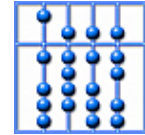


Specification of Distributed Systems

Dr. Bernhard Schätz
Leopold-Franzens Universität Innsbruck
Sommersemester 2005



Overview

1. Introduction

2. Basics: Behavior, Interaction, Concurrency

3. Coroutines

4. Communicating Processes

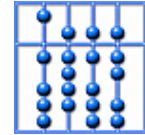
5. Data Flow Models

6. State-Based Models

7. Coordination

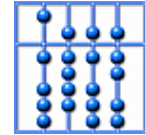
8. Executions

9. Property Descriptions

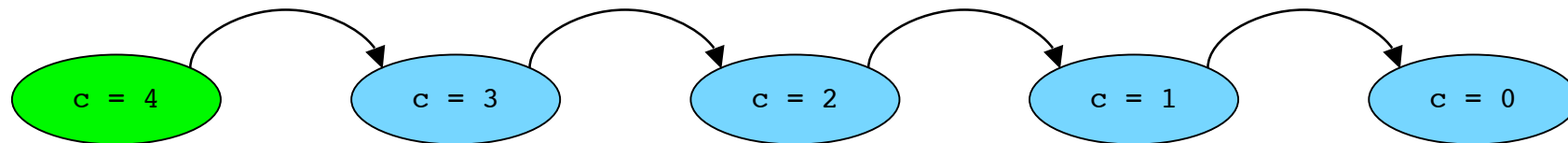


Overview

1. Introduction
2. Basics: Behavior, Interaction, Concurrency
 1. Modeling Computation: State Transition Systems
 2. Modeling Interaction: Labeled Transition Systems
 3. Modeling Concurrency: Synchronized Transition Systems
 4. Modeling Behavior: Streams of Observations
 5. Modeling Communication: Synchronized Behaviors
 6. Modelling Parallelism: Event Structures
 7. Modeling Executions: Kripke Structures
3. Coroutines
4. Communicating Processes
5. Data Flow Models
6. State-Based Models
7. Coordination
8. Executions
9. Property Descriptions



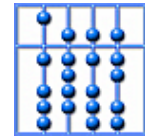
Recap: Modeling Computations



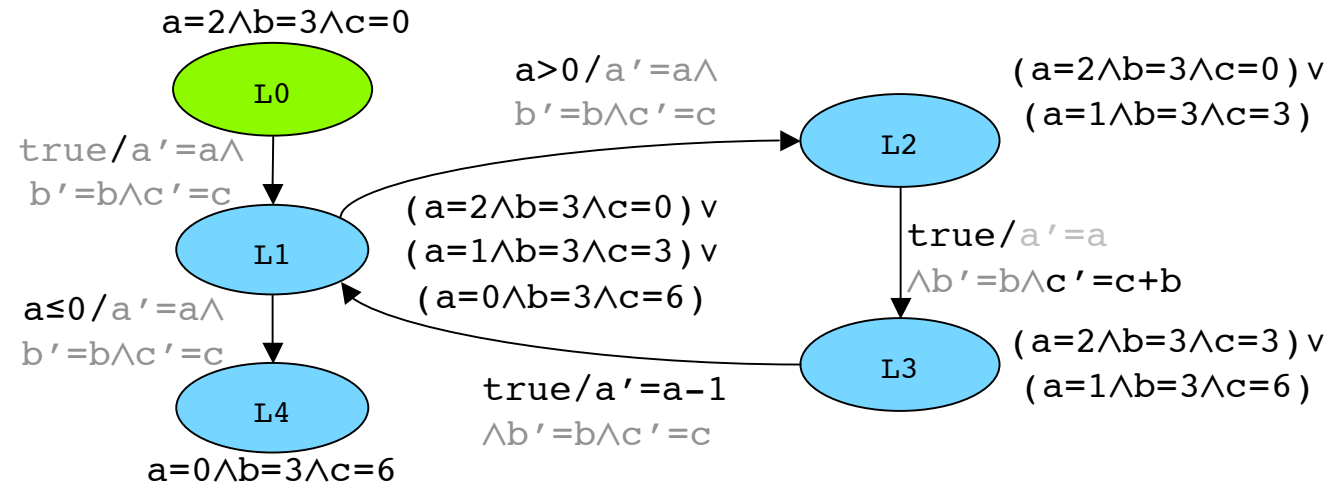
Concepts:

- State: Observation of a system at a specific instance of time
- Transition: Atomic action changing the state of a computation
- Transition relation: Set of possible actions of a computation
- Execution Trace: Sequence of states during a computation

Model: Transition System (S, s_0, T)



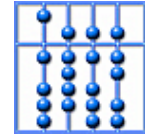
Recap: Modeling Complex Computations



Concepts:

- State: Observation control and data aspects of a system at a specific instance of time
- Invariant: Collection of data states (of a control location)
- Extended Transition relation: Set of possible actions of a computation characterized by pre- and postcondition

Model: Extended Transition System (S, S_0, T)



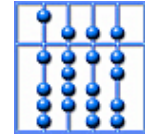
2.3 Modeling Executions: Kripke Structures

Goal: Define a model to describe the behavior of parallel paths of execution

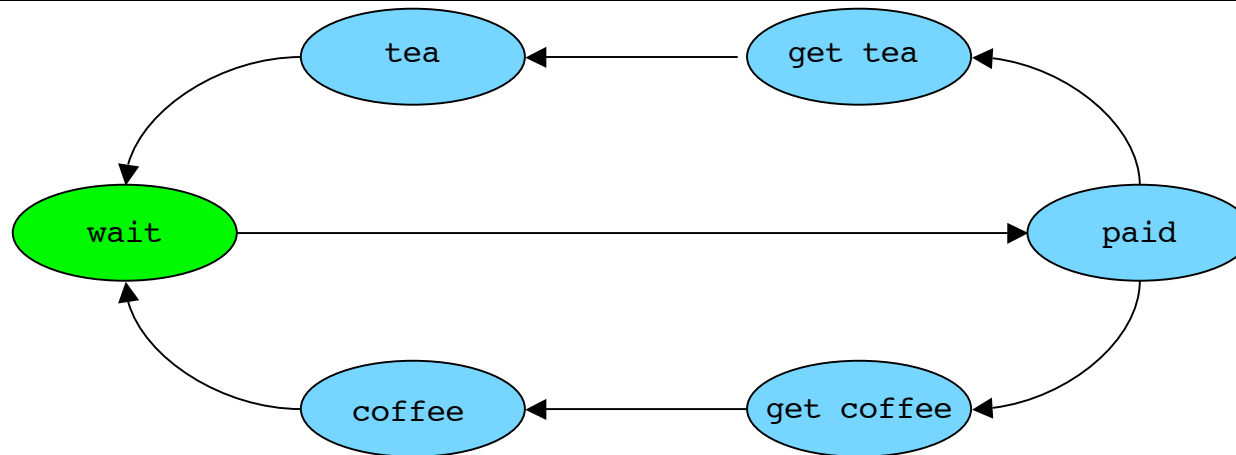
- Relevant: Address aspects evolving with alternative executions
- General: Cover aspects independent of specific model of computation
- Abstract: Ignore aspects like execution speed or platform

Concept: State, observation, transition, execution

Model: Kripke structures



Concept: State

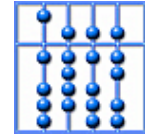


Purpose: Describe system in a stable condition

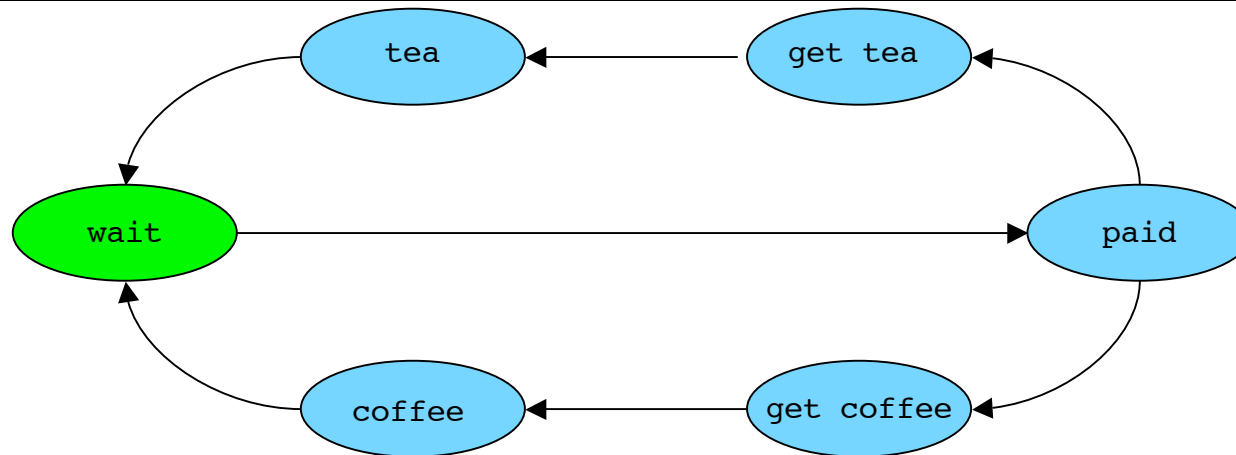
Concept: State $s \in S$

- Discernable: Different states can be distinguished
- Stable: Observations do not change in a state

Example: “Wait for order”, “Coin has been inserted”



Concept: Transition

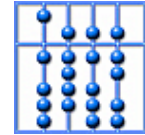


Purpose: Describe change from one state to another

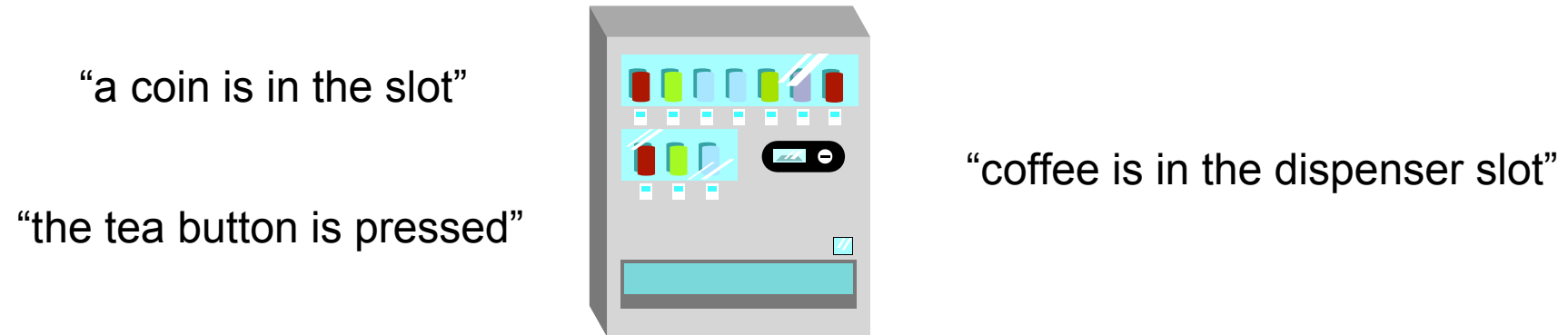
Concept: Transition $t \in T \subseteq S \times S$

- Atomic: Cannot be broken into sub-transitions
- Instantaneous: Cannot be interrupted

Example: $T = \{ (wait,paid), (paid, get tea), (paid, get coffee), \dots \}$



Concept: Proposition

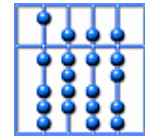


Purpose: Describe observations that can be made about the system

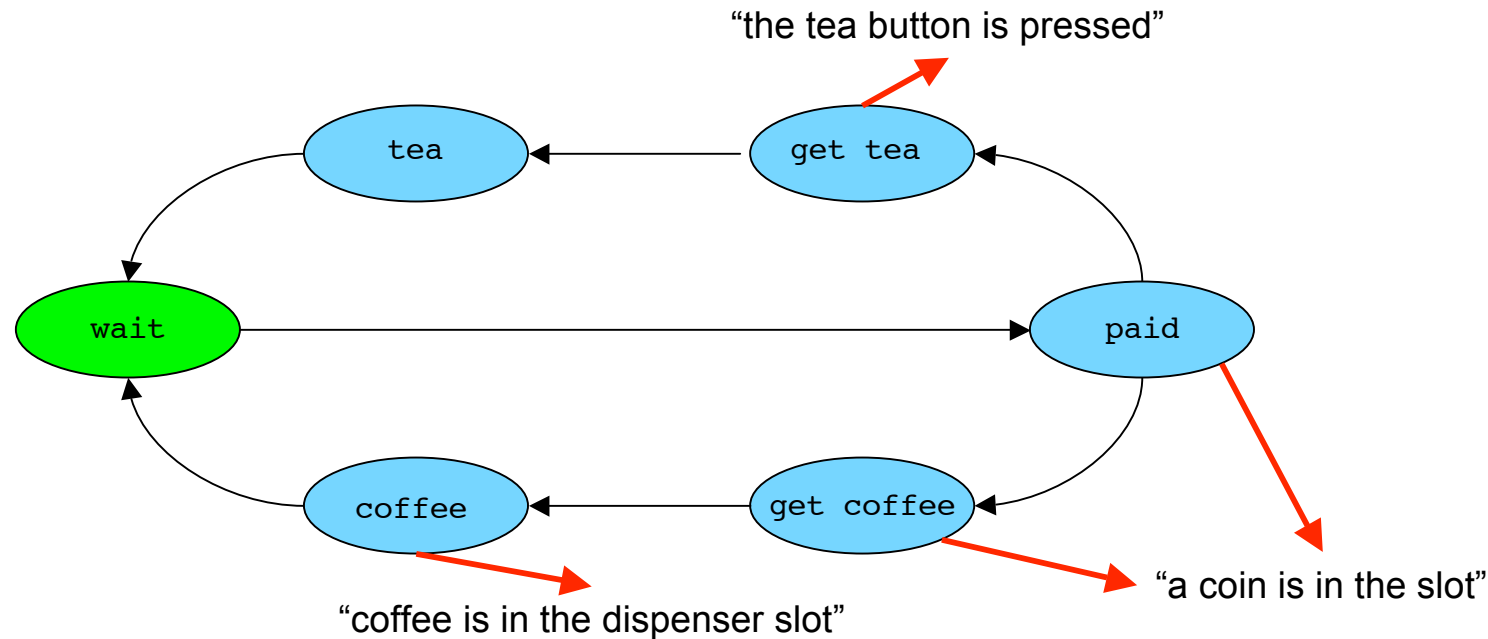
Concept: Proposition $p \in P$

- Observable: Proposition is either true or false
- Atomic: Proposition cannot be broken into sub-propositions

Example: “A coin is in the slot”, “The tea button is pressed”



Concept: Observation

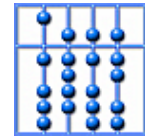


Purpose: Describe those propositions that hold for the current state

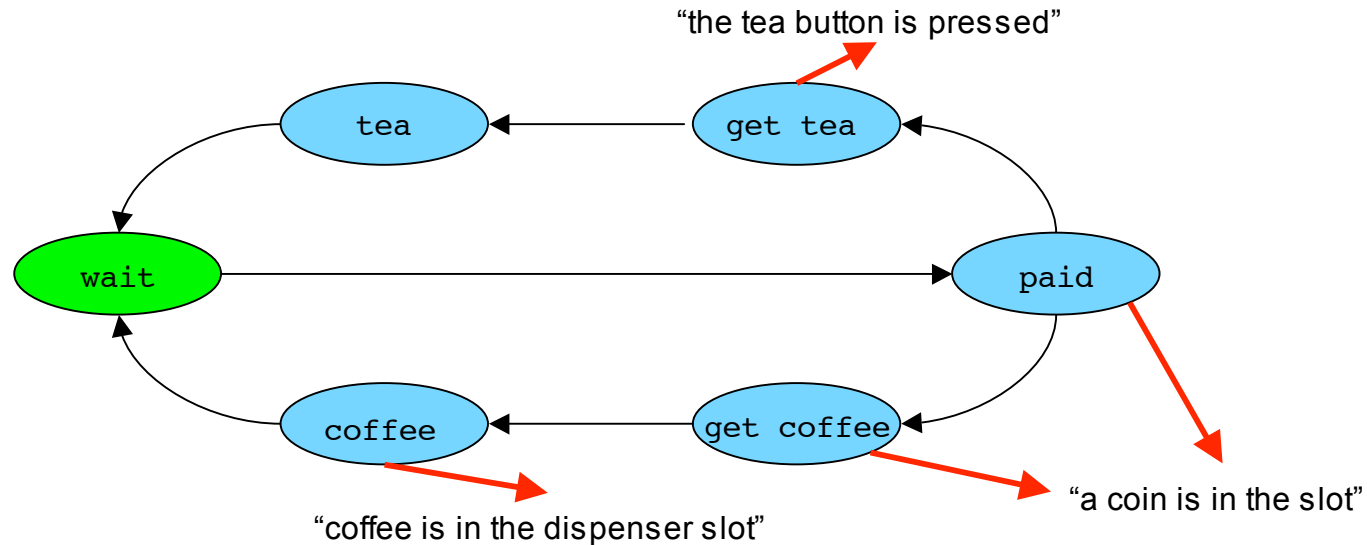
Concept: Observation $o \in O \subseteq S \times P$

- Assigns valid propositions to states
- Propositions may hold in several states

Example: $T = \{ (wait, paid), (paid, get\ tea), (paid, get\ coffee), \dots \}$



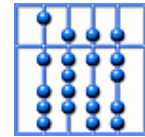
Model: Kripke Structure



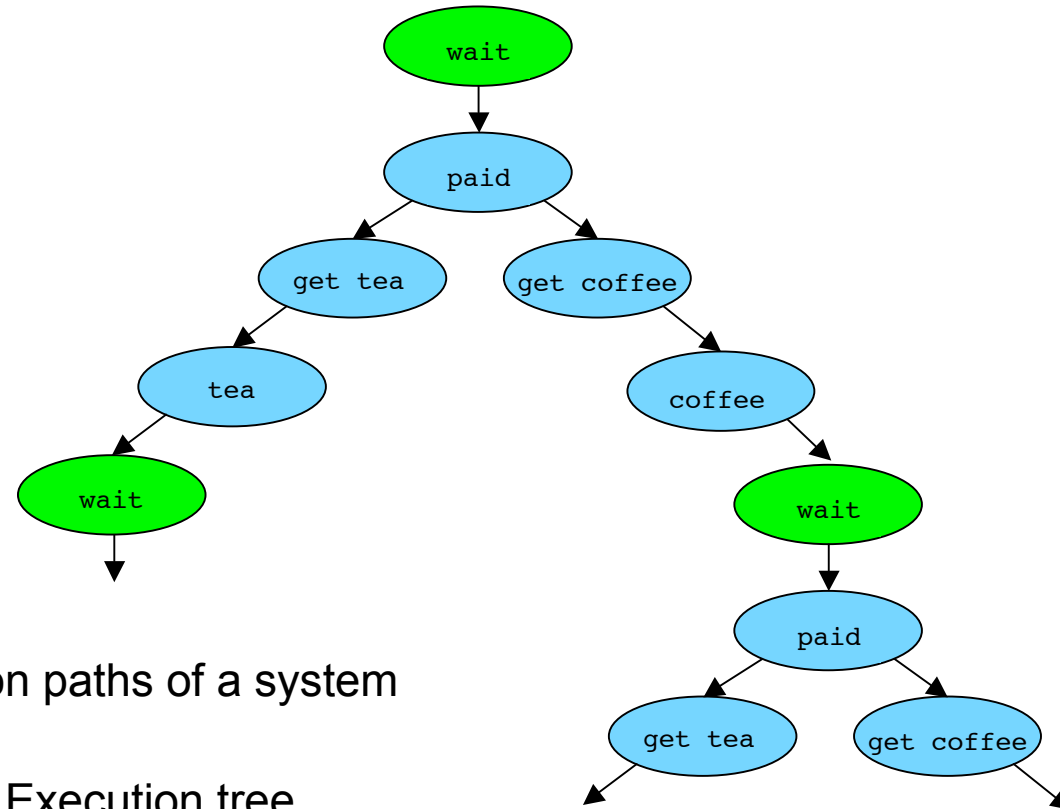
Purpose: Model the change between valid propositions through the changes of a system

Model: Kripke structure (S, S_0, T, P, O)

- S: States of the system
- S_0 : Initial states
- T: Transition relations
- P: Propositions
- O: Observation relations



Concept: Execution

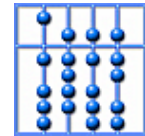


Purpose: Execution paths of a system

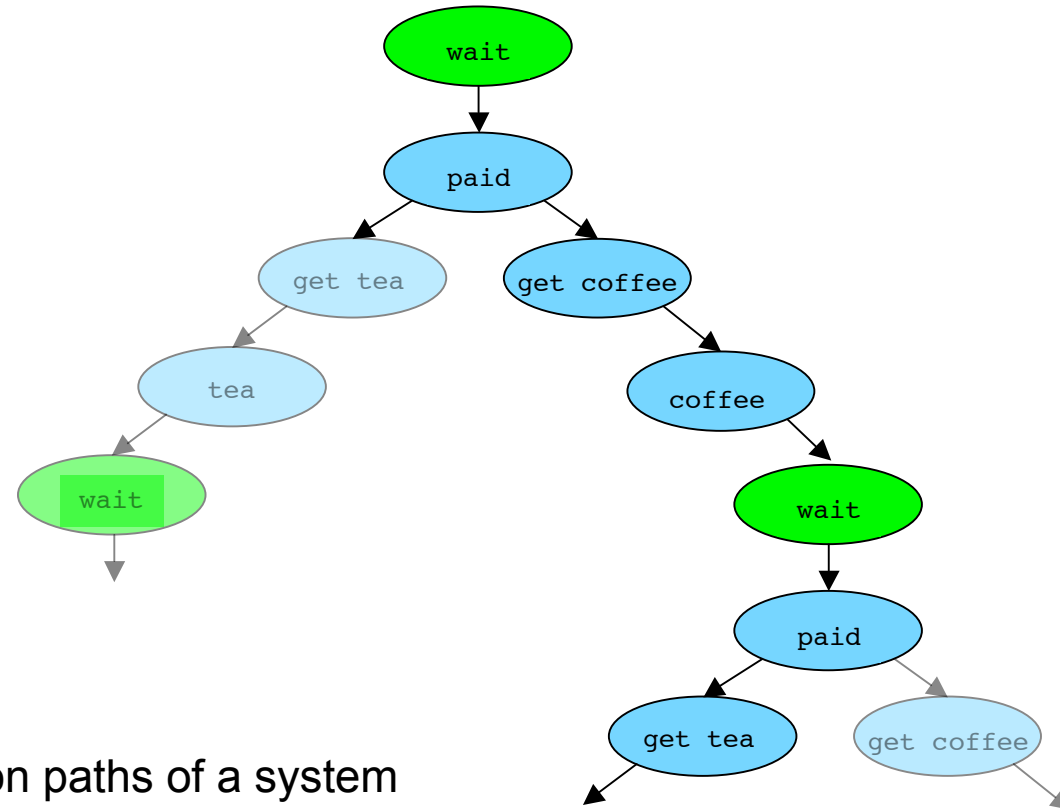
Concept: (Infinite) Execution tree

- Nodes: Correspond to states of the system
- Edges: Connect nodes corresponding to transition relation of system

Note: Finite trees often extended to infinite trees by “leave loops”



Concept: Execution



Purpose: Execution paths of a system

Concept: Execution path

- Linear sub-tree of the execution tree
- Generally complete (cannot be extended)