Laboratory on Reachability Analysis, Equivalence and Property Verification

Gianpiero Cabodi *Stefano Quer*

Politecnico di Torino

Torino, Italy

{gianpiero.cabodi,stefano.quer}@polito.it

http://staff.polito.it/{gianpiero.cabodi,stefano.quer}/

Outline

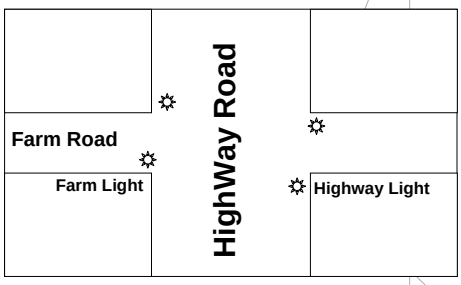
- **Benchmarks**
- **A Demo Traversal Package: nanotrav**
- **A Traversal & Verification Tool: VIS**

Benchmarks

- **Combinational benchmarks**
  - **`c17.blif`, c..., etc.**
- **Sequential benchmarks**
  - **`s713.blif`**
    `#PI=35, #PO=23, #FF=19, #Gate=393`
  - **`s1512.blif`**
    `#PI=29 , #PO=21, #FF=57 , #Gate=780`
  - **`s1423.blif`**
    `#PI=17, #PO=5, #FF=74, #Gate=657`

  (from now on <c>)

- **Traffic Light Controller (TLC)**
  - **`tlc.mv` (description), `tlc.ctl` (properties)**



nanotrav

- **Main features**
  - **Toy-tool within the CUDD distribution**
  - **Compiled with CUDD**
  - **In the nanotrav directory in CUDD**
  - **Given a combinatorial or sequential circuit it**
    - **Find a variable ordering through static and/or dynamic techniques**
    - **Performs reachability analysis (of sequential circuits)**
    - **Performs combinational verification (compare all next state and output functions)**
    - **Etc.**

- **Laboratory duty**
  - **Check the following reachability analysis commands out**
    - **`nanotrav –p 1 –trav <c>`**
    - **`nanotrav –p 1 –trav –ordering dfs <c>`**
    - **`nanotrav –p 1 -trav –autodyn <c>`**

    Analyse their effect!
  - **Check the following verification commands out**
    - **`nanotrav –p 1 –verify <c1> <c2>`**
    - **`nanotrav –p 1 –verify –ordering dfs <c1> <c2>`**
    - **`nanotrav –p 1 -verify –autodyn <c1> <c2>`**

    Where <c1> and <c2> are:
    - **The same circuit**
    - **One original circuit and one manually modified**

    Analyse their effect!

- **Advanced duty**
  - **Check the following (nanotrav) parameters out**
    - `–image part`
    - `–image mono`
    - `–approx over`
    - `-approx under`
    - `-reordering sifting`
    - **etc.**

VIS

- **Main features**
  - **UC-Berkeley, Brayton & co, 1996**
  - **Verification tool**
    - **Reachability Analysis**
    - **Approximate Reachability Analysis**
    - **Equivalence Checker**
    - **Property Checker**
    - **Synthesis Link through SIS (UC Berkeley)**
  - **The VIS tool includes two packages**
    - **glu: Includes libraries for arrays, hash-tables, BDDs, etc.**
    - **vis: Main package**

- **Laboratory Duty**
  - **Grab and uncompress it**
  - **Compile it (first the glu directory then the vis one)**
    - **"Run"**
      ./configure
      to build Makefile in the root directory
    - **"Run"**
      Make

- **Run the TLC**
  - **Directory**
    - **vis-2.0/vis-2.0/examples/tlc**

    (description in blif-mv format (tcl.mv file))
  - **Check CTL properties**
    - **tlc.ctl**

    Which is the meaning?
  - **Run command sequence**
    - `(help)`
    - `read_blif_mv tlc.mv`
    - `init_verify`
    - `model_check –i –v2 tlc.ctl`
    - `quit`

- **Advanced duty**
  - **Check reachability analysis commands within VIS**
  - **Run script**
    - `script_compute_reached.robust`

    in the vis-2.0/share directory for sequential circuits (e.g., `s1423.blif`)