LOTOS/CADP – based  AciD – VERIFICATION

Michael Yoeli
myoeli@cs.technion.ac.il

Ref. : Michael Yoeli and Abraham Ginzburg
LOTOS-based verification of asynchronous circuits
CS. Technion, Haifa, 1998, Rep.#CS0951
(being revised!)

See also: Michael Yoeli and Abraham Ginzburg
Petri – net based verification of asynchronous
circuits, CS. Technion, Haifa, 1999, Rep.#CS0959

The above reports are accessible at:
http://www.cs.technion.ac.il/Reports/

CADP = Caesar – Aldebaran Distrib. Package,
Vasy, Grenoble.
See: http://www.inrialpes.fr/vasy/cadp.html
Definition of Realization

Let IMPL := Implementation Process

SPEC := Specif. Process

Process := LOTOS Process or LTS (Labelled Trans. Syst.)
          or Non – Deterministic Automaton
          or Labelled Petri Net

We assume that IMPL and SPEC share the same alphabet
(i.e., set of observable actions) and that this alphabet is
partitioned into INPUTS and OUTPUTS.

IMPL realizes (notation: |= ) SPEC iff

Cond1: SPEC || IMPL is obs. equiv. to SPEC

Cond2: IMPL is live – lock free

Cond3: No “undesirable” outputs (to be formalized)

Note: Cond1 implies L(SPEC) subset of L(IMPL)

Motivation

Cond1 insures that any behaviour specified by SPEC can be
performed by IMPL.

Cond2 prevents IMPL to enter a cycle of i - transitions
Verifying Conditions 1 – 3 Using CADP

Cond1:

Method (1). Generate SIMPL.lotos =
SPEC.lotos || IMPL.lotos

Convert SIMPL.lotos into SIMPL.aut,
and SPEC.lotos into SPEC.aut.

Then check SIMPL.aut and SPEC.aut
w.r.t. obs. equiv.

Method (2). Generate SPEC.aut and IMPL.aut;
Then obtain SIMPL.aut =
SPEC.aut || IMPL.aut;
continue as above.

Cond2: Check IMPL.aut w.r.t. livelock.

Cond3:

Formalization. Let w1 be an action sequence of SPEC,
w2 an action sequence of IMPL, and assume
the two sequences coincide w.r.t. their
observable actions. Assume z is an output,
and that z is the observable action imme-
diately following w2. Then the same must
hold w.r.t. w1.

Verification. Here is one way to verify Cond.3.
Let iIMPL.lotos be the process obtained
from IMPL.lotos by replacing each output,
say z, by i;z. Then SPEC.lotos || iIMPL.lotos
should be deadlock – free.
However, frequently ad-hoc methods are
preferable.