Introduction to LOTOS

Samira Sadaoui

Contents

Definition

- LOTOS: Language Of Temporal Ordering Specifications
- Specify the temporal ordering of actions
- Describe the OSI[a] services, protocols and interfaces

[a] Open Systems Interconnection

Definition

- Two formalisms (independent but complementary):

  Process Algebra: describe the behavior of concurrent processes (CCS[a] and CSP[b])

  Abstract Algebraic Data Types: specify the data types manipulated and exchanged between processes (ACT ONE[c])

---

[a] Calculus of Communicating Systems

[b] Communicating Sequential Processes

[c] Algebraic specification technique for Correct and Trusty software systems, number ONE

Why LOTOS ?

- ISO[a] international standard
- Expressive power: LOTOS can be applied on different application domains (information systems, distributed & concurrent systems, hardware verification, etc)
- High abstraction level to promote the precise definition of the requirements
- Great modularity to structure, analyze and maintain complex specifications

[a] International Organization of Standardization

Why LOTOS ?

- Operational semantics allowing the execution of the specifications (early prototyping)
- Strong theory for the verification & validation (V&V) of specifications
- Many tools to compile, simulate and verify the specifications

Process Algebra

- Processes or behavior expressions
- Processes are specified and composed with operators
- A process P evolves with the execution of an action (interaction, gate or event)
- Interface of P is the set of the external gates that interact with the environment
- Environment of P: all other processes communicating with P
- Synchronization: many processes participate in the execution of an action at the same time
- Operational semantics of LOTOS: a set of inference rules that translates a LOTOS expression into a labeled transition system LTS

Operators of Basic LOTOS

- inaction: **stop**
- successful termination: **exit**
- internal unobservable action: **i** (to express a lapse of time)
- action prefix: **a ; B**
- nondeterministic choice: **B1 [ ] B2**
- sequential composition or enabling: **B1** $>>$ **B2**
- disabling: **B1 [** $>$ **B2**
- hiding: **hide gates in B**

Basic LOTOS

- parallel composition:
  - interleaving: **B1** $|||$ **B2**
  - full synchronization: **B1** $||$ **B2** (synchronization in all gates)
  - general synchronization: **B1** $|[\ ]|$ **B2**
- process instantiation: instantiate the formal gates with the actual gates

Example: Vending Machine

- High-level representation (black box)
- Interface of VM
- Environment (an observer) of VM
- Observable behavior of VM: all possible sequences of interactions of VM with O
- LTS of VM (action prefix and choice)
- Architecture of VM (internal structure/white box)
- Service and Protocol

Abstract Algebraic Data Types

- A data type:
  - set of values of some type
  - set of functions that can be applied on these values
- An abstract algebraic data type (ADT):
  - implementation-independent description: syntax (how to use the functions) and semantics (what do they mean ?)
  - implementation phase: decide a certain representation of the values

Classes of ADTs

- Basic types: booleans, integers, reals, characters, strings,
- Generic structuring types: set, sequence, table, stack, queue, etc
- Complex specifications (hierarchy of data types)

CADP Tool

- Compilation with *Caesar/Caesar.ADT*
- Verification with *Aldébaran*
- Validation with the simulator *XSimulator*
- Code generation in the language C

Caesar/Caesar.ADT

- Perform syntax and semantic analysis
- Compile ADTs into C types
- Compile LOTOS specifications into LTS using the BCG[a] format

[a] Binary Coded Graphs

Aldébaran

- Perform verification on the LTS generated by compiler
  - Reduction of the complexity of LTS
  - Comparison of two LTSs (e.g., service and protocol)
  - Search of deadlock and starvation
  - Search of particular execution sequences

Verification & Validation

- Verify the consistency and completeness of ADTs
- Prove theorems for ADTs
- Validation of the LTS: edit, print and study it
- Validation with a prototype: interactive or random simulation
- Verification with (Cléopâtre) of some properties (formulas in temporal logic) such as the safety