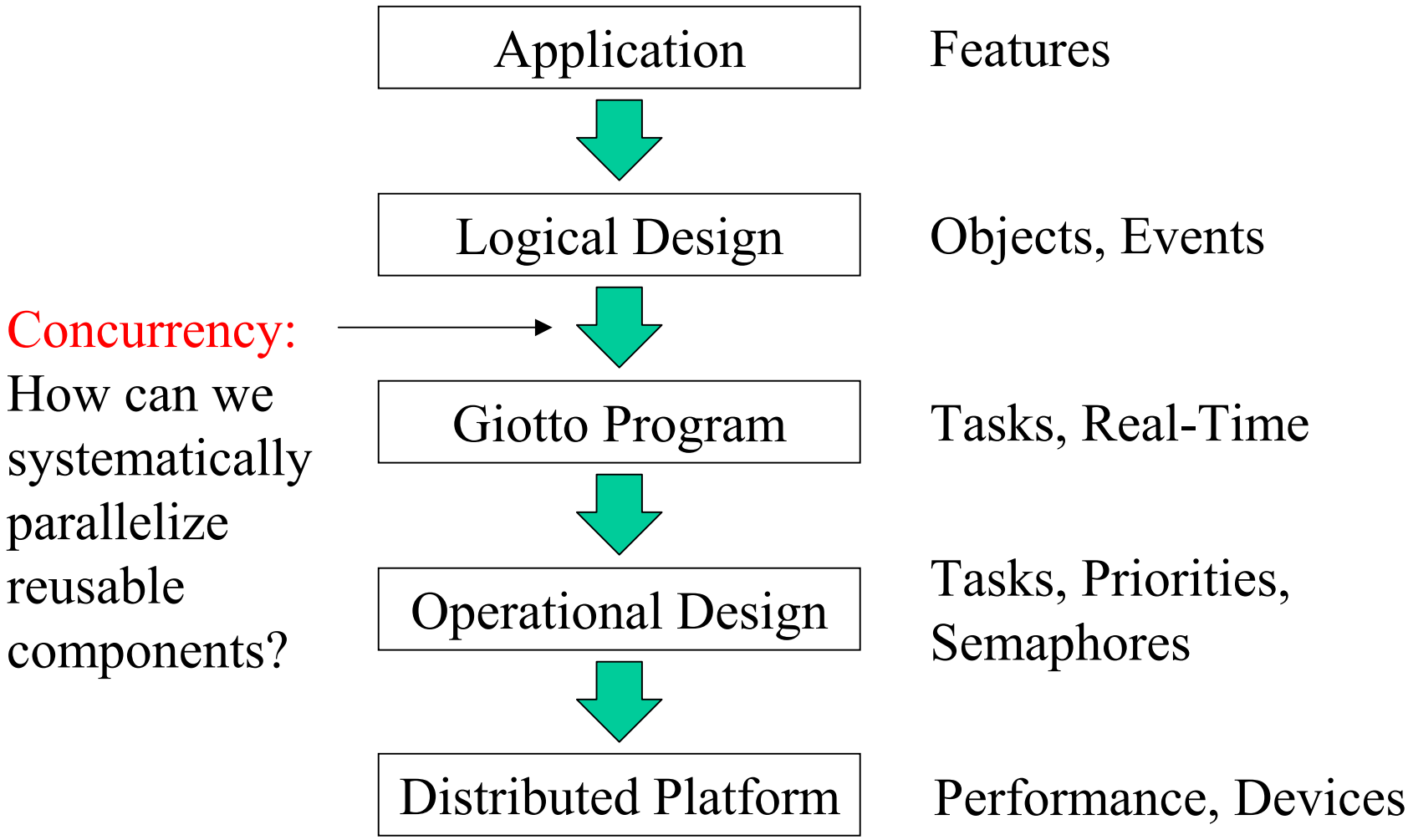
Distributed Platforms



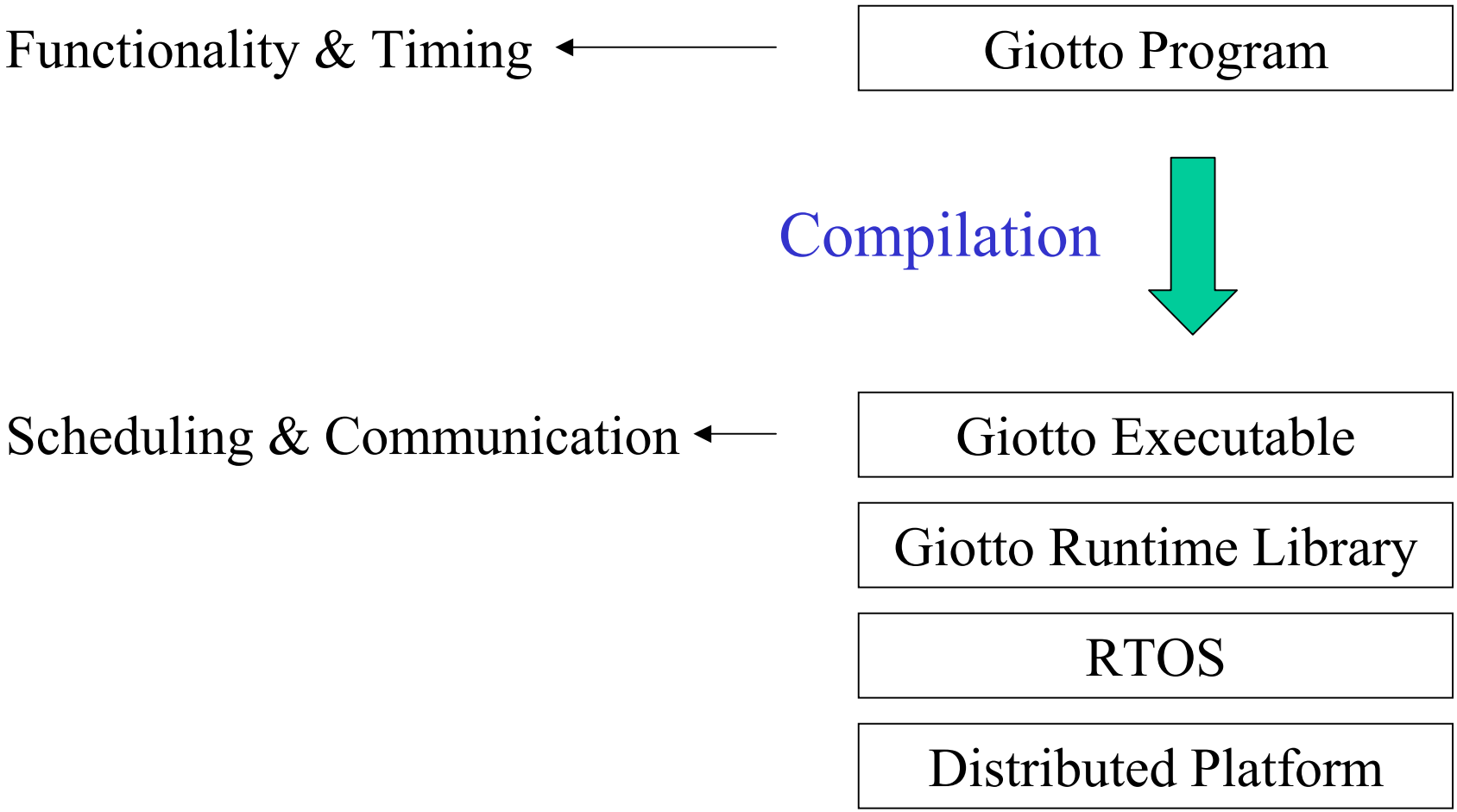
Scalability



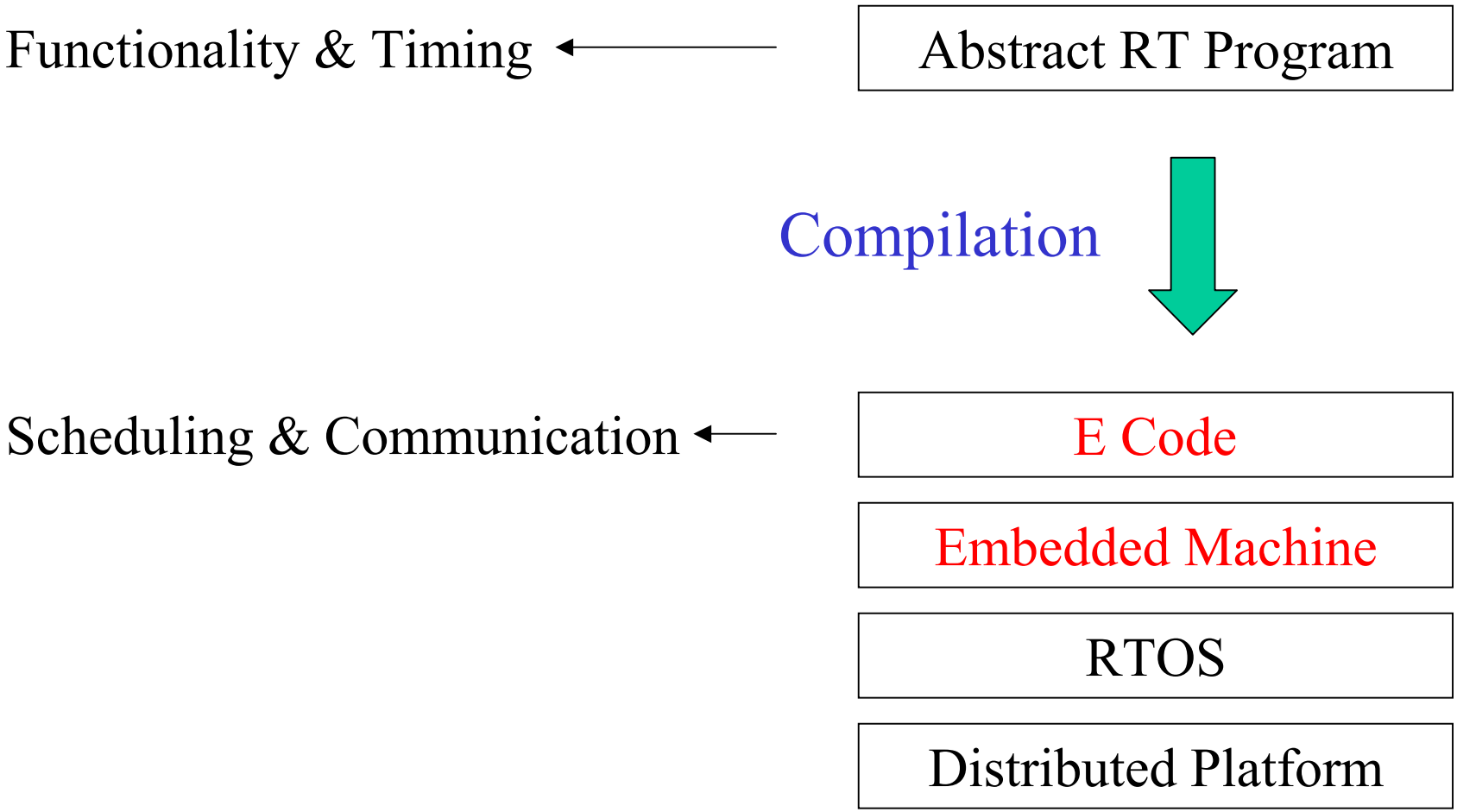
Berkeley - Constance



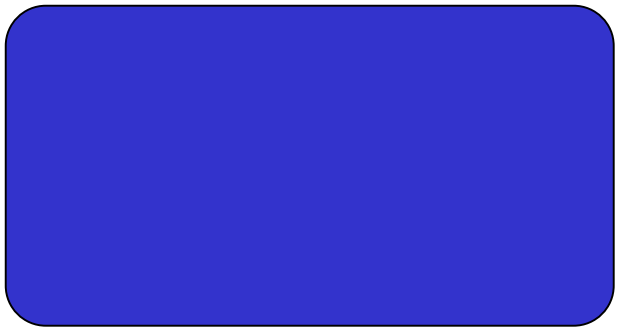
The Giotto Compiler



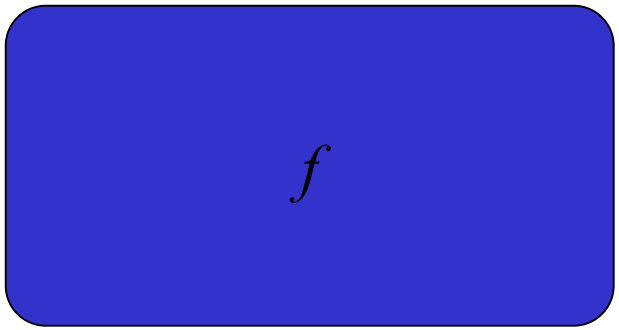
The Embedded Machine



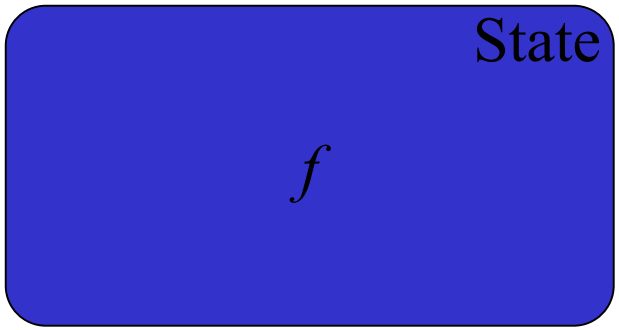
A Giotto Task



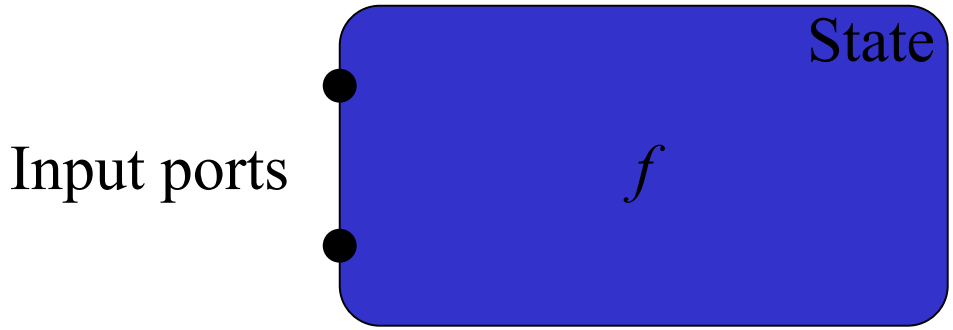
Abstract Syntax of a Task



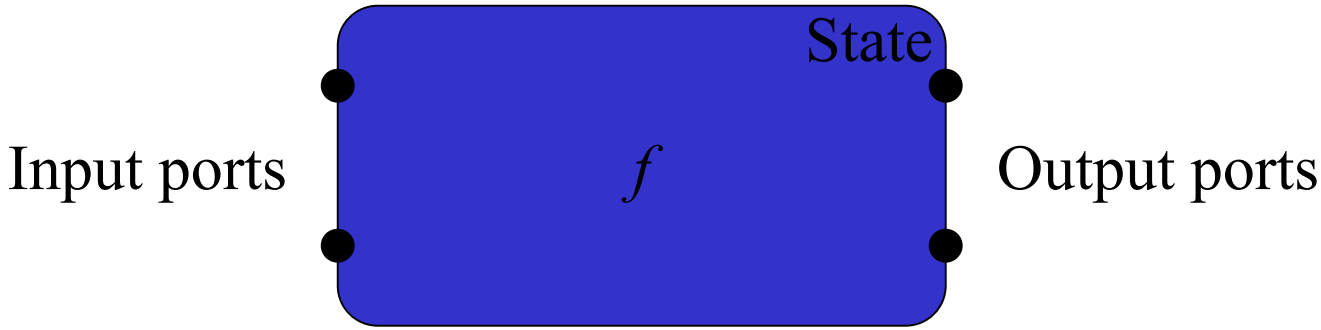
Abstract Syntax of a Task



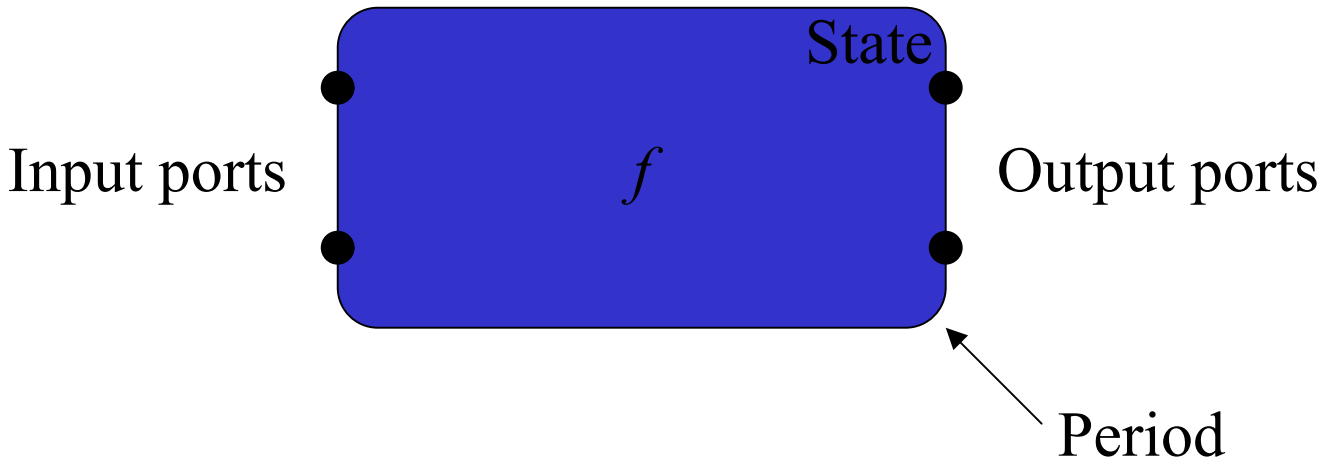
Abstract Syntax of a Task



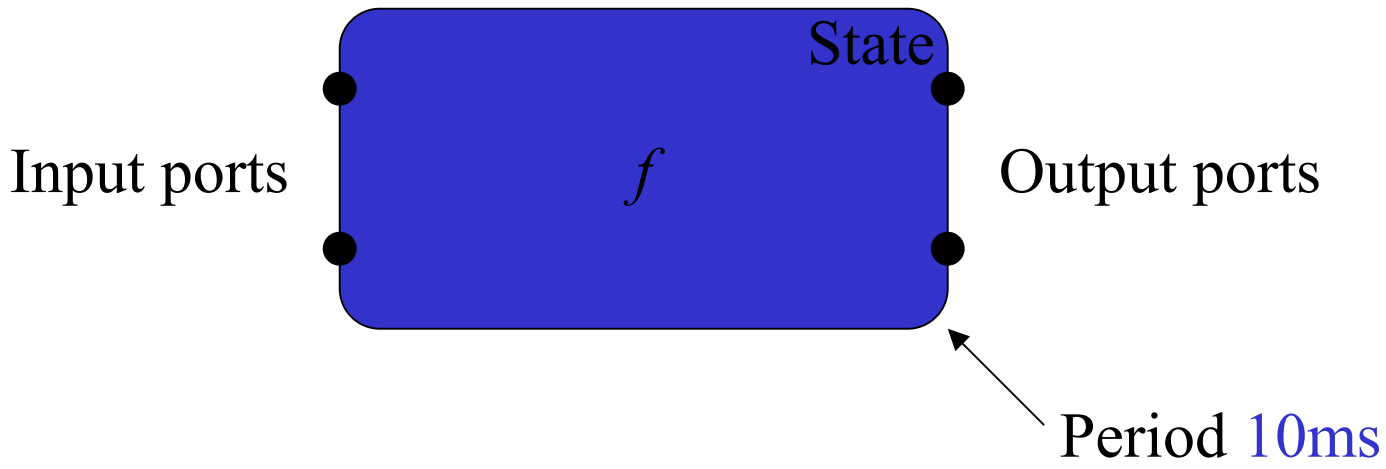
Abstract Syntax of a Task



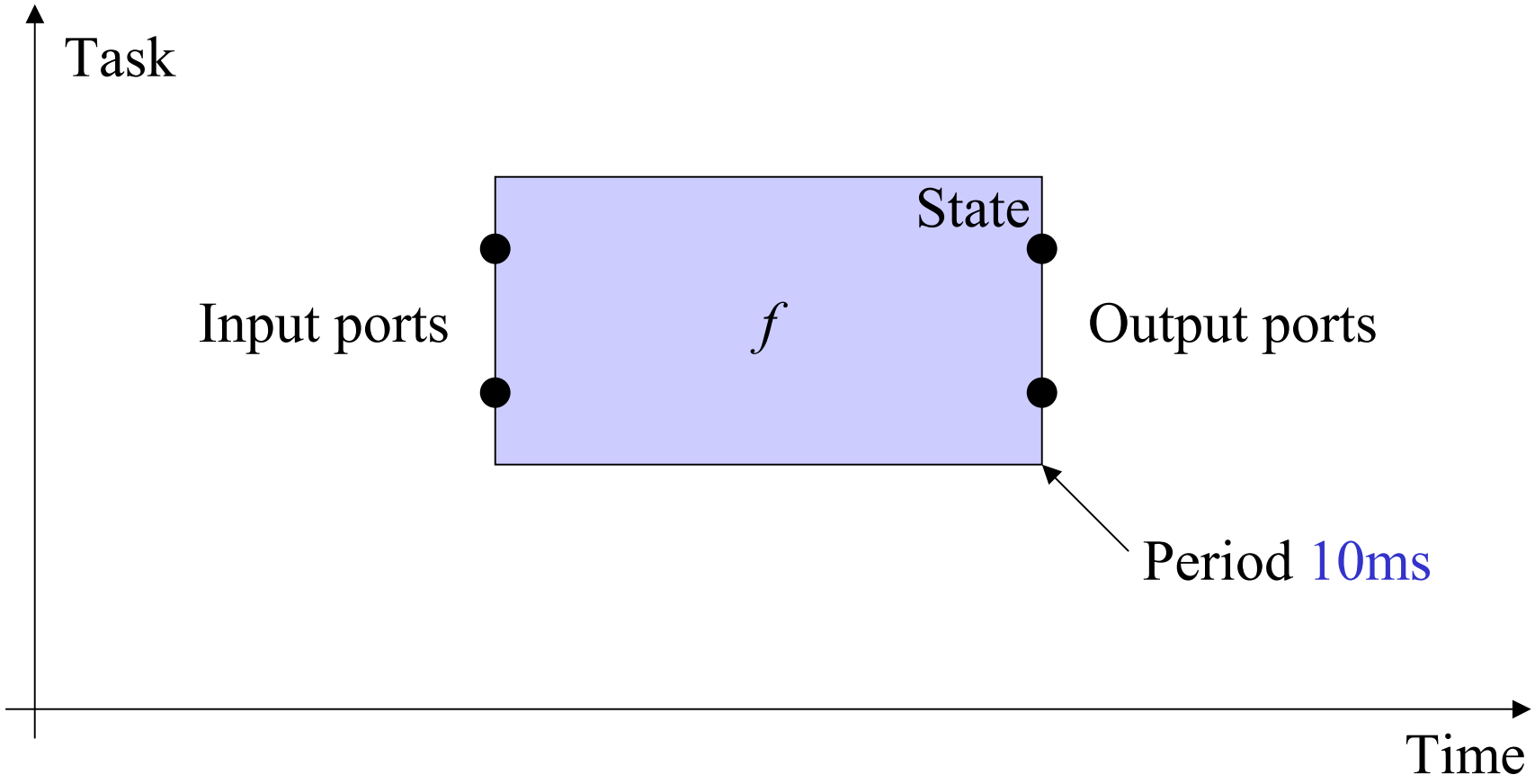
Abstract Syntax of a Task



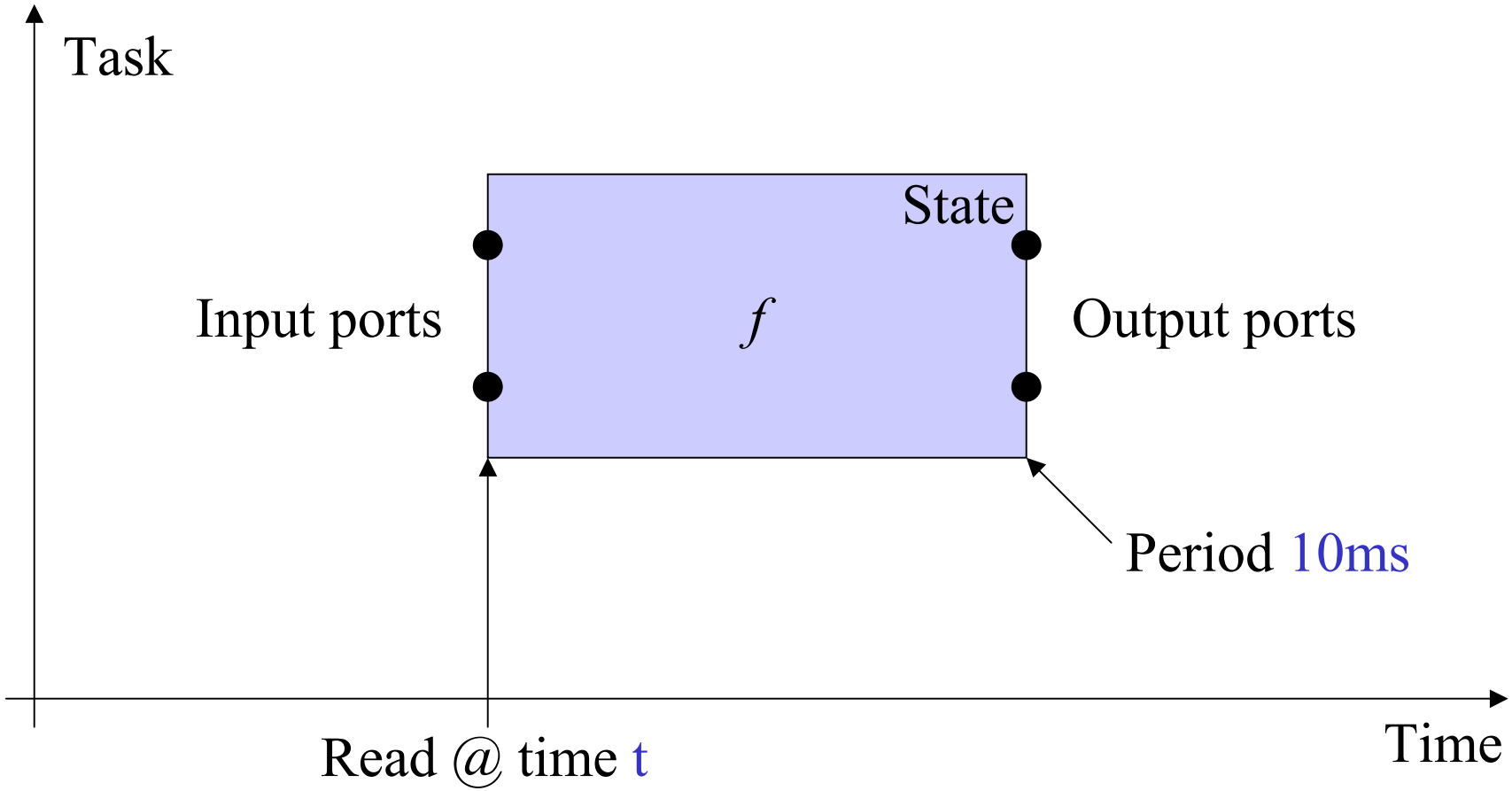
Semantics of a Task



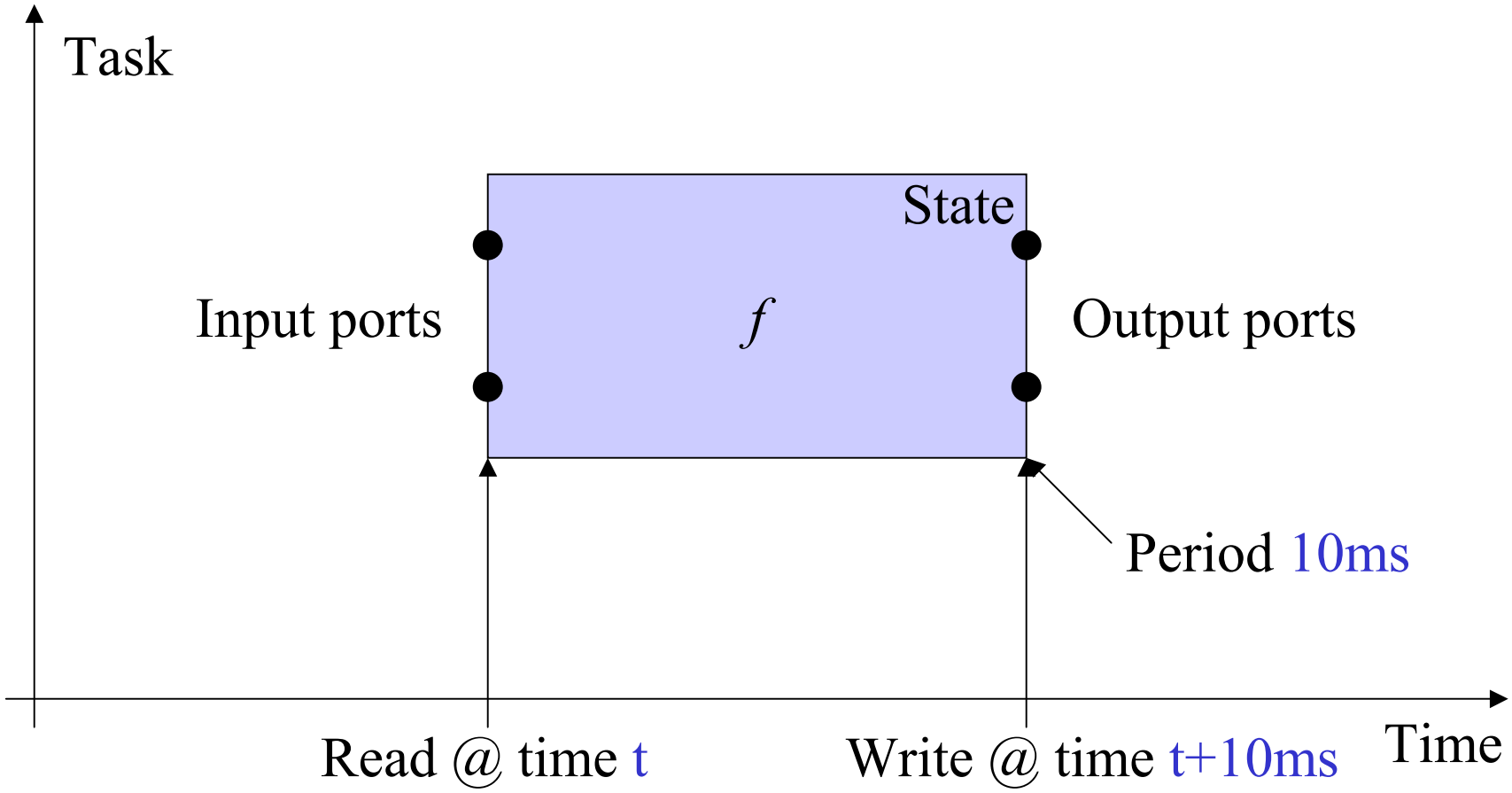
Semantics of a Task



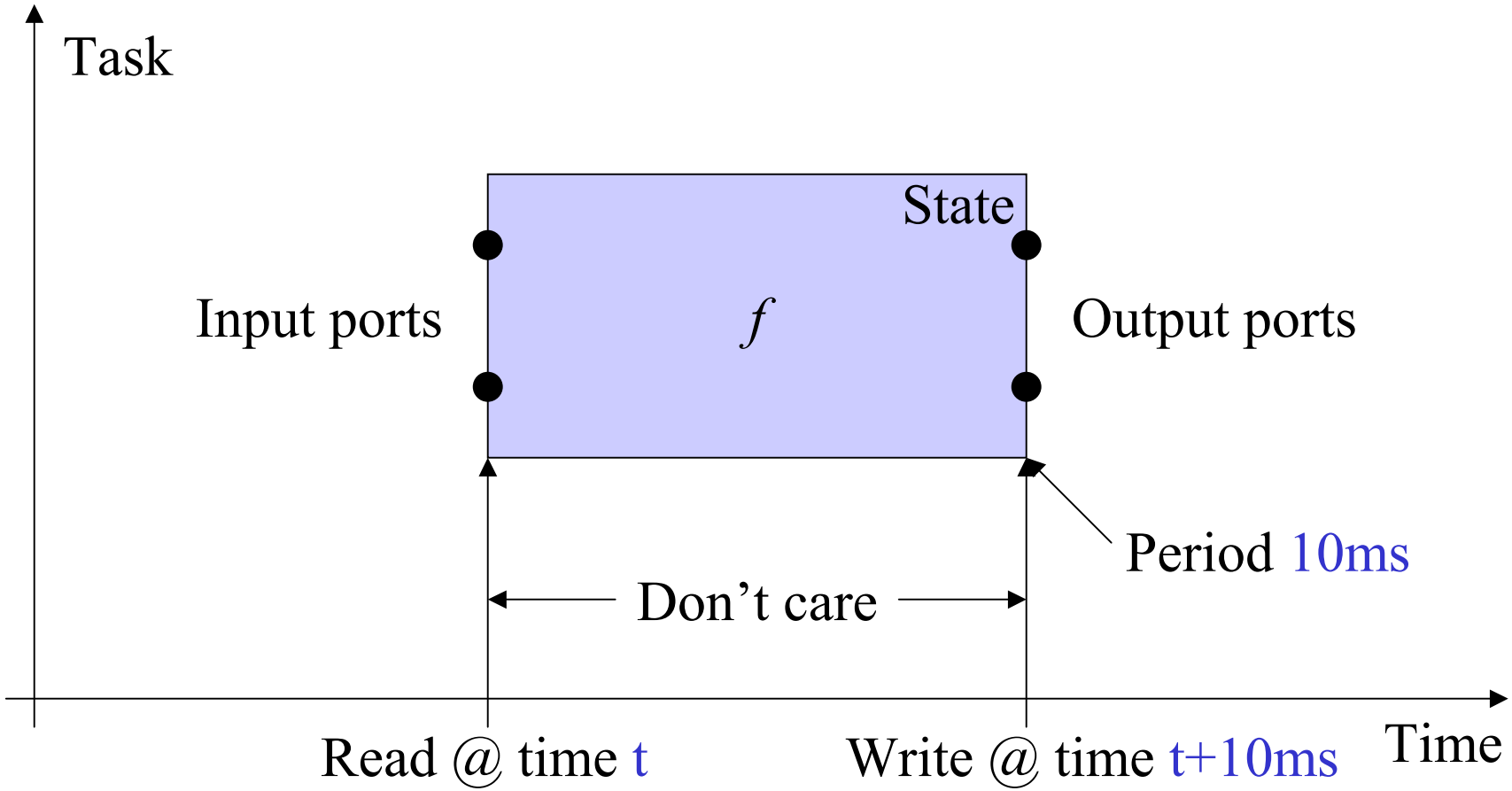
Semantics of a Task



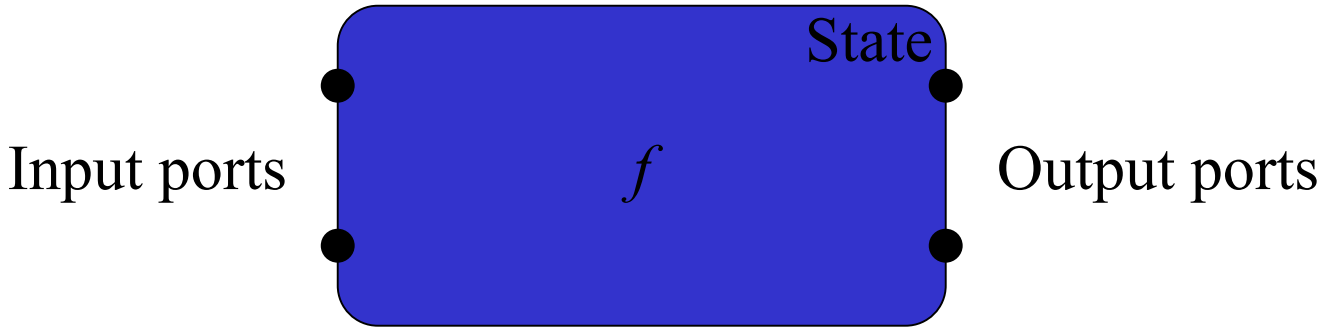
Semantics of a Task



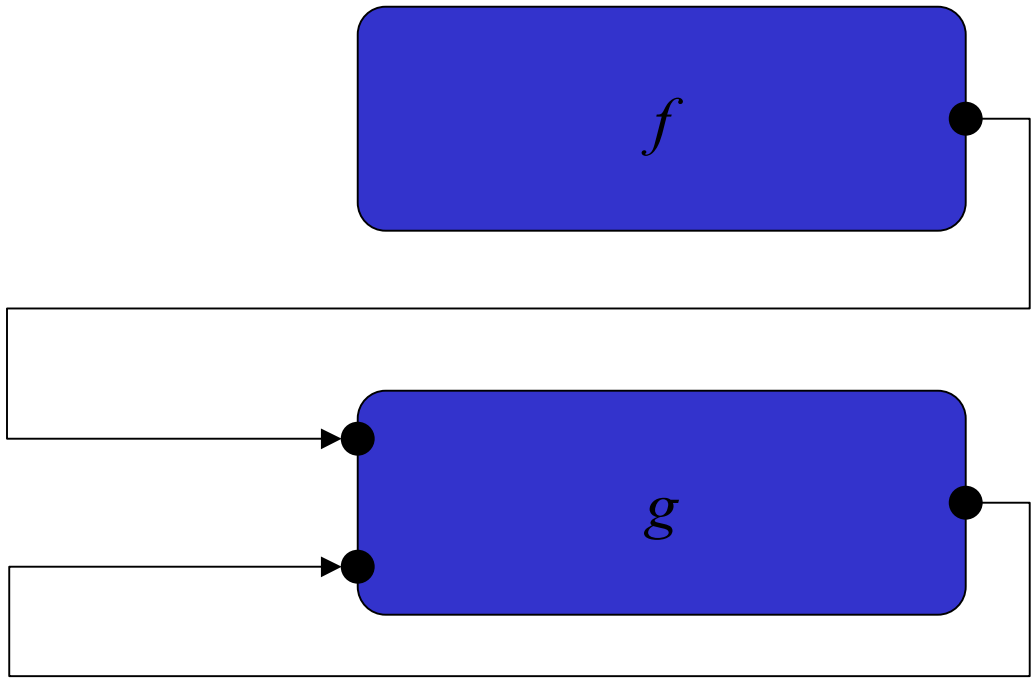
Semantics of a Task



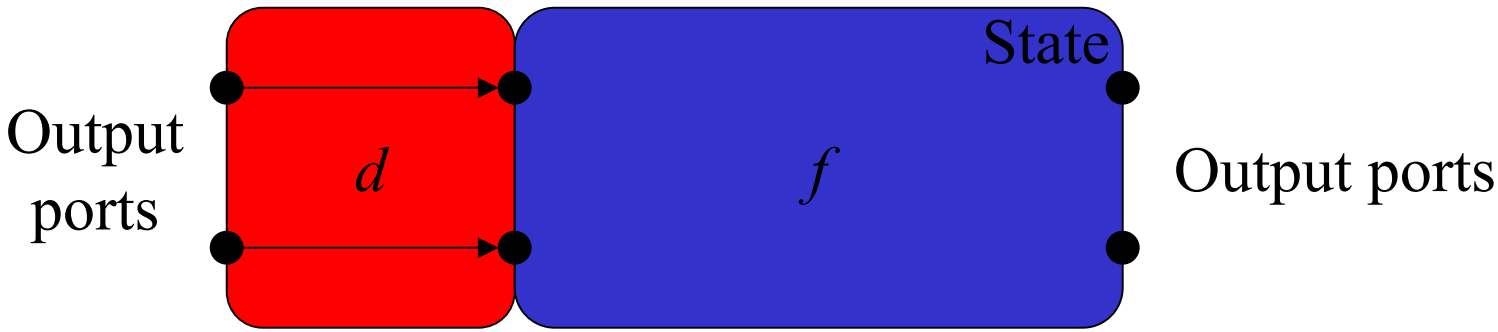
Communication



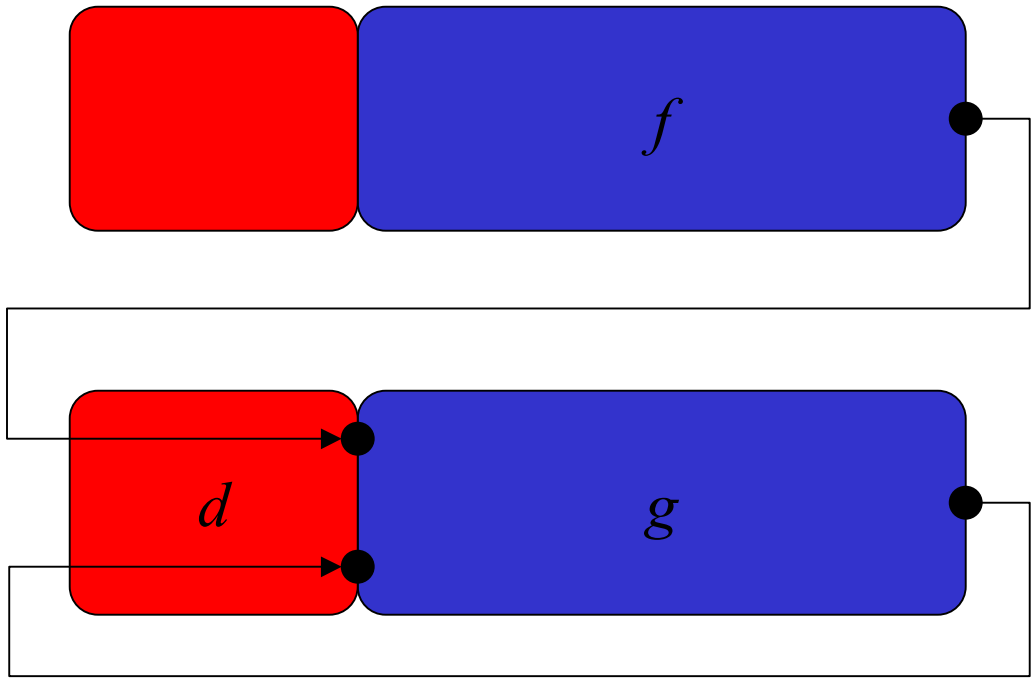
Intertask Communication



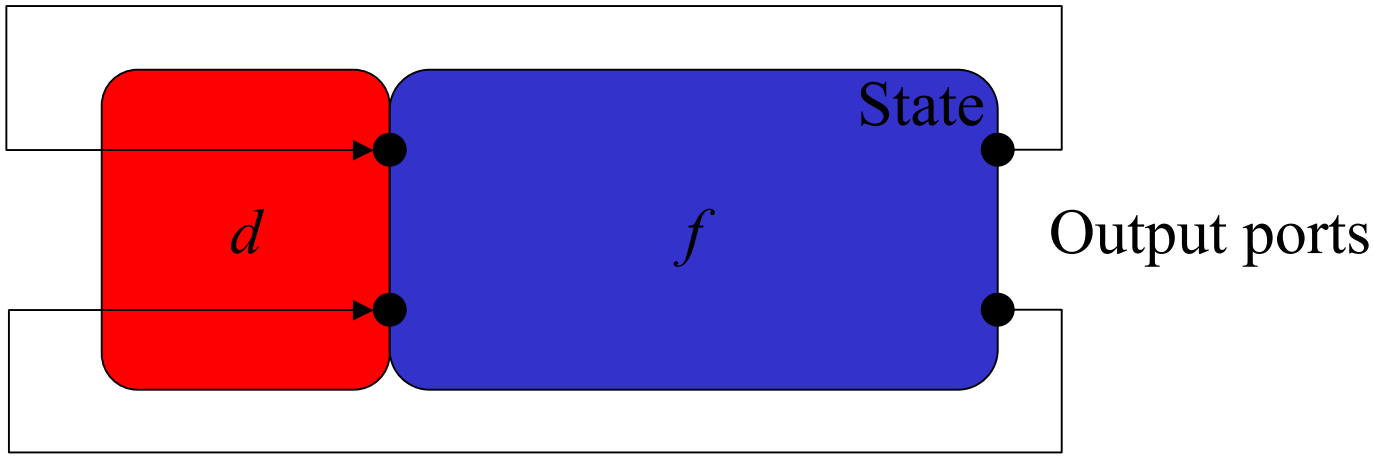
Task Driver



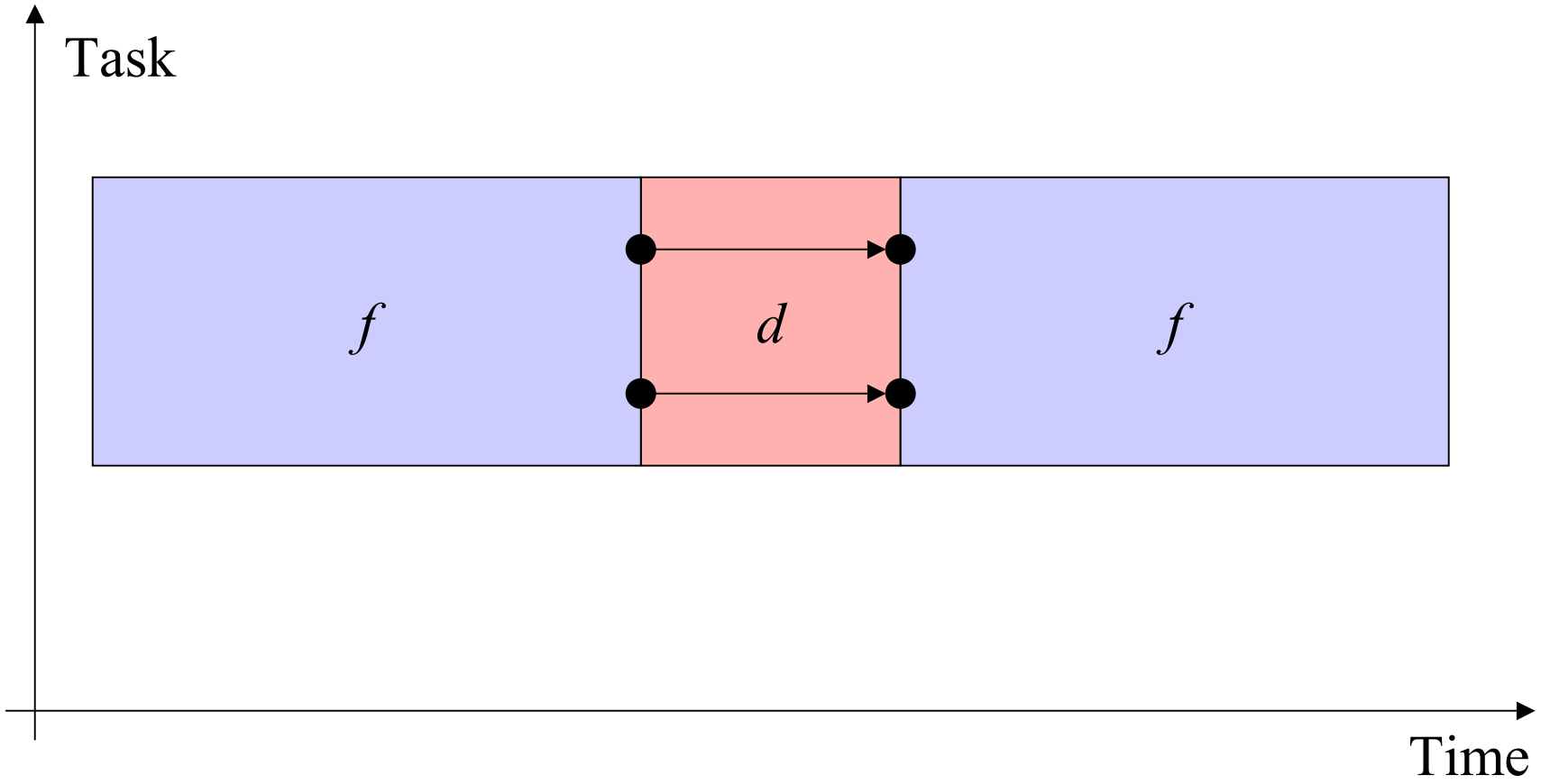
Communication



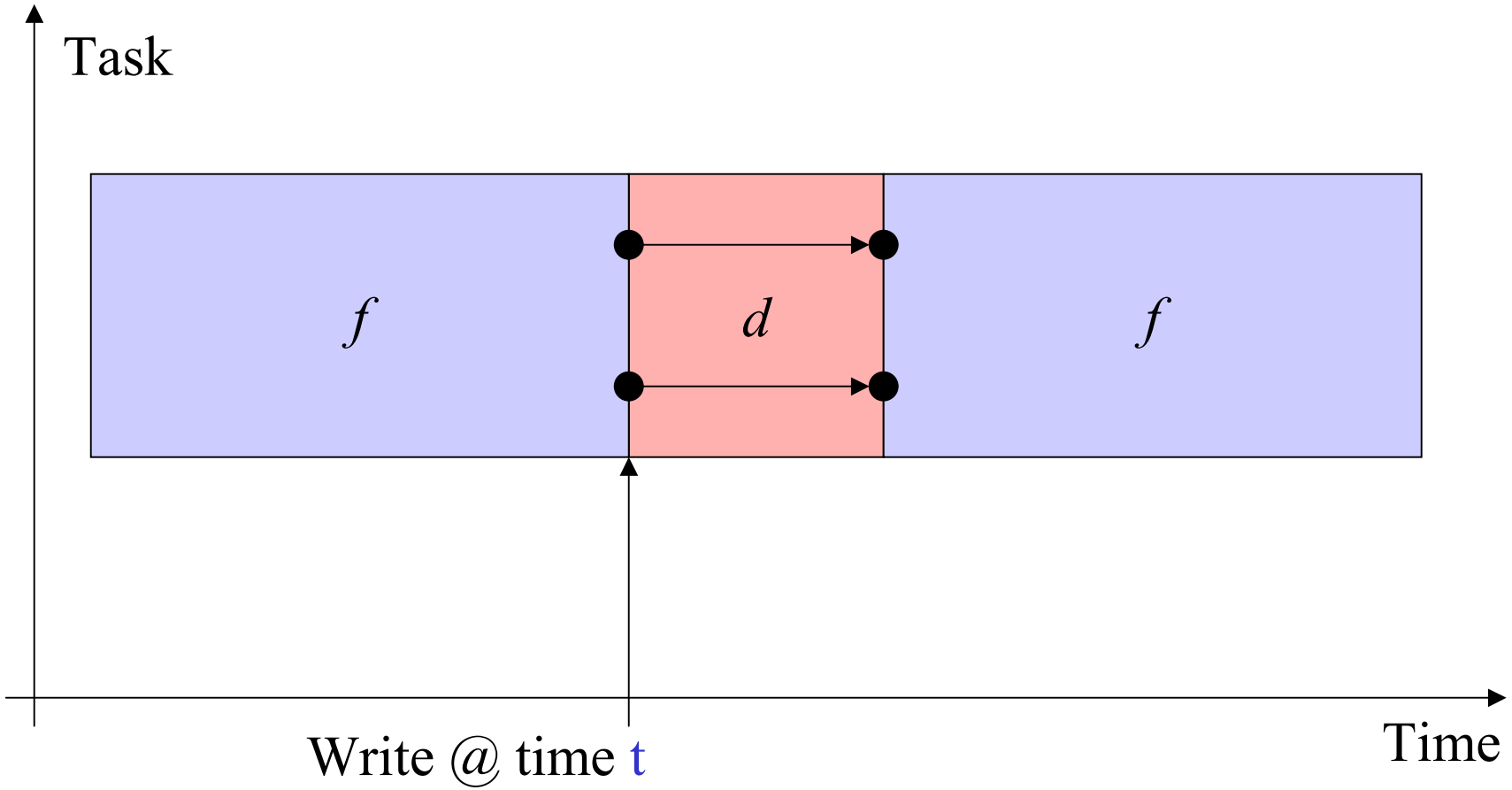
Communication Semantics



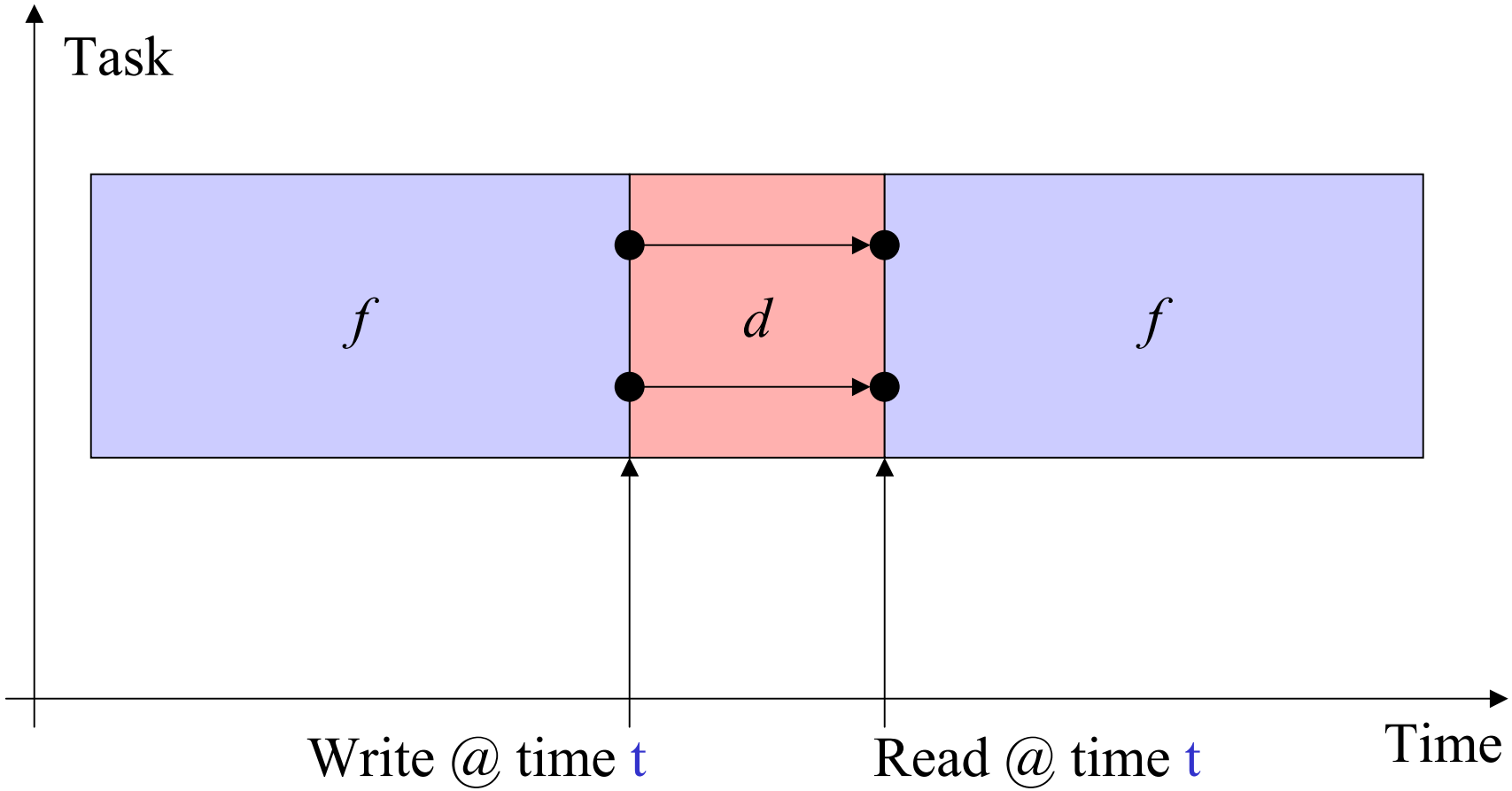
Communication Semantics



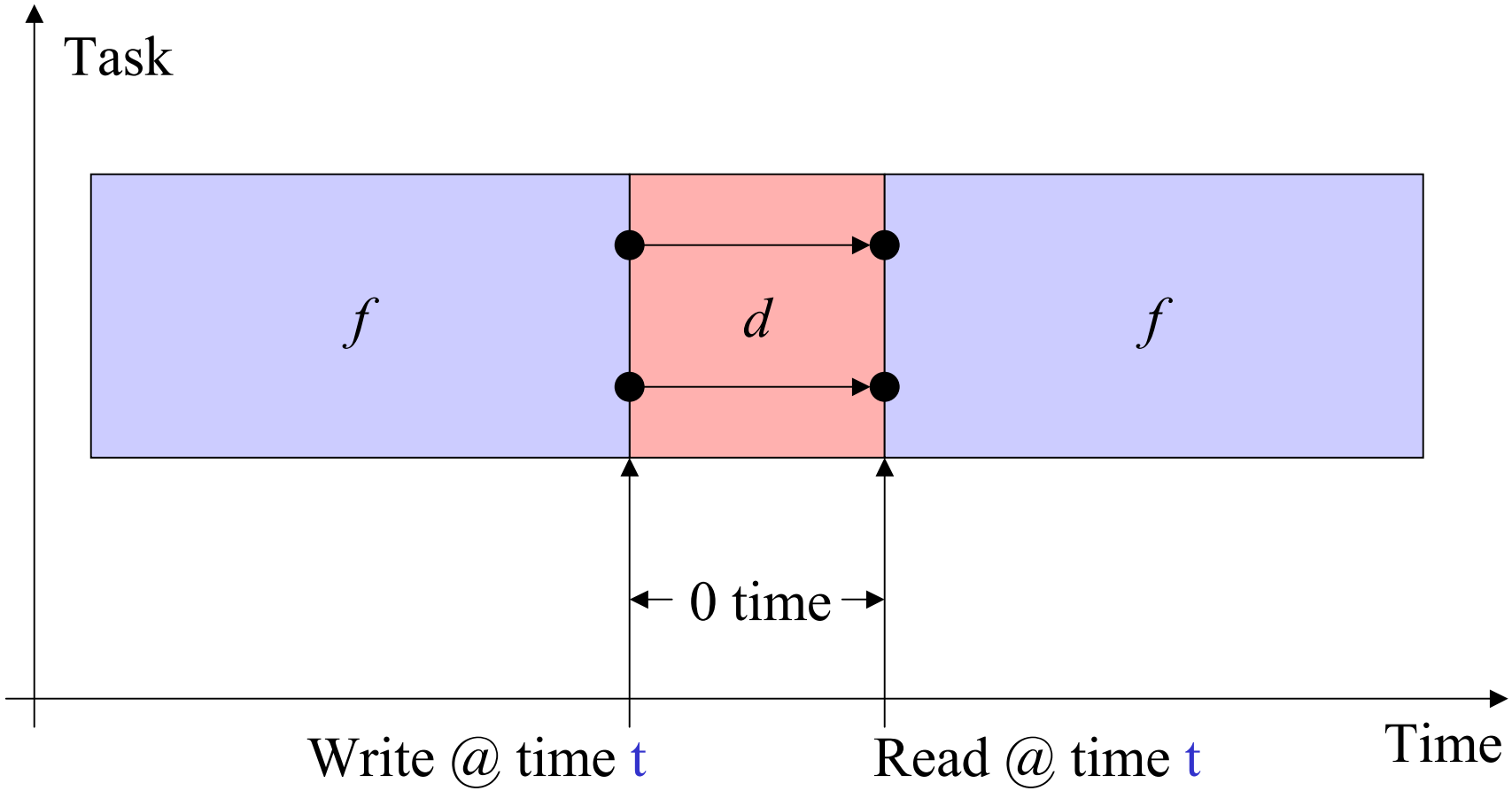
Communication Semantics



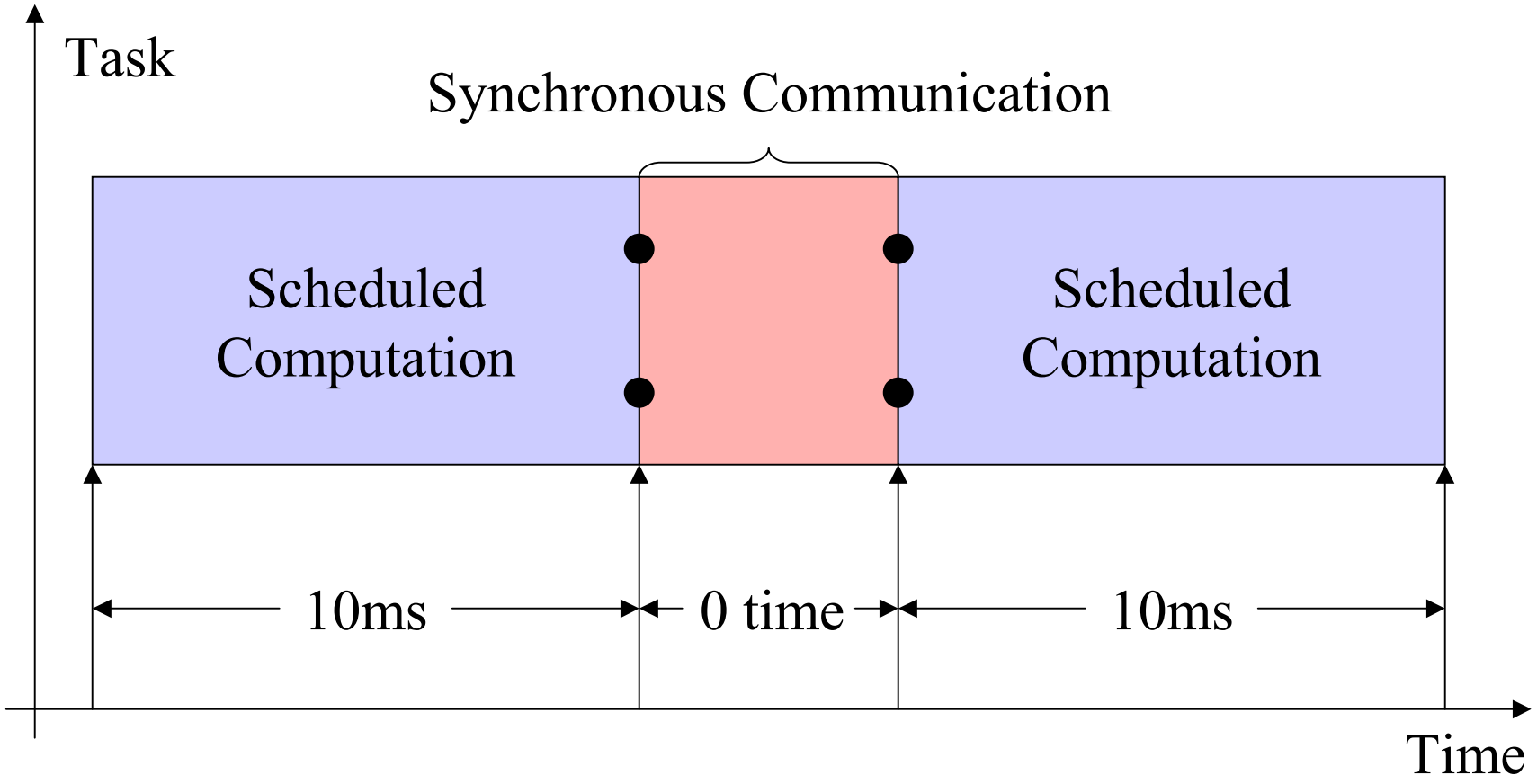
Communication Semantics



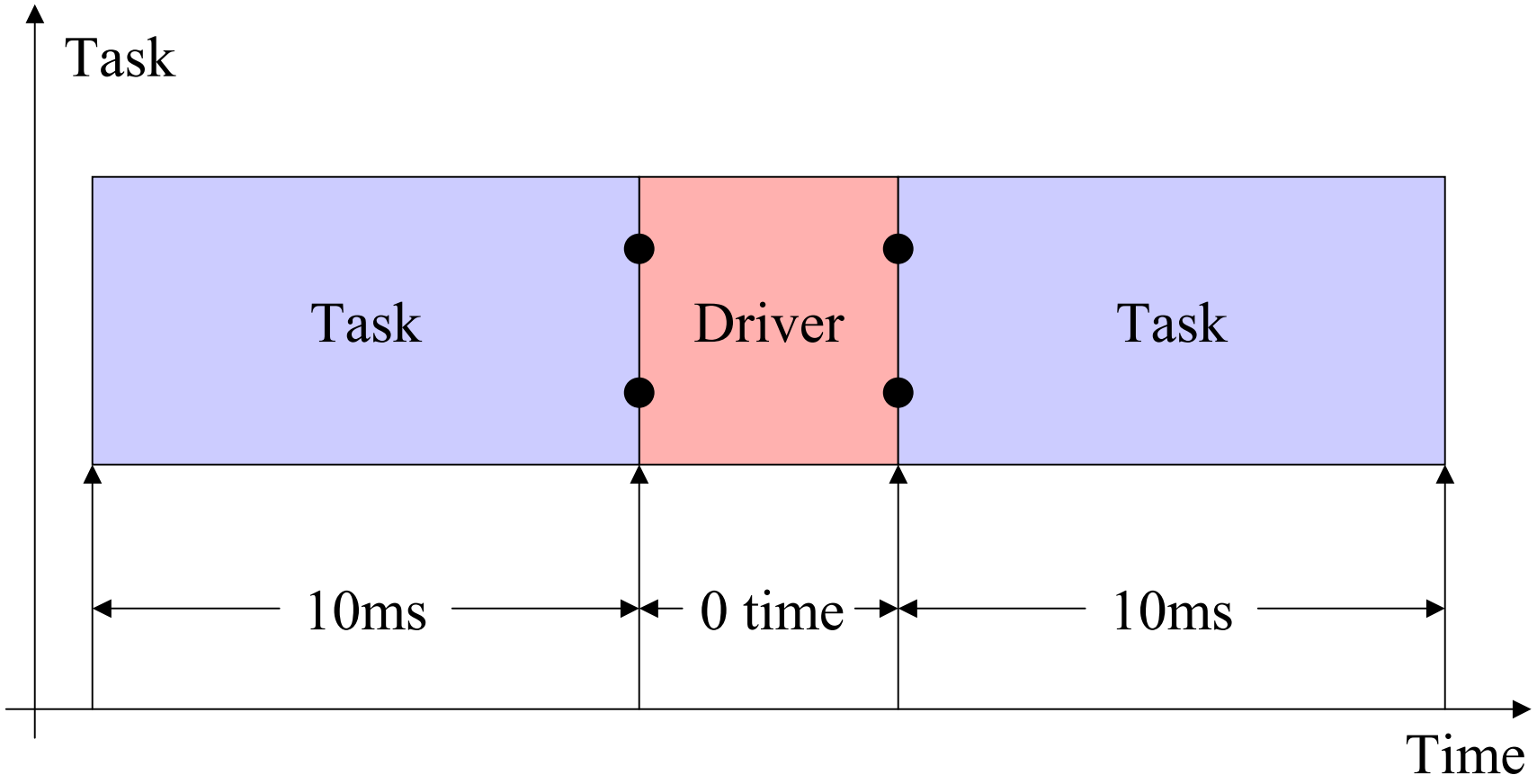
Communication Semantics



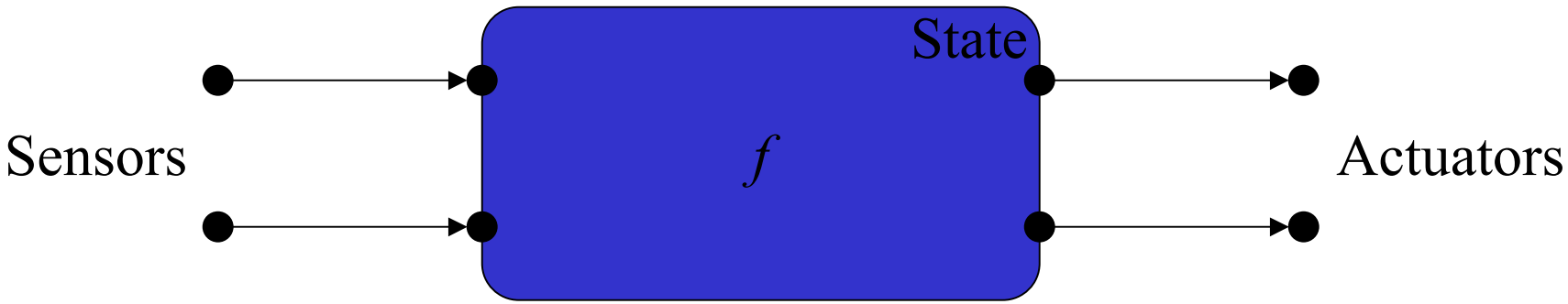
Scheduled vs. Synchronous



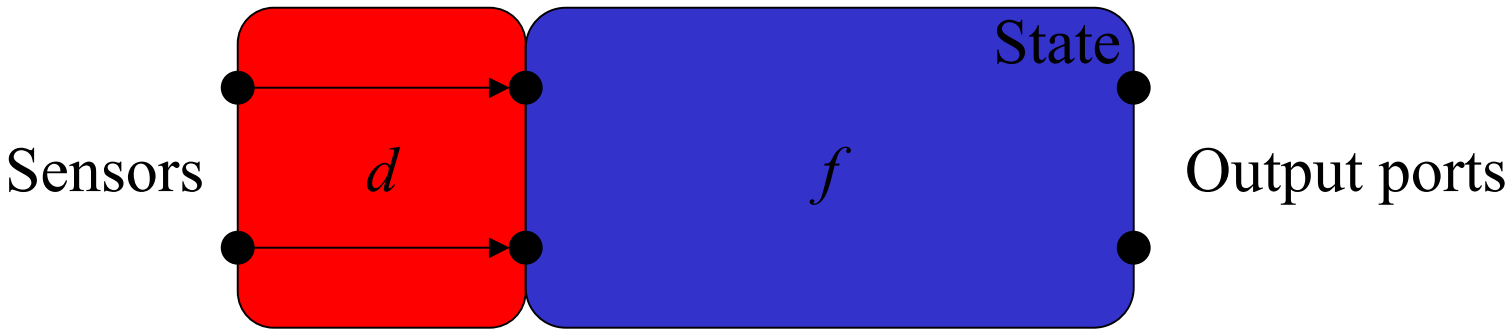
Task vs. Driver



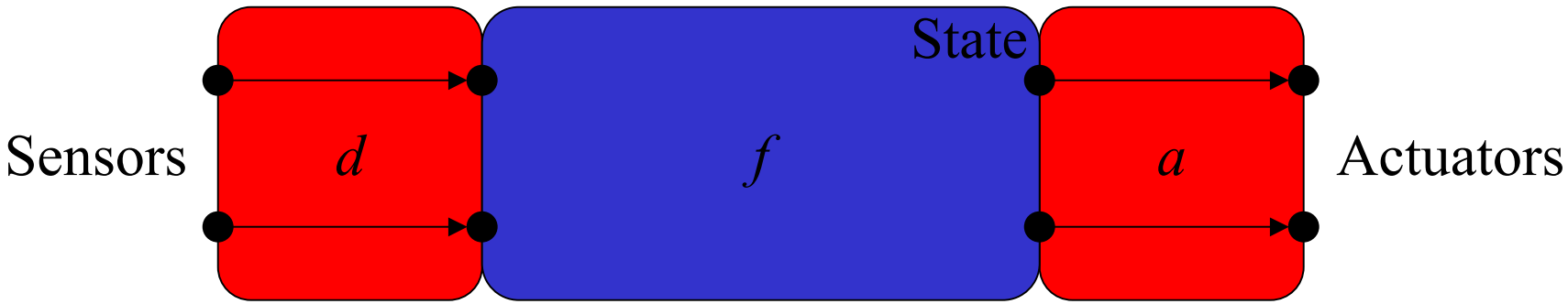
Sensor - Control Law - Actuator



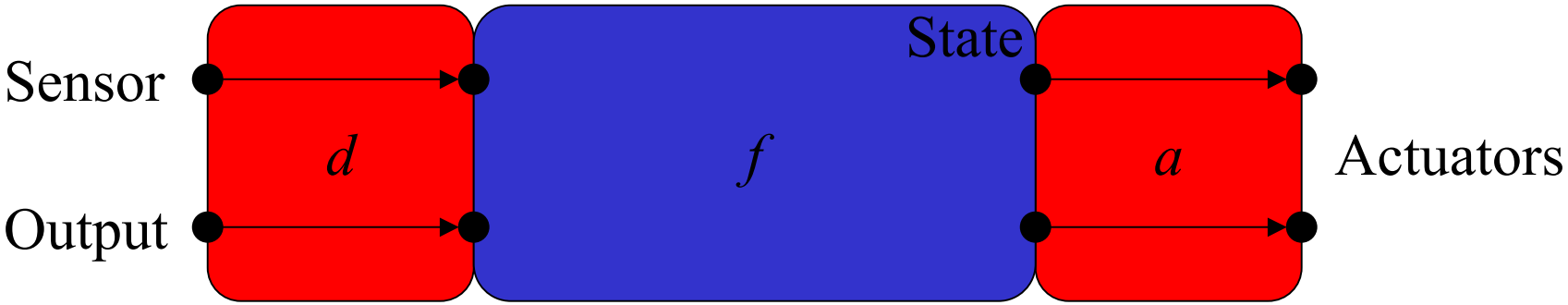
Task Driver



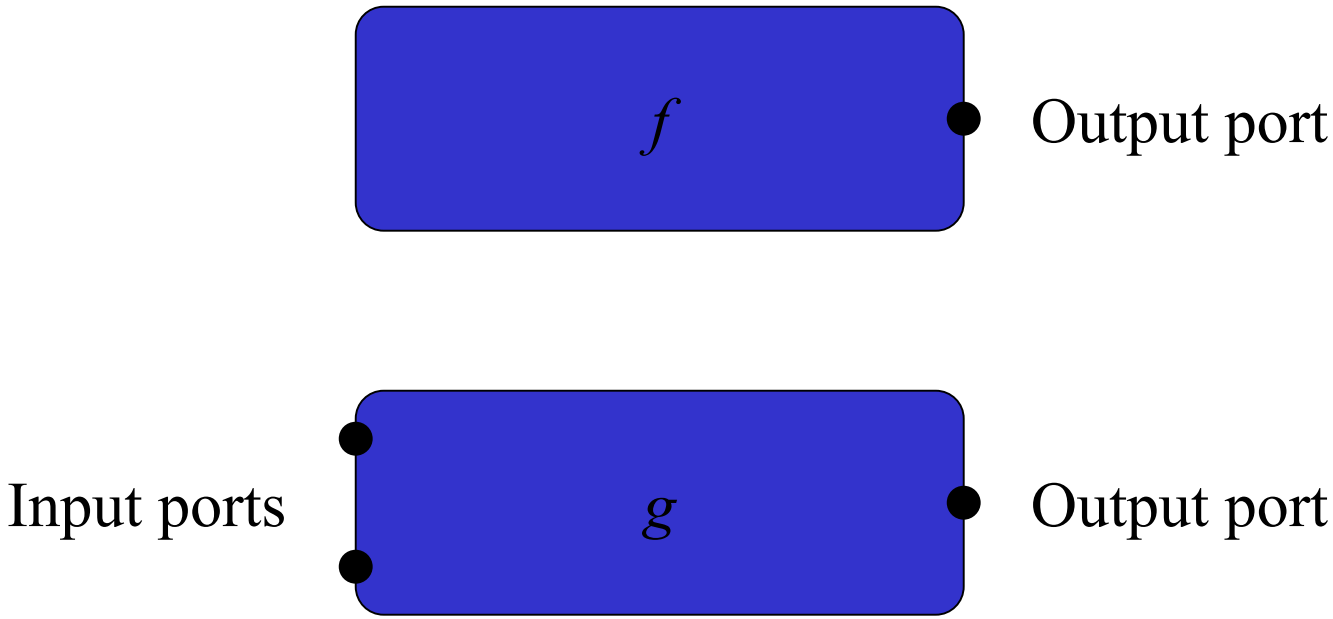
Actuator Driver



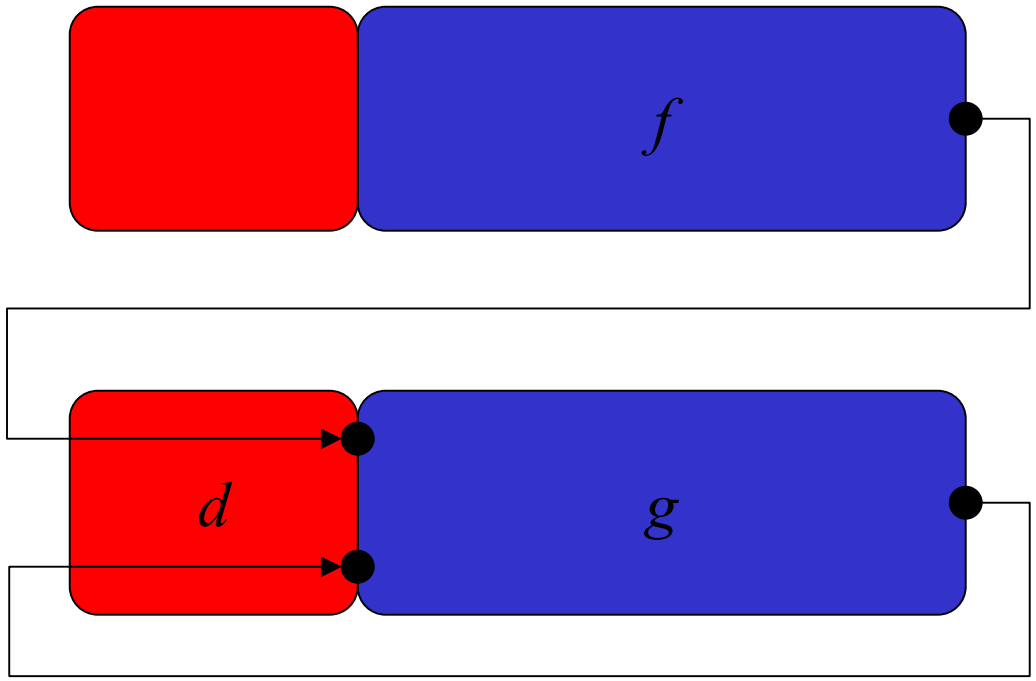
Sensors, Actuators, and Outputs



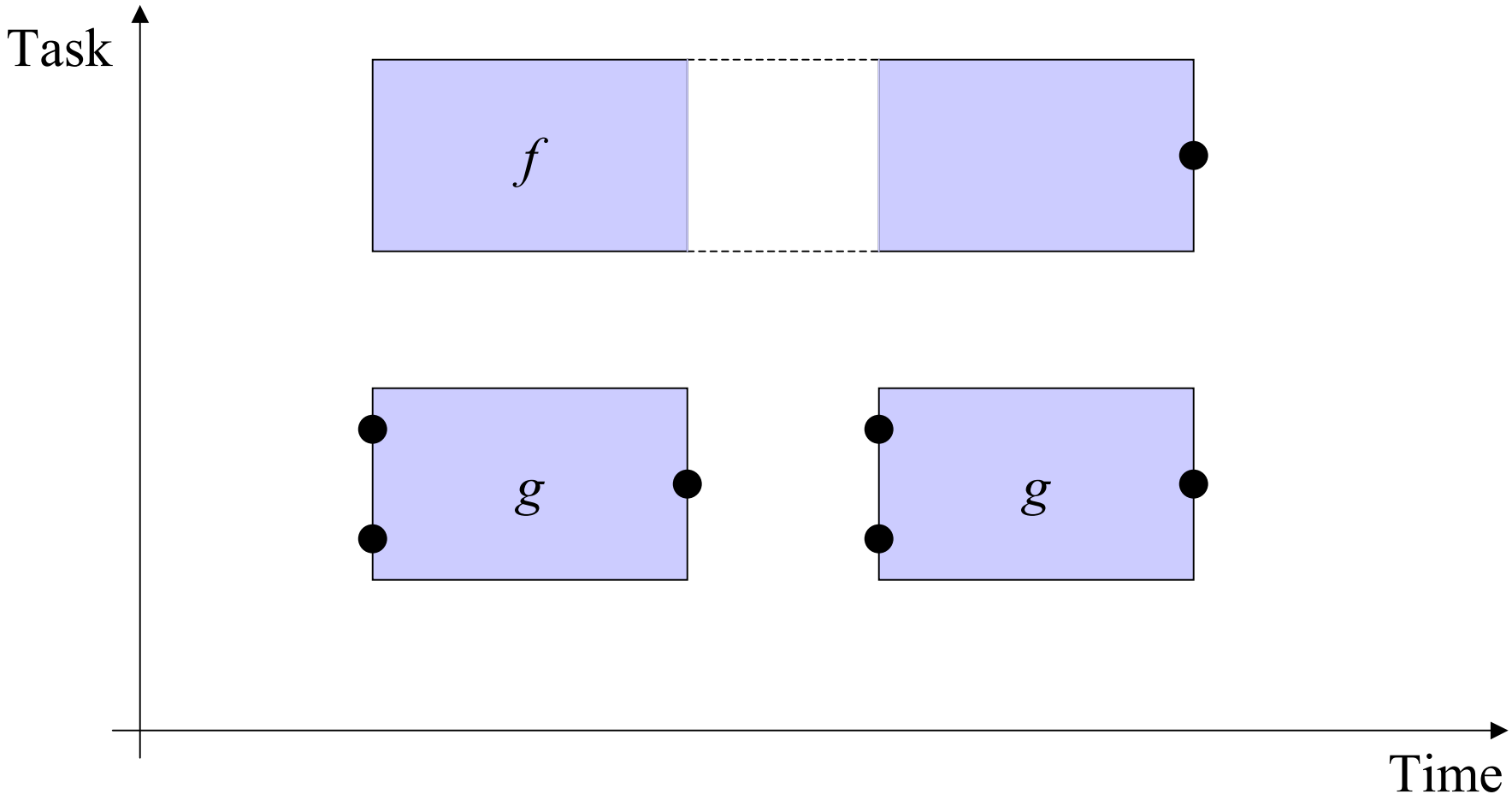
Concurrency



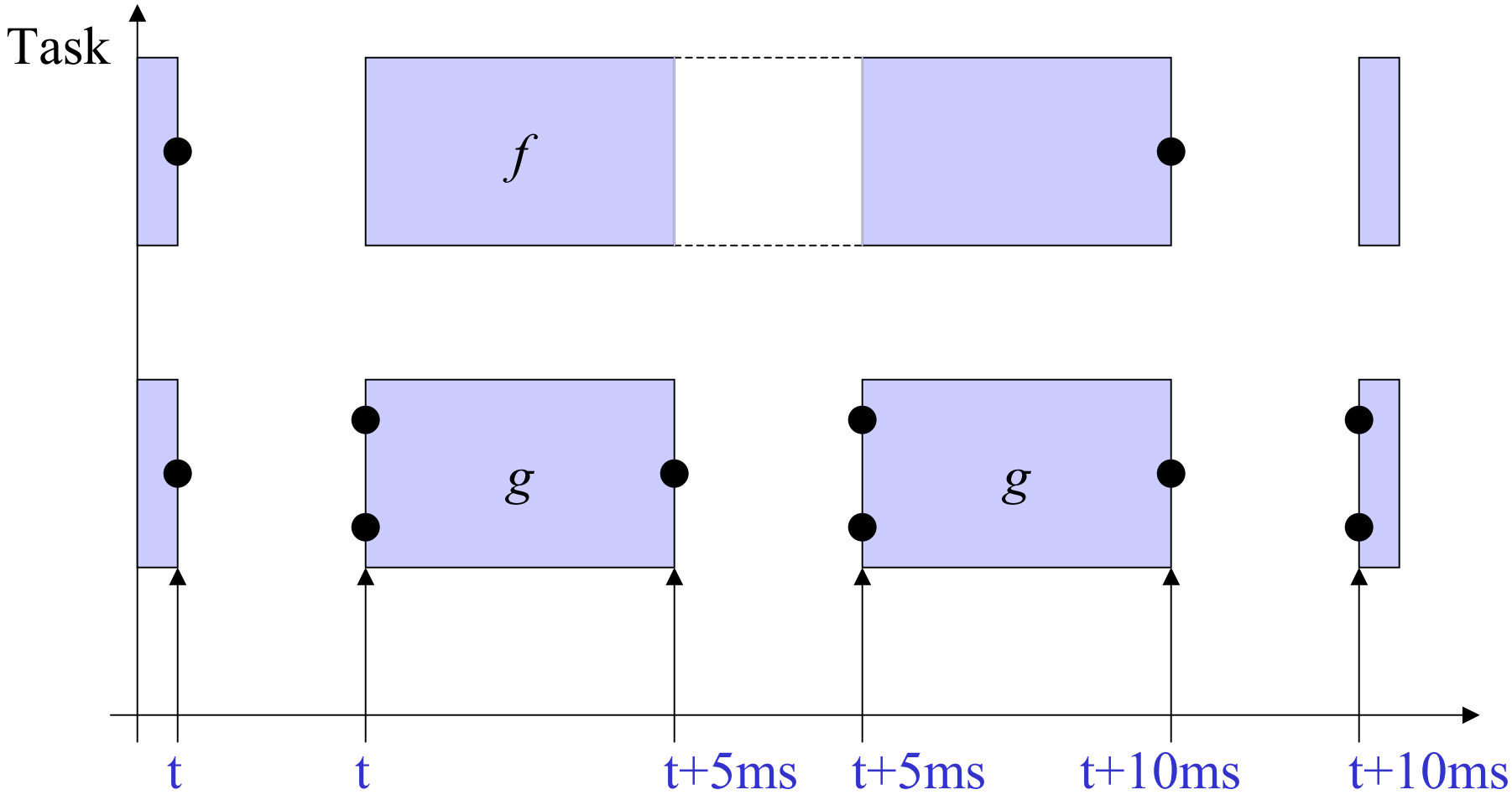
Communication



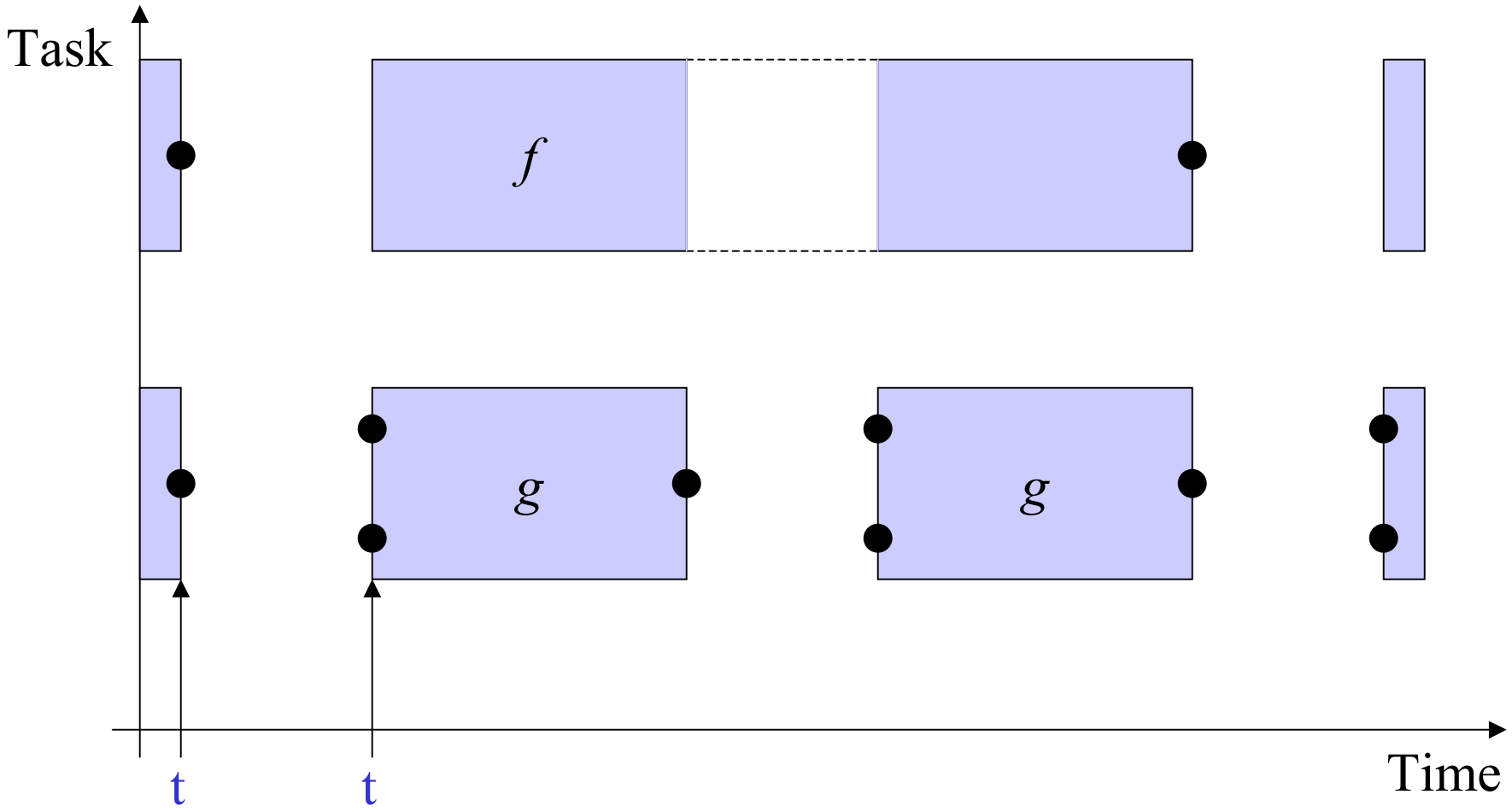
Different Periodicity



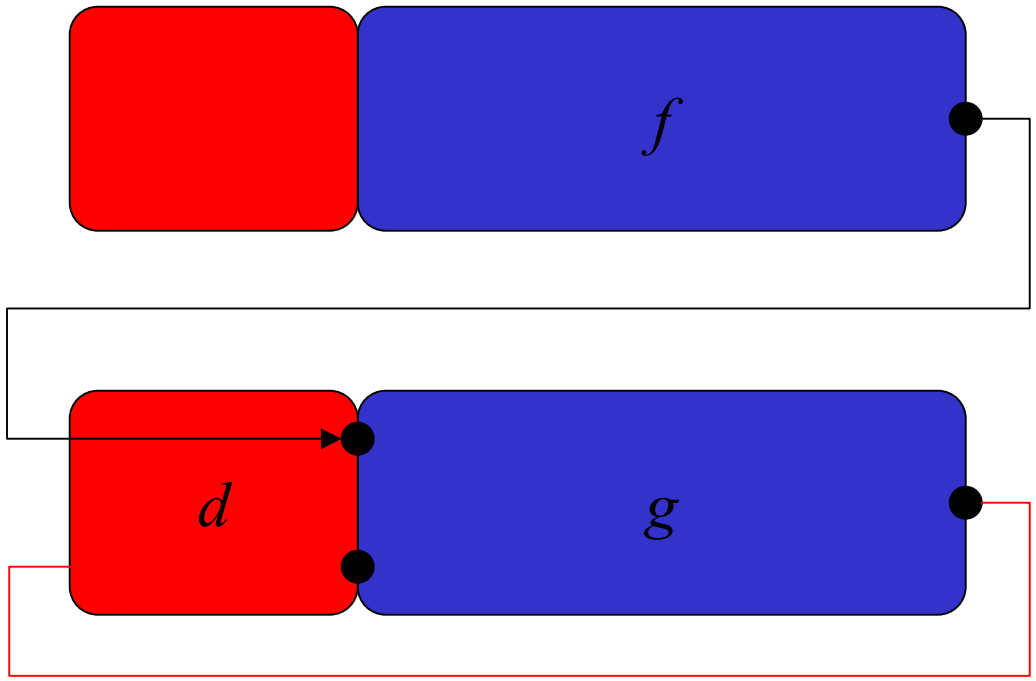
Semantics



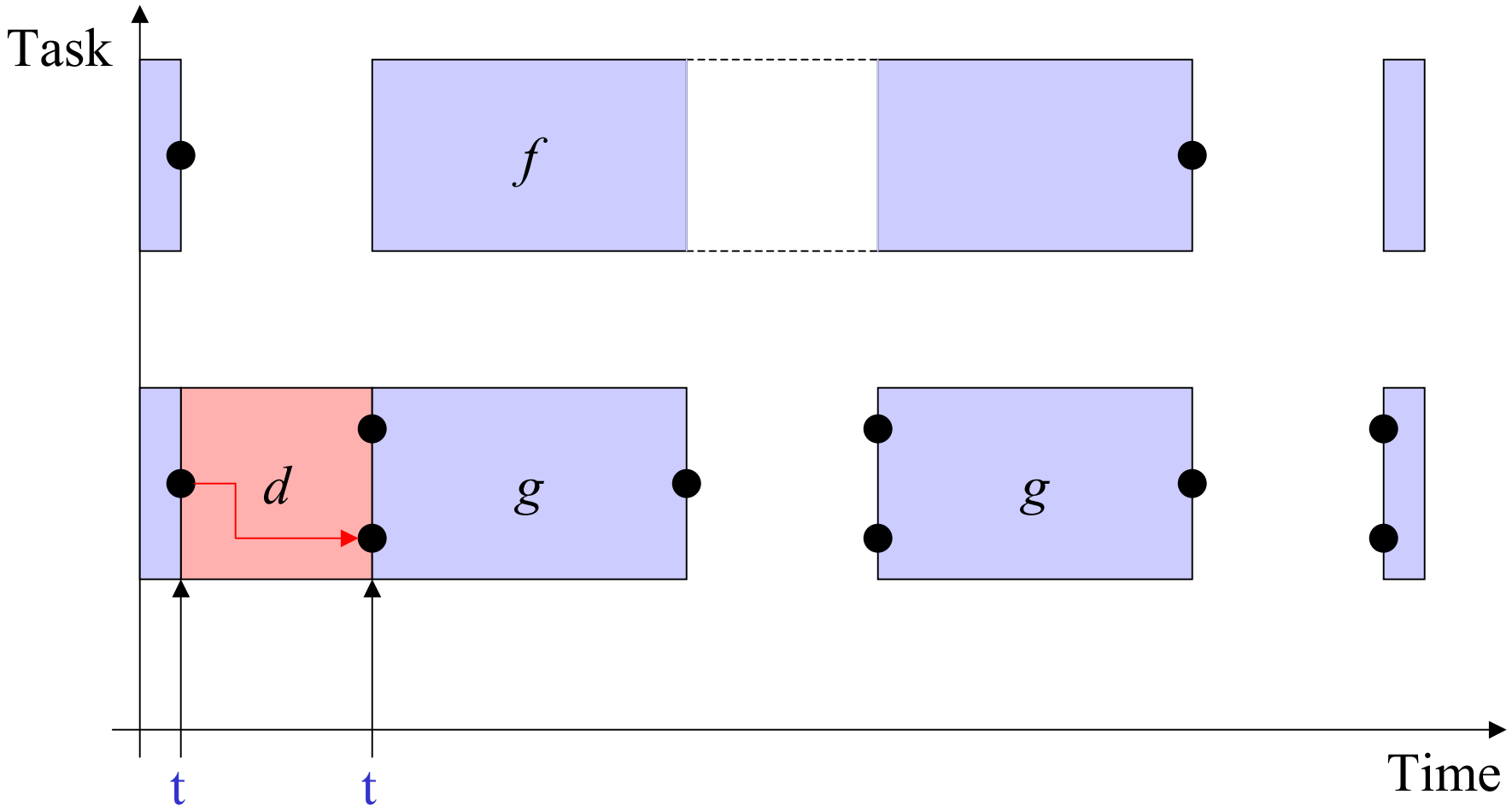
Semantics



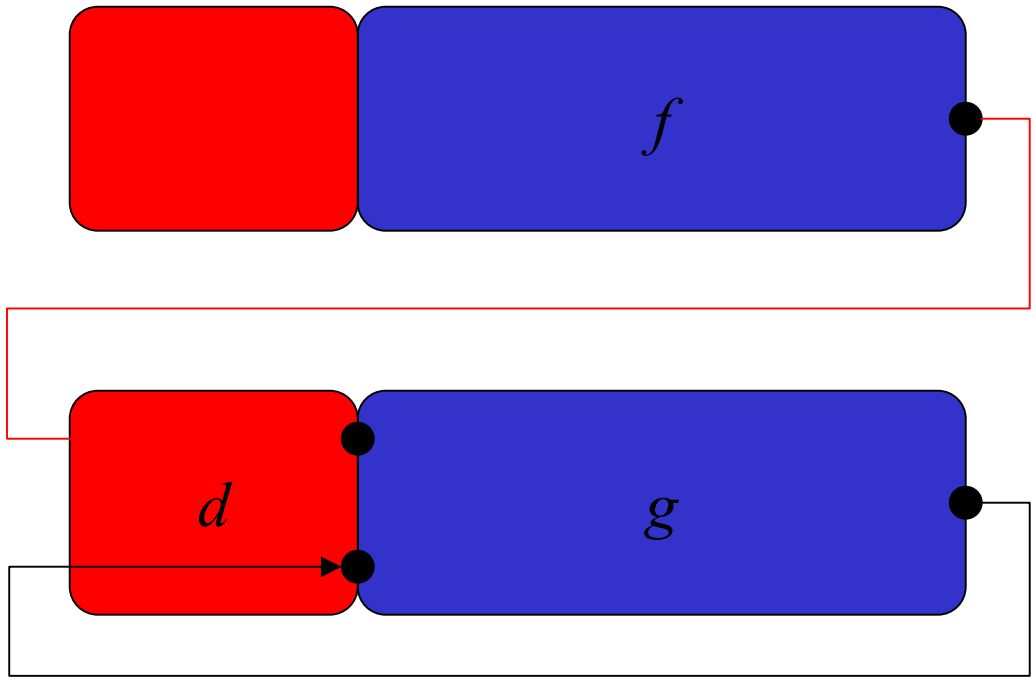
Communication



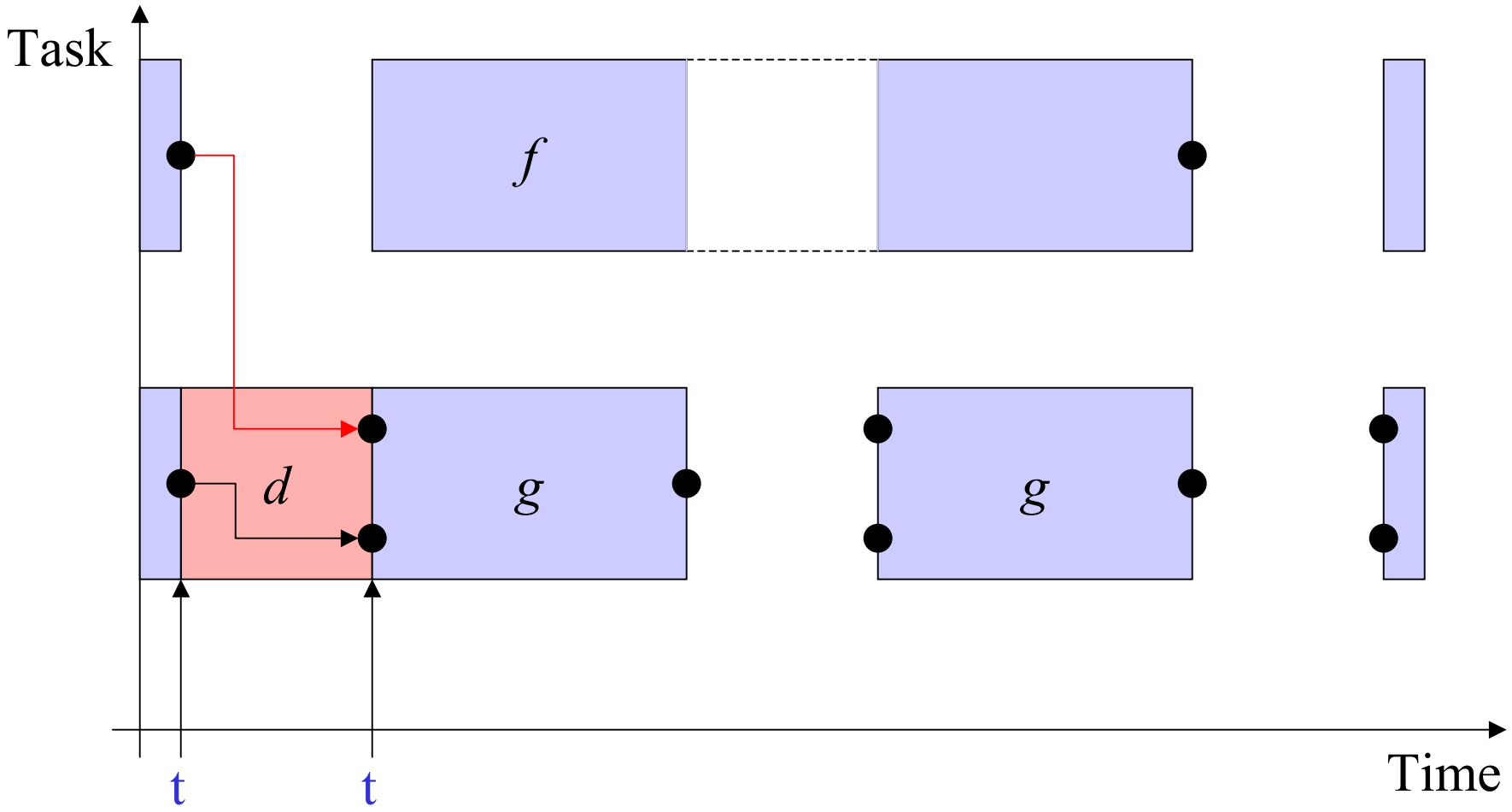
Semantics



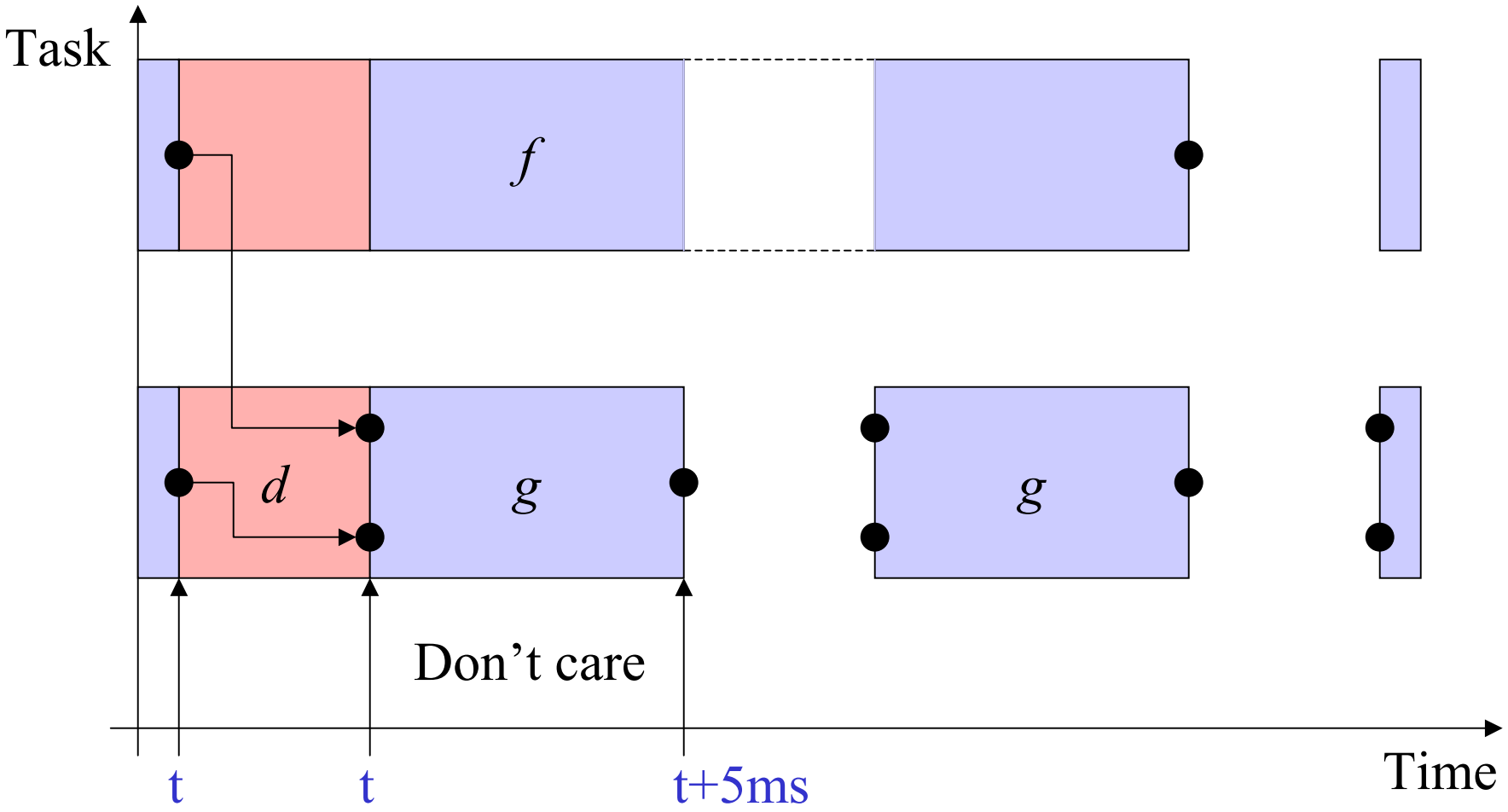
Communication



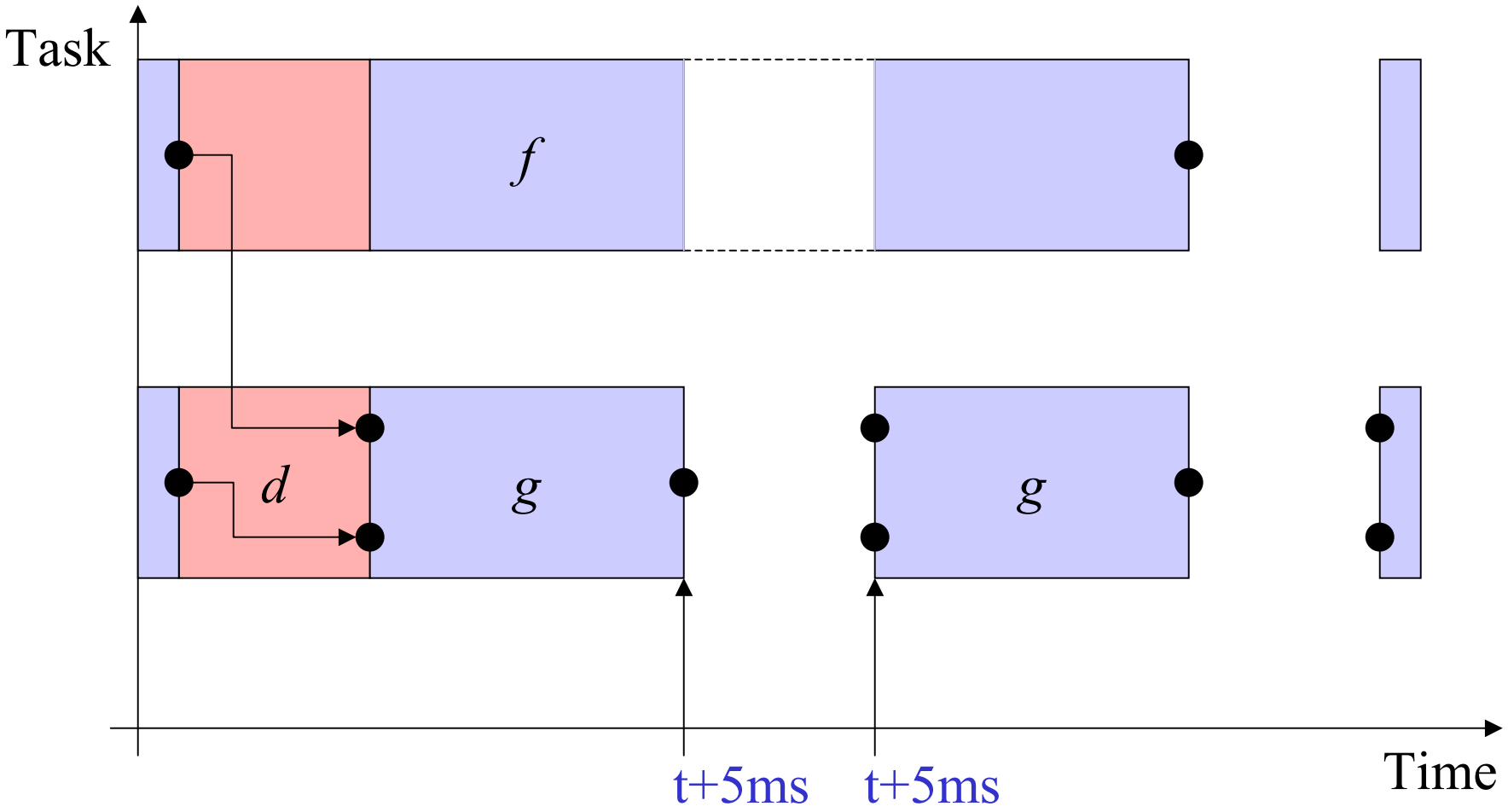
Semantics



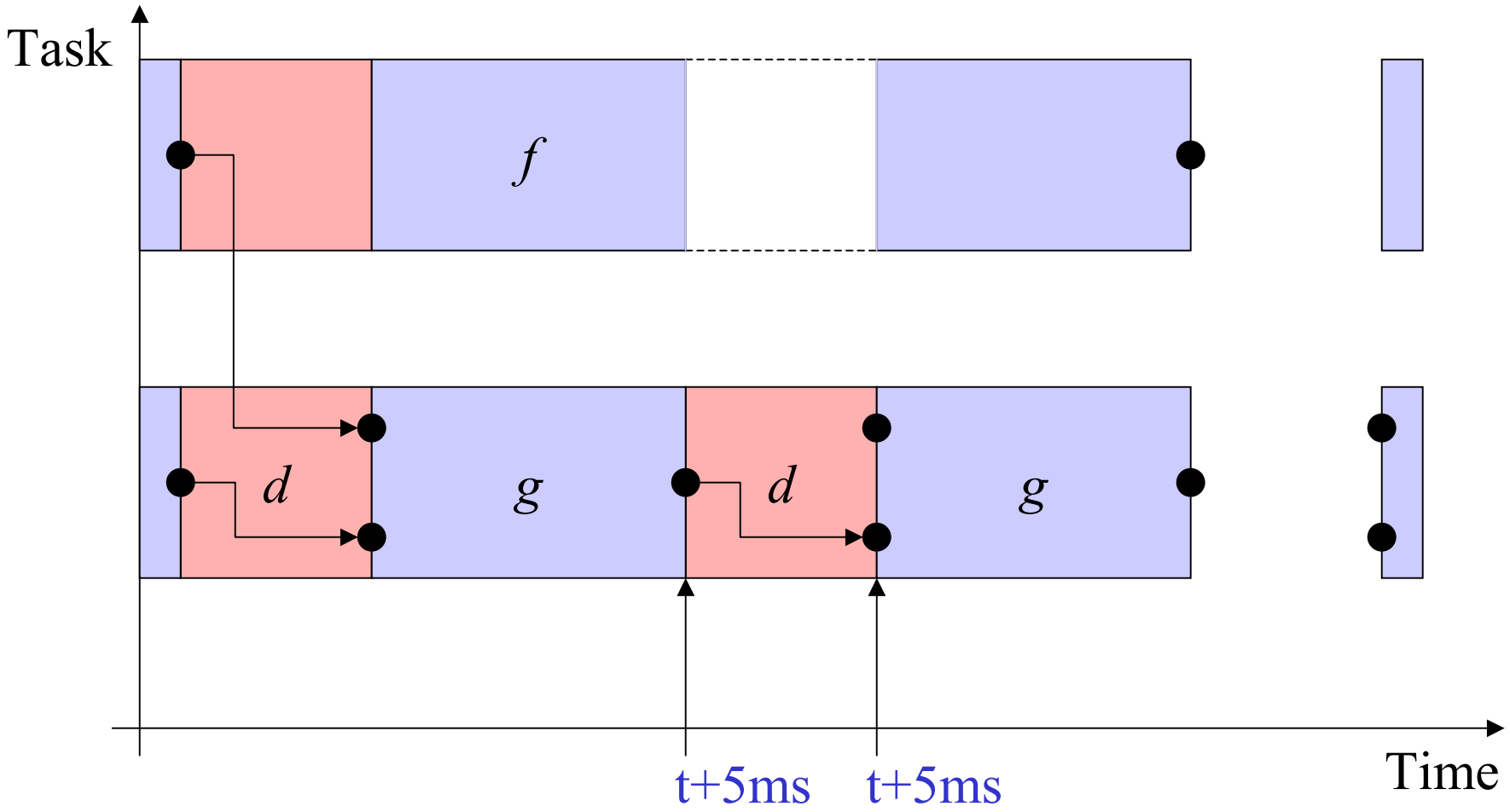
Semantics



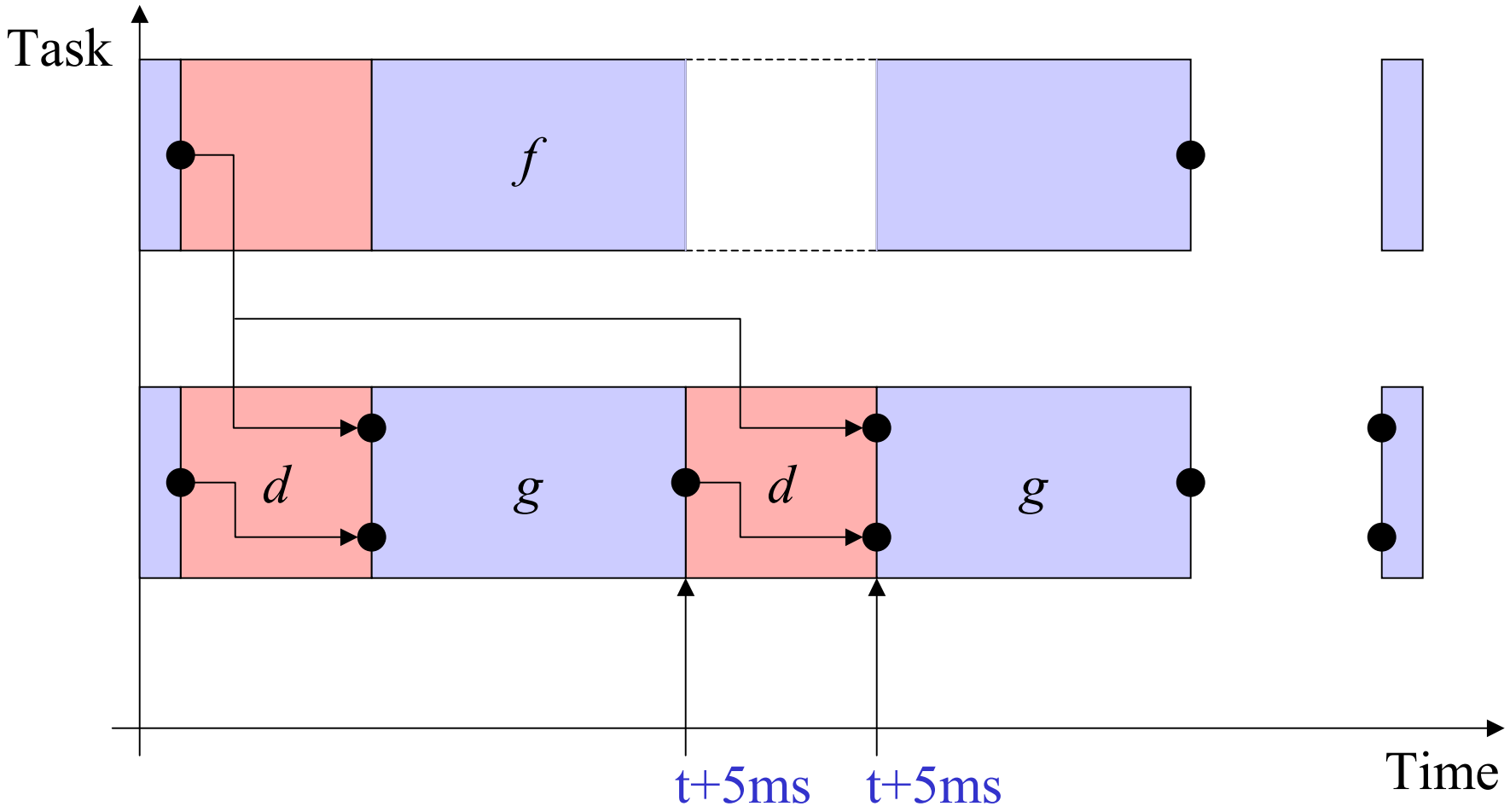
Semantics



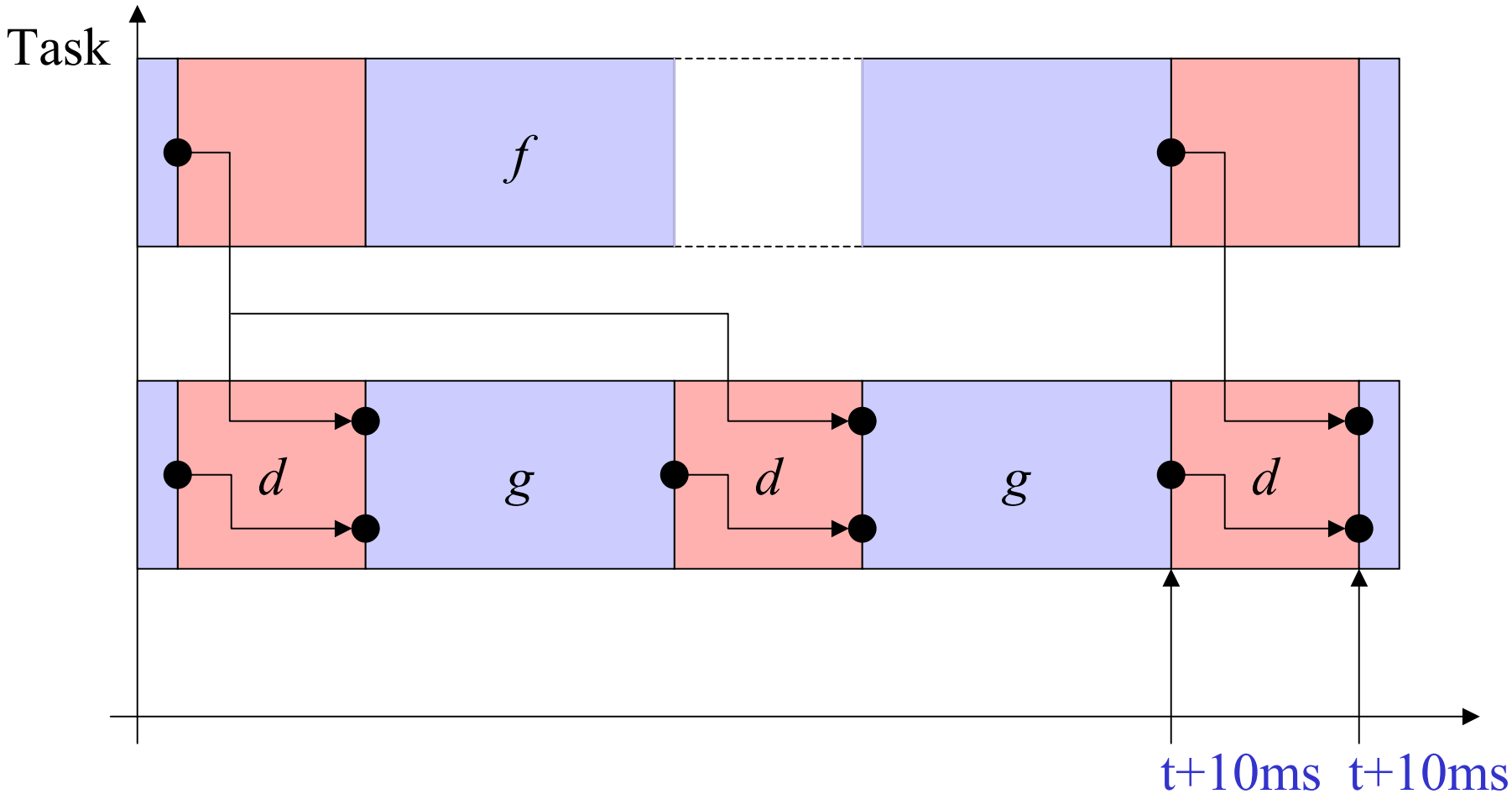
Semantics



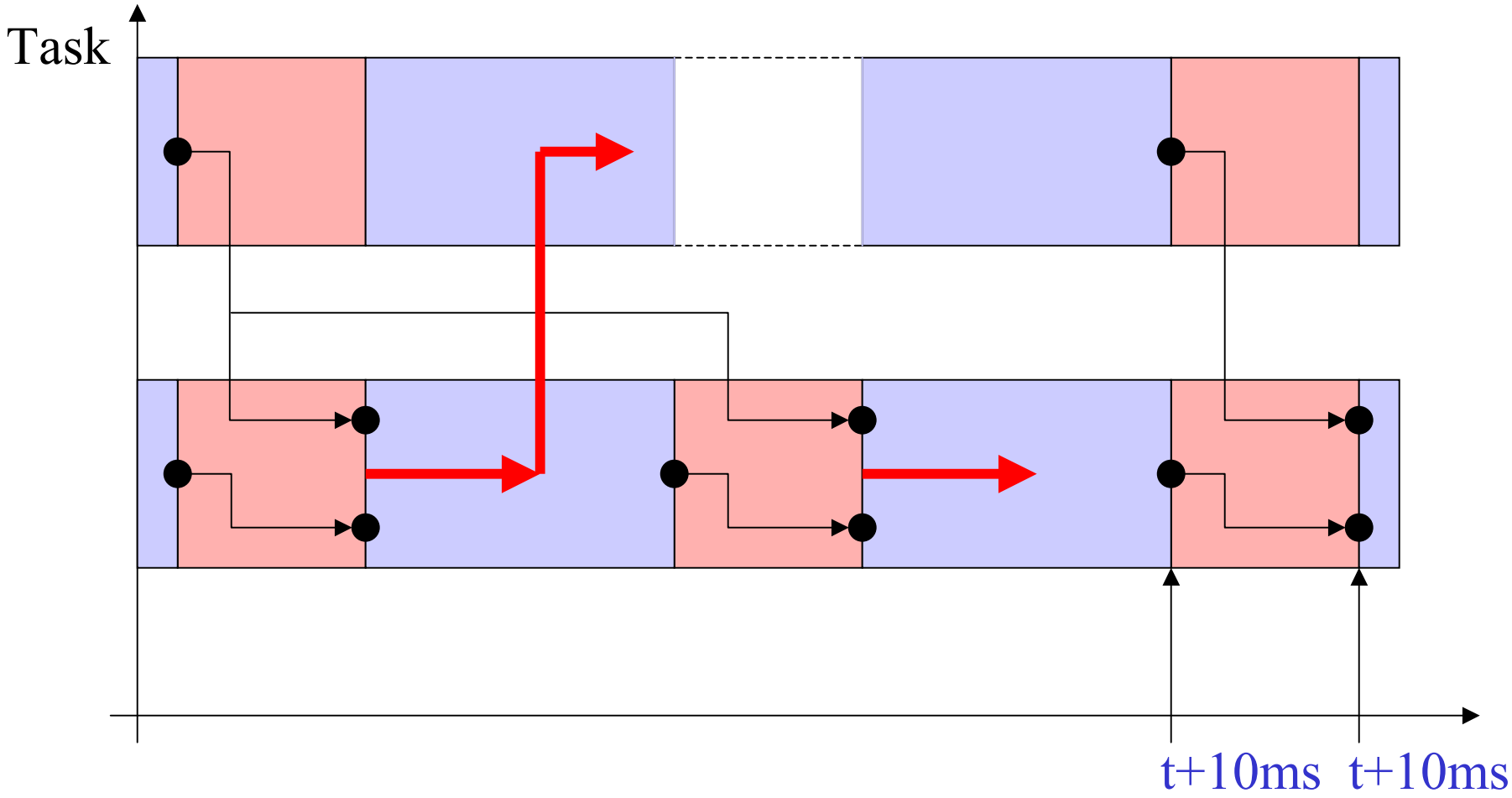
Semantics



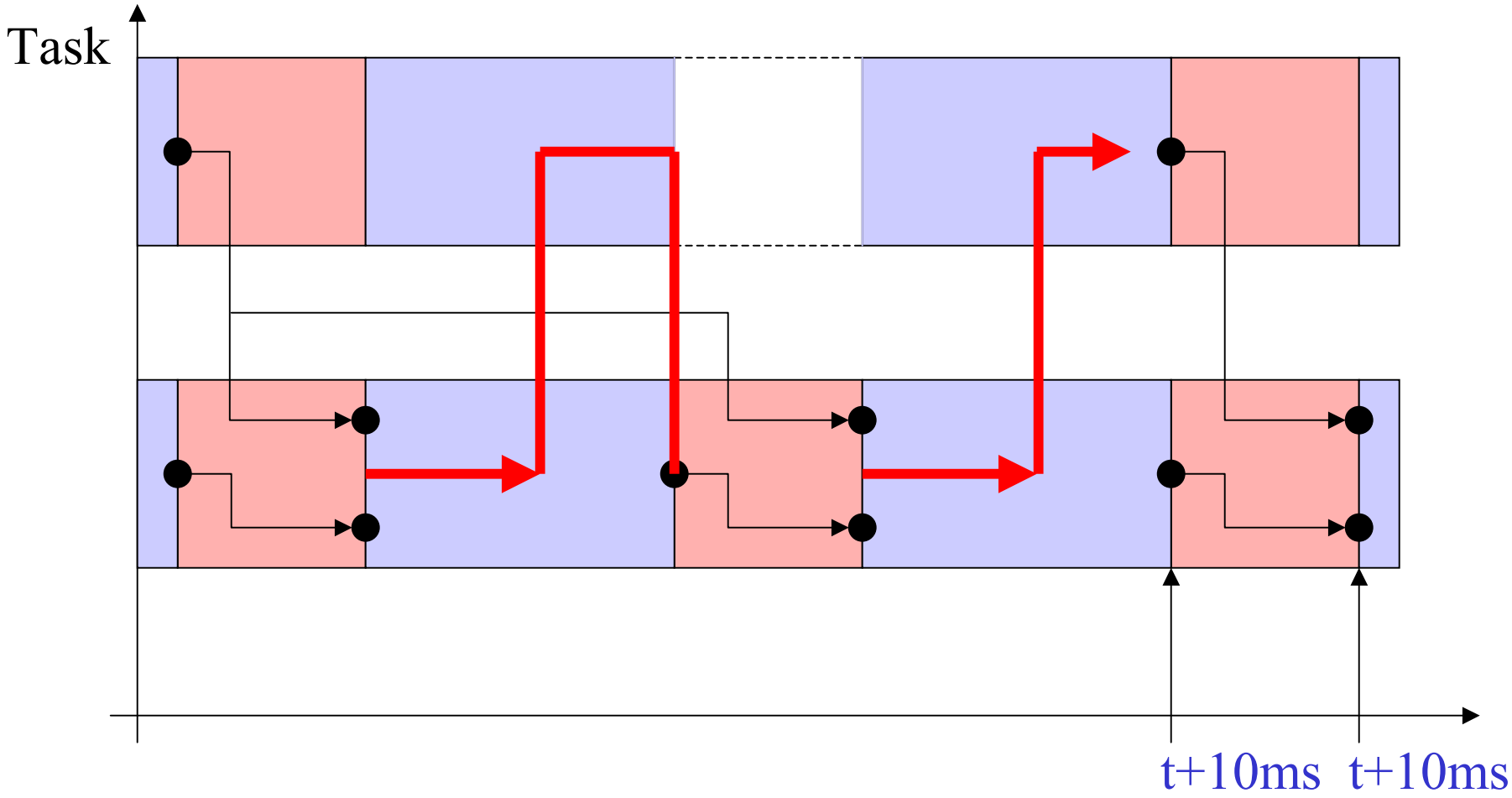
Semantics



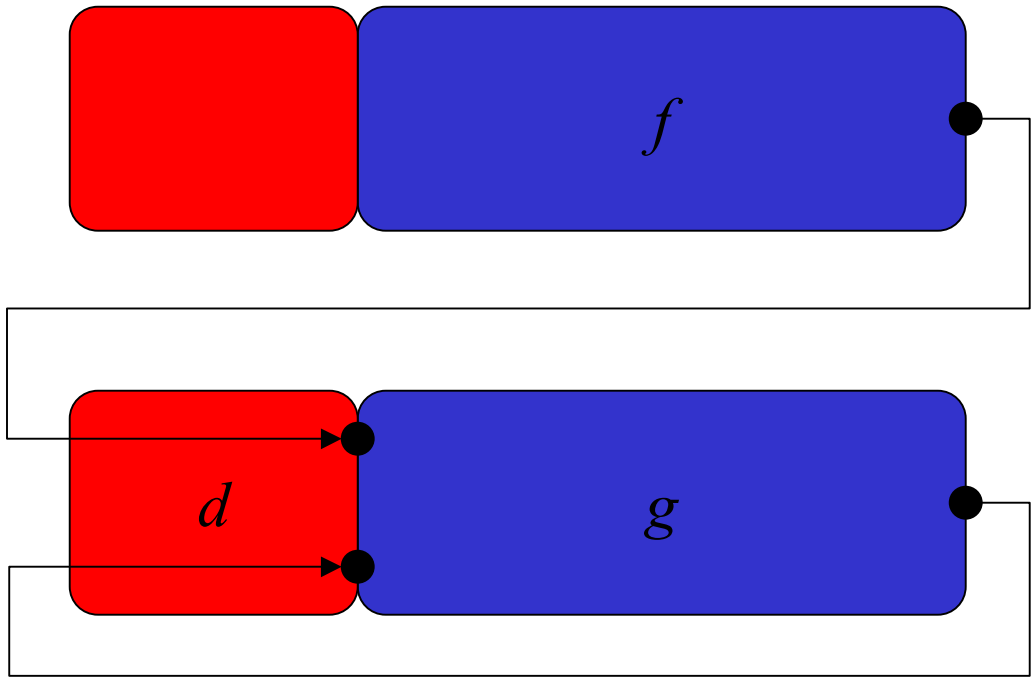
RMA



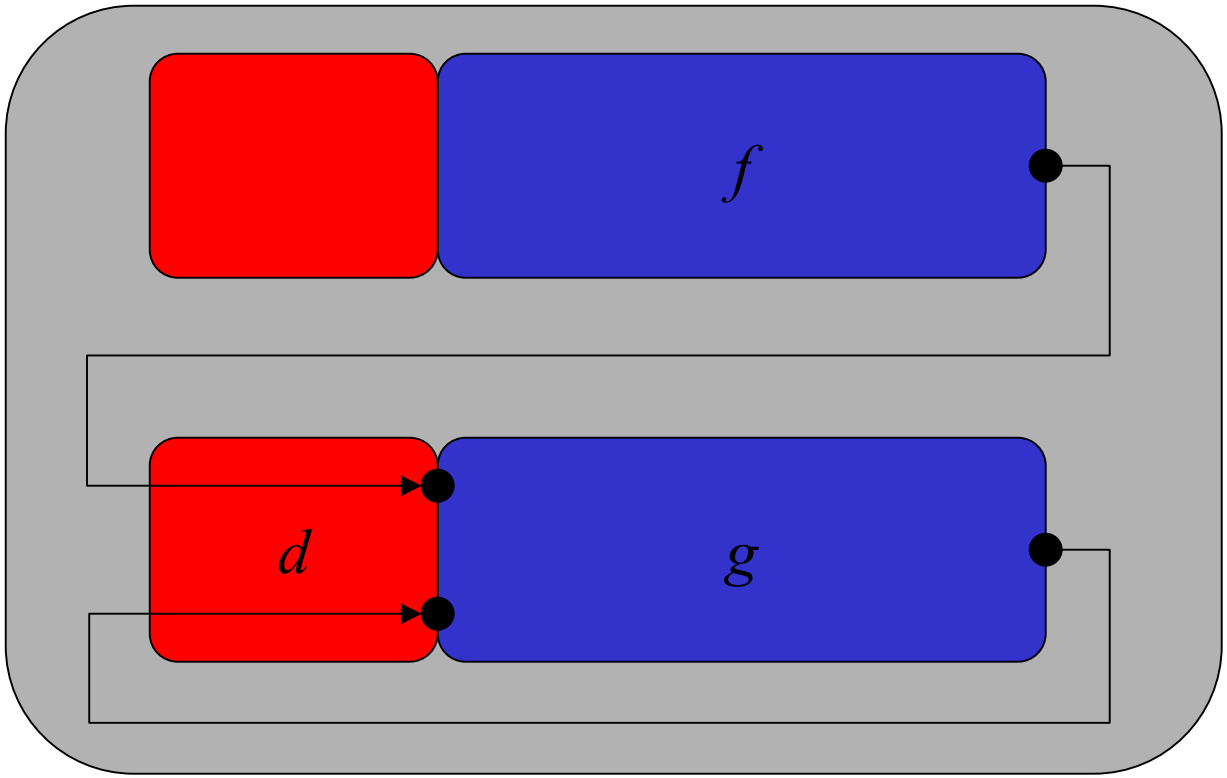
RMA and Preemption



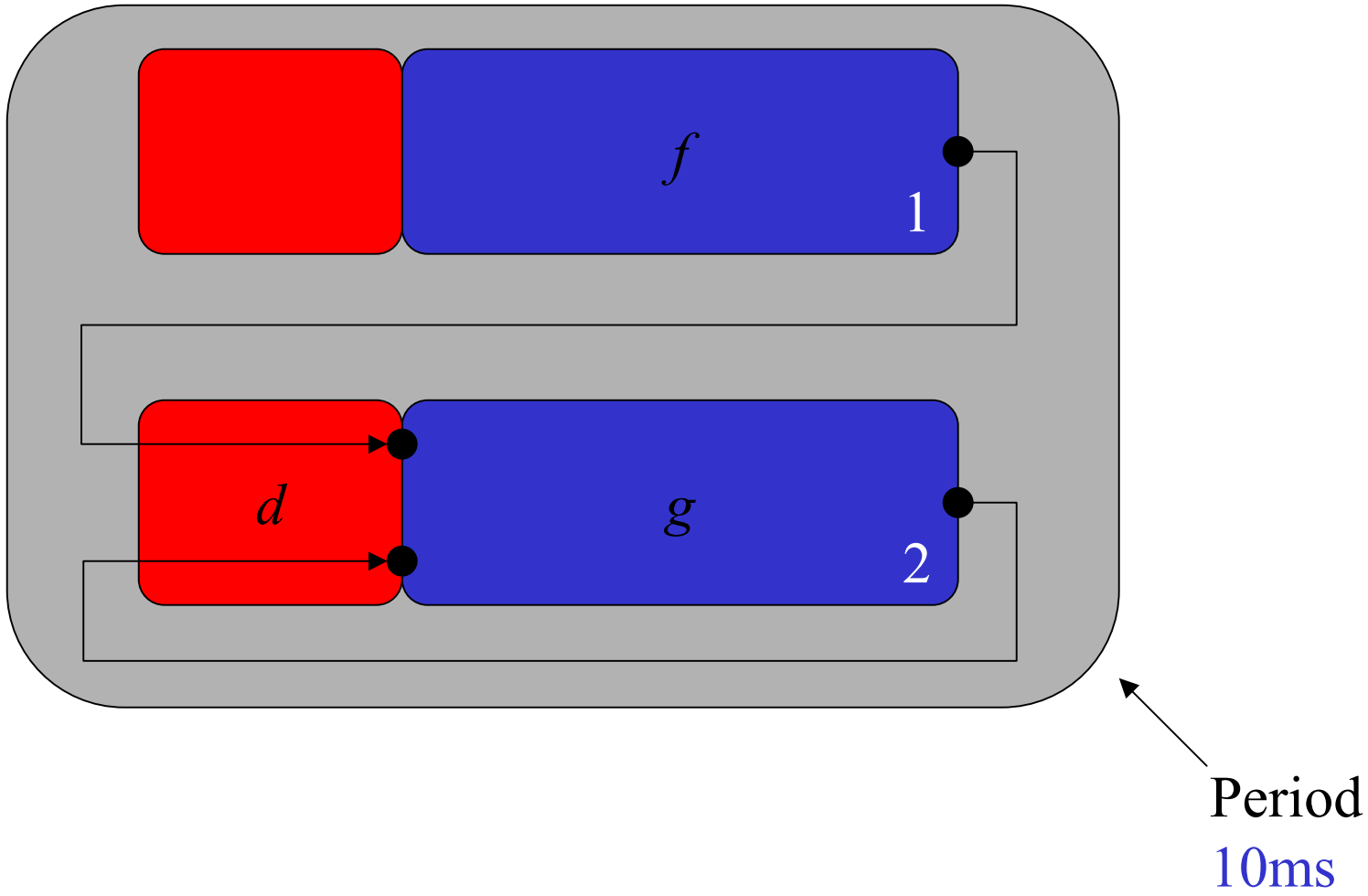
What's Next?



A Giotto Mode



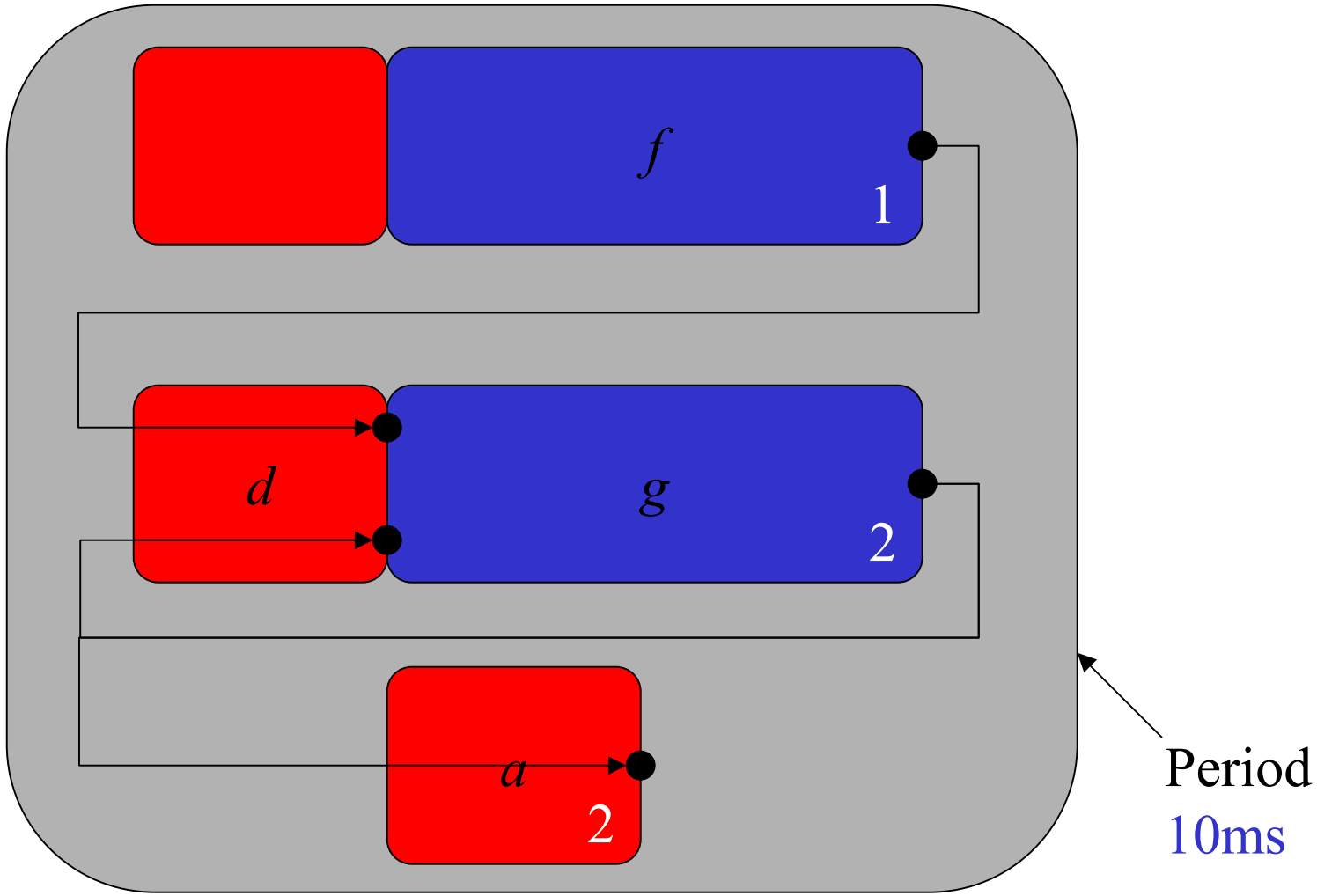
A Giotto Mode



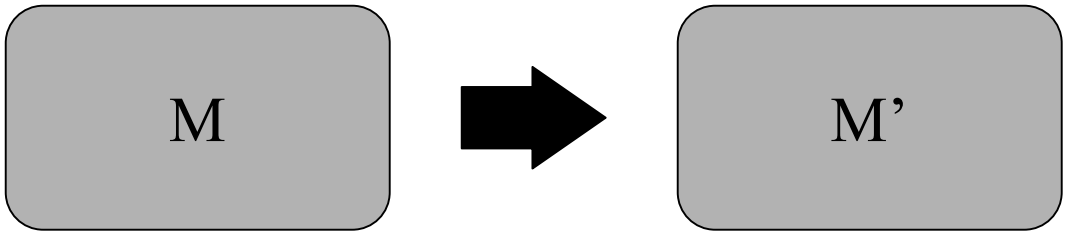
Concrete Syntax

```
mode m ( ) period 10 ms
  {
      taskfreq 1 do f ( ) output ( x );
      taskfreq 2 do g ( x, y ) output ( y );
  }
```

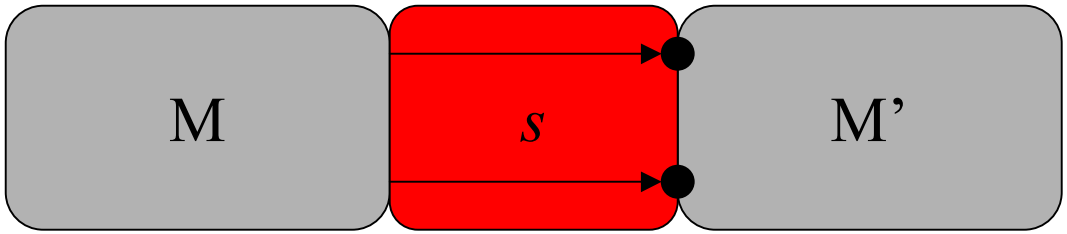
Actuator Driver



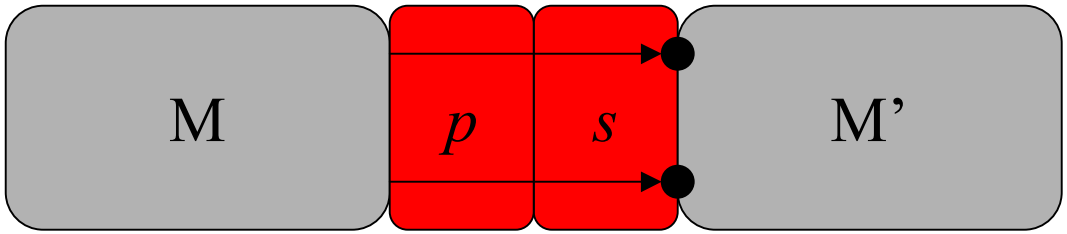
Mode Switch



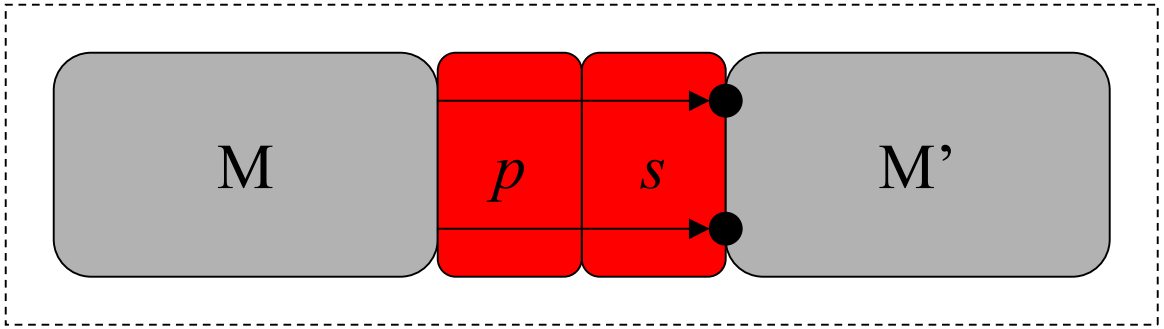
Mode Switch Driver



Guarded Driver



A Giotto Mode Switch

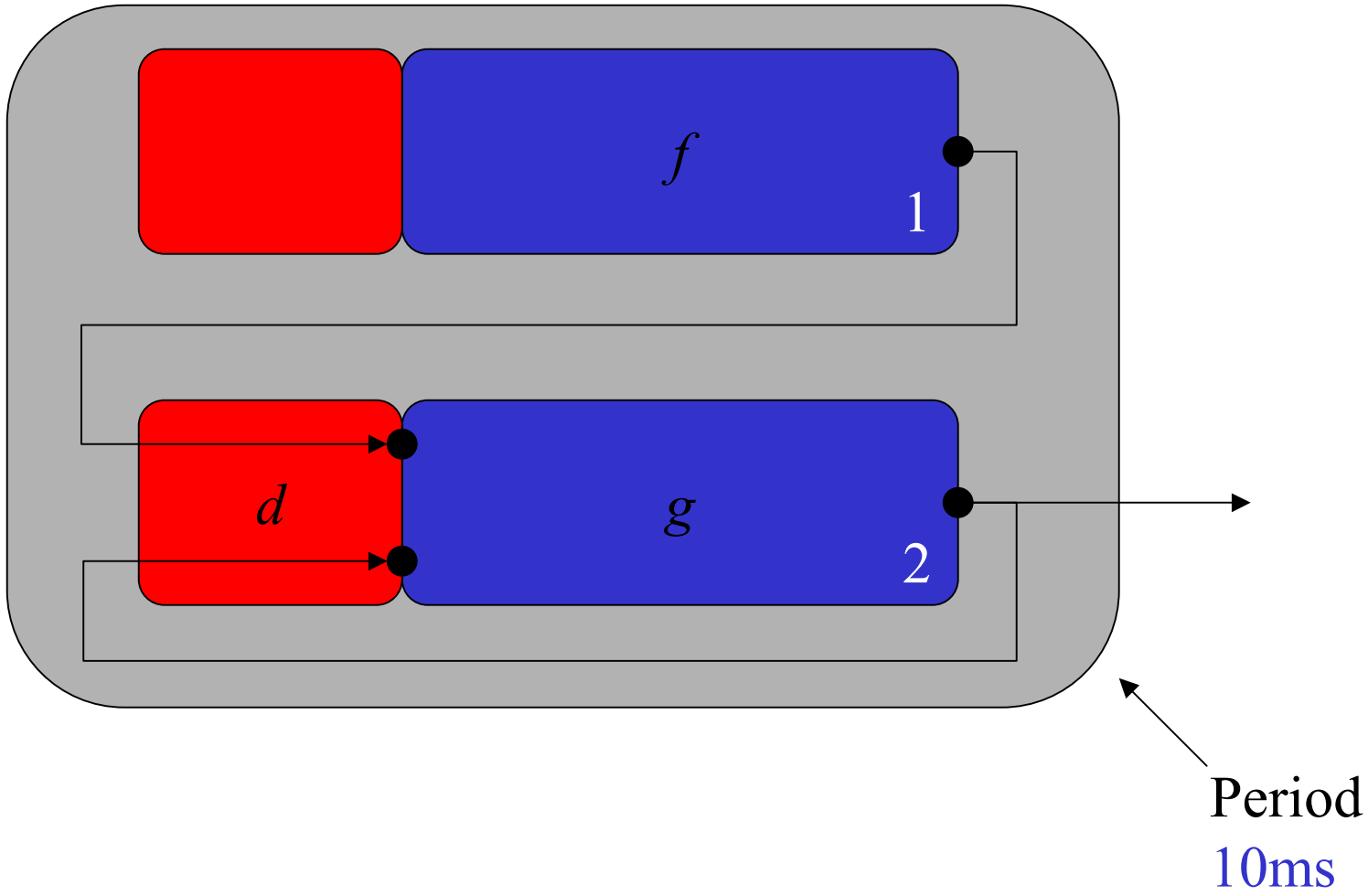


Giotto Mode

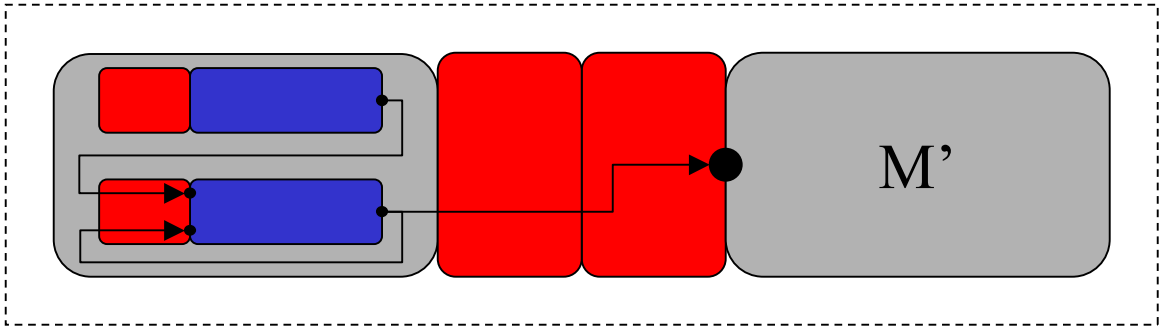
- A Giotto mode is a parameterized set of tasks and mode switches
- A Giotto program consists of a set of modes

- A Giotto system is in a single mode at the same time

Mode M

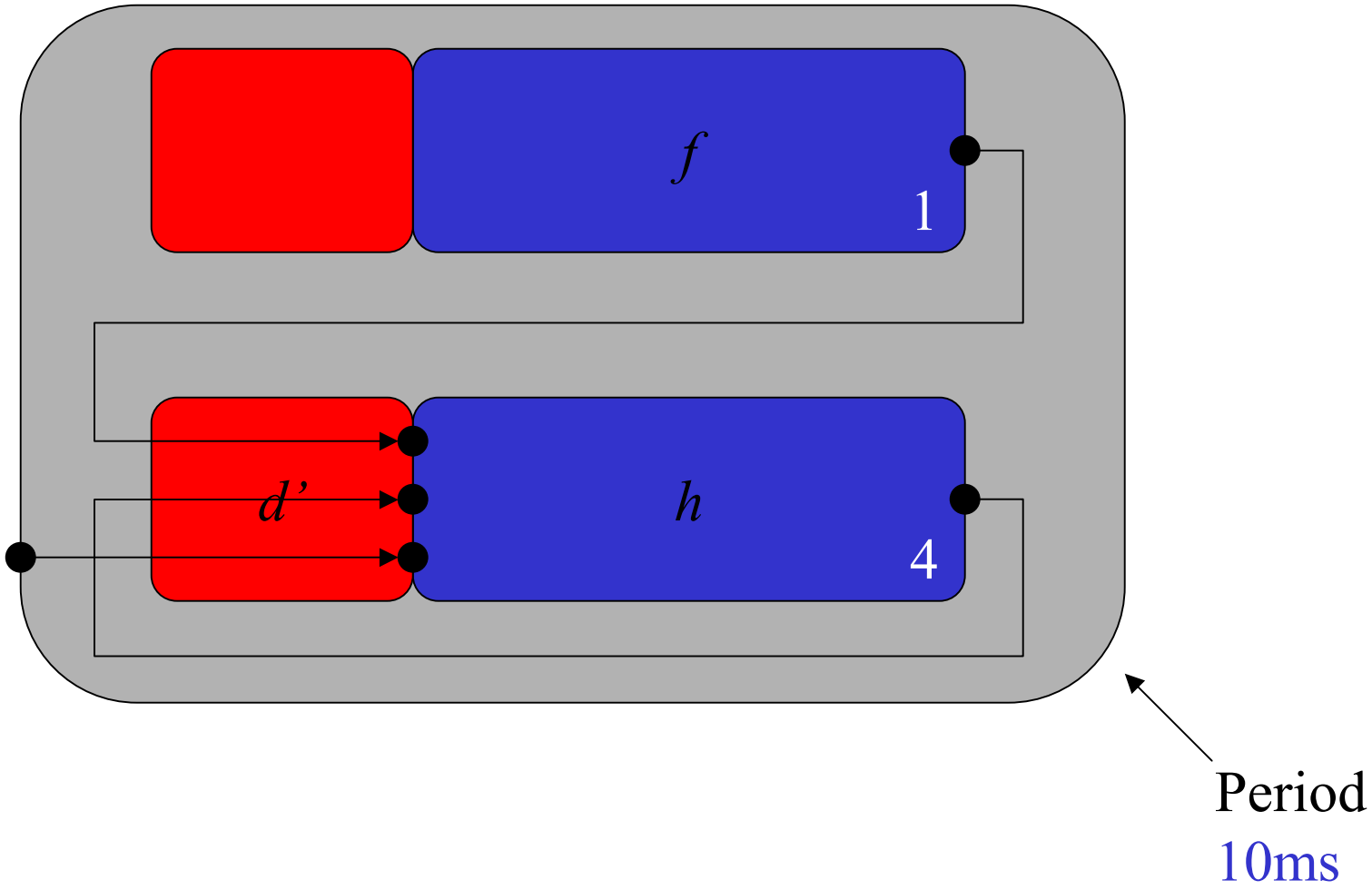


A Giotto Mode Switch

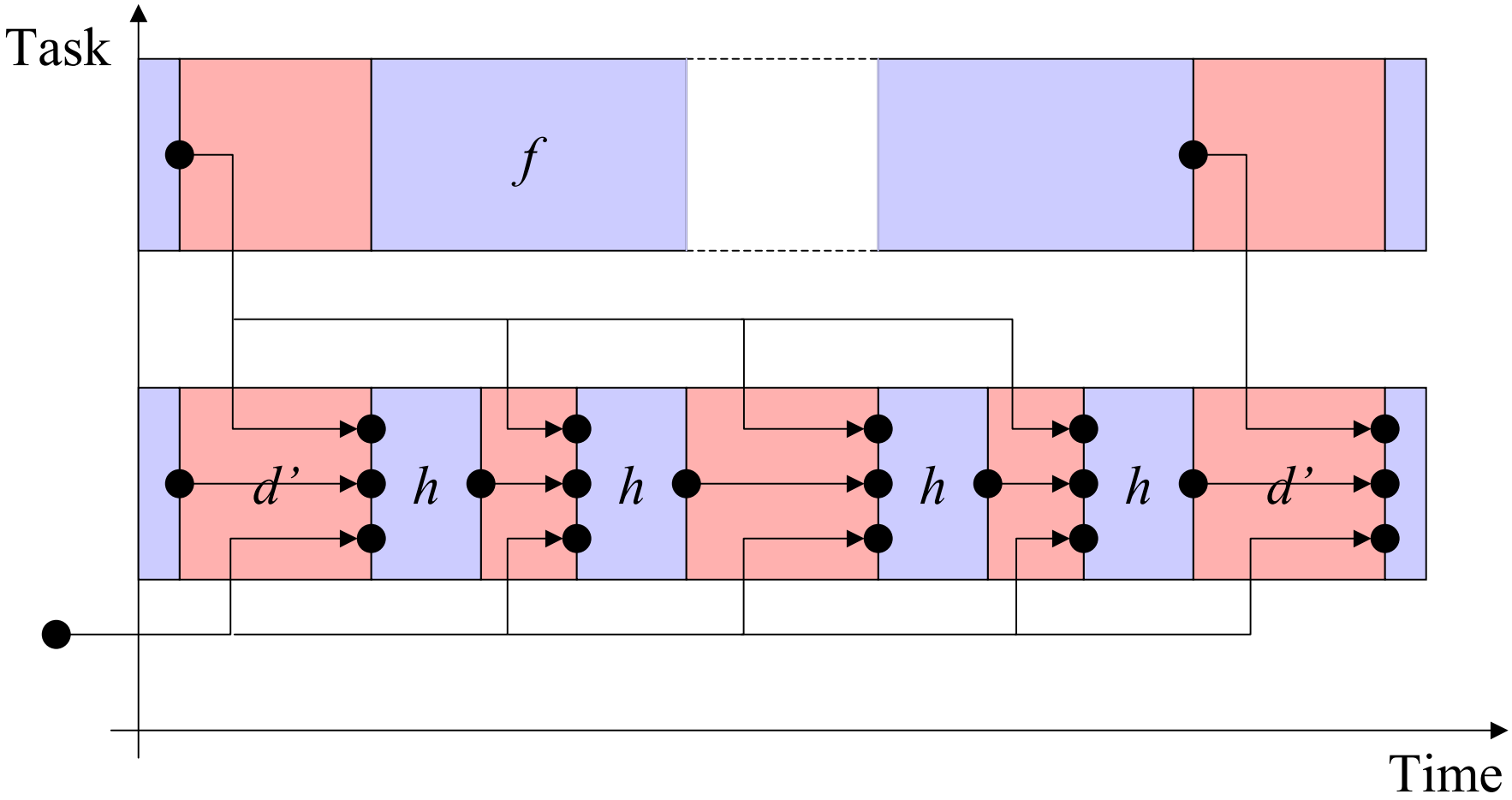


Period 5ms

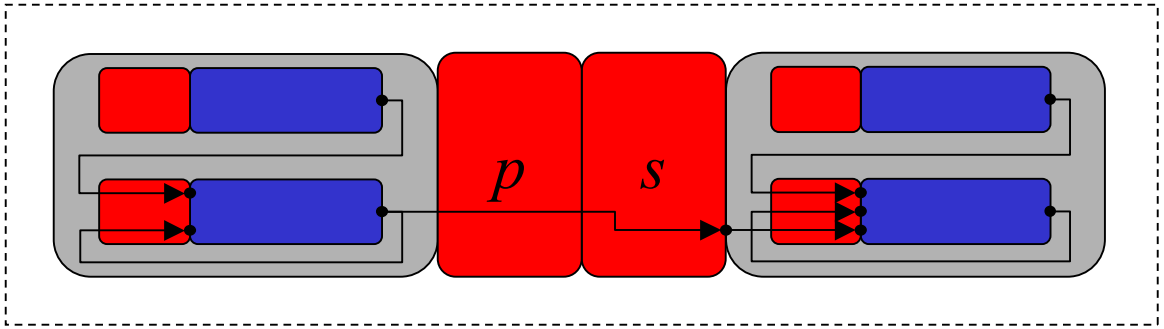
Mode M’



One Round of M’



A Giotto Mode Switch



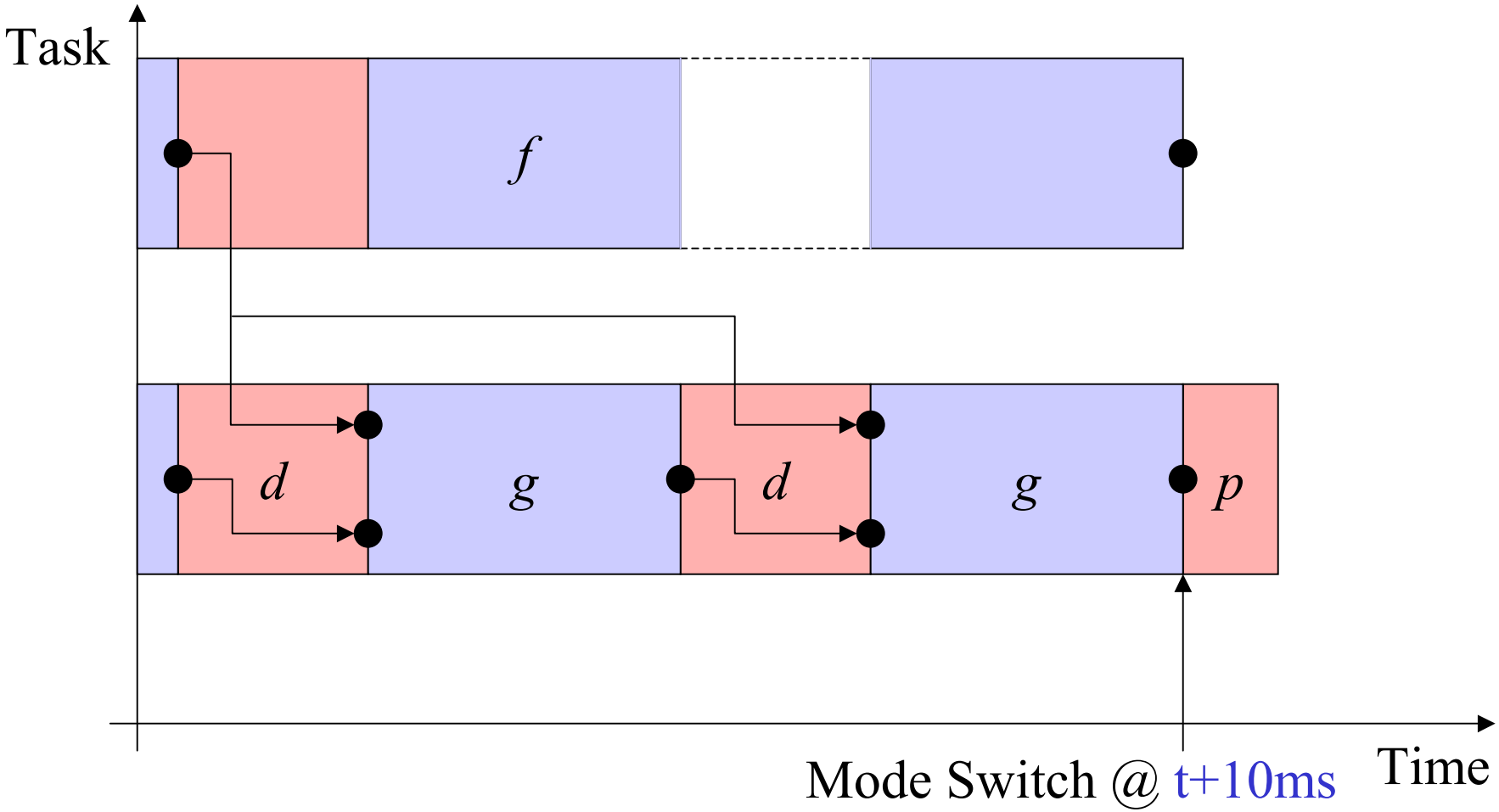
Period 5ms

Concrete Syntax

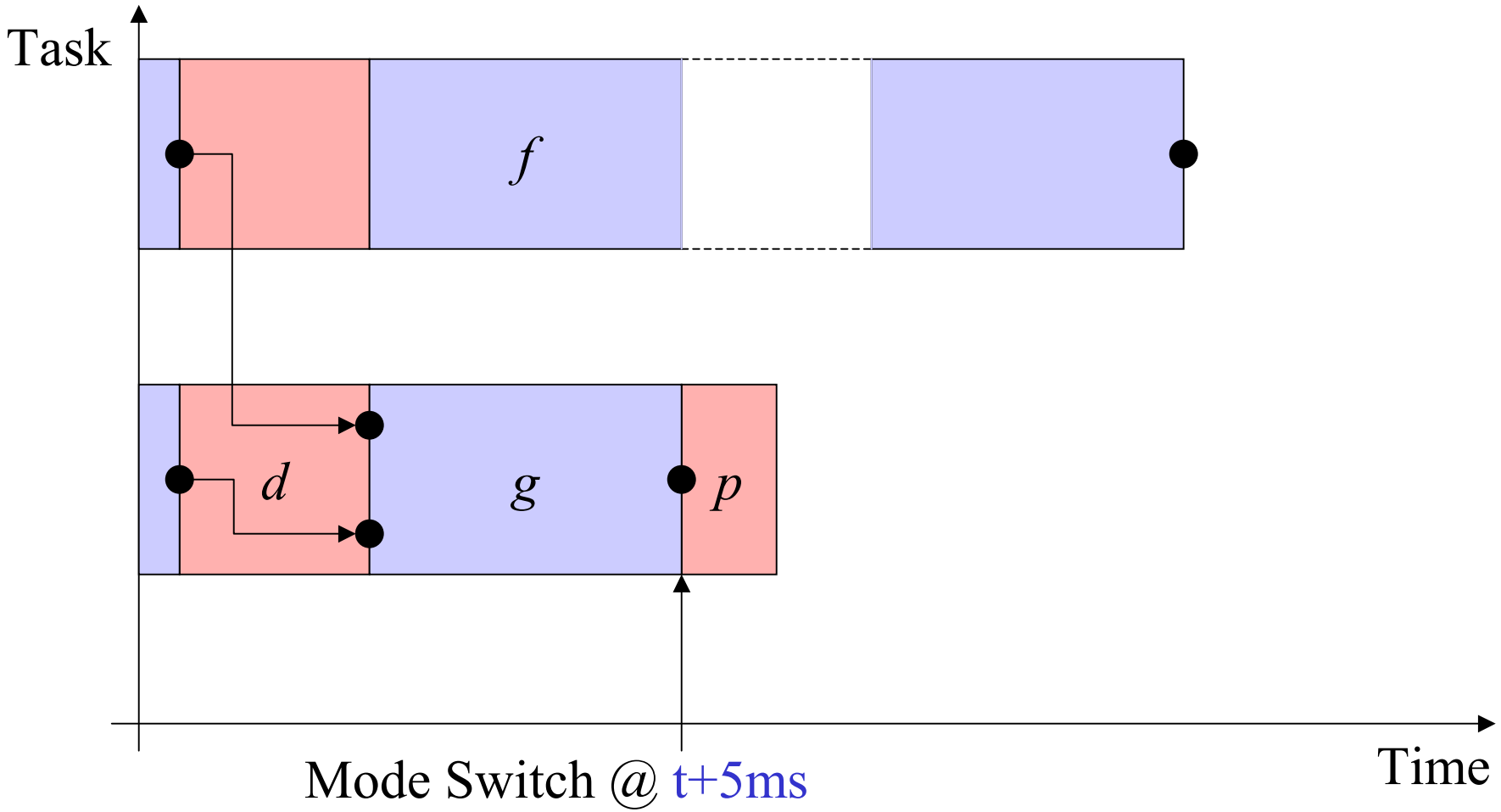
```
start m ( ) {
    mode m ( ) period 10 ms {
        taskfreq 1 do f ( ) output ( x );
        taskfreq 2 do g ( x, y ) output ( y );

        exitfreq 2 if p ( y ) then m' ( y );
    }
    mode m' ( z ) period 10 ms {
        taskfreq 1 do f ( ) output ( x );
        taskfreq 4 do h ( x, y, z ) output ( y );
    }
}
```

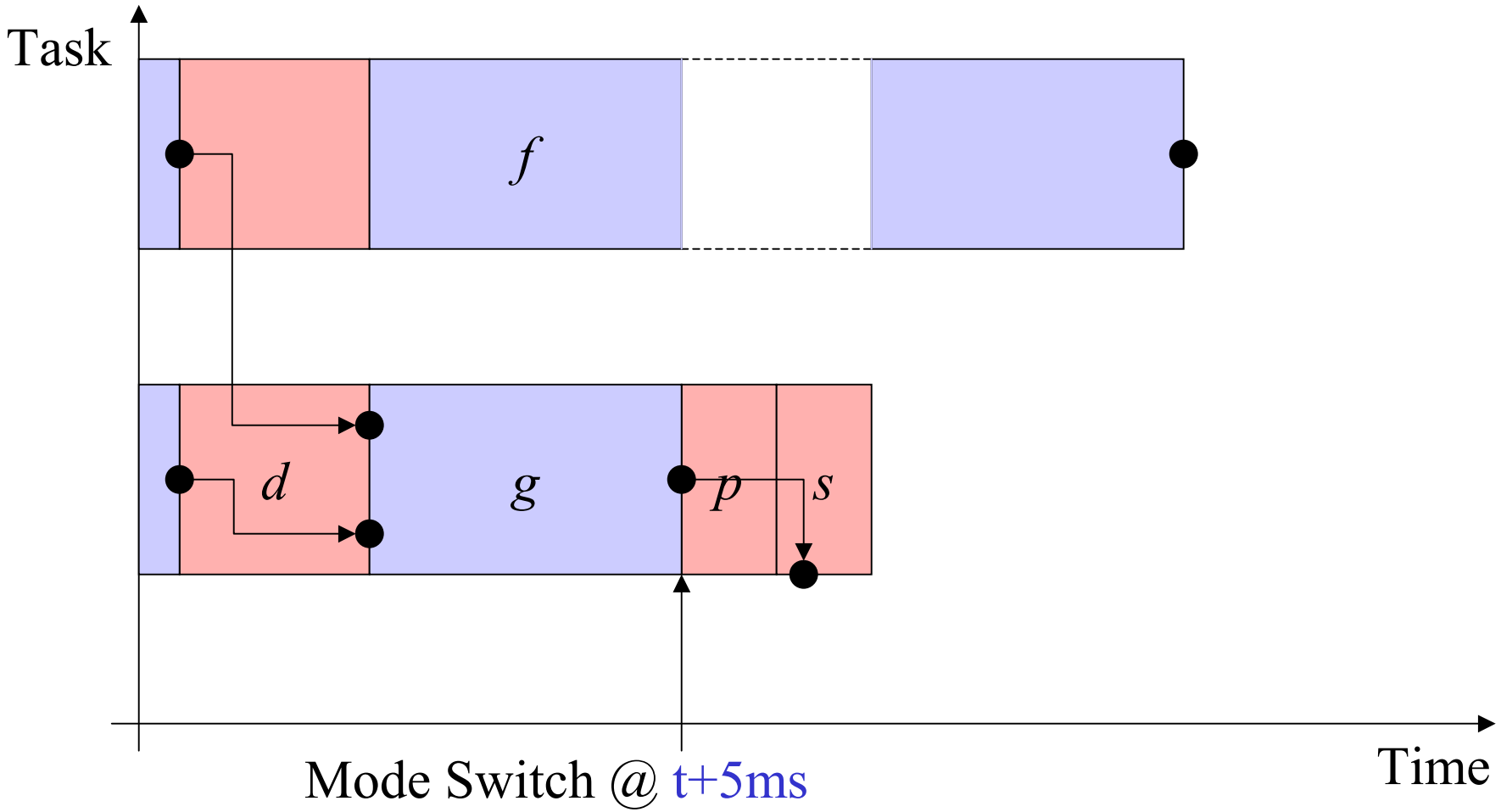
Semantics of a Mode Switch



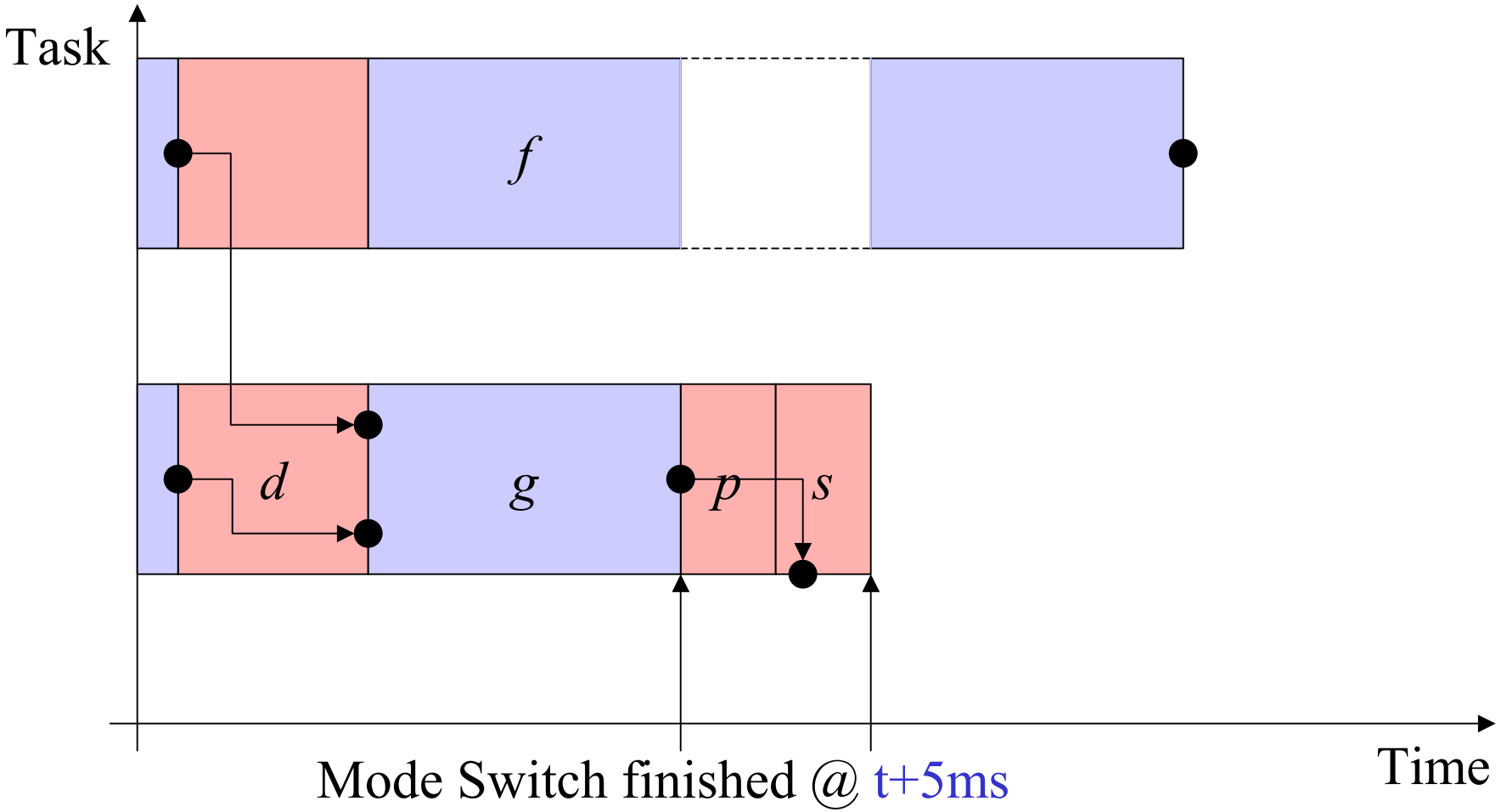
Semantics of a Mode Switch



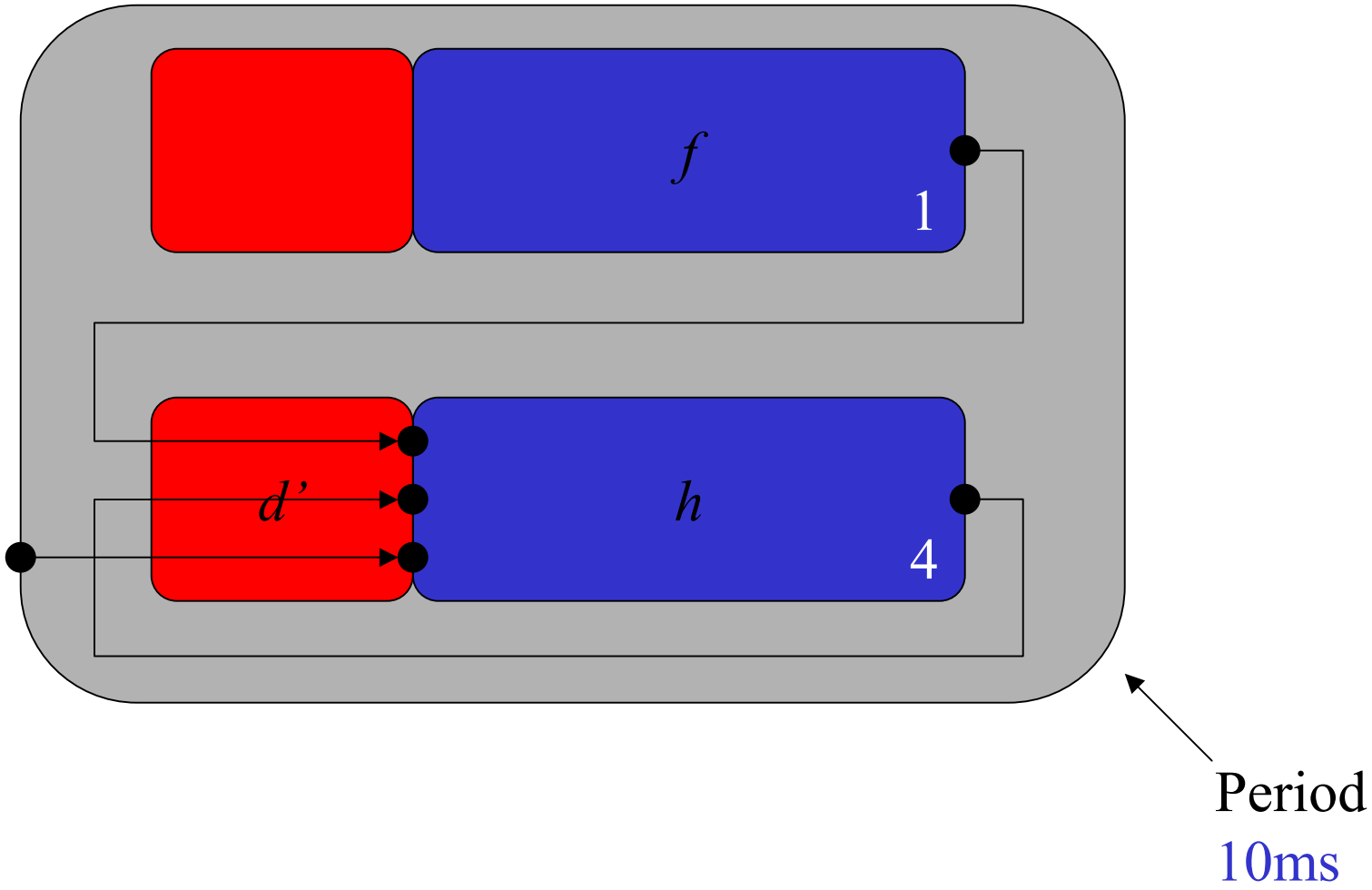
Switch Mode



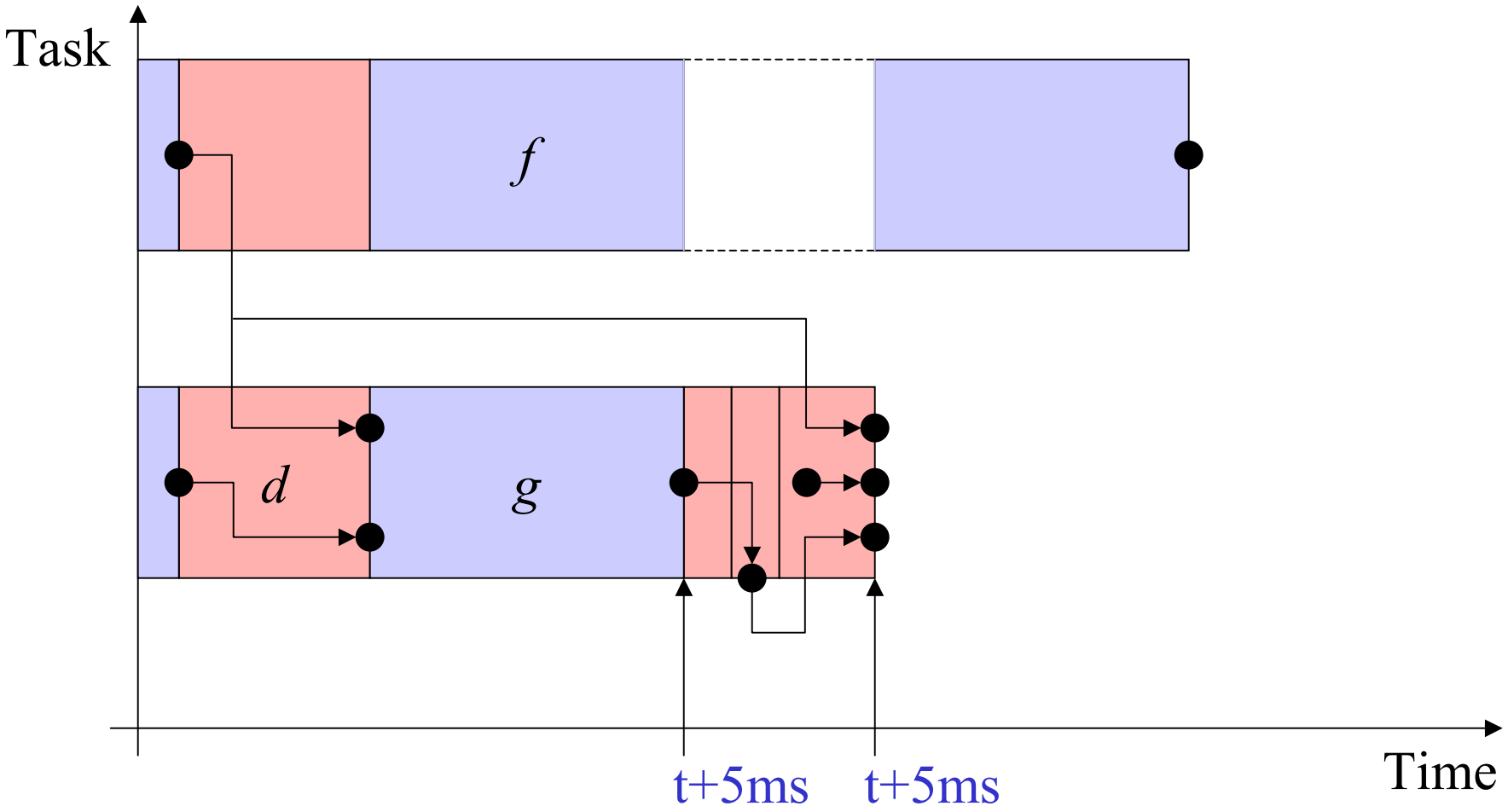
Switch to M’



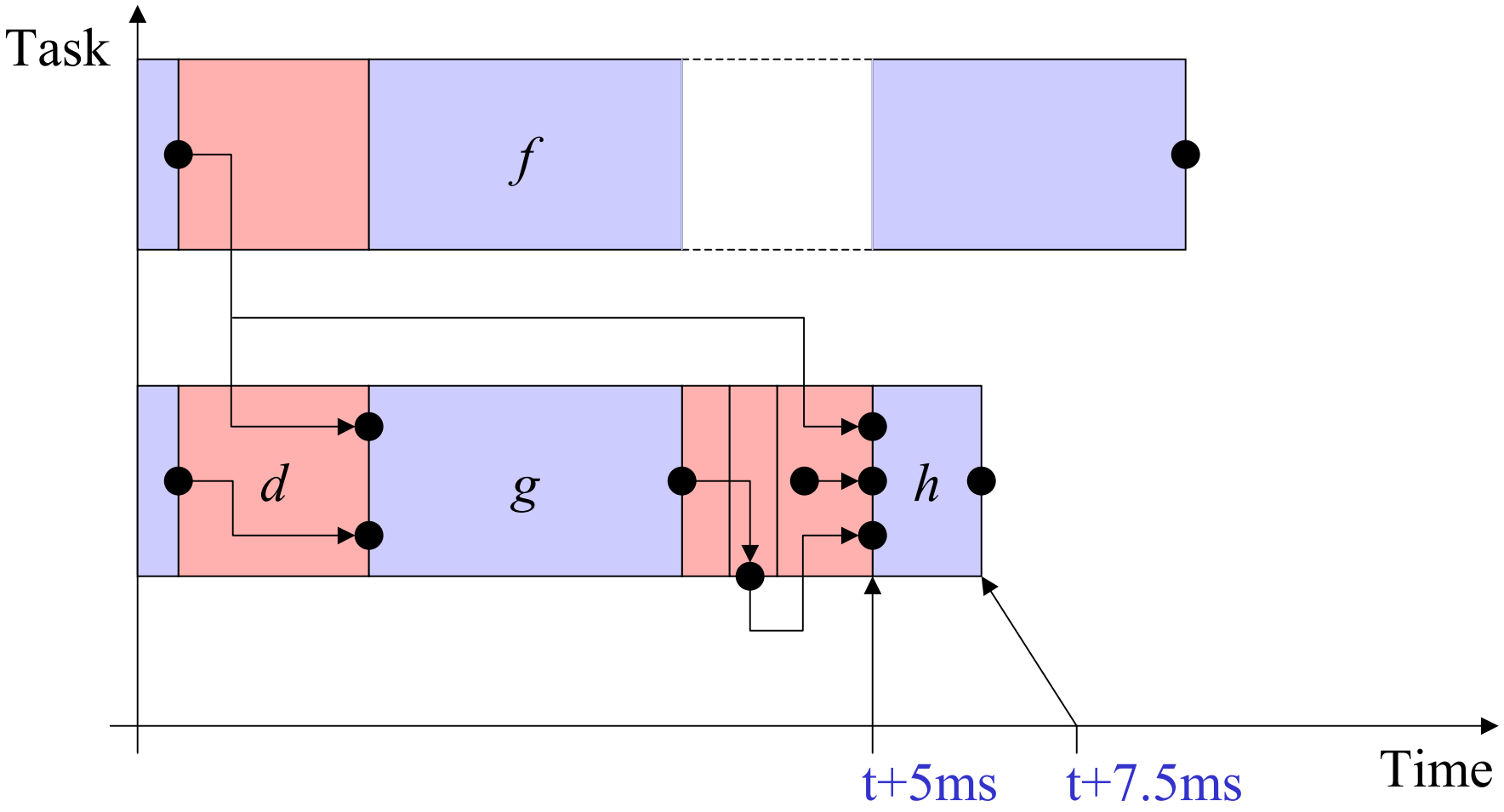
Mode M’



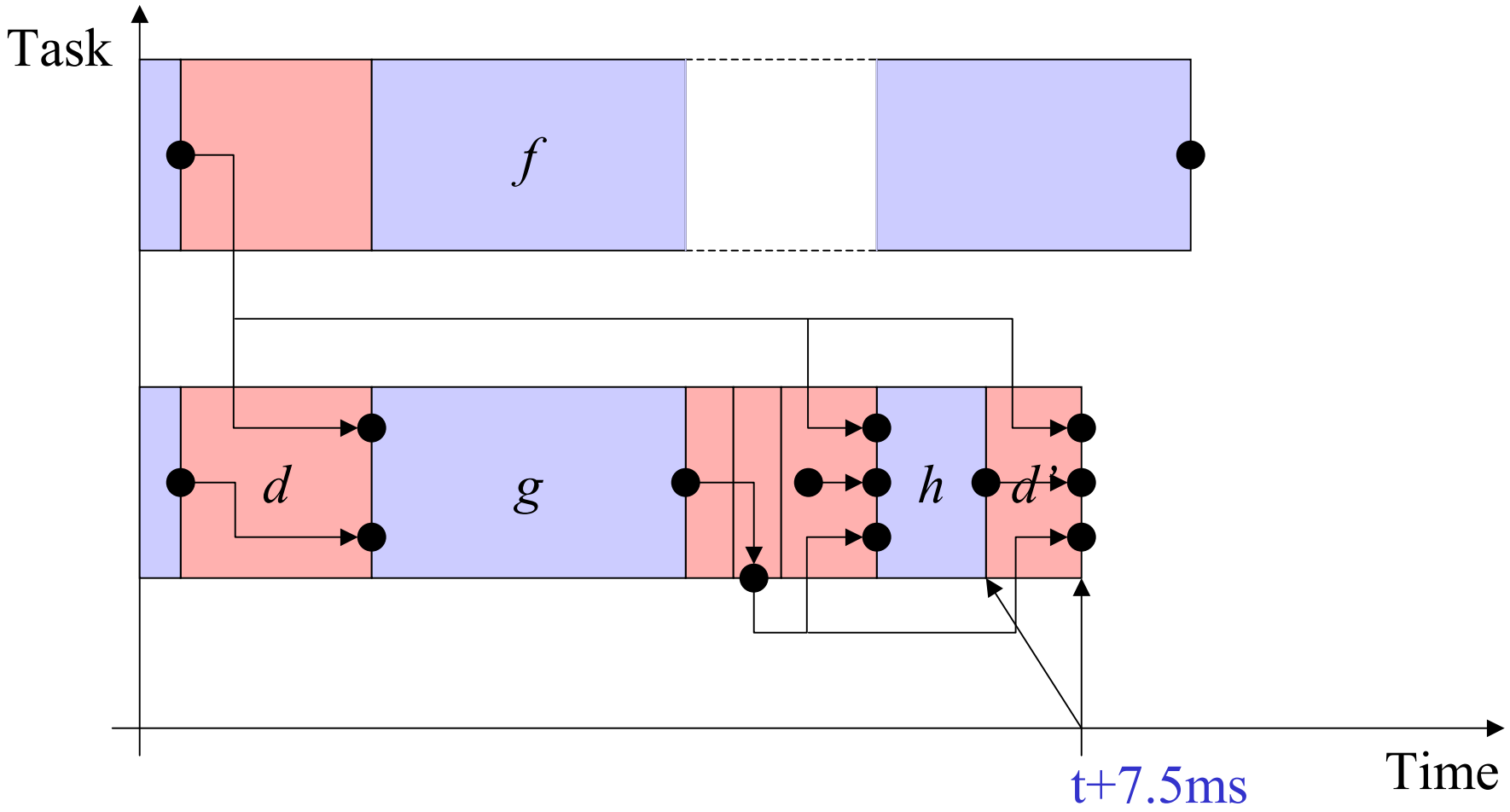
Drive task h



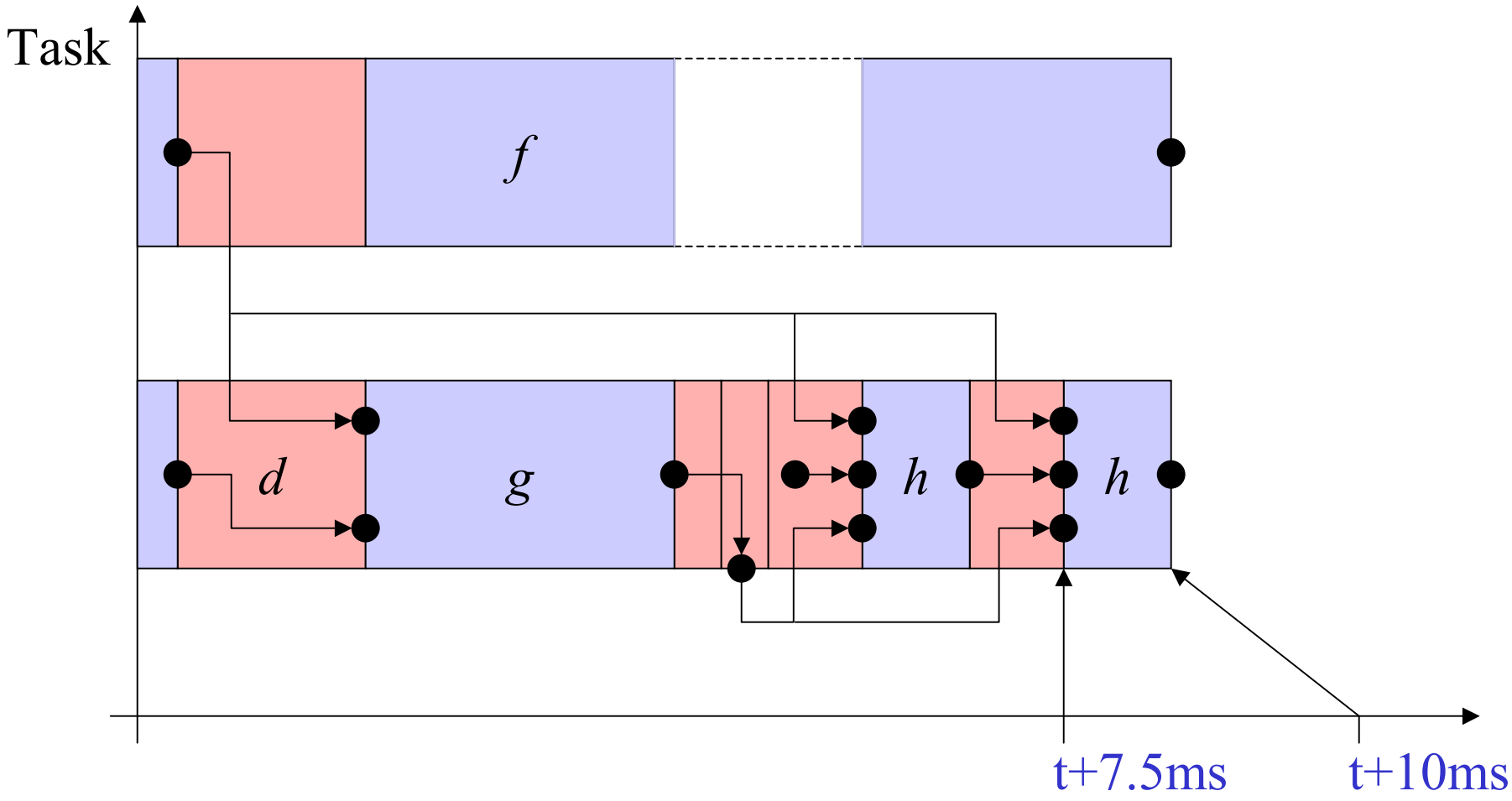
Schedule task h



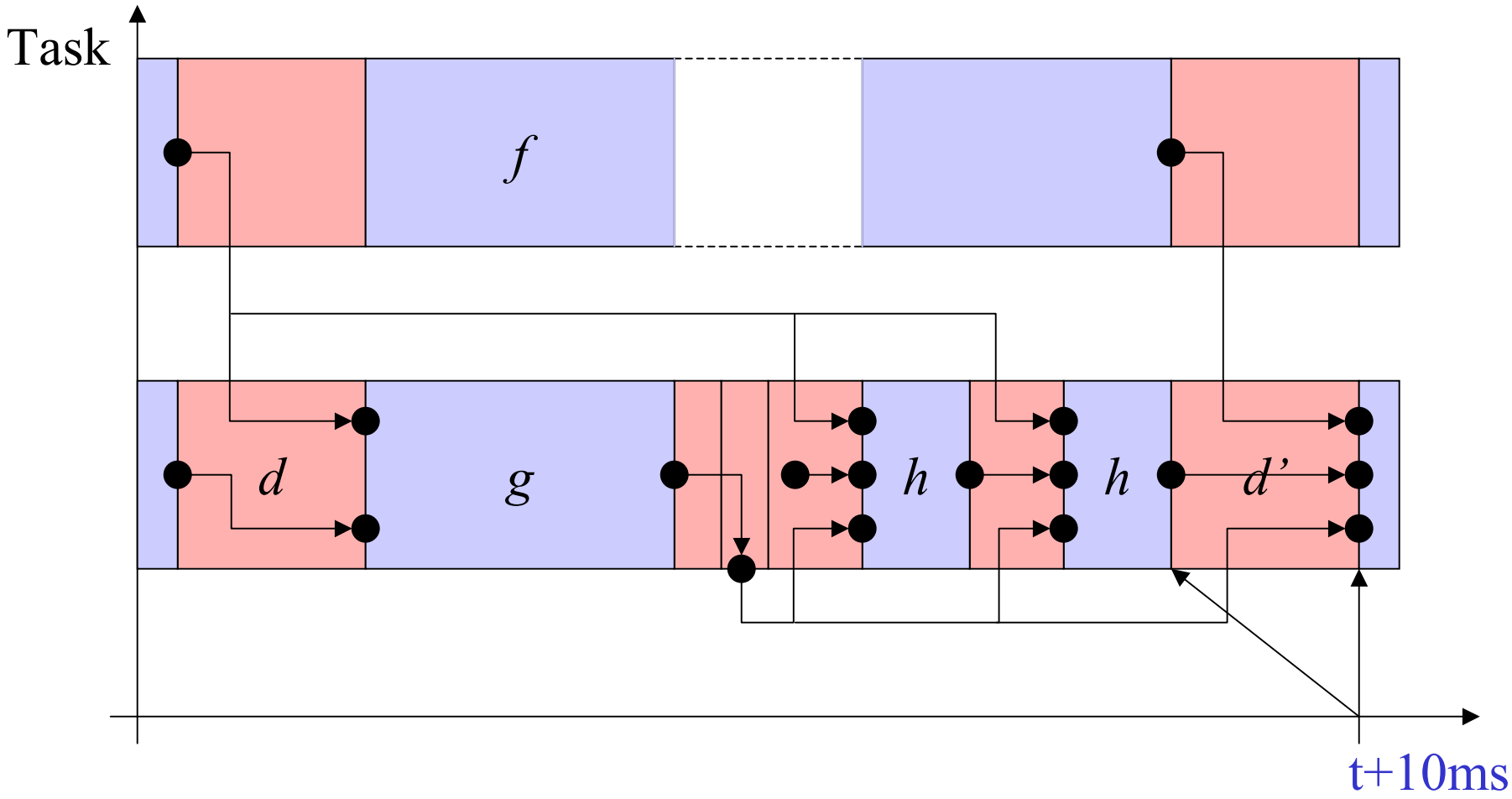
Drive task h



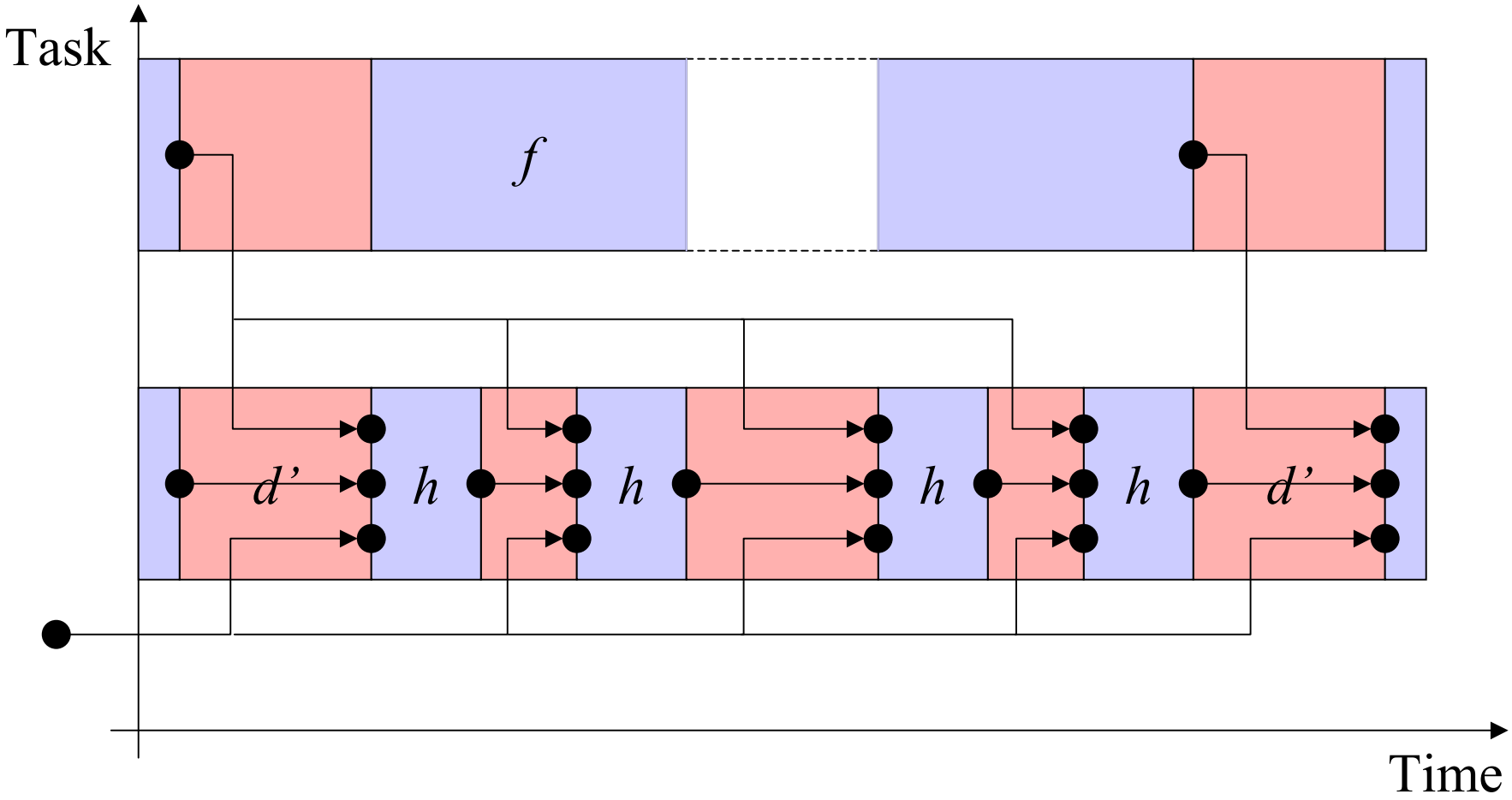
Schedule task h



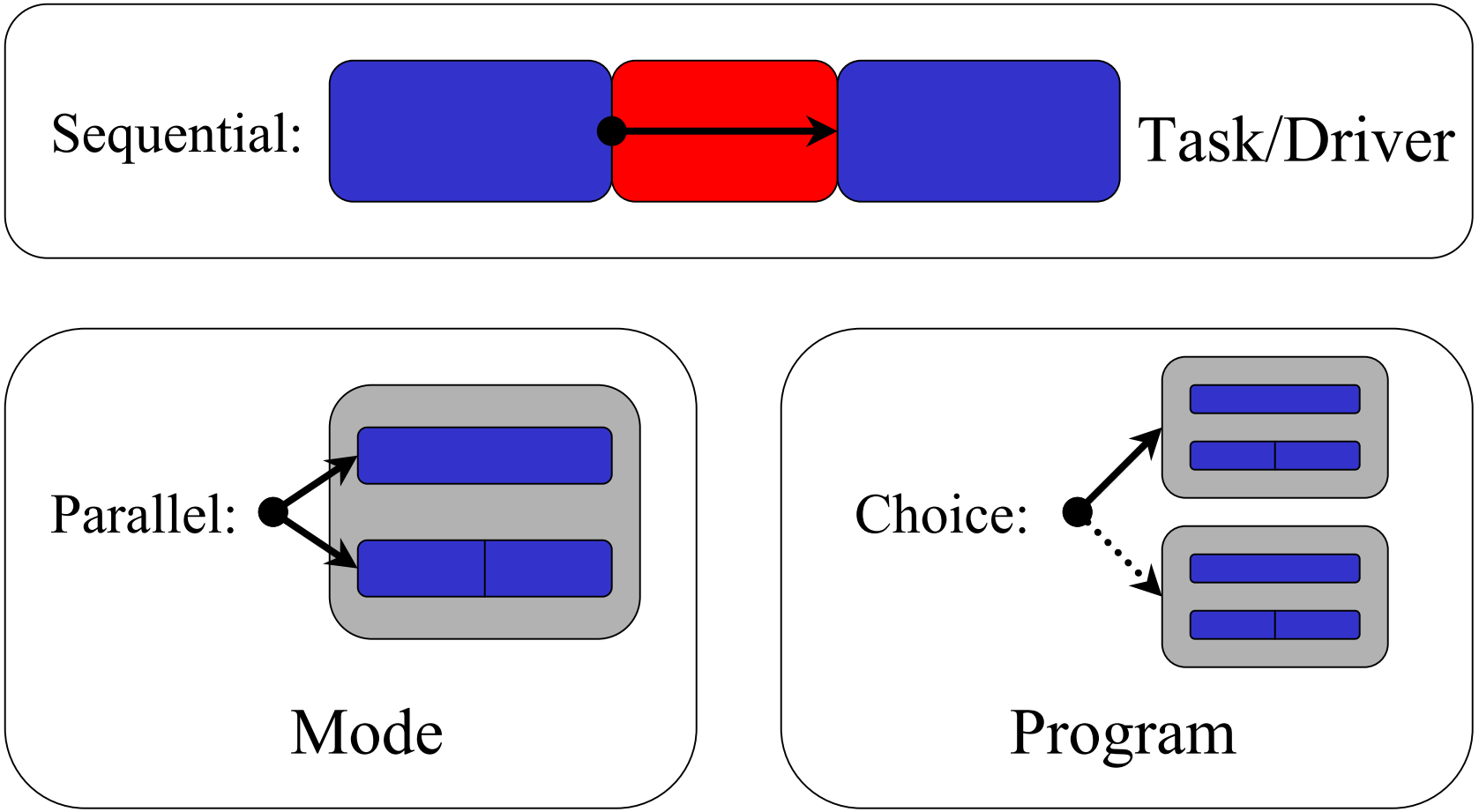
Next Round



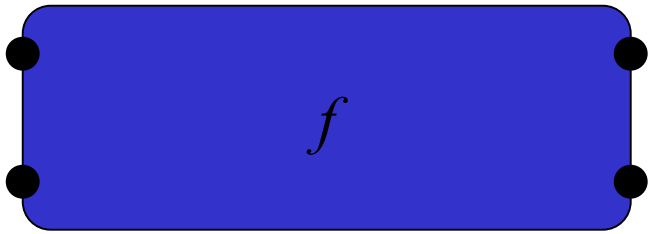
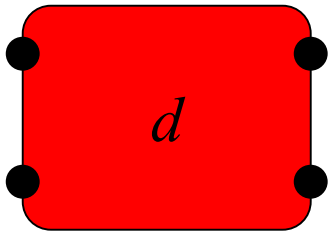
One Round of M’



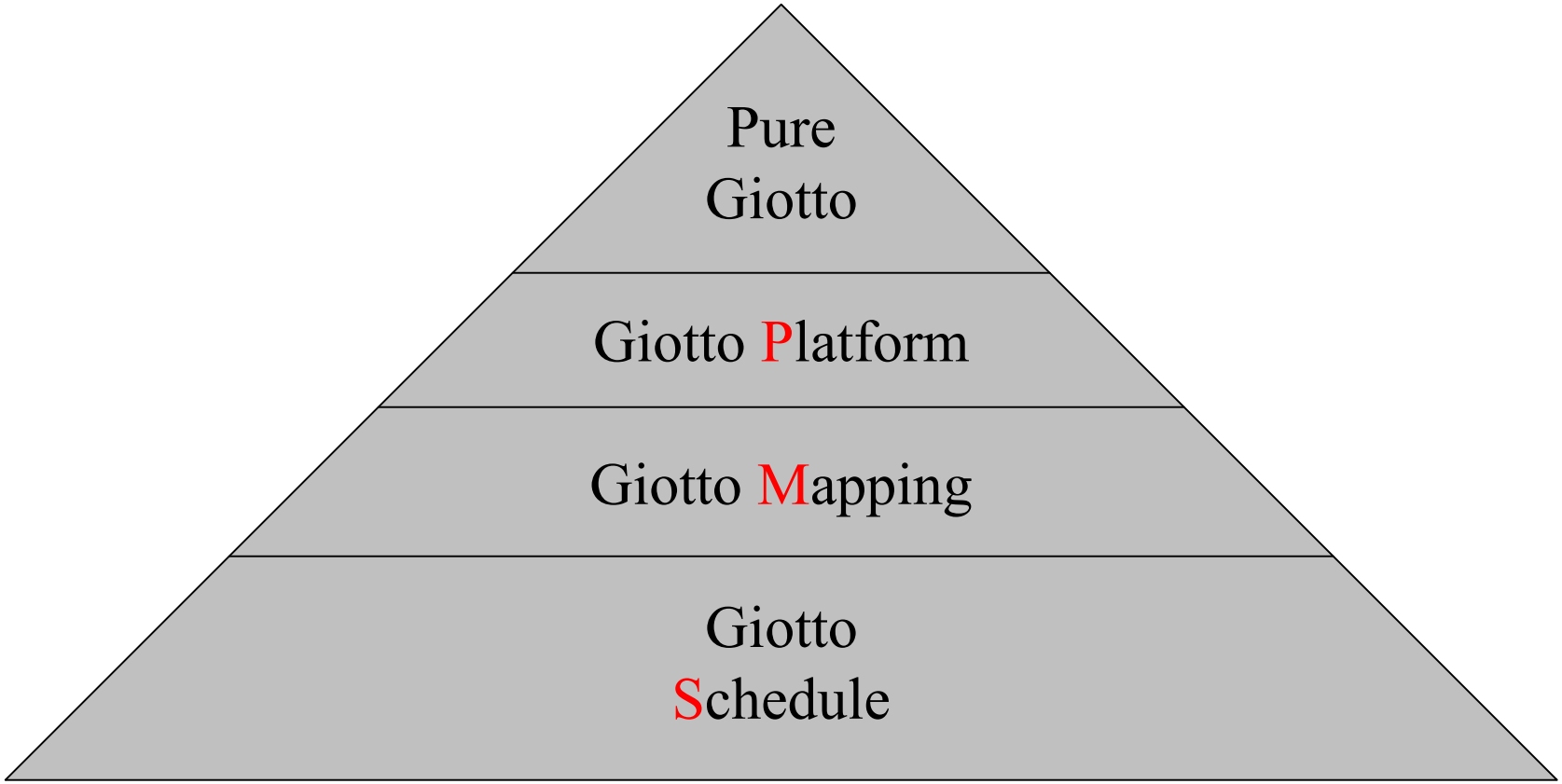
Summary: Operators



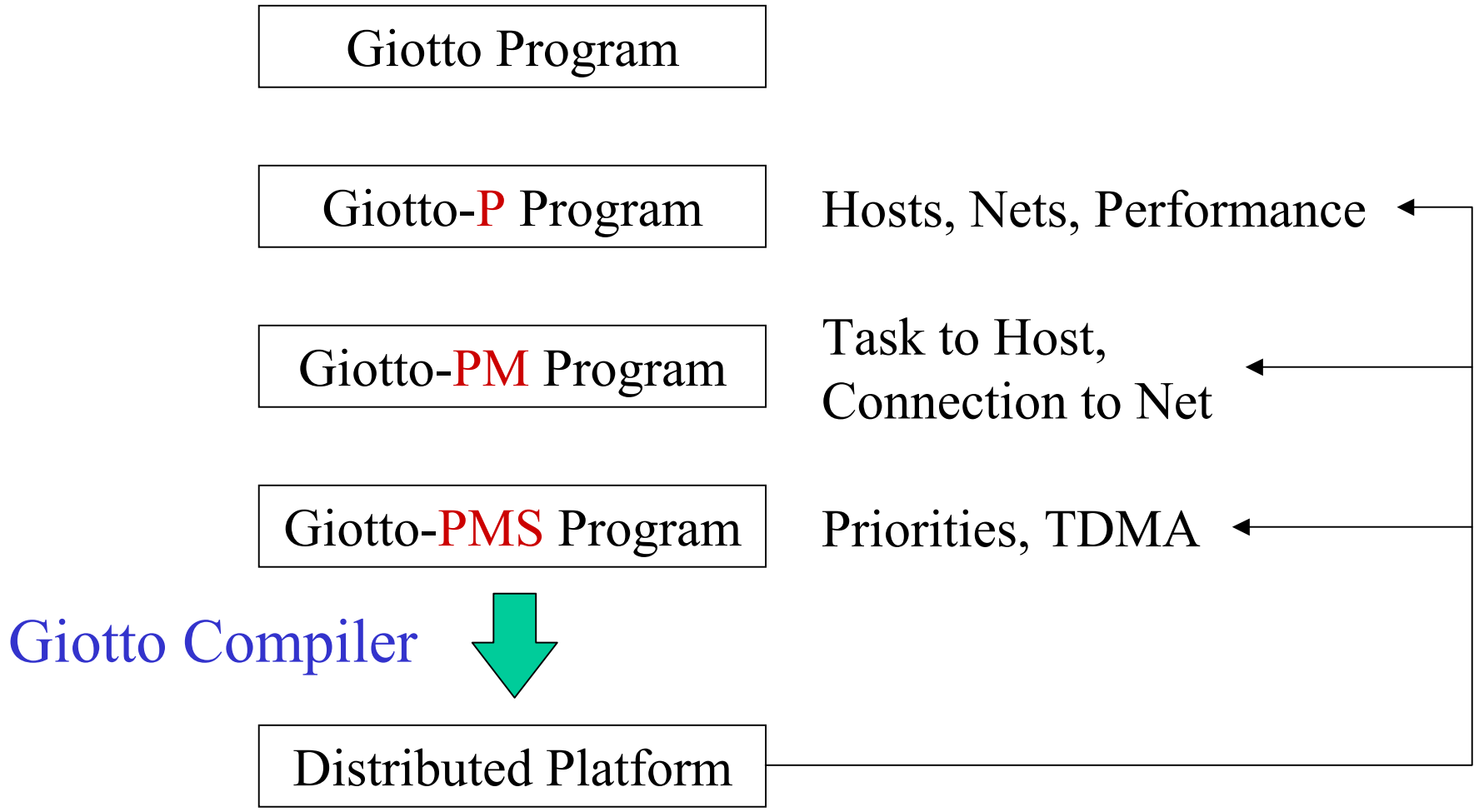
Summary: Functionality

	Synchronous	Scheduled
Computation	?	
Communication		?

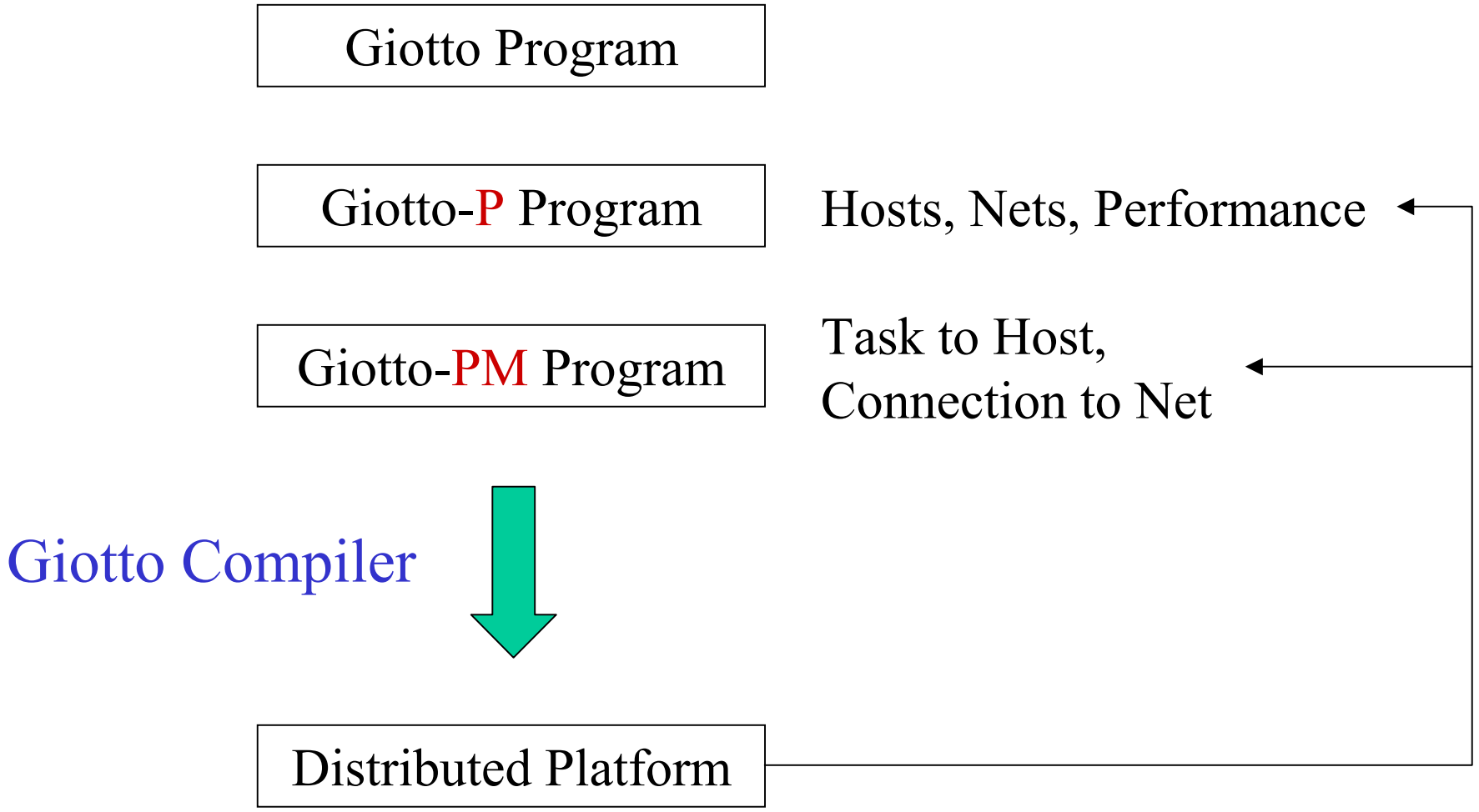
Pure vs. Annotated Giotto



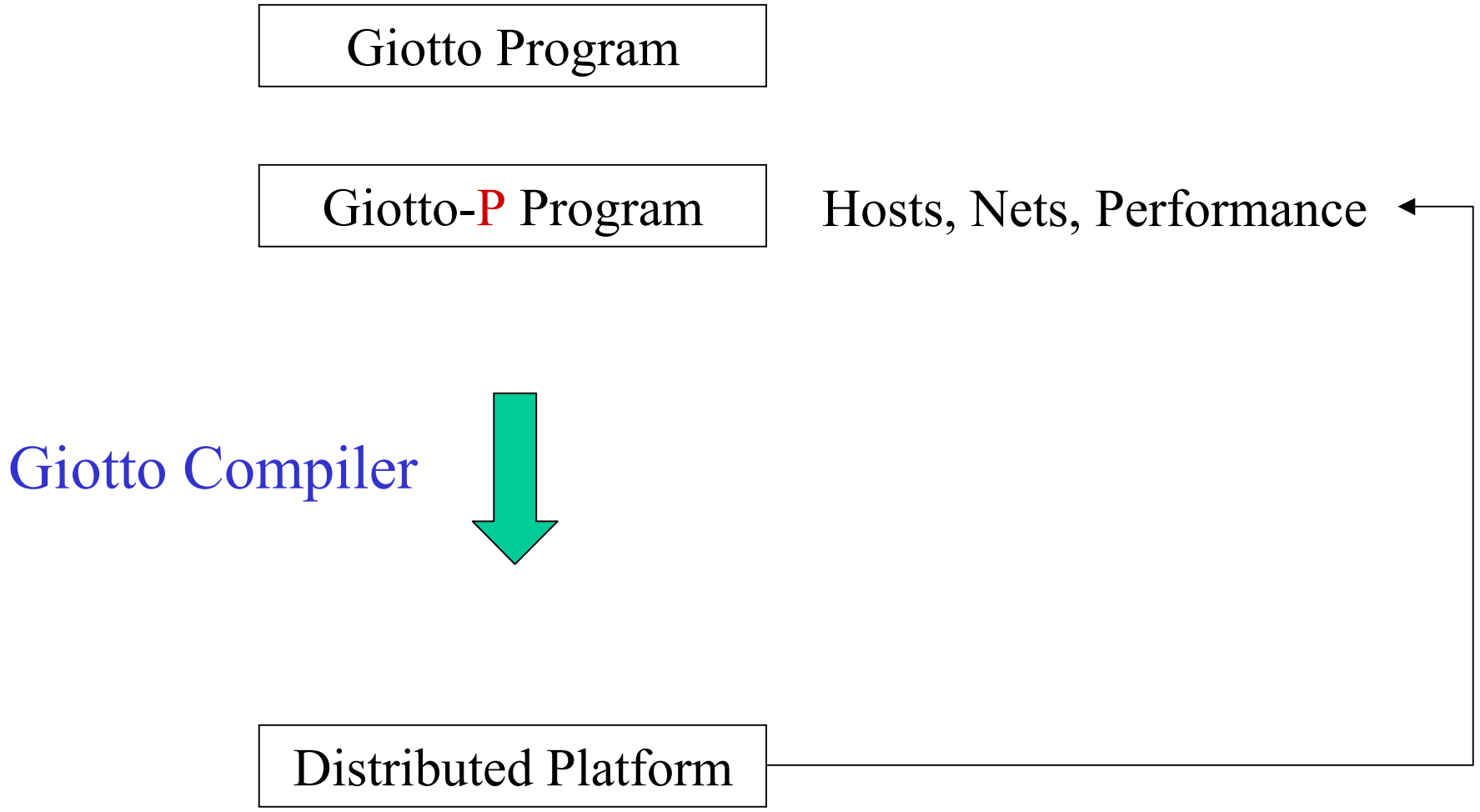
Giotto Schedule



Giotto Mapping



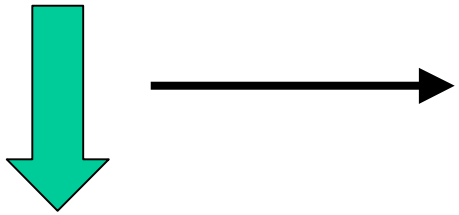
Giotto Platform



The Giotto Compiler

Giotto Program

Giotto Compiler

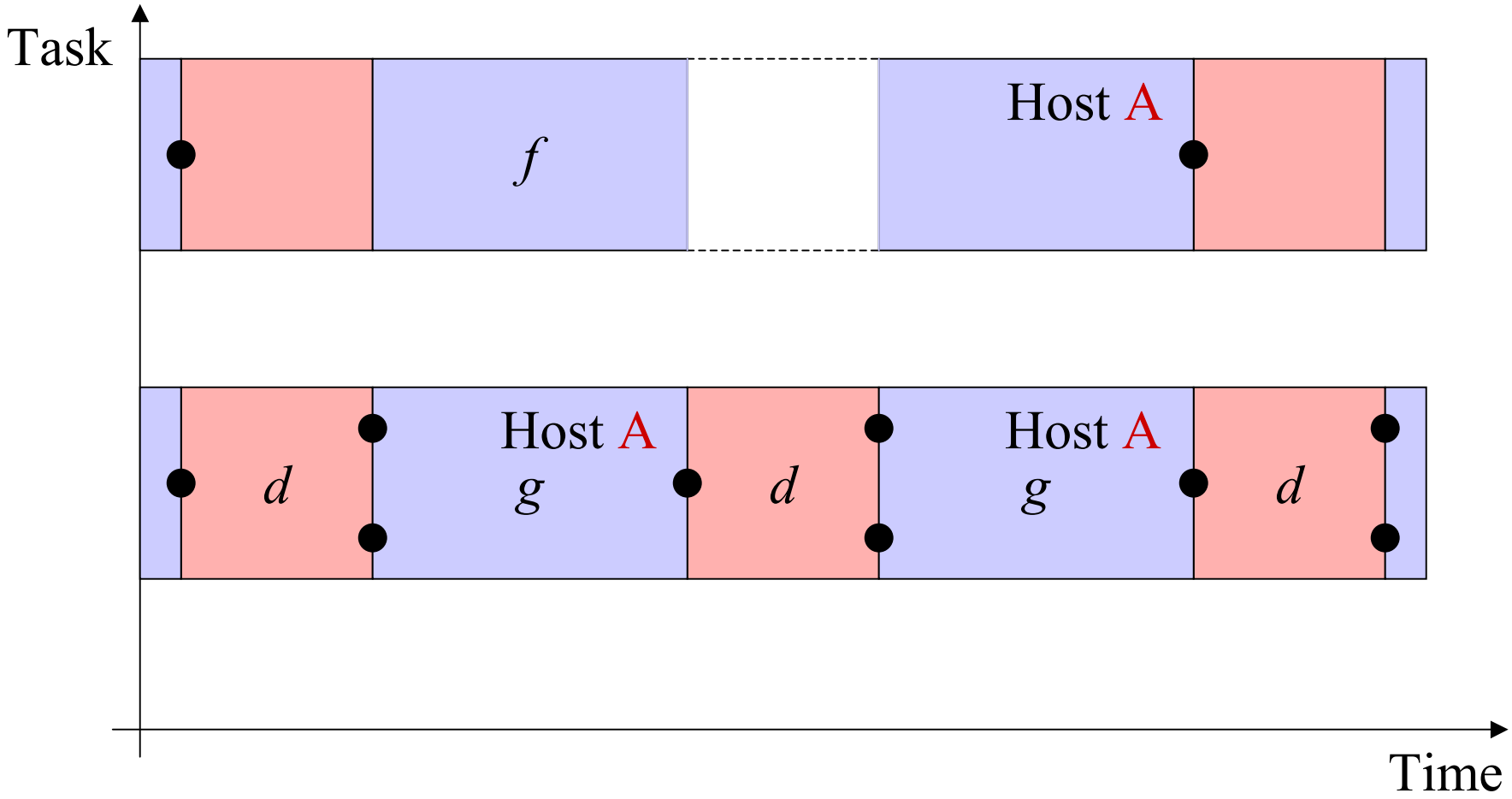


Possible Answers:

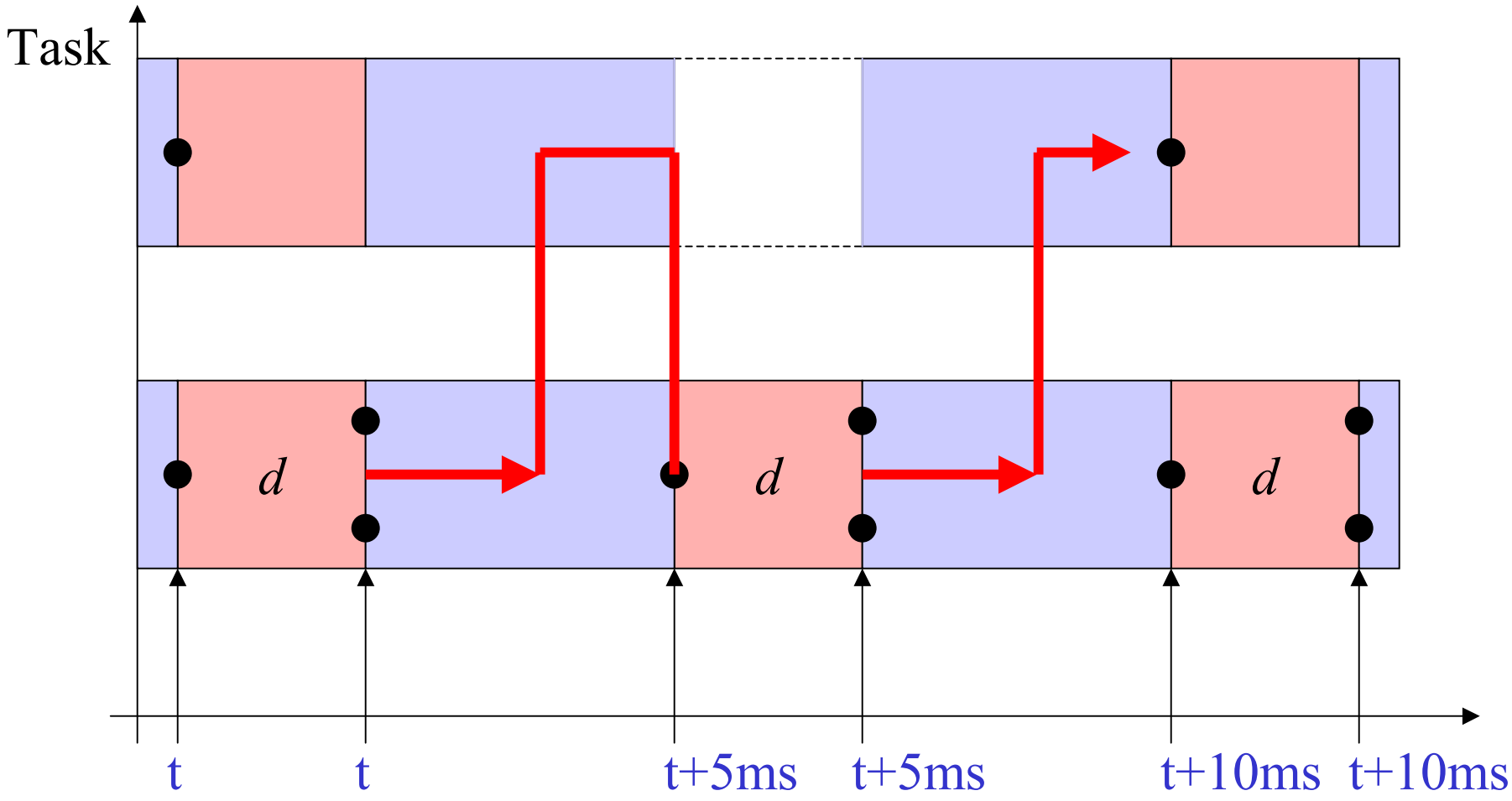
- Fully annotated Giotto program:
  Giotto Executable
- Annotated program invalid
- Cannot find schedule

Giotto Executable

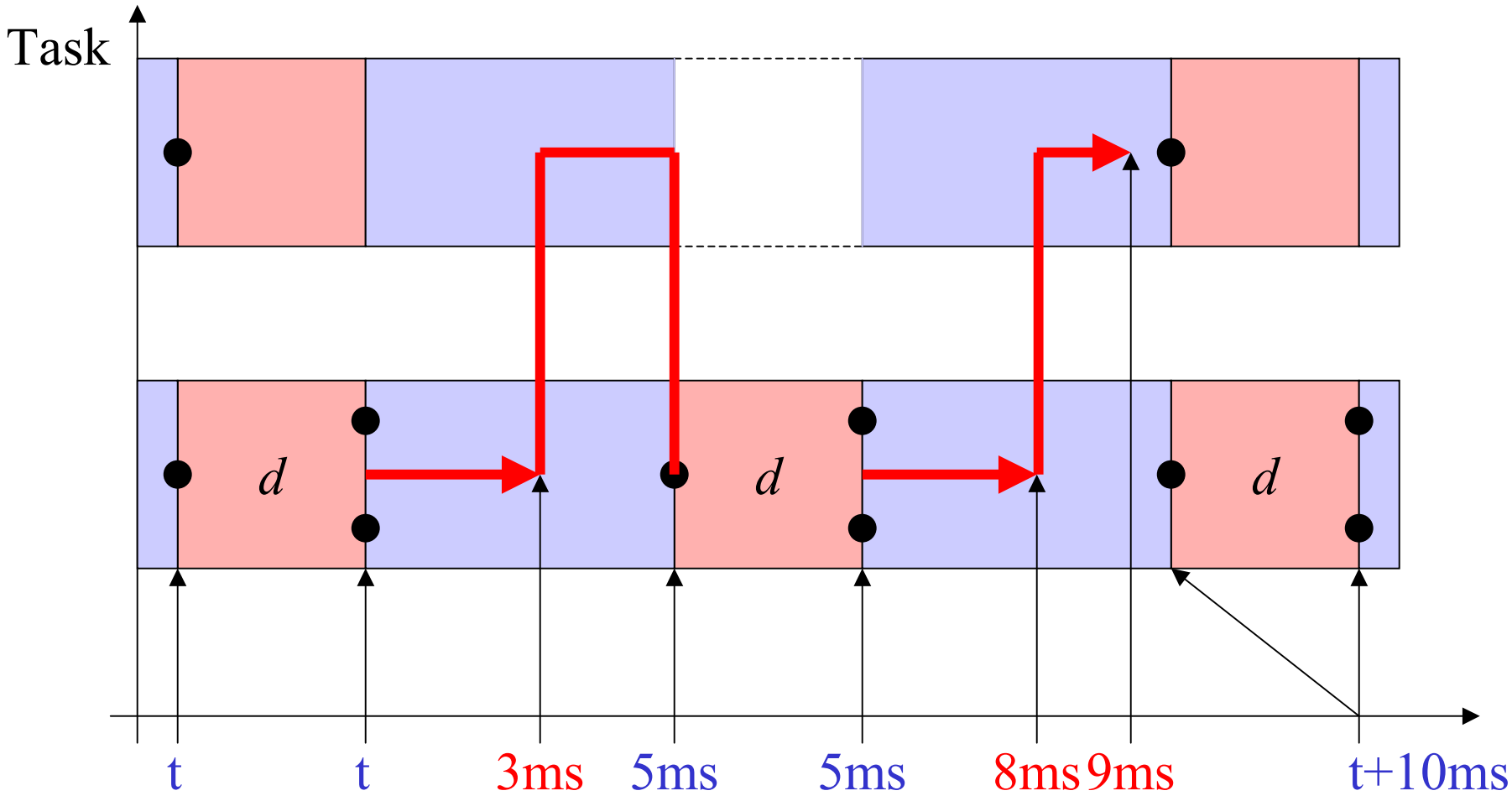
Giotto PM (Single CPU)



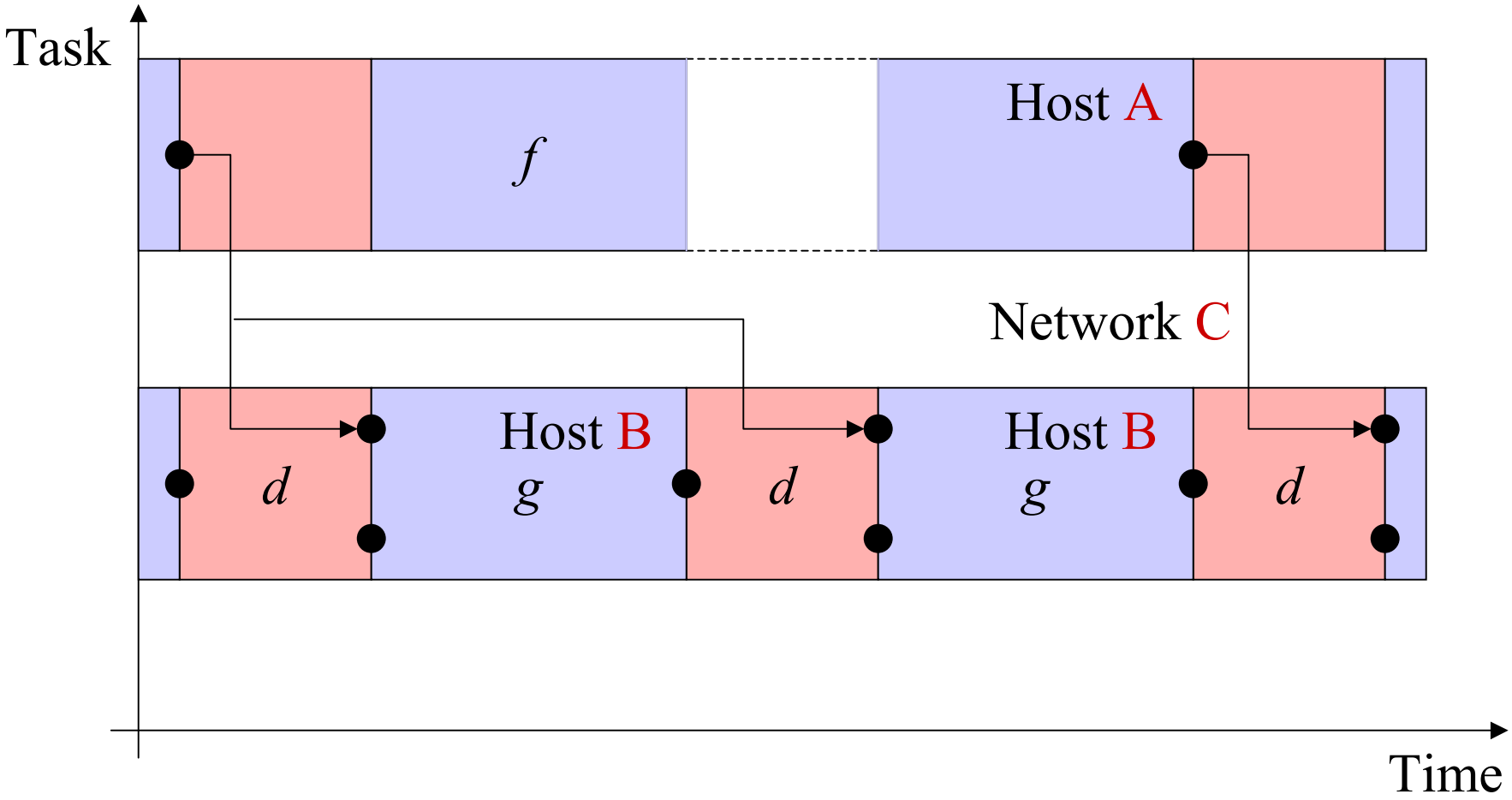
Giotto PMS (Task Priority)



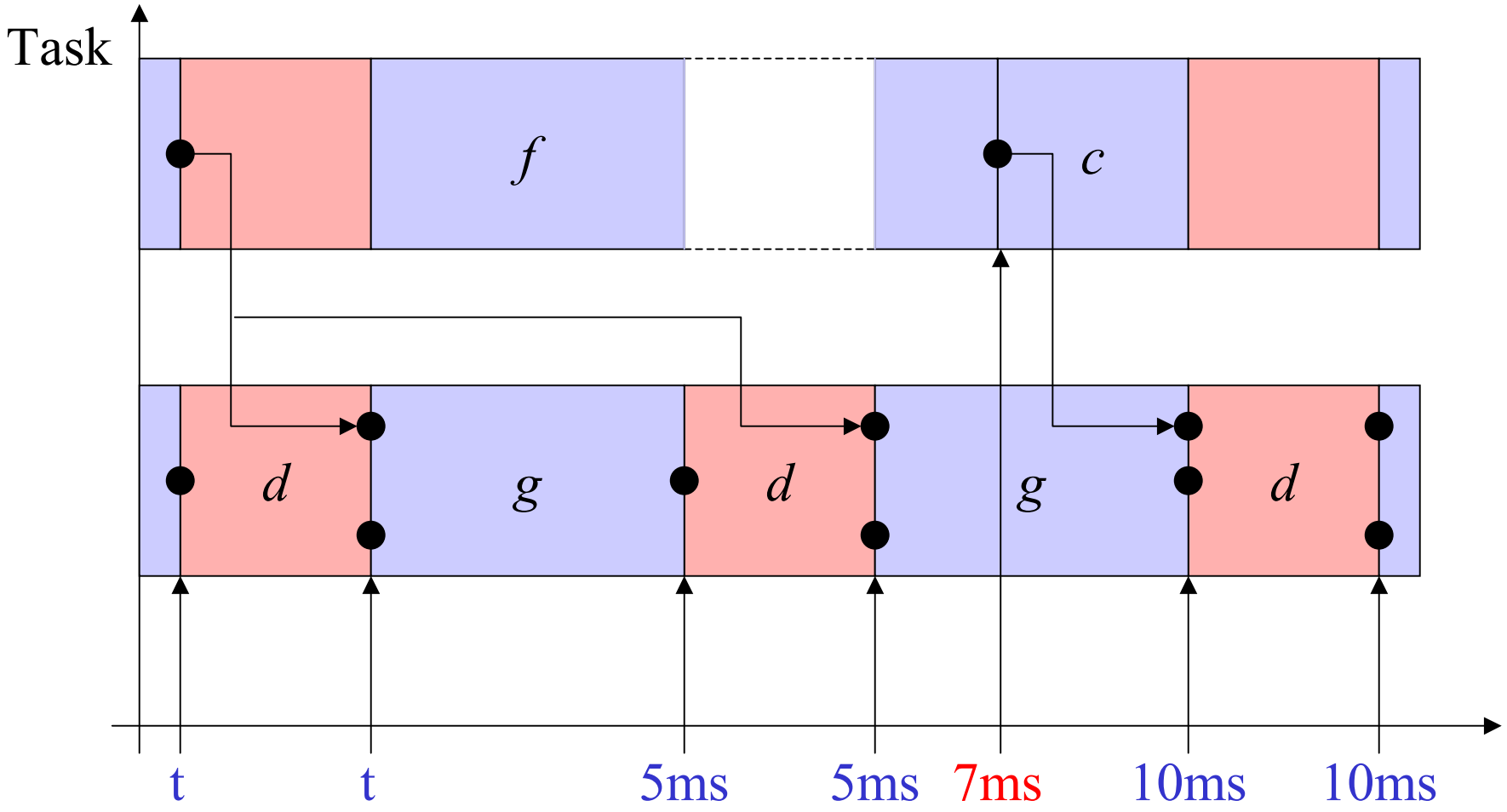
Refinement



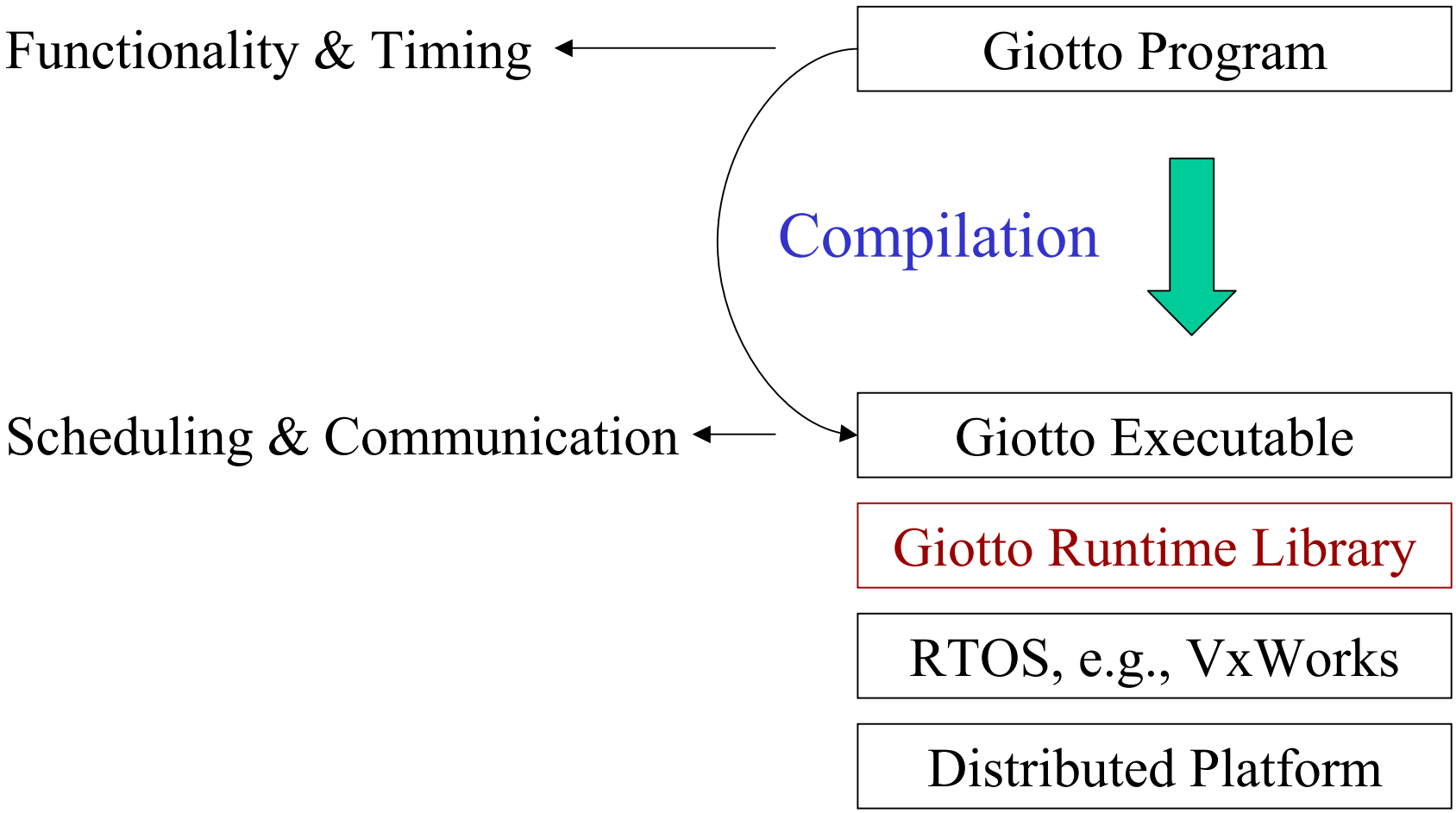
Giotto PM (Two CPUs)



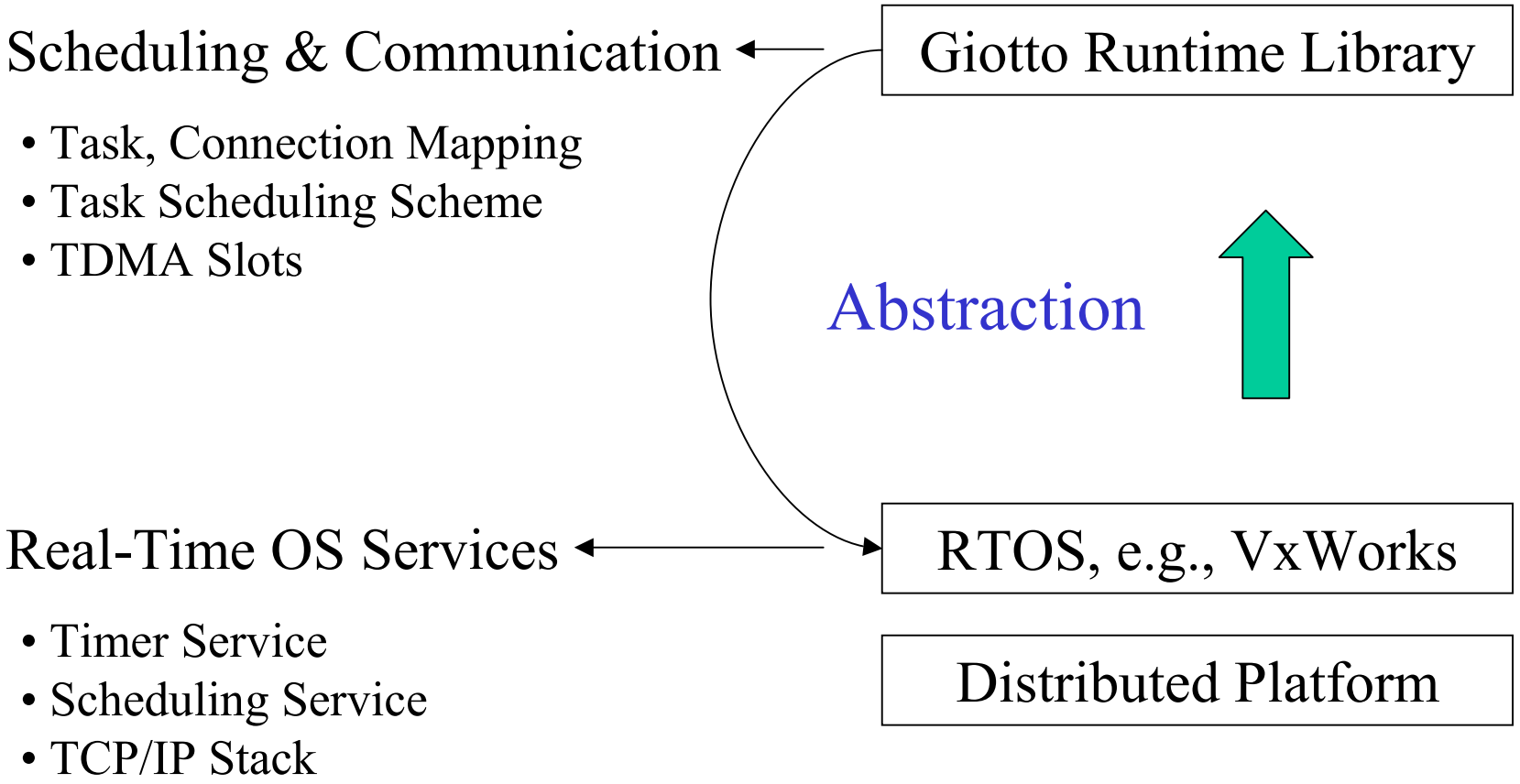
Giotto PMS (TDMA)



The Giotto Runtime Library



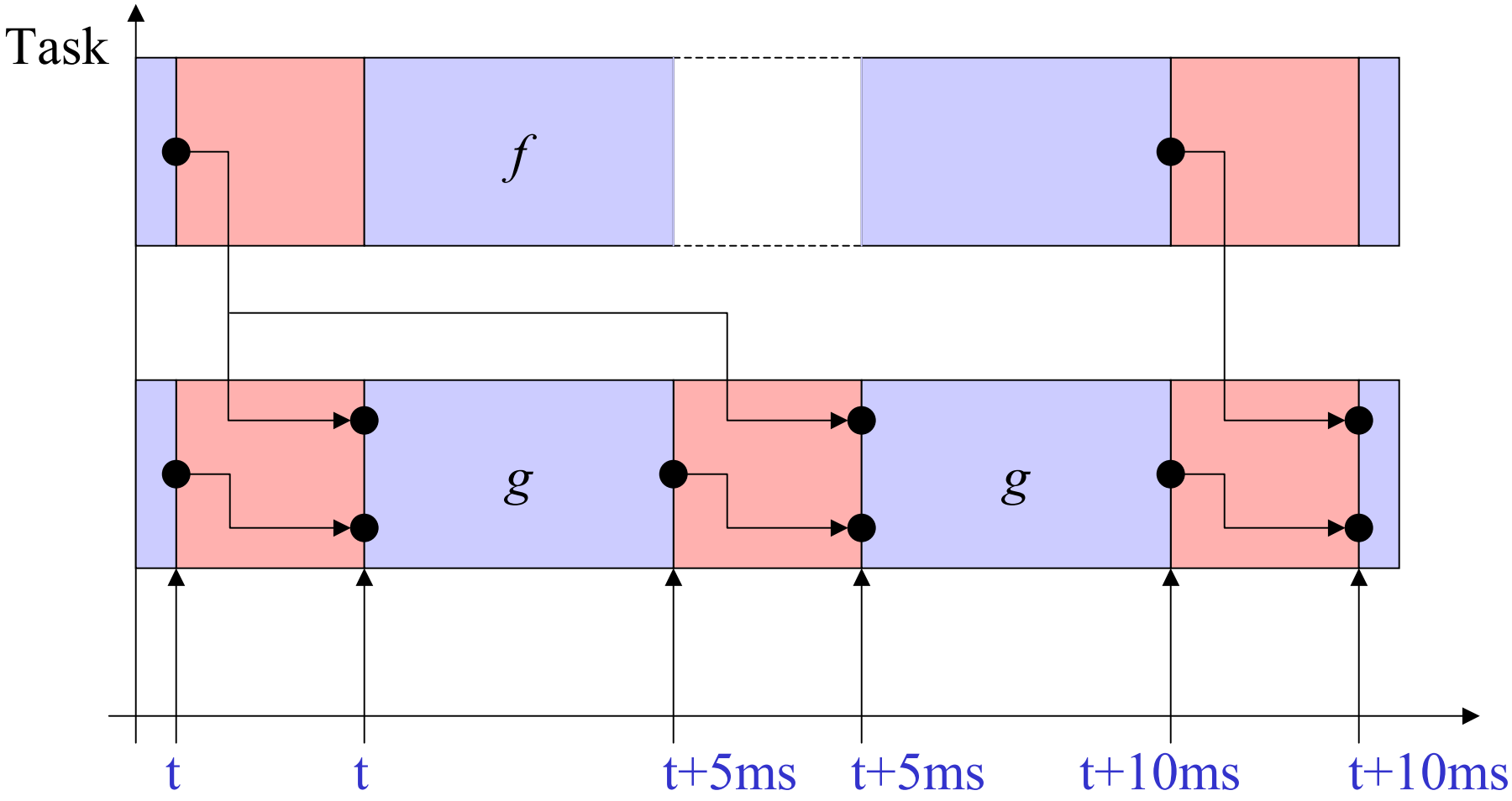
The Giotto Runtime Library



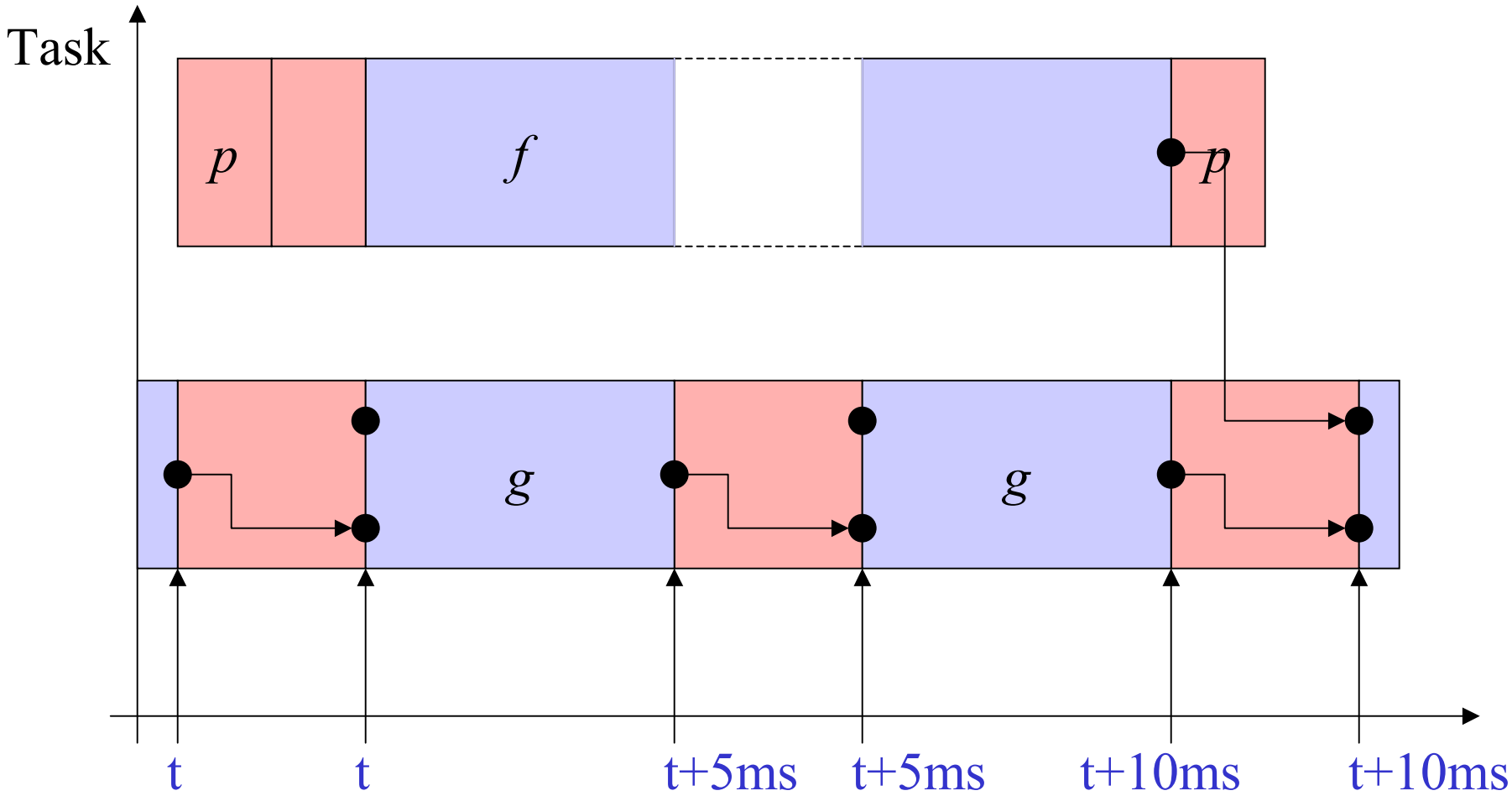
The Giotto Runtime Library

- Available on VxWorks and x86 targets
- Current Experimental Robots:
  - credit card size x86 single board computer running VxWorks
  - wireless Ethernet
  - Lego motors, sensor

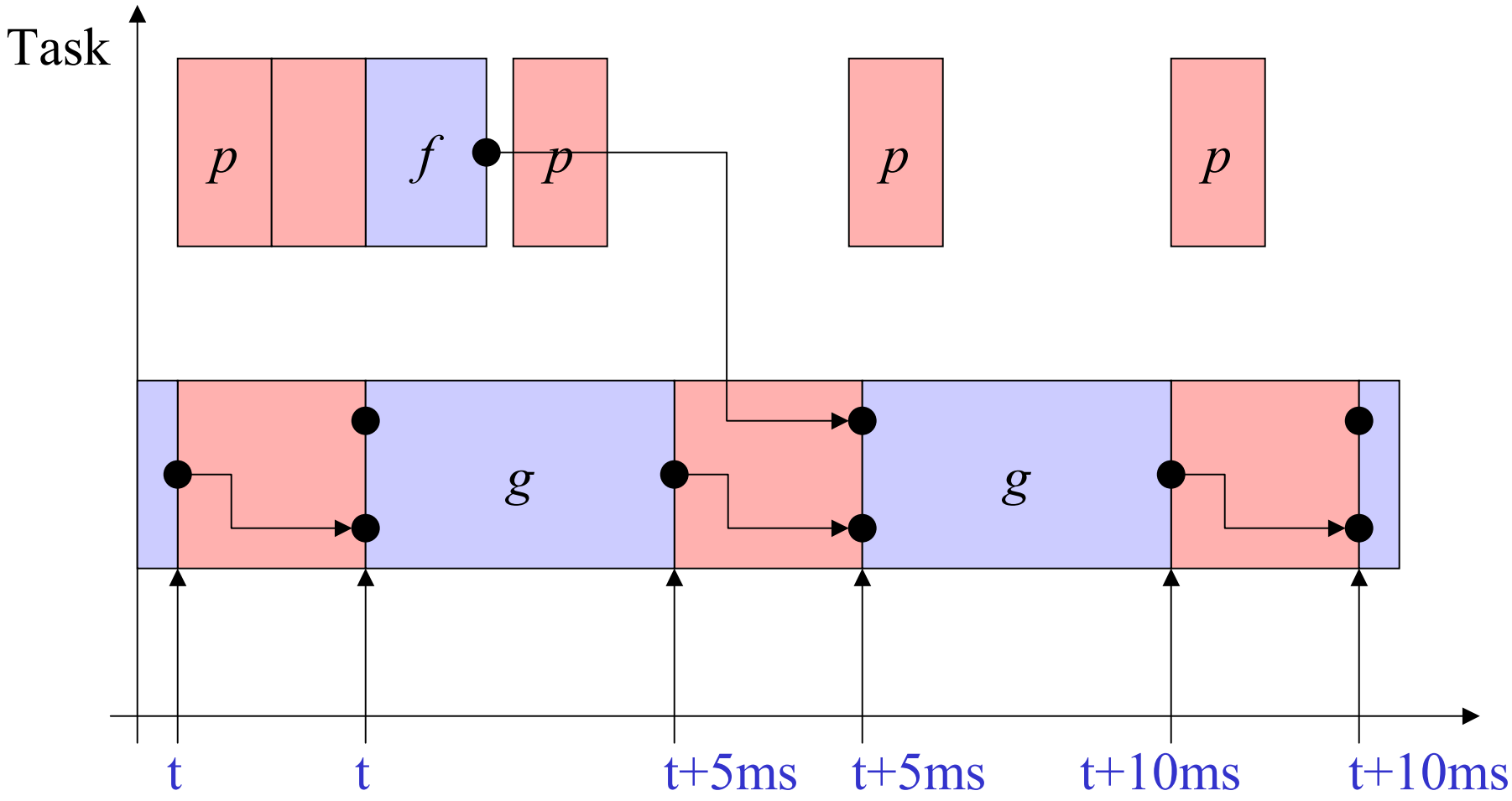
Event- vs. Time-Triggered Tasks



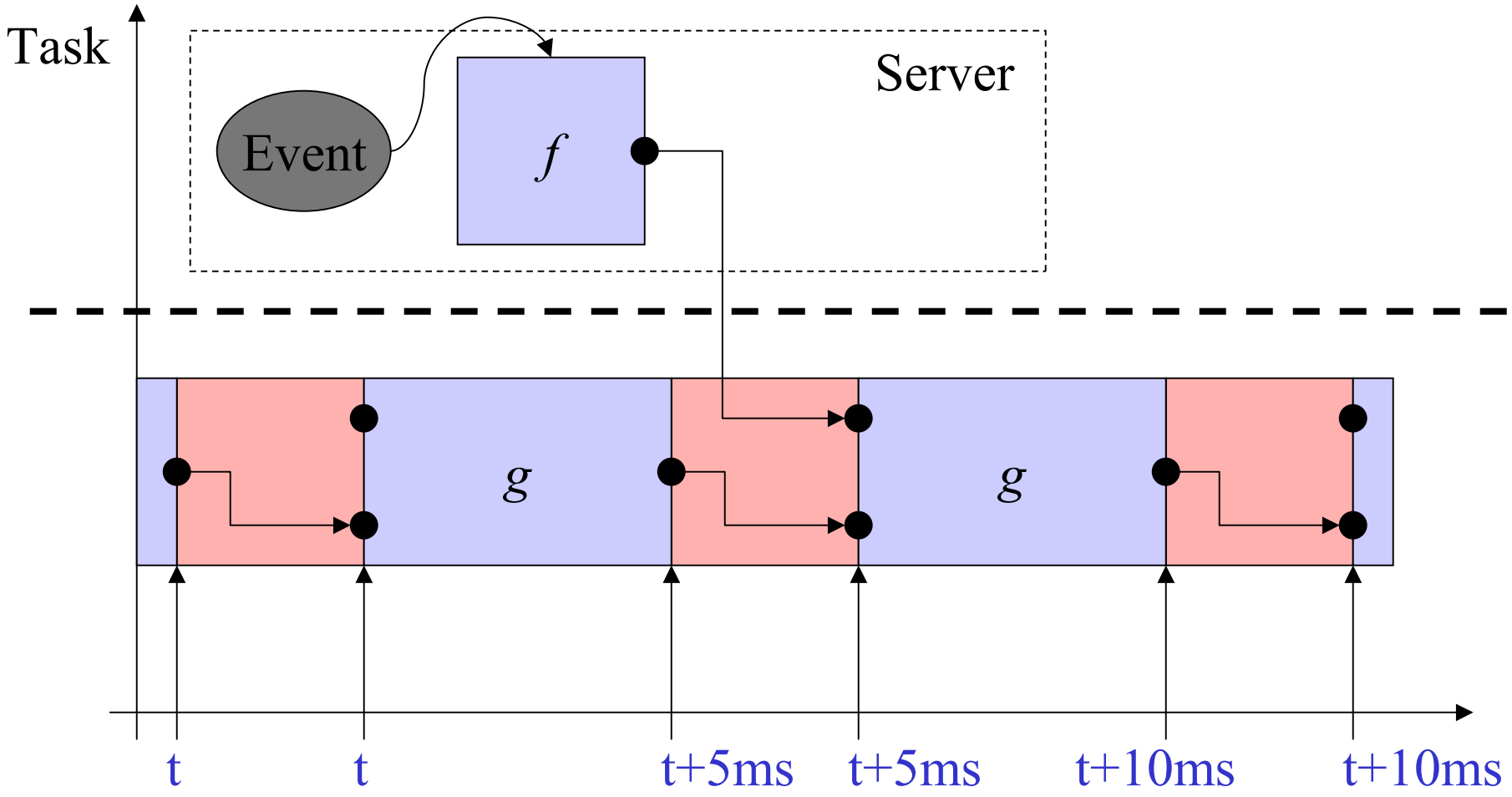
Guarded Task Invocations



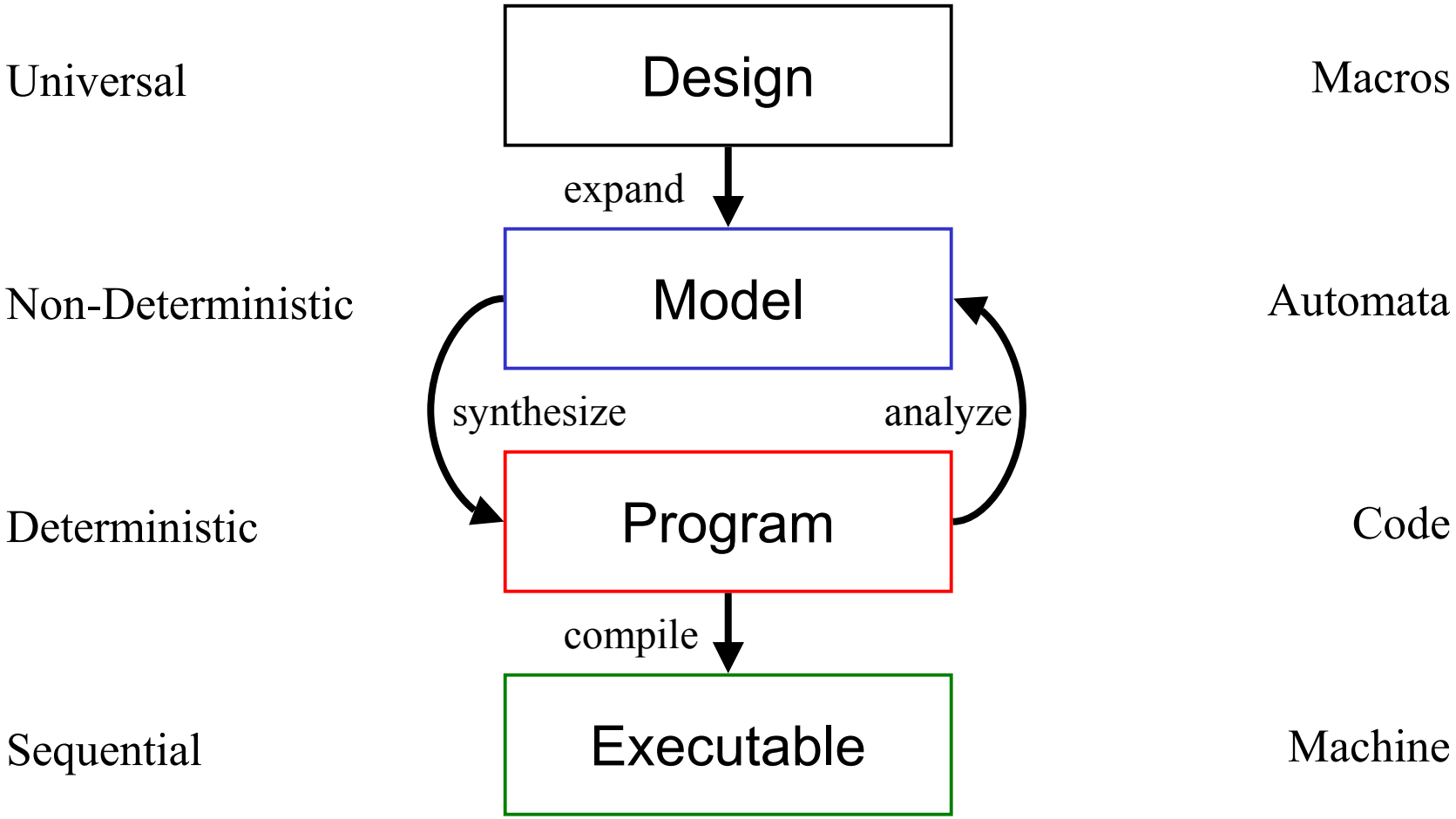
Polling



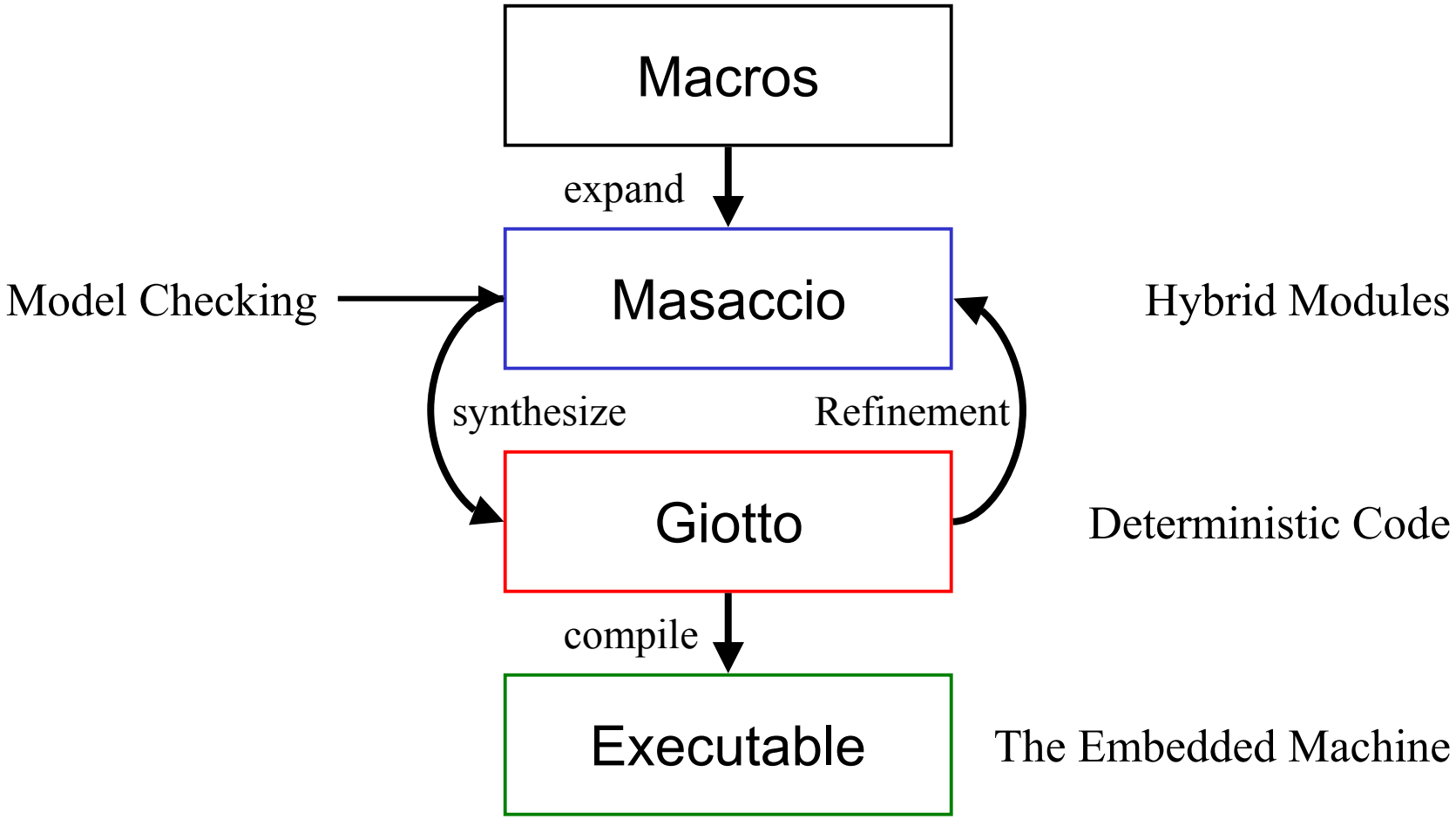
Sporadic Server



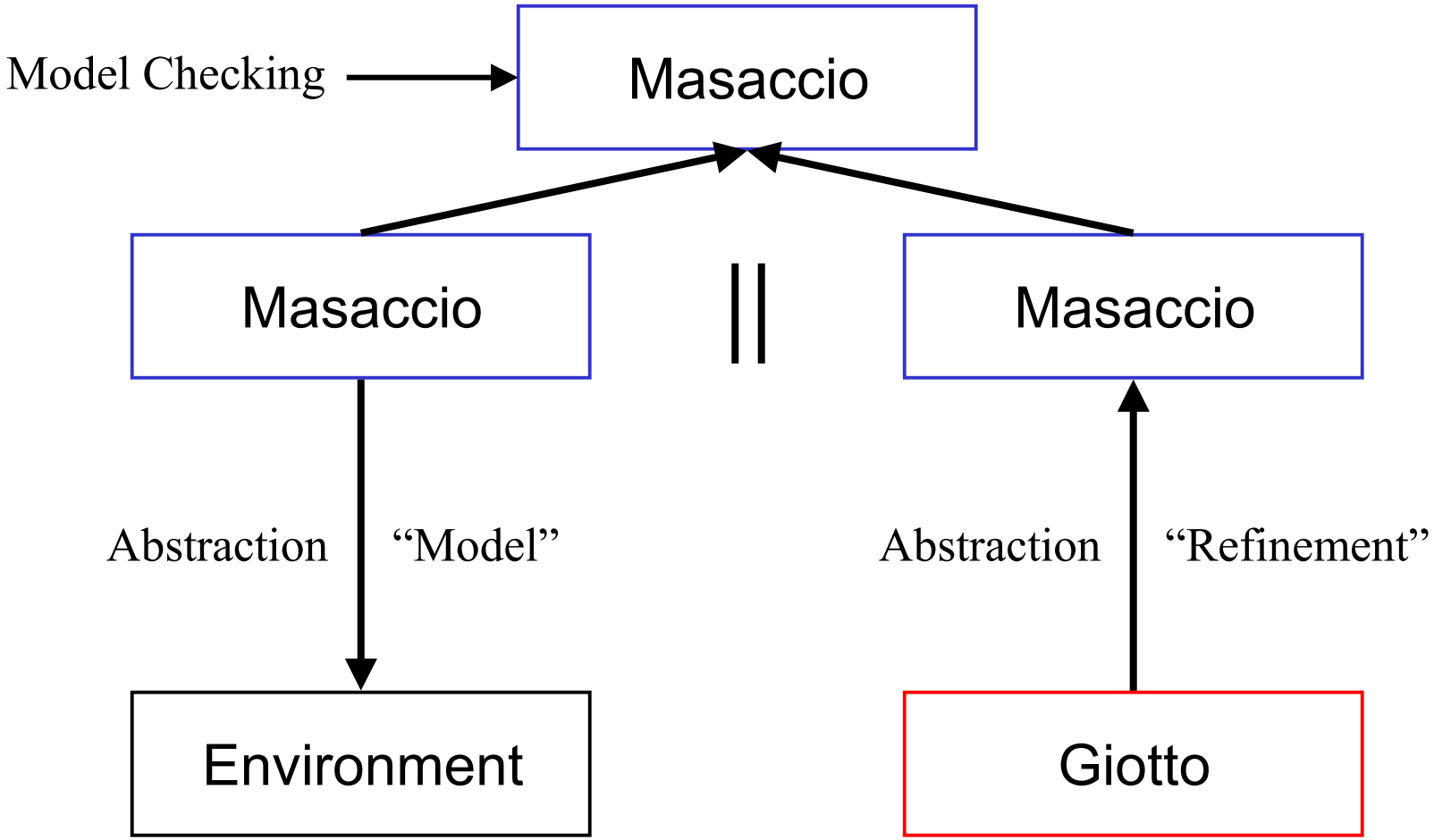
Language Hierarchy



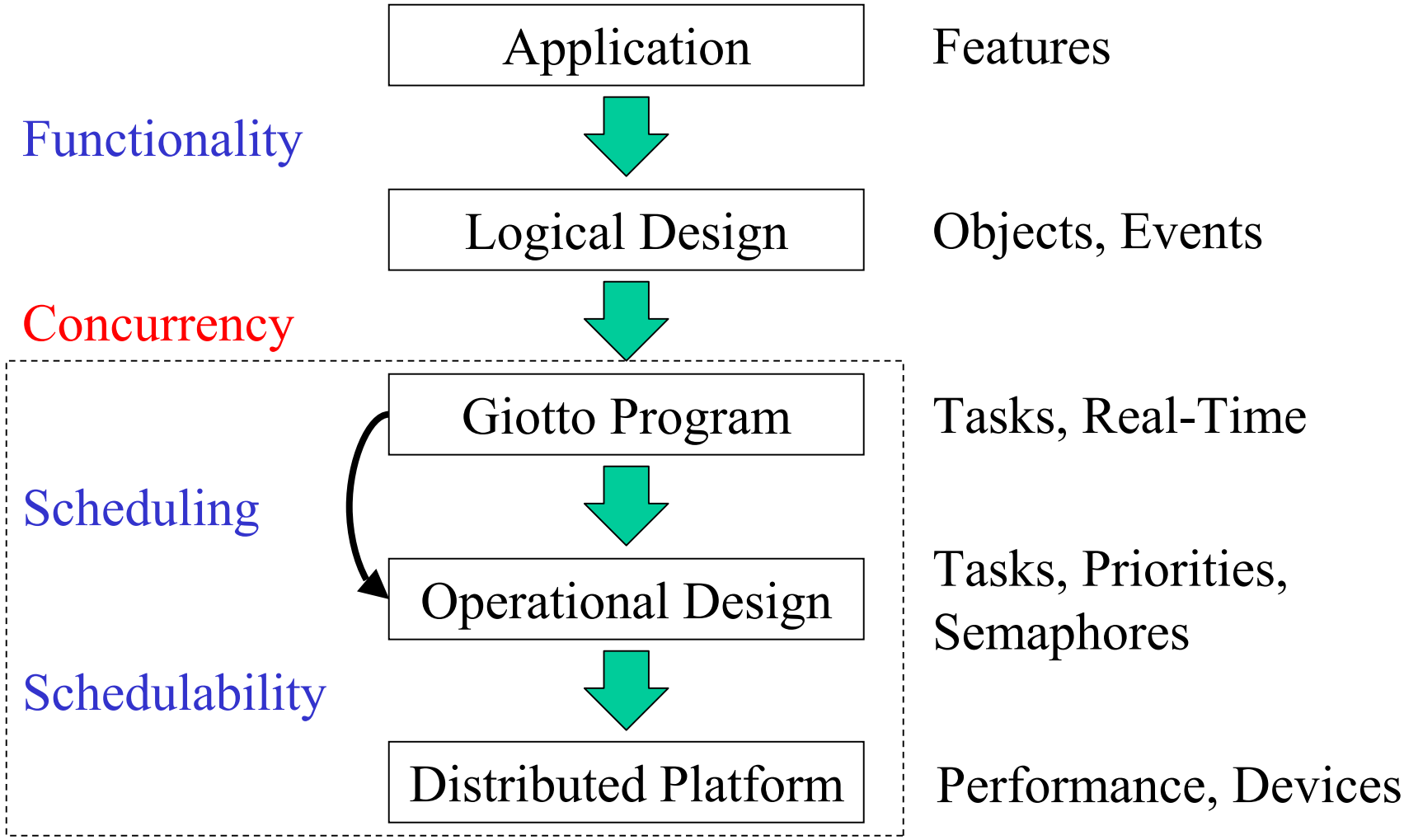
Giotto - Masaccio



Formal Verification



Summary


Application
Features
Functionality
Logical Design
Objects, Events
Concurrency
Giotto Program
Tasks, Real-Time
Scheduling
Operational Design
Tasks, Priorities,
Semaphores
Schedulability
Distributed Platform
Performance, Devices