Component Interaction Automata – equivalences, verification

Pavlína Vařeková

Verification of Component-Based Systems project group

Parallel and Distributed Systems Laboratory

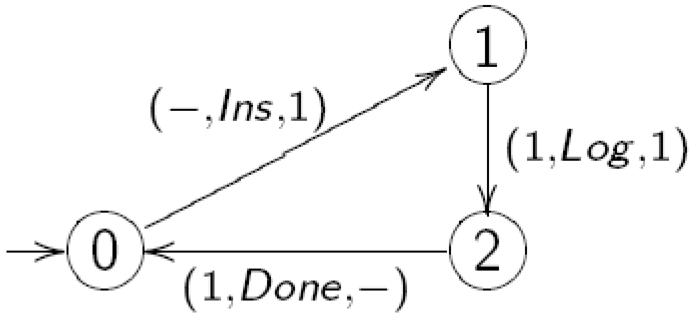
Seminar of ParaDiSe

10 April 2006

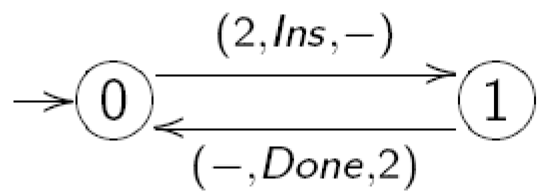
Content

- Examples
- Equivalences
  - Definition
  - Properties
- Automatic verification

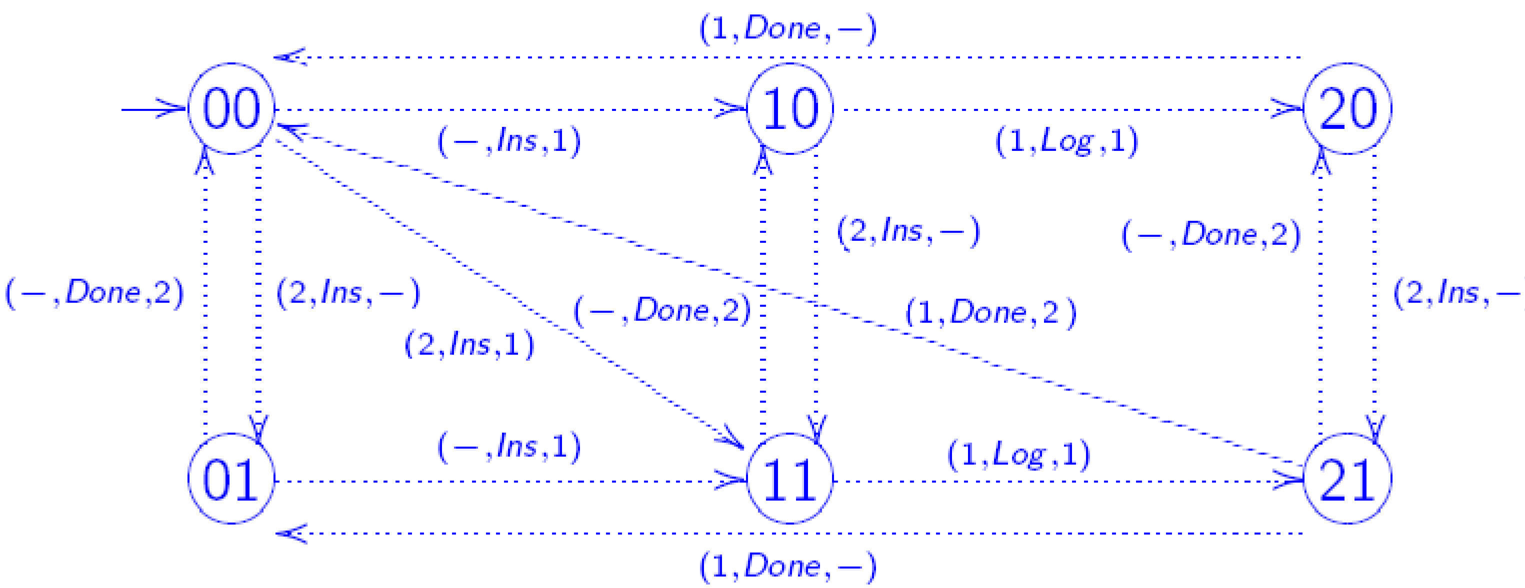
Examples



Hierarchy: (1)

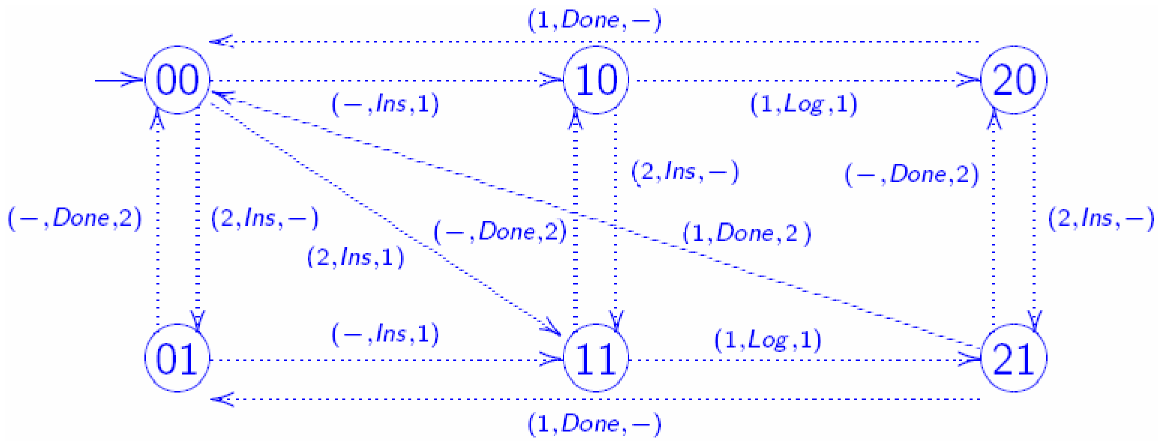


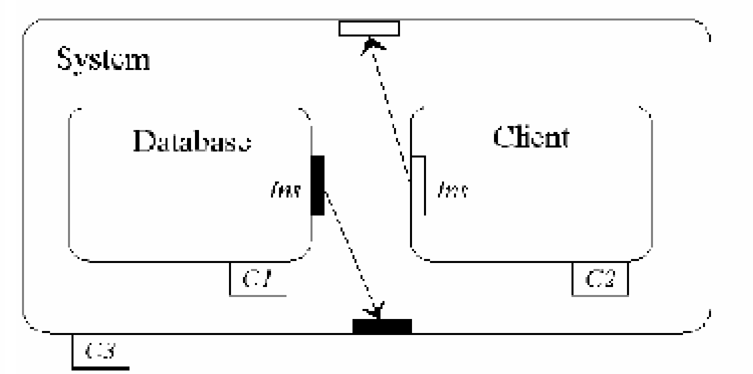
Hierarchy: (2)



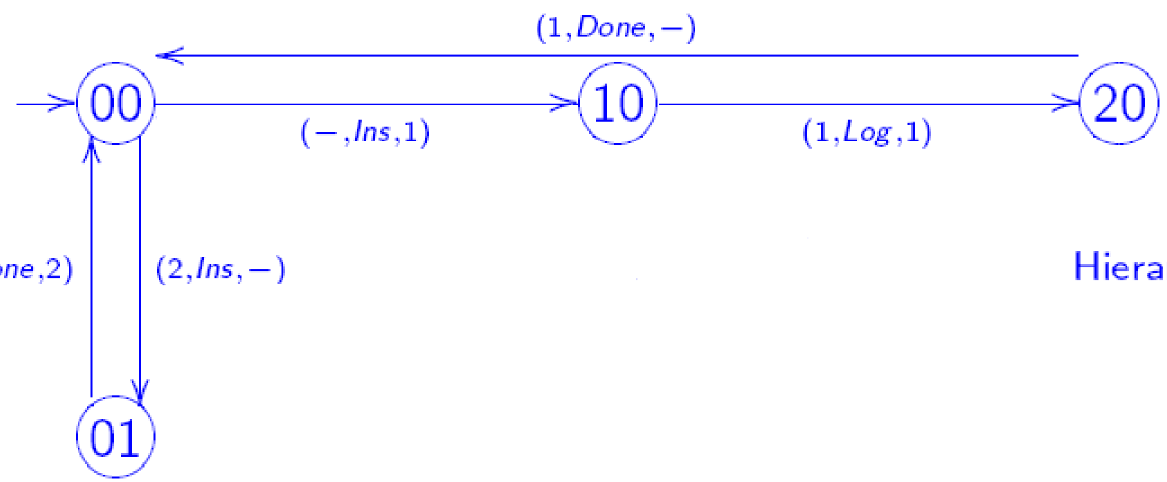
In figures, states ij stand for (i, j)

Examples



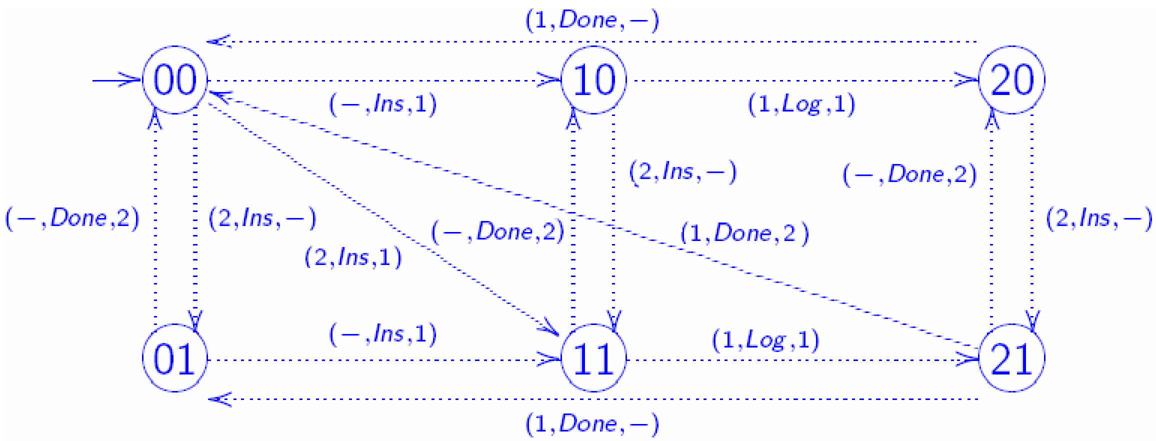


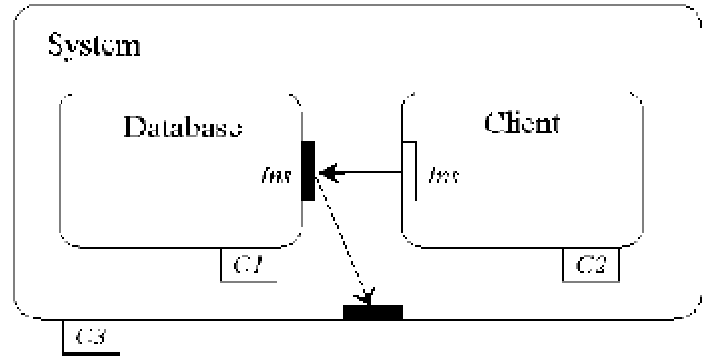
C_1 :



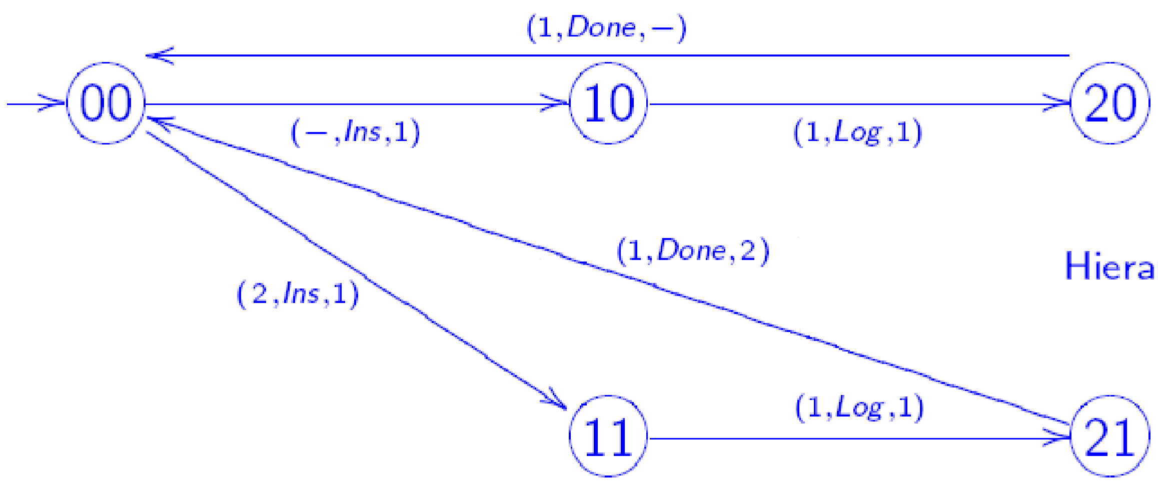
Hierarchy: ((1),(2))

Examples



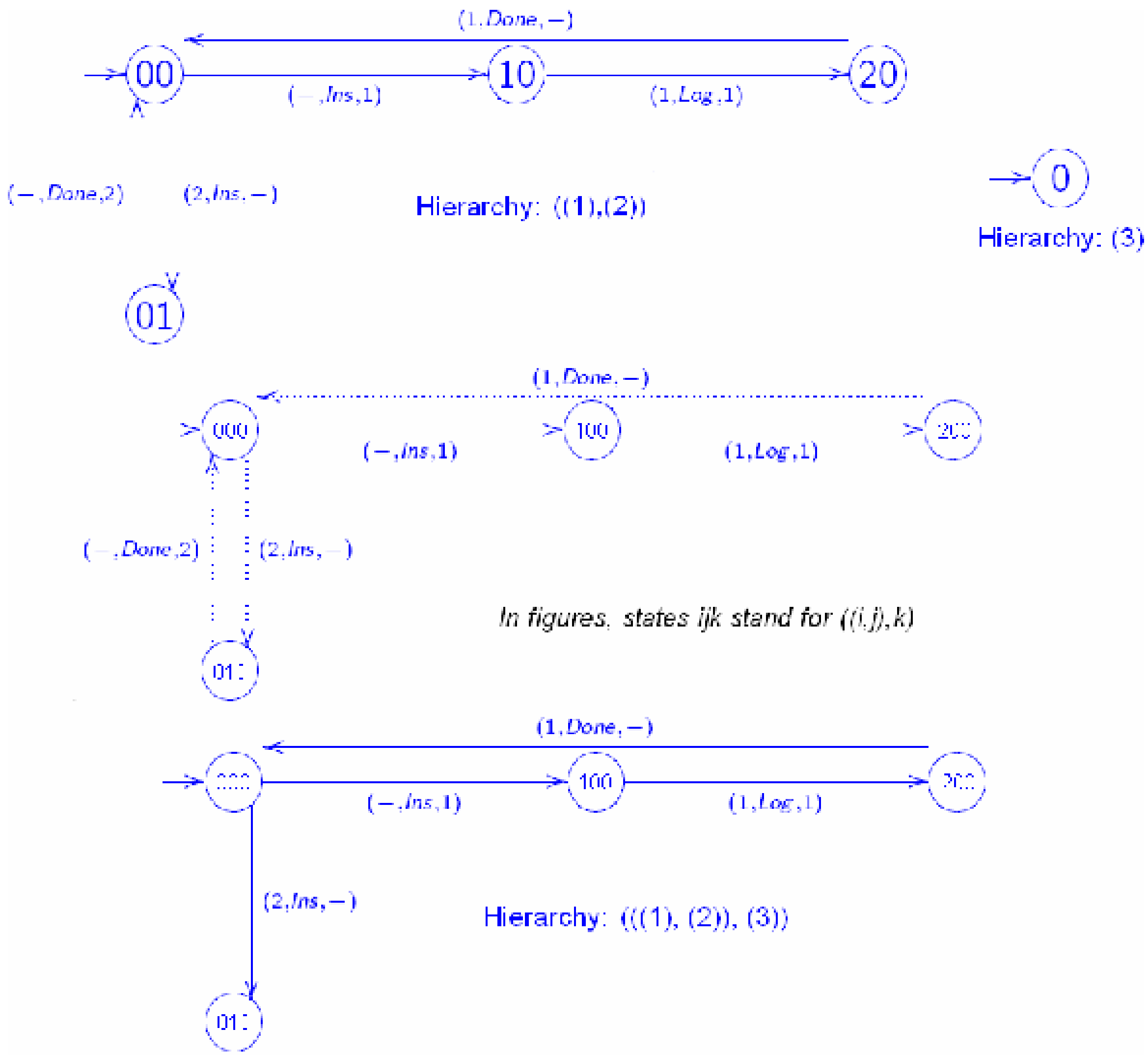


C_2 :



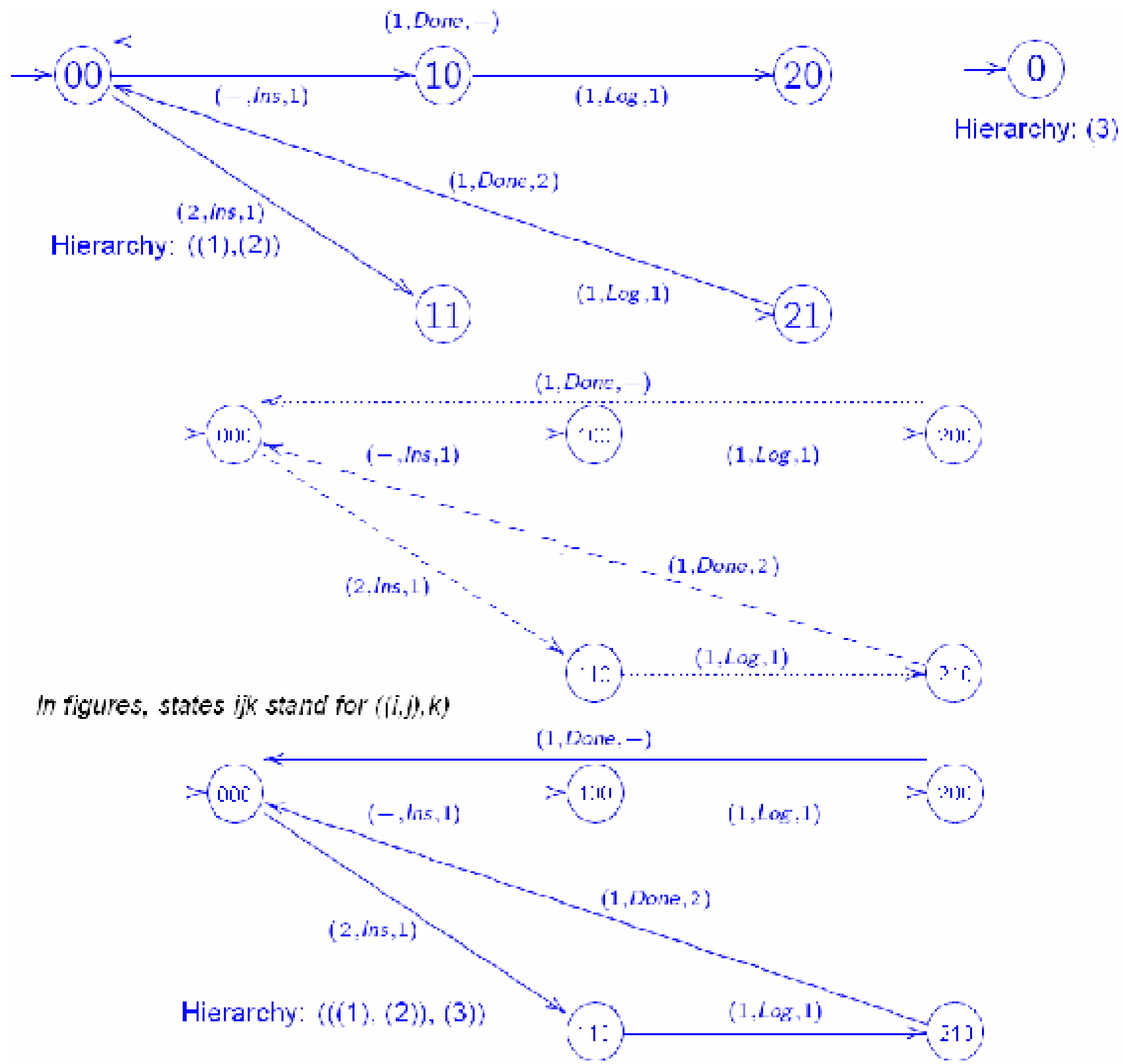
Hierarchy: ((1),(2))

Examples – composition



C´₁ :

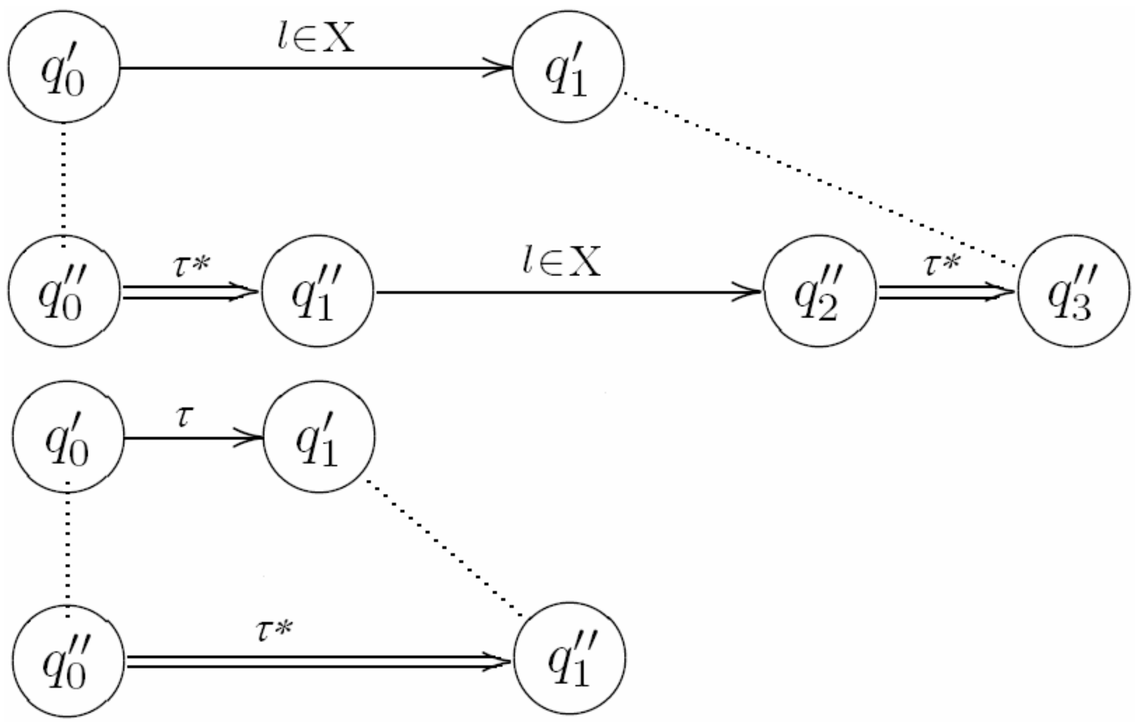
Examples – composition



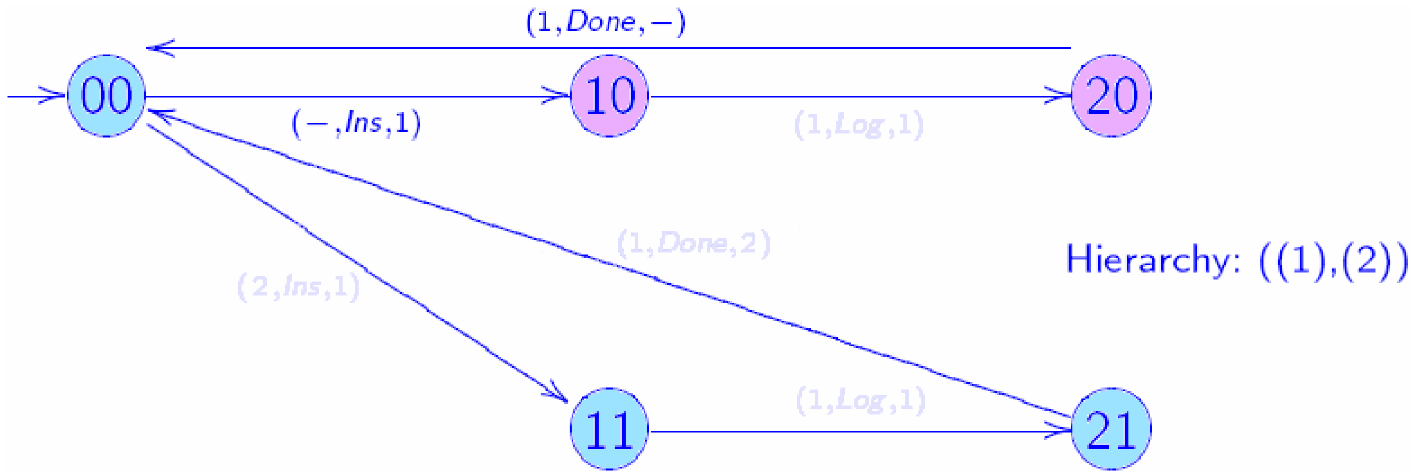
C'_2 :

Equivalences of CI automata

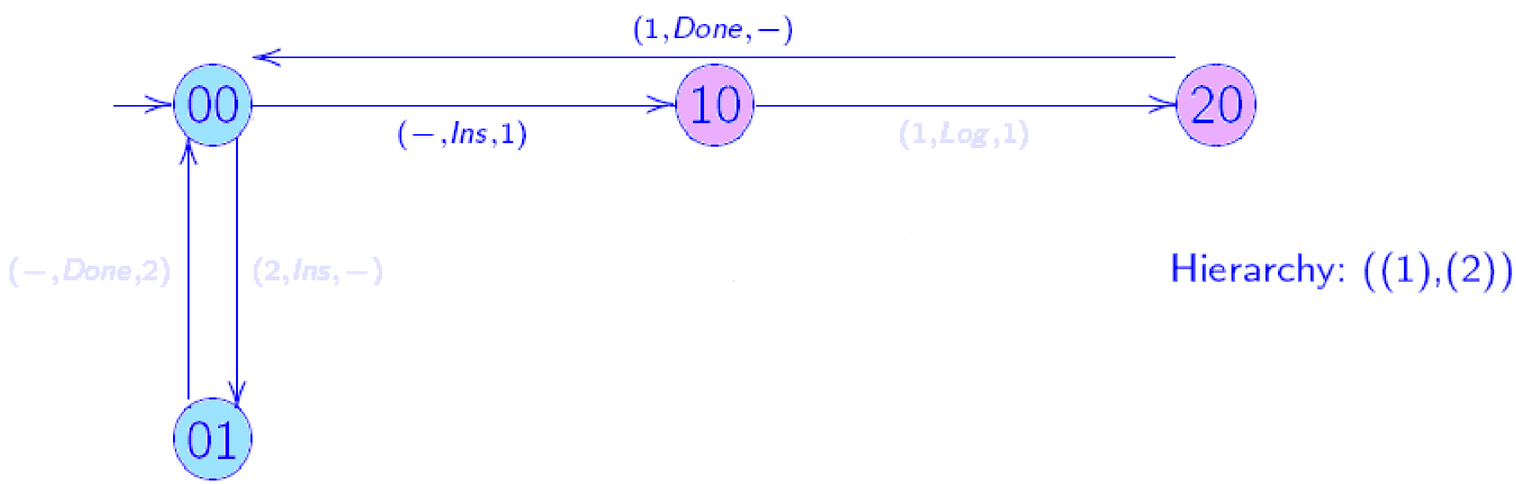
- For each set of labels X exists equivalence $\equiv_X$
- Similar to weak bisimulation of labelled transition systems with silent moves
  - transitions over labels which are not in X – **silent**,
  - transitions over labels which are in X – **observable transitions**.



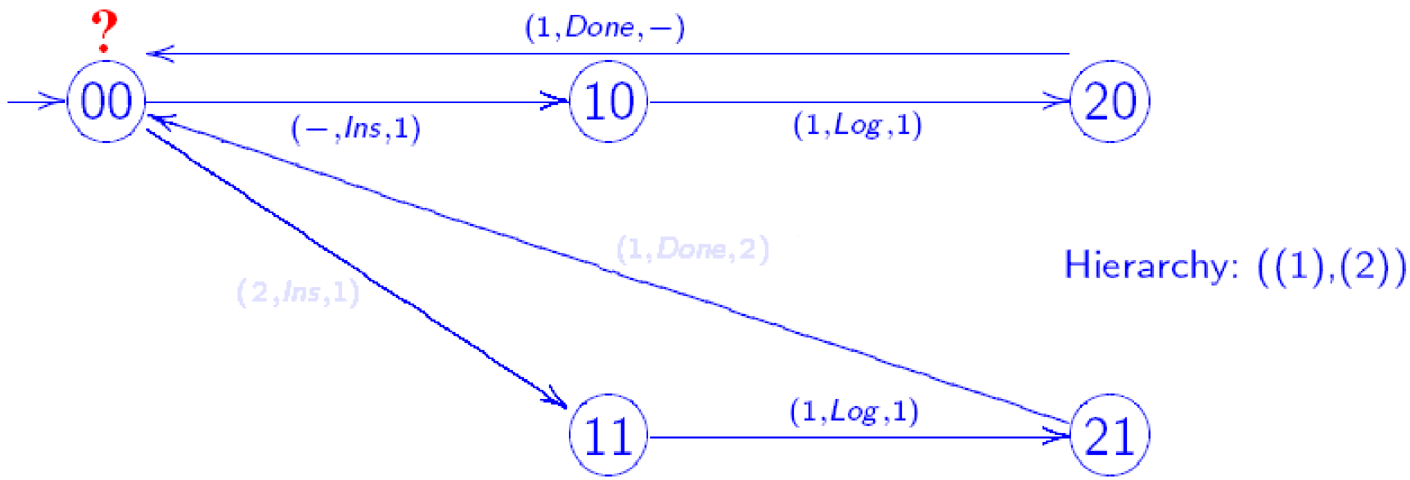
Equivalence $\equiv_X$ - example



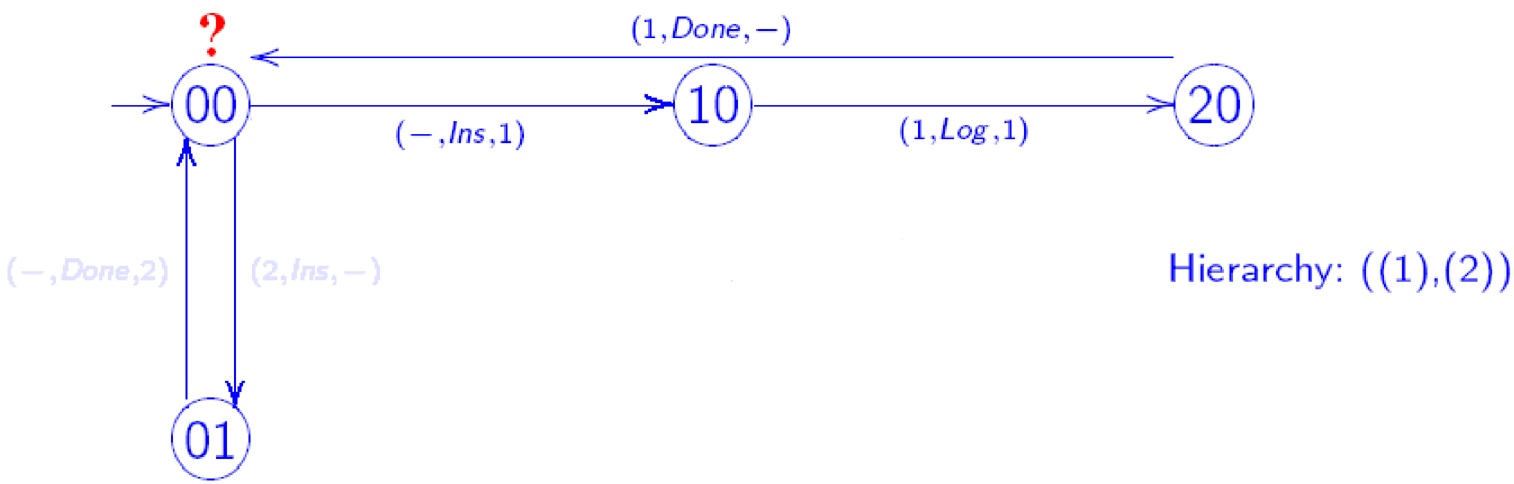
X = { (-, Ins, 1), (1, Done, -)}



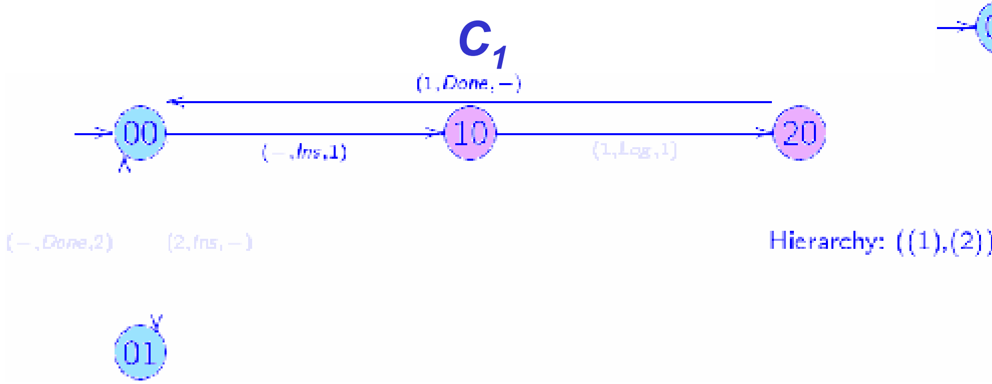
Equivalence $\equiv_x$ - example

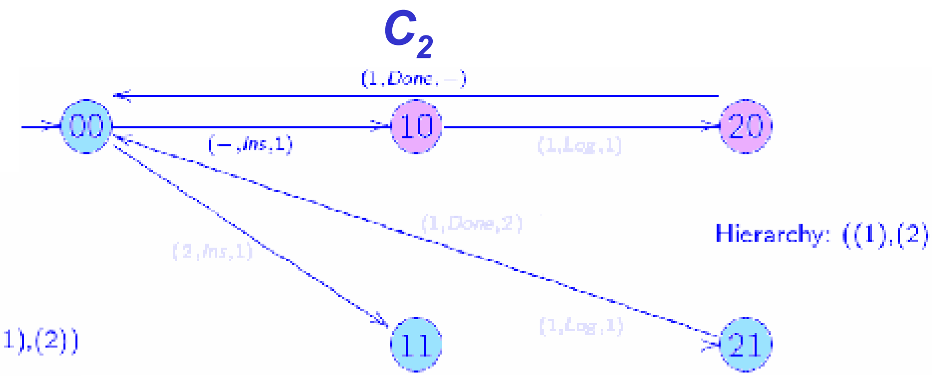


X = {(-, Ins, 1), (1, Log, 1), (1, Done, -)}



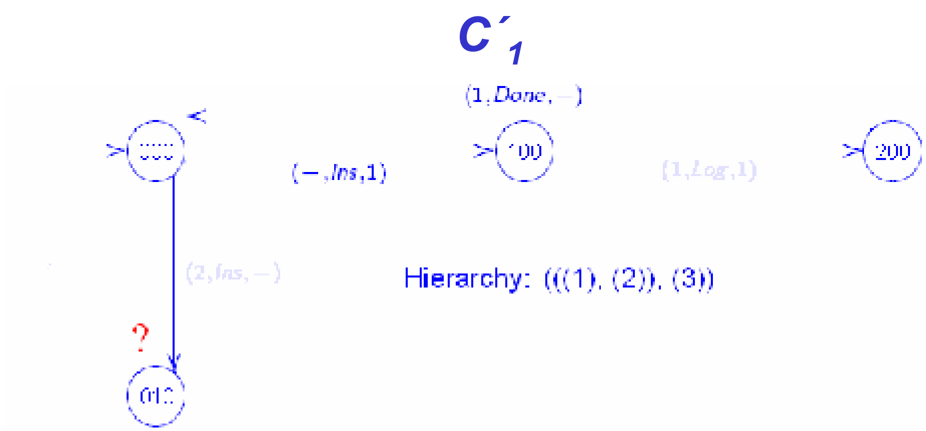
Example

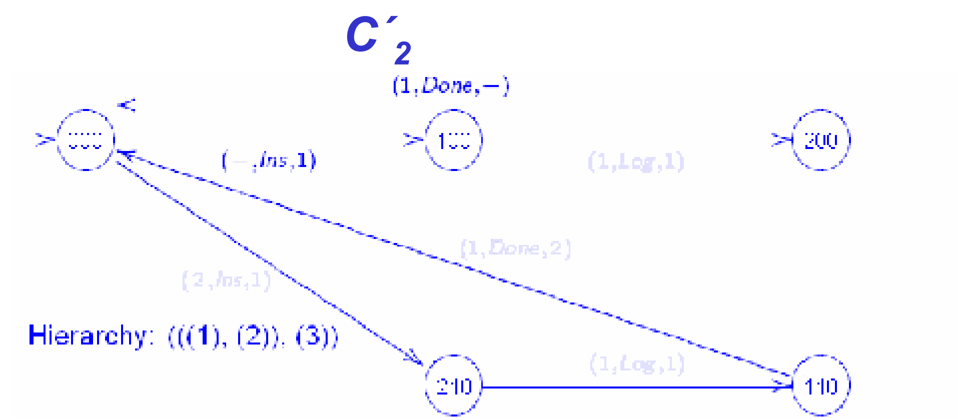




- Automata $C_1 \equiv_x C_2$, where X = { (-, Ins, 1), (1, Done, -)}

- Automata C_1' *and* C_2' *are automata* C_1, C_2 without transitions over label (-,Done,2)





- Automata $C'_1 \not\equiv_x C'_2$

Properties

- $\equiv_X$ - equivalence (X set of labels)
- C, C_1, C_2 - CI automata
- $\otimes_L$ - composition according to chosen labels

For which triples $\equiv_X$, C_1, C_2 *such that* $C_1 \equiv_X C_2$ *it is satisfied:*

$\forall$ C, $\otimes_L$: $\quad C_1 \otimes_L C \equiv_X C_2 \otimes_L C$

Necessary and sufficient condition: $C_{X'} \equiv_Y C_{X''}$

Properties

- $\equiv_X$ - equivalence (X set of labels)
- C, C_1, C_2 - CI automata
- $\otimes_L$ - composition according to chosen labels

For which 4-tuples $\equiv_X$, C_1, C_2, $\otimes_L$ *such that* $C_1 \equiv_X C_2$ *it is satisfied:*

$\forall$ C: $\quad C_1 \otimes_L C \equiv_X C_2 \otimes_L C$

Necessary and sufficient condition: $C_{X'} \equiv_Y C_{X''}$

Properties

- $\equiv_X$ - equivalence (X set of labels)
- C, C_1, C_2 - CI automata
- $\otimes_L$ - composition according to chosen labels

For which 5-tuples: $\equiv_X$, $C_1, C_2, C, \otimes_L$ *such that* $C_1 \equiv_X C_2$ *it is satisfied:*

$$C_1 \otimes_L C \equiv_X C_2 \otimes_L C$$

Properties

Let be C_1, C_2 CI automata and X set such that

- *X contains all reachable labels in C_1,*
- *X contains all reachable labels in C_2,*
- *$C_1 \equiv_X C_2$,*

then C_1 satisfy LTL formula φ iff C_2 satisfy formula φ.

Automatic verification of LTL properties

- **CI automaton** $\rightarrow$ process in **DiVinE specification language**
- **Formula** in LTL $\rightarrow$ process in **DiVinE specification language**

- Transformation should be effective (with the respect to number of states of transformed automata)
- The second transformation depends on the first transformation

Verification

```
Int pinsert=1, pINSERT=1, pdone=0, pDONE=0, pLog=0;
Channel  insert, INSERT, done, DONE, Log;

process System
{
state q00, q01, q02, q10;
init q00;
trans
   q00 -> q01 {sync insert; pINSERT=0, pinsert=0, pLog=1;},
   q01 -> q02 {sync Log; pLog=0, pdone=1;},
   q02 -> q00 {sync done; pdone = 0, pINSERT=1, pinsert=1;},
   q00 -> q10 {sync INSERT; pINSERT=0, pinsert=0, pDONE=1;},
   q10 -> q00 {sync DONE; pDONE = 0, pINSERT=1, pinsert=1;};
}
```