Orthogonal Finite-State Representations

Bernard Boigelot
Xavier Hainaut
Sébastien Jodogne

Université de Liège

Finite-State Representations

Principles :

- The elements of a domain D are *encoded* by finite words over some finite alphabet Σ;
- The encoding relation E needs to be complete and injective;
- A set S is *representable* if $L = E(S)$ forms a regular language. Any finite-state automaton accepting L is a representation of S.

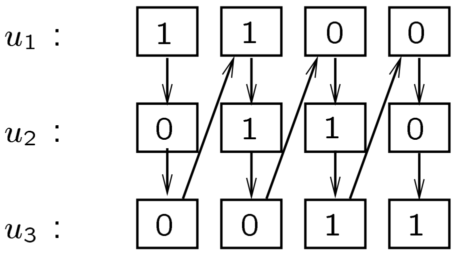
Notes :

- The generalization to infinite words is direct if the representations are limited to *weak* Büchi automata.
- Simple encoding relations are known for some useful domains : integer and real numbers, unbounded FIFO channels.

Representing Multidimensional Sets

Let $D = D_1 \times D_2 \times \cdots \times D_n$. An encoding relation for D can be obtained by combining relations suited for the D_i.

The *serial* combination method consists of reading one symbol in each component's encoding, repeatedly and in a fixed order :



Notes :

- This method can be used when the component encodings must be read at the same rate;

- The combined relation is a valid encoding relation if and only if for every tuple $(d_1, \ldots, d_n) \in D$, there exist same-length encodings of all the d_i.

Advantages :

- Expressiveness is well suited for intended applications;
- Manipulation algorithms are simple;
- Representations have a canonical form.

Main drawback : In some specific cases, simple sets may have an unnecessarily large representation.

In particular, because of the interleaving of the encodings, the representation of a Cartesian product $S_1 \times S_2 \times \cdots \times S_p$ of simple sets S_i can be exponentially large in p.

Idea : This problem could be avoided by making the terms of the product recognized by separate finite-state machines.

Set Orthogonality

Let $S \subseteq D = D_1 \times D_2 \times \cdots \times D_n$.

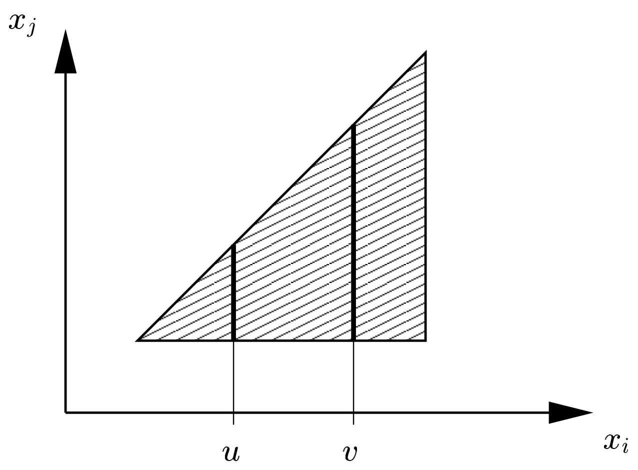
Definition : *The variables x_i and x_j are* correlated *in S if there exist*

- $u, v \in D_i$, *and*
- $d_k \in D_k$ *for each $k \notin \{i, j\}$*

such that the sets

$$\begin{array}{rl} & \{d_j \in D_j \mid (d_1, \ldots, d_{i-1}, u, d_{i+1}, \ldots, d_n) \in S)\} \\ \textit{and} & \{d_j \in D_j \mid (d_1, \ldots, d_{i-1}, v, d_{i+1} \ldots, d_n) \in S)\} \end{array}$$

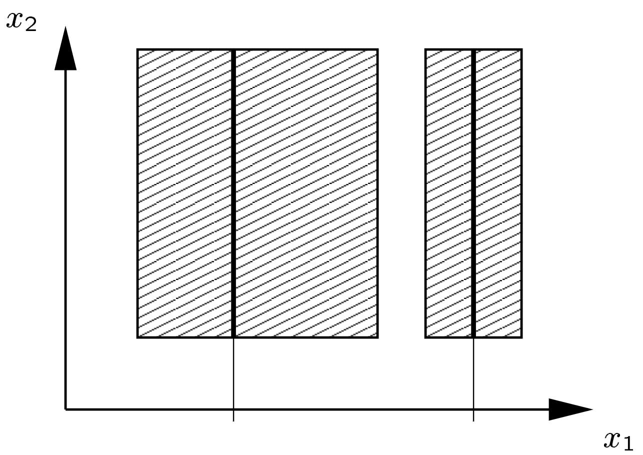
are nonempty and different from each other.



Let $S \subseteq D = D_1 \times D_2 \times \cdots \times D_n$, and P be an ordered partition of $\{1, 2, \ldots, n\}$.

Definition : *The set S is* orthogonal *w.r.t. P if for all distinct $P_l, P_m \in P$ and for all $i \in P_l$ and $j \in P_m$, the variables x_i and x_j are not correlated in S.*

Example : Set orthogonal w.r.t. $\{\{1\}, \{2\}\}$:



Orthogonal Sets : Properties

- A set S orthogonal w.r.t. $P = \{P_1, P_2, \ldots, P_p\}$ can be expressed as a generalized ordered Cartesian product

  $$S = \times_P (S_1, S_2, \ldots, S_p)$$

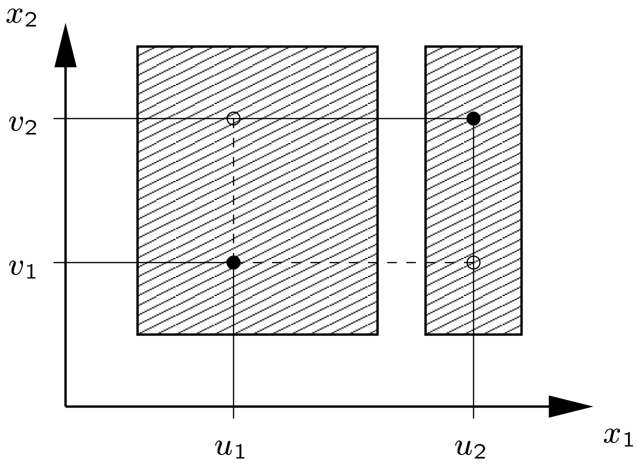
  of sets of smaller dimension;

- For each orthogonal set, there exists a unique finest partition w.r.t. which it is orthogonal;

- If x_i, x_j are not correlated in S, then for each $u_1, u_2 \in D_i$, $v_1, v_2 \in D_j$ and assignment of values to the other variables,

$$\begin{array}{rl} & (\ldots, u_1, \ldots, v_1, \ldots) \in S \\ \text{and} & (\ldots, u_2, \ldots, v_2, \ldots) \in S \\ \text{implies} & (\ldots, u_1, \ldots, v_2, \ldots) \in S. \end{array}$$

Illustration :



Language Orthogonality

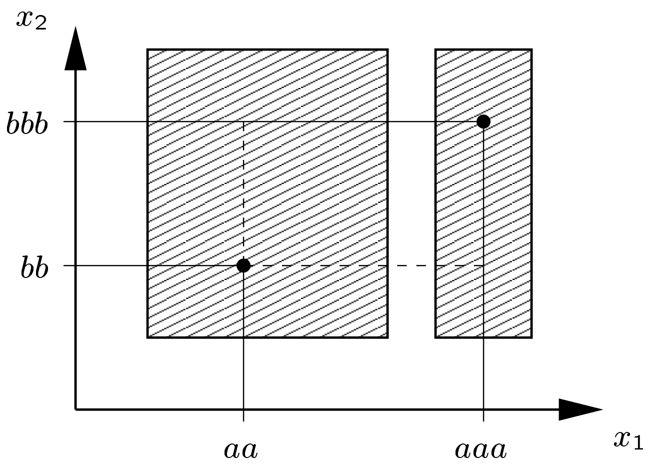
The notions of variable correlation and set orthogonality can be extended to languages accepted by finite-state representations.

Definition : *A language L is orthogonal w.r.t. $P = \{P_1, P_2, \ldots, P_p\}$ if it can be expressed as the serial combination*

$$L = \times_P (L_1, L_2, \ldots, L_p)$$

of languages of smaller dimension.

Problem : The last property of Slide 8 cannot be directly generalized to languages :



Solution : Use only encoding relations for which two arbitrary encodings can always be combined together.

Examples :

- fixed-length encodings

  $\hookrightarrow$ positional number system for bounded integers ;

- infinite words

  $\hookrightarrow$ bounded real numbers.

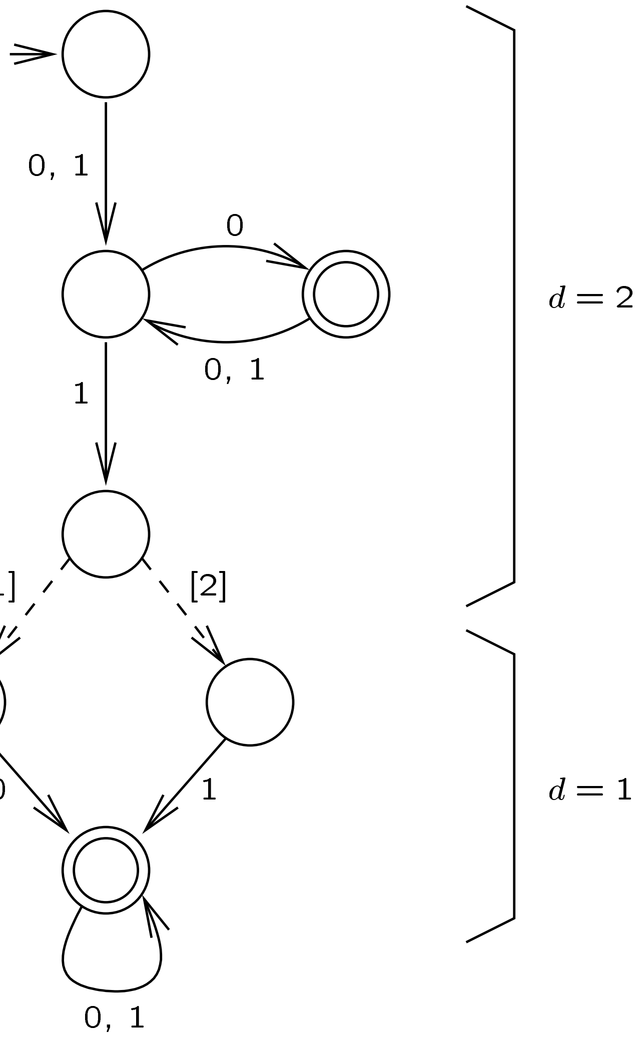
Orthogonal Finite-State Machines

Principles :

- An *Orthogonal Finite-State Machine (OFSM)* is a finite-state machine extended with *decomposition states*;
- Each state s of an OFSM has a *dimension* $d(s)$;
- A decomposition state is associated with an ordered partition $P(s) = \{P(s)_1, \ldots, P(s)_{p(s)}\}$ of $\{1, 2, \ldots, d(s)\}$.

  For each $P(s)_i$, there is a *decomposition* outgoing from s, and ending in a state s' such that $d(s') = |P(s)_i|$;
- When one reaches a decomposition state, the operation of the OFSM proceeds concurrently in all its successors, each of them dealing with its proper subset of the domain variables.

Example :



Properties :

- OFSMs are exactly as expressive as finite-state automata;
- OFSMs admit a minimal form, unique up to isomorphism.

Minimization Procedure

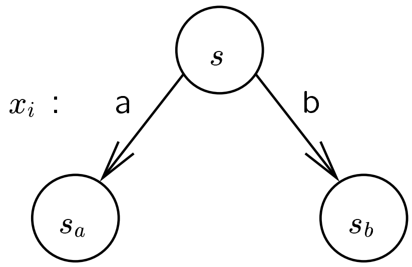
Goal : Developing an algorithm for turning an arbitrary OFSM into its minimal form.

Approach :

- Assume the OFSM deterministic and complete;
- Proceed layer by layer, lower dimension first;
- At each step, try to detect correlations in the languages accepted from each state;
- Once the language accepted from a state s is known to be orthogonal, turn s into a decomposition state, and merge its successors with the appropriate lower layers;
- Finally, minimize separately each layer using the classical Hopcroft's algorithm.

Detecting Correlations

Correlations are detected thanks to the following rule :



If

- $L(s_a) \neq \emptyset$,
- $L(s_b) \neq \emptyset$, and
- $L(s_a)|_{x_j} \neq L(s_b)|_{x_j}$,

then x_i and x_j are correlated in $L(s)$.

Proof :

- If the hypotheses are satisfied, then there exists $w'_a \in L(s_a)|_{x_j}$ such that $w'_a \notin L(s_b)|_{x_j}$ (or symmetrical);
- Let w_a be such that $(w_a, w'_a) \in L(s_a)|_{\{x_i,x_j\}}$. Let w_b, w'_b be such that $(w_b, w'_b) \in L(s_b)|_{\{x_i,x_j\}}$;

- $w'_a \notin L(s_b)|_{x_j}$ implies $(w_b, w'_a) \notin L(s_b)|_{\{x_i,x_j\}}$;
- So we have :

$$
\begin{aligned}
& (aw_a, w'_a) \in L(s)|_{\{x_i,x_j\}}, \\
& (bw_b, w'_b) \in L(s)|_{\{x_i,x_j\}}, \\
\text{but} \quad & (bw_b, w'_a) \notin L(s)|_{\{x_i,x_j\}};
\end{aligned}
$$

- Thus, x_i and x_j are correlated in $L(s)$.

Other properties :

- The previous rule is sufficient for detecting all correlations;
- Propagation rule : If $s \to^* s'$ and x_i, x_j are correlated in $L(s')$, then they are also correlated in $L(s)$.

Partitioning Automata States

Problem : The correlation detection rule needs to be able to distinguish accepted languages w.r.t. groups of correlated variables.

Strategy :

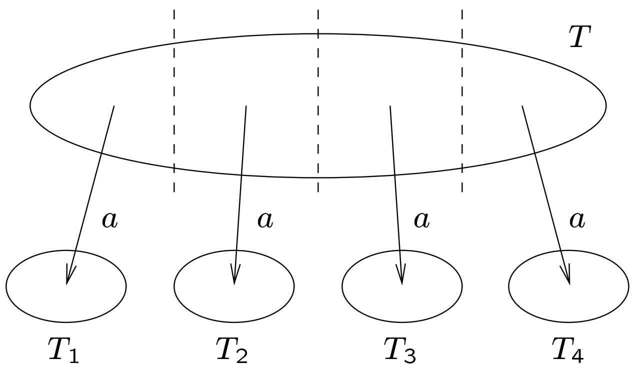
- Maintain in each state a partition of the variables according to the correlations detected so far;
- For each element P_i of such partitions, keep a partition of the states. Partition the states according to their accepted language, considering only the variables in P_i;
- After each batch of correlation detections (including applications of the propagation rule), reset state partitions corresponding to new groups of correlated variables.

Partial Minimization

Problem : Partitioning automata states w.r.t. their accepted language, considering only some subset V of the domain variables.

Solution : Apply the classical Hopcroft's algorithm :

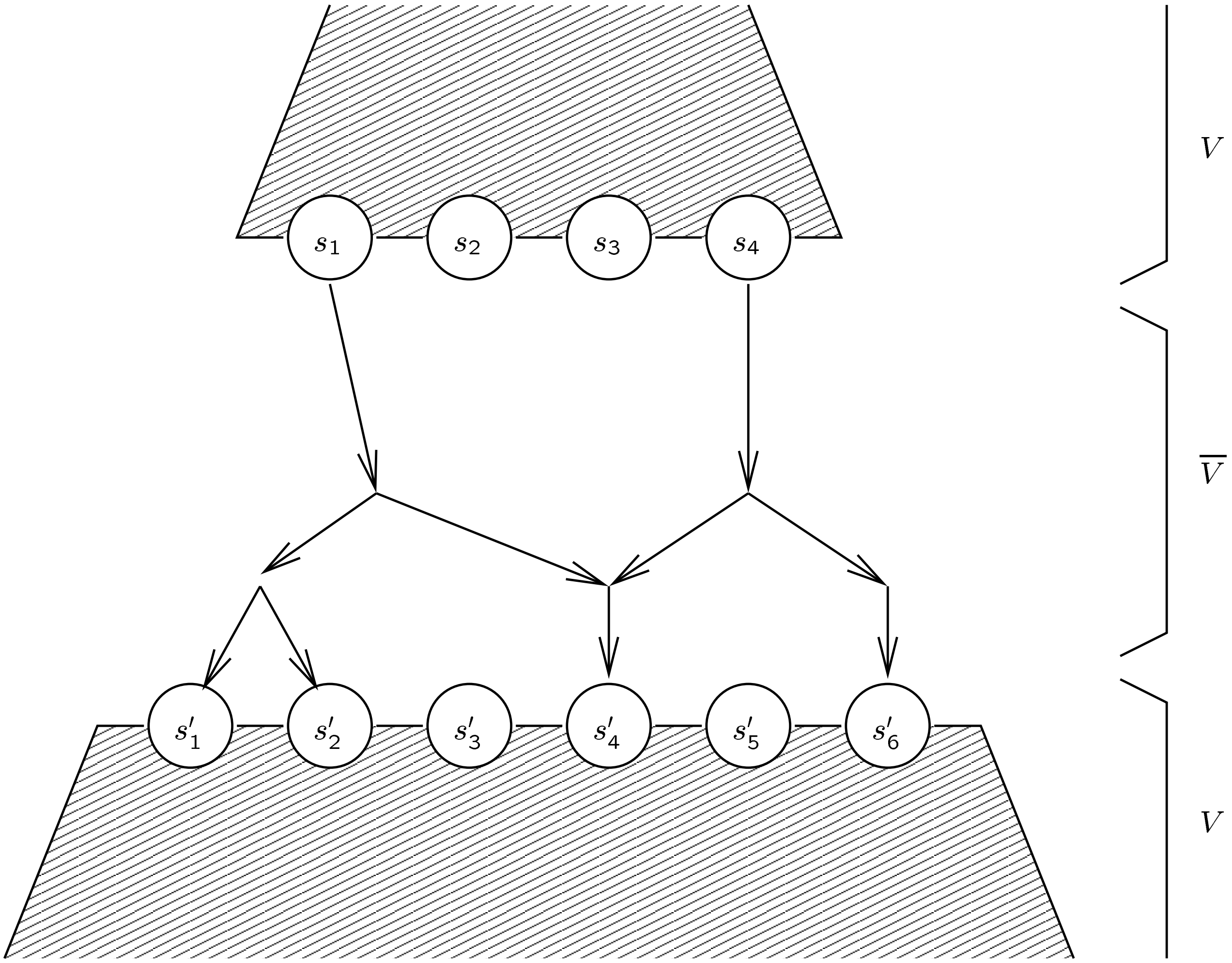
- Start with a partition distinguishing only accepting and non-accepting states;
- When the successors of some part T w.r.t. some symbol a (corresponding to a variable in V) belong to different partition elements, split T accordingly;



- Handle the transitions reading variables that are not in V as ε-transitions.

Property : In the absence of correlations between V and other variables, ε-transitions do not produce non-determinism.

Illustration :



If the variables in V are not correlated with those in $\overline{V}$, then the connected states accept either the same language w.r.t. the variables in V, or the empty language.

Other Operations

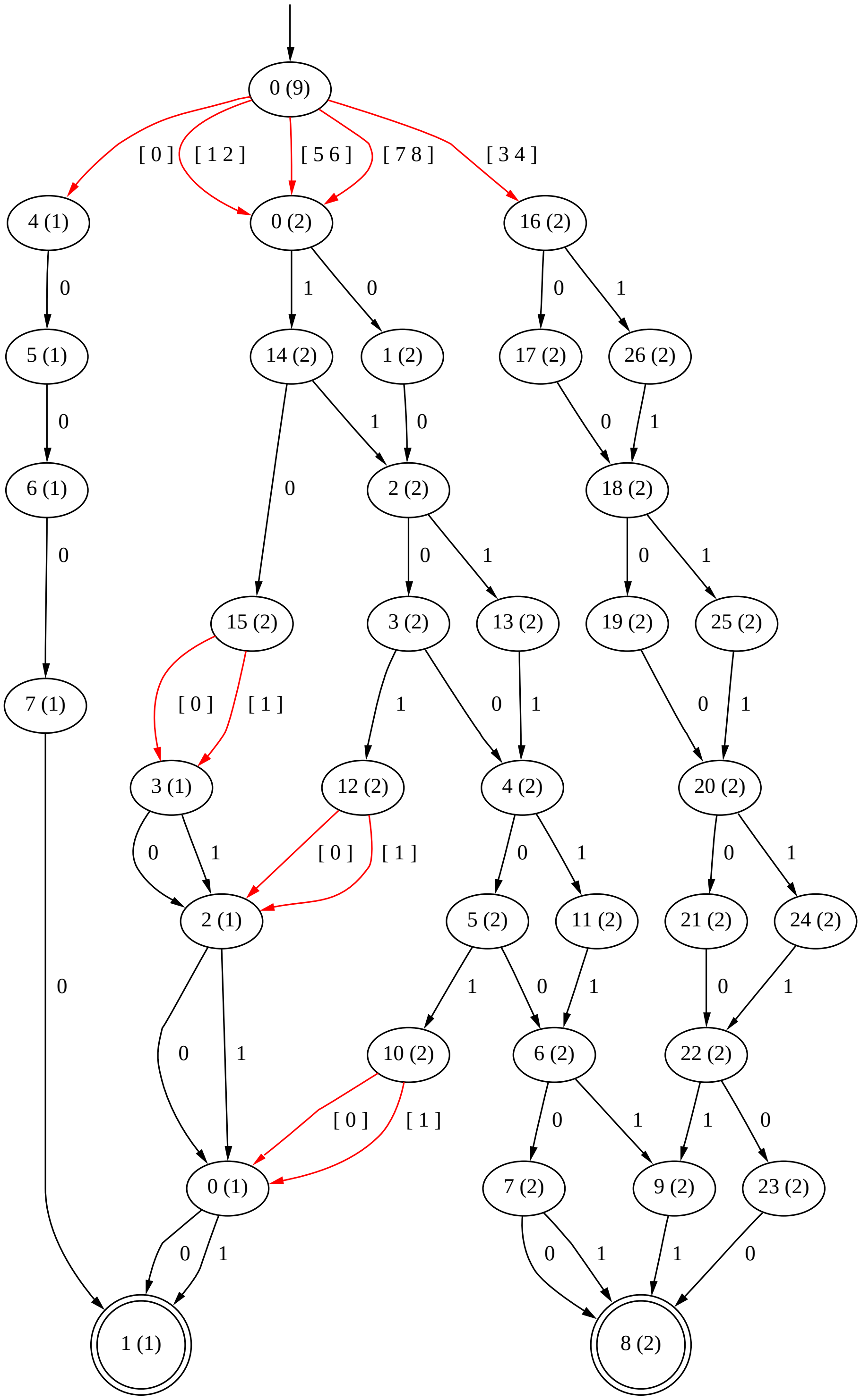
- *Boolean combinations* : The product construction used with classical finite-state machines can be generalized to OFSMs. Instead of pairs of states, one has pairs of either states or conjunctive combinations of states. Orthogonality subsists only if it is consistent in both operands.

  Each operations is followed by a minimization step.

- *Projection* : Amounts to replacing by ε the symbols corresponding to the variables that are projected out. The necessary determinization step is done by the classical subset construction, which generalizes to OFSMs in a way similar to the product construction.

Experimental results

$S' = \{(x_0, \ldots, x_8) \in \mathbb{Z}^9 \mid x_0 = 0 \wedge x_1 \leq x_2 \wedge x_3 = x_4 \wedge x_5 \leq x_6 \wedge x_7 \leq x_8\}$ over 4 bits :



S' over 64 bits : 5882 states $\Rightarrow$ 585 states

Conclusions

- This approach targets one specific source of inefficiency in finite-state representations.
- There are still a lot of problems to solve :
  - How can more general encoding relations be dealt with ?
  - Is there more efficient way of minimizing automata incrementally ?
  - Can the resetting of state partitions be avoided ?