Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

DYNAMO

HS 2023

Prof. Dr. Lana Josipović and Jiahui Xu
based on Prof. Lothar Thiele's material

Discrete Event Systems

Exercise Sheet 12

1 Specifications in Computation Tree Logic (CTL)

An **elastic system** is a collection of computation and storage modules interconnected by channels. Every channel can propagate data from one module to the other. Every channel has a pair of bidirectional control signals that implement the handshake between the sender and receiver; the control signals guarantee that data transfer happens only if the sender has valid data and the receiver is ready to receive it. An elastic system is tolerant to variations of computational and communication delays, thus, they are correct by construction. In this exercise, we are interested in deriving some specifications that an elastic system must respect; any violation might lead to the corruption of computation data or a system deadlock.

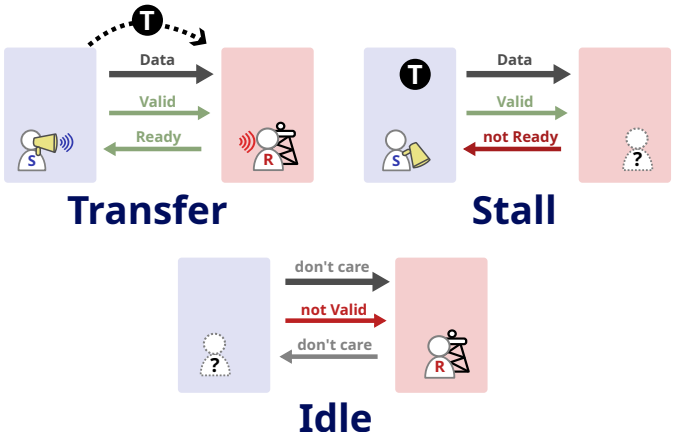


Figure 1: The states of a channel described by the *synchronous elastic flow* (SELF) protocol—a standard handshake protocol for elastic systems.

Determine the CTL formulas for the following properties. The solution should be expressed only using the CTL operators (i.e., $\mathbf{AG}, \mathbf{AF}, \ldots$), standard logical operators (i.e., $\neg, \vee, \wedge, \rightarrow$), and handshake signal values (i.e., *valid* and *ready*). For example, you can denote the property "*the receiver is not ready*" as $\neg$*ready*.

a) **Liveness**: each request (sender asserts a *valid*) in the channel should eventually be acknowledged (receiver asserts *ready*).

b) **Fairness**: the receiver *ready* signal should assert infinitely often.

c) **Persistency**: when the sender asserts its *valid* signal high, then it should be remained high until its respective *ready* is also high.

2 Model Checking CTL Specifications (I)

Consider the state machine depicted in Figure 2; property a holds in the shaded states (i.e., state 0 and 3).

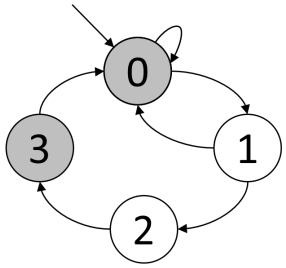


Figure 2: The state machine used in Exercise 2.

To distinguish a CTL formula and the set of states that satisfy the formula, we denote the set of states that satisfy a CTL formula ϕ as $[\![\phi]\!]$. Moreover, we say that a state machine T satisfies ϕ if $Q_0 \subseteq [\![\phi]\!]$, where Q_0 is the set of initial states of T. For each of the following properties described using CTL formulas, determine the set of states that the property holds:

- $Q_a := [\![\mathbf{EF}\, a]\!]$
- $Q_b := [\![\mathbf{EG}\, a]\!]$
- $Q_c := [\![\mathbf{EX}\, \mathbf{AX}\, a]\!]$
- $Q_d := [\![\mathbf{EF}\, (a \wedge \mathbf{EX}\, \neg a)]\!]$

3 Model Checking CTL Specifications (II)

Consider a state machine T with a set of states S, the transition relation $R : S \times S$ with its characteristic function $\psi_R(q, q')$ (which evaluates to *true* only if the state transition $(q, q') \in R$), and the set $Z \subseteq S$ with its characteristic function $\psi_Z(q)$ (which evaluates *true* only if state $q \in Z$). Our goal is to devise an iterative algorithm to find the characteristic function $\psi_{[\![\mathbf{AF}\, Z]\!]}(q)$ of the set of all states satisfying $\mathbf{AF}\, Z$, $[\![\mathbf{AF}\, Z]\!]$.

a) Determine the relation between $\mathbf{EG}\, Z$ and $\mathbf{AF}\, Z$.

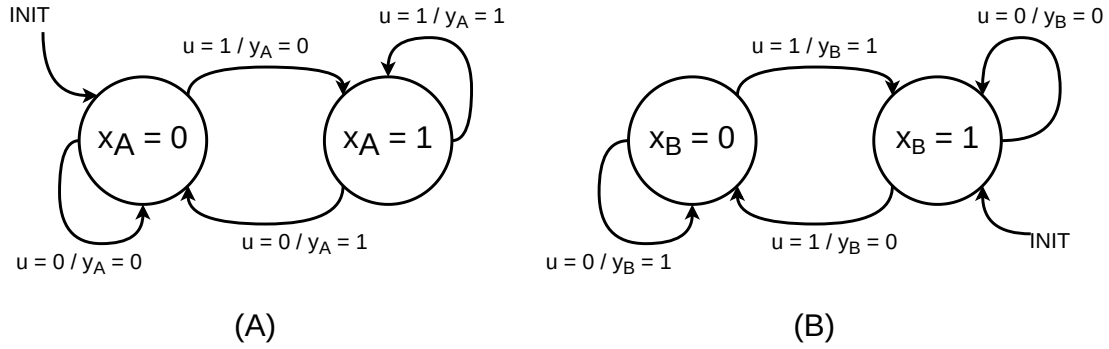
b) Using this relation, formulate an iterative procedure to find the set of states that satisfy $\mathbf{AF}\, Z$. Use regular set operations to find the procedure. You can use the predecessor function $Pre(Q, R)$, which returns the set of states from which we can reach states in Q using the transition relation R in one (backward) step.

$$Pre(Q, R) = \{q' : \exists q,\, \psi_R(q', q) \cdot \psi_Q(q) = 1\}$$

c) Translate the iterative procedure from **b)** into an algorithm using Boolean expressions. Assume that you are given with the characteristic function $\psi_{Pre(Q,R)}$ for each set Q.

4 Sequential Equivalence Checking

Here are two simple finite state machines:



For each state machine, the state is represented using 1-bit encoding (x_A and x_B), one 1-bit output (y_A and y_B), and one common 1-bit input (u). We want to verify whether these two state machines are equivalent. For this, we would like to investigate into the following steps:

a) Determine the characteristic function $\psi_A(x_A, x'_A, u)$ and $\psi_B(x_B, x'_B, u)$ of the transition relation for the two state machines A and B.

b) Determine the characteristic function $\psi_f(x_A, x'_A, x_B, x'_B)$ of the transition relation for the product of the two state machines.
Reminder: $\psi_f(x_A, x'_A, x_B, x'_B) := (\exists u : \psi_A(x_A, x'_A, u) \cdot \psi_B(x_B, x'_B, u) = 1)$

c) Determine the characteristic function $\psi_X(x_A, x_B)$ of the set of reachable states of the product state machines.

d) Determine the characteristic function $\psi_Y(y_A, y_B)$ of the set of reachable states of the product state machines.

e) Are the two state machines equivalent? *Hint:* Evaluate, for example, $\psi_Y(0, 1)$