Statecharts, Transition Structures and Transformations*

Adriano Peron

Dipartimento di Matematica ed Informatica — Università di Udine
Via Zanon 6 — I-33100 Udine — Italy
E_mail: peron@dimi.uniud.it

Abstract. Statecharts are state-transition machines endowed with hierarchy on states and parallelism on transitions. It is shown that a statechart is described by a pair of relations over transitions (a transition structure), the former describing causality and the other describing a notion of asymmetric independence. A statechart can be effectively constructed from its transition structure. Transition structures corresponding to a subclass of Statecharts are characterized. Natural notions of morphisms among transition structures allow to define classes of statechart transformations which preserve behaviour.

1 Introduction

Statecharts (see [2]) is a graphical specification formalism which enriches state-transition diagrams with a hierarchical structure (i.e., tree-like) on states and with graphical conventions for explicitly representing parallelism and communication among parallel components. In recent years, a great effort has been devoted to semantics of Statecharts (e.g., [3, 4, 1, 7, 5, 8]). A minor attention has been paied to the investigation of equivalence notions and to the related issues of behaviour preserving transformations and "a priori" correct development of specifications. This paper gives a contribution in this sense.

The classical representation of Statecharts emphasizes the structure of the specification (it is a graphical structured formalism), but, if behaviour properties are the main concern, structure might be an useless complication. For this reason we provide a basic representation of a statechart in terms of its set of transitions, a relation of causality and a relation of asymmetric independence on transitions (a *transition structure*). We show that a transition structure is sufficient to describe a statechart and that states, hierarchy on states and representation of parallelism can be derived from the transition structure. In particular, we provide a recursive algorithm which associates, with the transition structure of a statechart Z, a statechart Z' having the same transition structure. In general, the statechart Z' is not isomorphic to Z, since it has minimal

* Partially supported by project ESPRIT Basic Research 8130 LOMAPS and by project CNR 94.01874.CT7 Specifica di Alto Livello e Verifica Formale di Sistemi Digitali.

structure on states. The proposed algorithm is useful for removing redundances in the graphical representation of a statechart. We also characterize transition structures corresponding to "well structured statecharts" -a widely adopted restriction of statecharts (e.g., see [8])- and we call them *good structures*. Good structures are dual with respect to well structured statecharts and it is possible to shift effortless from one formalism to the other.

Since transition structures are simple objects which emphasize behavioural properties of a statecharts, they are a proper representation for studying transformations. We investigate statechart transformations by investigating morphisms of transition structures which preserve behaviour. Implicitly, each notion of morphism defines a class of transformation of statecharts, in the sense that if there is a morphism from the transition structure of a statechart Z to that of Z', then Z can be transformed into Z'. We provide two different notions of morphism which support two meaningful classes of structural transformations: transformations which allow (in some cases) to reshape a statechart into a well structured statechart and a form of top-down refinement.

For the sake of simplicity, we consider statecharts in a version where many communication features are simplified (e.g., we do not consider shared variables and negative events). However, since our main concern is not communication, we stress that the results of this paper does not depend on restricting communication. The semantics we enforce slightly modifies the original one (see [3]).

In section 2 Statecharts, transition structures and semantics are defined. In section 3 the technique for reconstructing a statechart from its transition structure is described and it is proved correct. In section 4 transition structures corresponding to well structured statecharts are characterized. In section 5 notions of behaviour preserving morphisms of transition structures are defined.

2 Statecharts

The graphical convention is that states are depicted as boxes, and the box of a substate of a state b is drawn inside the area of the box of b. States are either of type OR, called *or-states*, or of type AND, called *and-states*. And-states are depicted as boxes whose area is partitioned by dashed lines. Each element of the partition is the root state of a statechart describing a *parallel component* of the and-state. When an or-state is entered, also one and only one of its immediate substates is. When an and-state is entered, all of its immediate substates are. Each non basic or-state has a privileged immediate substate: the *default substate* (graphically, it is the target of a dangling arc). For instance, in Fig.1, L is an and-state whose parallel components are K and J. State B is the default substate of A. Transitions are represented graphically by arrows and are labelled by a pair of sets of events, one representing a "triggering", the other an "action". Events are interpreted as pure signals communicated by the environment. A transition is *enabled* if the set of events of the triggering are currently communicated. When a transition t is performed, the set of events in the action of t are instantaneously communicated, so augmenting the set of events offered by the environment. As

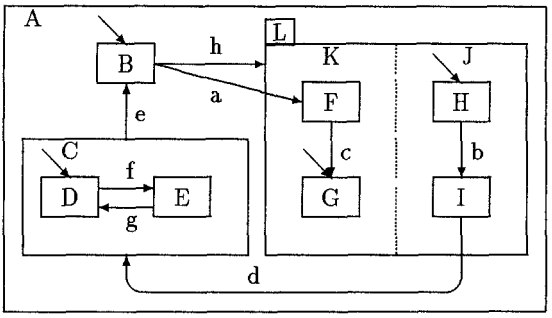

Fig. 1. A statechart.

an example, if transition a in Fig.1 is labelled by a pair $(\{\alpha, \beta\}, \{\gamma\})$, then a is enabled when events α and β are communicated. When transition a is performed, event γ is instantaneously communicated.

Definition 1. A *statechart* Z is a 8-tuple $(B, \rho, \phi, T, in, out, P, \chi, \delta)$, where:

- B is the nonempty finite set of *states*;
- $\rho : B \rightarrow 2^B$ is the *hierarchy function* which gives, for each state, the set of its immediate substates; for $s \in B$, $\rho^\star(s)$ denotes the least set $S \subseteq B$ such that $s \in S$ and $\rho(s') \in S$ for all $s' \in S$, and $\rho^+(s)$ denotes the set $\rho^\star(s) - \{s\}$; ρ describes a tree-like structure, namely:
  1. there exists a unique $s \in B$ s.t. $\rho^\star(s) = B$, denoted as $root_Z$;
  2. $s \notin \rho^+(s)$, for all $s \in B$;
  3. if $s' \notin \rho^+(s)$, $s \notin \rho^+(s')$, then $\rho^\star(s) \cap \rho^\star(s') \neq \emptyset$ implies $s = s'$, for all $s, s' \in B$.

  Given two states b and b', *the lowest common ancestor* of b and b', denoted as $\mathcal{L}(b, b')$, is the state $\bar{b} \in B$ such that $\rho^\star(\bar{b}) \supseteq \{b, b'\}$ and, $\bar{b} \in \rho^\star(b'')$, for each b'' with $\rho^\star(b'') \supseteq \{b, b'\}$;
- $\phi : B \rightarrow \{AND, OR\}$ is the *state type function*;
- T is a finite set of *transitions*;
- $in : T \rightarrow B$ is the function which gives the *target state* of a transition;
- $out : T \rightarrow B$ is the function which gives the *source state* of a transition; functions in and out are such that $\phi(\mathcal{L}(t)) = OR$,[2] $in(t) \notin \rho^\star(out(t))$ and $out(t) \notin \rho^\star(in(t))$, for all $t \in T$;
- P is the finite set of *events*;
- $\chi : T \rightarrow 2^P \times 2^P$ is the *transition labelling function*; $Ev(t)$ and $Act(t)$ denote the first and the second component of $\chi(t)$, respectively;
- $\delta : B \rightarrow B$ is the (partial) *default function* giving, for a non basic or-state, the default substate, i.e., $\delta(b) \in \rho(b)$, for all $\phi(b) = OR$ s.t. $\rho(b) \neq \emptyset$.

[2] $\mathcal{L}(t)$ is a short writing for $\mathcal{L}(in(t), out(t))$.

In the following, we shall write B_Z intending the set of states of statechart Z (and analogously for all the other statechart components). We shall write $\mathcal{L}_Z$ to stress that relation $\mathcal{L}$ is over Z and we omit the index when the context is clear (and analogously for all the other relations we shall define).

When a state b is entered a subset of its substates is entered consistently with the requirement that if an and-state (resp.: an or-state) is entered, then all of its (resp.: exactly one) immediate substates are entered. In particular, one of these sets of states is the one induced by the default function (i.e., the chosen immediate substate of an or-state is its default substate), called *default closure of* b. As an example, the default closure of state L is the set $\{L, K, J, G, H\}$.

For $K \subseteq B_Z$, the *default closure of* K, denoted as $\Downarrow_Z K$, is the least superset of K such that

1. $\rho_Z(d) \subseteq \Downarrow_Z K$, for each and-state $d \in \Downarrow_Z K$,
2. $\delta_Z(d) \in \Downarrow_Z K$, for each or-state $d \in \Downarrow_Z K$ such that $\rho_Z(d) \cap K = \emptyset$.

When the target state b of a transition is entered, states in the default closure of b are entered. When an and-component of an and-state is entered, states in the default closure of all the other components are entered. As an example, when transition a is performed the set of entered states is $\{L, K, F, J, H\}$. In general, when a transition t is performed, the set of entered states is given by the default closure of the set of substates of $\mathcal{L}(t)$ which are ancestors of $in(t)$.

On the transition set we define a *sequentiality relation*. A transition t' *follows* t if the source state of t' belongs to the set of states entered by performing t. Actually, we enrich the set of transitions T by a transition *imp* representing the "start up" of the statechart activities. We can assume that *imp* is a transition without source and leading to the root. The set of states entered by *imp* is the downward closure of the root of the statechart.

Definition 2. Let *imp* be a symbol not in T_Z, called *initial transition*, and let T_Z^i denote $T_Z \cup \{imp\}$. *Sequentiality relation* is $\rightarrow_Z \subseteq T_Z^i \times T_Z$, with

$$t \rightarrow_Z t' \text{ iff } out_Z(t') \in \Downarrow_Z \{b : b \in \rho_Z^+(\mathcal{L}(t)), in_Z(t) \in \rho_Z^*(b)\}, \text{ for } t, t' \in T_Z$$

$$\text{and } imp \rightarrow_Z t \text{ iff } out_Z(t) \in \Downarrow_Z \{root_Z\}.$$

If $t \in T_Z$, then $t^{\rightarrow_Z}$ denotes the set $\{t' \in T_Z : t \rightarrow_Z t'\}$.

With reference to Fig.1, $imp^{\rightarrow} = e^{\rightarrow} = \{h, a\}$, $a^{\rightarrow} = \{c, b\}$, $c^{\rightarrow} = \emptyset$, $d^{\rightarrow} = \{f, e\}$, $b \rightarrow d$, $f \rightarrow g$, $g \rightarrow f$ and $h \rightarrow b$.

On the transition set we define a *concurrency* relation. Two transition are concurrent if both of their source states can be simultaneously entered. If t and t' are concurrent, then they may be enabled simultaneously (it does not imply that they can be performed simultaneously).

Definition 3. *Concurrency relation* is the symmetric relation $\|_Z \subseteq T_Z \times T_Z$, with $t' \|_Z t''$ and $t'' \|_Z t'$ iff one of the two following constraints is satisfied:

1. there is a transition t such that $t \rightarrow_Z t'$, $t \rightarrow_Z t_1 \rightarrow_Z \ldots \rightarrow_Z t_m \rightarrow_Z t''$ and $\mathcal{L}(\mathcal{L}(t_i), out(t')) = out_Z(t')$, for all t_i with $1 \leq i \leq m$;
2. there is a transition t such that $t \rightarrow_Z t_1 \rightarrow_Z \ldots \rightarrow_Z t_m \rightarrow_Z t'$ and $t \rightarrow_Z h_1 \rightarrow_Z \ldots \rightarrow_Z h_n \rightarrow_Z t''$, with t_i and h_j transitions such that $\mathcal{L}_Z(\mathcal{L}_Z(t_i), \mathcal{L}_Z(h_j))$, $\mathcal{L}_Z(\mathcal{L}_Z(t_i), out(t''))$ and $\mathcal{L}_Z(\mathcal{L}_Z(h_i), out(t'))$ are and-states, for all $1 \leq i \leq m$ and $1 \leq j \leq n$.

With ref. to Fig.1, $a \parallel h$, $b \parallel c$, $d \parallel c$, $f \parallel e$ and $g \parallel e$ and the symmetric cases.

A transition t' is *independent* w.r.t a transition t if t and t' are concurrent and the performance of t does not disable the performance of t' (i.e., the performance of t does not cause exiting the source state of t').

Definition 4. *Independence relation* is $\rhd_Z \subseteq T_Z \times T_Z$ with $t \rhd_Z t'$ iff

$$\text{either } \mathcal{L}_Z(\mathcal{L}_Z(t), out_Z(t')) = out_Z(t')$$

$$\text{or } \mathcal{L}_Z(\mathcal{L}_Z(t), out_Z(t')) \text{ is an and-state and } t \parallel_Z t'.$$

With reference to Fig.1, $b \rhd c$, $c \rhd b$, $c \rhd d$, $f \rhd e$, $g \rhd e$, and $d \not\rhd c$, $e \not\rhd f$ and $e \not\rhd g$ (the relation $\rhd$ is not symmetric). If $t \rhd t'$ and not $t' \rhd t$, then t' can be seen as being an *interrupt* for t. If $t \rhd t'$ and $t' \rhd t$ (we write shortly $t \diamond t'$ and we say that t and t' are *parallel*), then t and t' can be performed, if enabled, in any order as well as in parallel. With reference to Fig.1, transition e is an interrupt for f and g, and d is an interrupt for c. Transitions c and b are parallel.

The *transition structure* of Z is the quadruple

$$\mathcal{TS}(Z) \stackrel{def}{=} (T_Z^i, \rightarrow_Z, \rhd_Z, \chi_Z).$$

We shall introduce now a slight variant of the standard semantics of Statecharts (see [3]). At each instant of time (a discrete time domain is enforced, for instance natural numbers $\mathbb{N}$) the environment prompts a statechart with a set of events. So, we assume to have a function $Env : \mathbb{N} \rightarrow 2^P$ describing environment. At a fixed instant of time n, a statechart *configuration* is characterized by the set of states which are currently entered and by the set of events communicated at that time. A statechart reacts by simultaneously performing a number of transitions (i.e., a *microstep*) enabled by events $Env(n)$. The effect is that the set of currently entered states is changed and the set $Env(n)$ is augmented by adding events in the action part of the performed transitions. As a consequence, a larger set of transitions might be enabled at time n (than that enabled by $Env(n)$) and a chain reaction might occur (i.e., a sequence of microsteps). Events can be sensed only at the instant of time they have been communicated (i.e., they are instantaneous), and only sets of parallel transitions can be performed instantaneously. Thus, the sequence of microsteps triggered by $Env(n)$ is finite. In our definition, a configuration consists of a set D of transitions and a function describing the environment. Transitions belonging to D are those having source state in the set of currently entered set of states. Semantics is defined only by exploiting the transition structure of a statechart without any direct reference to the notion of state and hierarchy on states.

Definition 5. A *configuration* is a pair $C = (D, Env)$, where $D \subseteq T_Z$ and $Env : \mathbb{N} \to 2^{P_Z}$ is the *environment function*; C is *initial* iff $D = imp^{\rightarrow z}$.
A transition t is *enabled at time* $\tau \in \mathbb{N}$ in C iff $t \in D$ and $Ev(t) \subseteq Env(\tau)$.
A set $\Psi \subseteq T_Z$ of transitions is a *microstep at time* τ from C iff each transition in Ψ is enabled at time τ in C and $t \diamond t'$, for any $t, t' \in \Psi$ with $t \neq t'$.
For a microstep Ψ at time τ from C, the configuration *reached from* C *by* Ψ is (D', Env'), with:

$$D' = \{t \in D : t' \rhd t, \text{ for all } t' \in \Psi\} \cup \bigcup_{t \in \Psi} t^{\rightarrow}$$

$$Env'(n) = \begin{cases} Env(\tau) \cup \bigcup_{t \in \Psi} Act(t) & \text{for } n = \tau \\ Env(n) & \text{for } n \neq \tau. \end{cases}$$

A sequence (possibly null) $\mathcal{S} = \Psi_0.\Psi_1 \ldots \Psi_n$ of pairwise disjoint transition sets such that $t_i \rhd t_j$, for each $t_i \in \Psi_i$ and $t_j \in \Psi_j$ with $0 \leq i < j \leq n$, is a *step at time* τ from a configuration C to a configuration C' iff there exists a sequence $C_0 \ldots C_{n+1}$ of configurations such that $C_0 = C$, $C_{n+1} = C'$, C_{i+1} is reached from C_i by microstep Ψ_i at time τ, for $0 \leq i \leq n$. Step $\mathcal{S}$ is *maximal* iff $\mathcal{S}.\Psi$ is not a step, for each non-empty microstep Ψ from C'.
A sequence $\mathcal{S}_0 \ldots \mathcal{S}_n$, where $\mathcal{S}_i$ is a sequence (possibly null) of transition sets, for $0 \leq i \leq n$, is a *behaviour* from an initial configuration C iff there exists a sequence of configurations $C_0 \ldots C_{n+1}$, with $C_0 = C$ and $\mathcal{S}_i$ is a maximal step at time i from C_i to C_{i+1}, for all $0 \leq i \leq n$.

In the standard semantics (see [3]) a step is a sequence $\Psi_0.\Psi_1 \ldots \Psi_n$ of pairwise disjoint transition sets such that $t_i \diamond t_j$, for each $t_i \in \Psi_i$ and $t_j \in \Psi_j$ with $0 \leq i \leq j \leq n$. With reference to Fig.1, if the transitions f and e (resp.: c and d) are both enabled, then they can be performed in the same step provided that f is performed before e (resp.: b is performed before d). In the classical semantics they cannot be performed in the same step and the choice between them is non-deterministic. As shown in [6], the advantages (with respect to the standard semantics) are that (high-level) transitions can be interpreted as interrupts and that some natural refinement techniques are supported.

3 Statecharts from Transition Structures

In this section we show that a transition structure completely describes a statechart. We present a recursive algorithm which associates, with the transition structure of a statechart Z, a statechart Z' having the same transition structure of Z. Actually, the algorithm is given for a slight restriction of the class of Statecharts, namely, for Statecharts all of whose states are *downward reachable*. A state $b \in B_Z$ is *downward reachable* iff there exists a transition t such that $b \in \Downarrow \{in(t)\}$. But in few pathological cases, a statechart can be transformed into a statechart with downward reachable states. In order to give a recursive definition of the algorithm, we have to extend the notion of Statecharts given in Def.1, thus allowing Statecharts possibly containing transitions either without source or without target state (*dangling transitions*). In particular, each state b

in a statechart Z *induces* an (incomplete) statechart whose set of states is $\rho^*(b)$ and whose set of transitions is

$$\{t \in T_Z : in_Z(t) \in \rho_Z^\star(b) \text{ or } out_Z(t) \in \rho_Z^\star(b)\} \cup \{t \in T_Z : b \in \Downarrow_Z \{in_Z(t)\}\}.$$

The other components are the obvious restrictions of the corresponding components of Z.

As concerns the transition structure $\Omega = (T_\Omega, \rightarrow_\Omega, \rhd_\Omega, \chi_\Omega)$ of an incomplete statechart Z, we need to distinguish dangling transitions from the others. In particular, we are interested in the following subsets of T_Ω:

1. $K_\Omega \subseteq T_\Omega$ is the set of non dangling transitions;
2. $O_\Omega \subseteq T_\Omega$ is the set of transitions without target;
3. $W_\Omega \subseteq O_\Omega$ is the set of transitions whose source is the root;
4. $I_\Omega \subseteq T_\Omega$ is the set of transitions without source;
5. $U_\Omega \subseteq I_\Omega$ is the set of transitions not having a substate of the root as target.

Note that $T_\Omega = I_\Omega \cup K_\Omega \cup O_\Omega$. Note, also, that if Z is a *complete* statechart (i.e., as in Def.1), then the initial transition *imp* can be regarded as an incomplete transition leading to the root of Z, and so, our convention is that, for $\Omega = \mathcal{TS}(Z)$, $K_\Omega = T_Z$, $I_\Omega = U_\Omega = \{imp\}$, and $O_\Omega = W_\Omega = \emptyset$. Given a transition structure Ω, we associate a pair of symbols $o(t)$ and $i(t)$ with each transition $t \in K_\Omega$, a symbol $i(t)$ with each transition $t \in I_\Omega$, and a symbol $o(t)$ with each transition $t \in O_\Omega$. Symbols $o(t)$ and $i(t)$ represent a source and a target state, respectively, for transition t. Over that set of symbols we define an equivalence relation $\equiv_\Omega$ and we consider the quotient set. If the quotient set is not a singleton, then we associate with Ω a statechart whose root is an or-state. The set of the immediate substates of the root is in one-to-one correspondence with the quotient set. If the quotient set is a singleton, then we associate with Ω a statechart whose root is an and-state and the set of parallel components of the root is in one-to-one correspondence with the quotient set of the set of transition T under an equivalence relation $\approx_\Omega$ (defined below). Each element of the quotient sets under $\equiv_\Omega$ and $\approx_\Omega$ induces a transition structure, and the required statechart can be obtained by recursively applying the outlined step to the transition structures induced by the elements of the quotient set.

Definition 6. For a transition structure Ω, assume that $i, o : T_\Omega \rightarrow M$ are fixed injective and image disjoint maps, with M a suitable alphabet;
then, $\equiv_\Omega \subseteq M \times M$ is the least equivalence relation s.t.:

1. if $t^{\rightarrow_\Omega} = t'^{\rightarrow_\Omega} = \emptyset$, then $i(t) \equiv_\Omega i(t')$, for $t, t' \in I_\Omega \cup K_\Omega$;
2. if $t \rhd_\Omega t'$, then $o(t) \equiv_\Omega o(t')$ and $o(t) \equiv_\Omega i(t)$, for $t, t' \in T_\Omega$;
3. if $t \rightarrow_\Omega t'$, then $i(t) \equiv_\Omega o(t')$, for $t \in I_\Omega \cup K_\Omega$ and $t' \in (O_\Omega - W_\Omega) \cup K_\Omega$;

The quotient set of $\equiv_\Omega$ over $i(I_\Omega \cup K_\Omega) \cup o((O_\Omega - W_\Omega) \cup K_\Omega)$ is denoted by $\mathcal{V}_\Omega$.

Example 1. Let Ω be the transition structure of the statechart of Fig.1, then $I_\Omega = U_\Omega = \{imp\}$, $O_\Omega = \emptyset$ and $K_\Omega = \{a, b, c, d, e, f, g, h\}$.

The quotient set is $\mathcal{V}_\Omega = \{v_1, v_2, v_3\}$, with:
$v_1 = \{i(e), i(imp), o(a), o(h)\}$;
$v_2 = \{i(a), i(h), i(c), i(b), o(b), o(c), o(d)\}$;
$v_3 = \{i(d), i(f), i(g), o(f), o(g), o(e)\}$.
A statechart with an or-state root corresponds with Ω.

For a transition structure Ω, each element $v \in \mathcal{V}_\Omega$ *induces a transition structure* Ω_v, where

$$T_{\Omega_v} \stackrel{def}{=} \{t \in T_\Omega : \text{ either } i(t) \in v \text{ or } o(t) \in v\}$$

and $\rightarrow_{\Omega_v}$, $\rhd_{\Omega_v}$, and χ_{Ω_v} are obvious restrictions of $\rightarrow_\Omega$, $\rhd_\Omega$, and χ_Ω, respectively, to T_{Ω_v}. Moreover,

1. $K_{\Omega_v} \stackrel{def}{=} \{t \in T_\Omega : o(t), i(t) \in v\}$;
2. $O_{\Omega_v} \stackrel{def}{=} \{t \in T_\Omega : o(t) \in v, i(t) \notin v\}$;
3. $W_{\Omega_v} \stackrel{def}{=} \{t \in O_{\Omega_v} : h \rhd_\Omega t \text{ for all } h \in K_{\Omega_v}, h \rightarrow_\Omega t \text{ for all } h \in I_{\Omega_v}\}$;
4. $I_{\Omega_v} \stackrel{def}{=} \{t \in T_\Omega : o(t) \notin v \text{ and } i(t) \in v\}$;
5. $U_{\Omega_v} \stackrel{def}{=} \begin{cases} \text{if } i(U_\Omega) \cap v \neq \emptyset, \text{ then} \\ U_\Omega \cup \{t : t \in I_{\Omega_v} - U_\Omega, t^{\rightarrow_{\Omega_v}} = h^{\rightarrow_{\Omega_v}}, \text{ for all } h \in U_\Omega\}; \\ \text{otherwise, a maximal set of transitions } V \subseteq I_{\Omega_v} \text{ s.t.} \\ t, h \in V \text{ implies } t^{\rightarrow_{\Omega_v}} = h^{\rightarrow_{\Omega_v}}. \end{cases}$

Example 2. With reference to the quotient set of Ex.1,
$K_{\Omega_{v_1}} = \emptyset$; $O_{\Omega_{v_1}} = W_{\Omega_{v_1}} = \{a, h\}$; $I_{\Omega_{v_1}} = U_{\Omega_{v_1}} = \{e, imp\}$;
$K_{\Omega_{v_2}} = \{c, b\}$; $O_{\Omega_{v_2}} = \{d\}$; $W_{\Omega_{v_2}} = \emptyset$; $I_{\Omega_{v_2}} = \{a, h\}$;
there are two possible choices for $U_{\Omega_{v_2}}$: either $U_{\Omega_{v_2}} = \{a\}$ or $U_{\Omega_{v_2}} = \{h\}$;
$K_{\Omega_{v_3}} = \{f, g\}$; $O_{\Omega_{v_3}} = W_{\Omega_{v_3}} = \{e\}$; $I_{\Omega_{v_3}} = U_{\Omega_{v_3}} = \{d\}$.

The sets O_{Ω_v} and W_{Ω_v} allow us to determine whether the source of a dangling outgoing transition t in Ω_v is the root of the statechart associated with Ω_v (i.e., $t \in W_{\Omega_v}$) or a substate of the root (i.e., $t \in O_{\Omega_v} - W_{\Omega_v}$). The transitions leading to the root of the statechart associated with Ω_v are the ones belonging to $U_{\Omega_v} - U_\Omega$. The default substate of the root is associated with the element $v \in \mathcal{V}_\Omega$ such that $U_\Omega \subseteq U_{\Omega_v}$.

We define now the equivalence $\approx_\Omega$.

Definition 7. For a transition structure Ω, $\approx_\Omega \subseteq (T_\Omega - (U_\Omega \cup W_\Omega)) \times (T_\Omega - (U_\Omega \cup W_\Omega))$ is the least equivalence relation s.t.:

1. if $t \rightarrow_\Omega t'$, then $t \approx_\Omega t'$, for $t \in K_\Omega$ and $t' \in (O_\Omega - W_\Omega) \cup K_\Omega$;
2. if either $t_1 \rightarrow_\Omega h$, $t_2 \not\rightarrow_\Omega h$ or $t_2 \rightarrow_\Omega h$, $t_1 \not\rightarrow_\Omega h$, then $t_2 \approx_\Omega h$, for $t_1 \in U_\Omega$, $t_2 \in I_\Omega - U_\Omega$ and $h \in (O_\Omega - W_\Omega) \cup K_\Omega$;
3. if there exists a chain $t_1 \rightarrow_\Omega \ldots \rightarrow_\Omega t_k$ such that $t_1 \in I_\Omega$, $t_i \in K_\Omega$ for $2 \leq i \leq k$, then $t_k \approx_\Omega h$, for all $h \in K_\Omega \cup (O_\Omega - W_\Omega)$ such that $t_1 \rightarrow_\Omega h$ and $t_k \not\rhd_\Omega h$.

The quotient set of $\approx_\Omega$ is denoted as $\mathcal{A}_\Omega$.

For a transition structure Ω, each element $v \in \mathcal{A}_\Omega$ *induces a transition structure* Ω_v, where $T_{\Omega_v} \stackrel{def}{=} I_\Omega \cup v$ and $\rightarrow_{\Omega_v}$, $\rhd_{\Omega_v}$, and χ_{Ω_v} are the obvious restrictions of $\rightarrow_\Omega$, $\rhd_\Omega$, and χ_Ω, respectively, to T_{Ω_v}.
Moreover, $K_{\Omega_v} \stackrel{def}{=} K_\Omega \cap v$; $O_{\Omega_v} \stackrel{def}{=} O_\Omega \cap v$; $W_{\Omega_v} \stackrel{def}{=} \emptyset$; $I_{\Omega_v} \stackrel{def}{=} I_\Omega$; $U_{\Omega_v} \stackrel{def}{=} I_\Omega - v$.

Example 3. Let Ω' (resp.: Ω'') be the transition structure induced by v_2 (see Ex.1 and Ex.2) with $U_{\Omega_{v_2}} = \{a\}$ (resp.: $U_{\Omega_{v_2}} = \{h\}$). The quotient sets $\mathcal{V}_{\Omega'}$ and $\mathcal{V}_{\Omega''}$ are singletons. The quotient set $\mathcal{A}_{\Omega'}$ (resp.: $\mathcal{A}_{\Omega''}$) has two elements $z_1 = \{h, c\}$ and $z_2 = \{b, d\}$ (resp.: $w_1 = \{a, c\}$ and $w_2 = \{b, d\}$). Moreover, $K_{\Omega'_{z_1}} = K_{\Omega''_{w_1}} = \{c\}$; $K_{\Omega'_{z_2}} = K_{\Omega''_{w_2}} = \{b\}$; $O_{\Omega'_{z_1}} = O_{\Omega''_{w_1}} = \emptyset$; $O_{\Omega'_{z_2}} = W_{\Omega''_{w_2}} = \{d\}$; $I_{\Omega'_{z_1}} = I_{\Omega''_{w_1}} = I_{\Omega'_{z_2}} = I_{\Omega''_{w_2}} = \{h, a\}$; $U_{\Omega'_{z_1}} = \{a\}$; $U_{\Omega''_{w_1}} = \{h\}$; $U_{\Omega'_{z_2}} = U_{\Omega''_{w_2}} = \{a, h\}$.

We define now the algorithm associating statecharts with transition structures. Since the way for defining the set U_{Ω_v}, for $v \in \mathcal{V}_\Omega$, is not unique, more than one statechart can be associated with a transition structure.

Definition 8. For a transition structure Ω, a statechart Z and a symbol sym, $\Omega \Rightarrow_{sym} Z$ iff $T^i_Z = T_\Omega$, $\chi_Z = \chi_\Omega$, and

1. if $T_\Omega = U_\Omega \cup W_\Omega$, then $B_Z = \{sym\}$, $\rho_Z = \{(sym, \emptyset)\}$, $\phi_Z = \{(sym, OR)\}$, $in_Z = \delta_Z = \emptyset$ and $out_Z = \{(t, sym) : t \in W_\Omega\}$.
2. if $T_\Omega \neq U_\Omega \cup W_\Omega$ and $\mathcal{V}_\Omega$ is a singleton, then, for $v \in \mathcal{A}_\Omega$,

$$Z_v \text{ is a statechart such that } \Omega_v \Rightarrow_{(sym,v)} Z_v, \text{ and}$$

$B_Z = \{sym\} \cup \bigcup_{v \in \mathcal{A}_\Omega} B_{Z_v}$;
$\rho_Z = \{(sym, sym \times \mathcal{A}_\Omega)\} \cup \bigcup_{v \in \mathcal{A}_\Omega} \rho_{Z_v}$;
$in_Z = \bigcup_{v \in \mathcal{A}_\Omega} in_{Z_v}$ and $out_Z = \bigcup_{v \in \mathcal{A}_\Omega} out_{Z_v} \cup \{(t, sym) : t \in W_\Omega\}$;
$\delta_Z = \bigcup_{v \in \mathcal{A}_\Omega} \delta_{Z_v}$; $\phi_H = \{(sym, AND)\} \cup \bigcup_{v \in \mathcal{A}_\Omega} \phi_{Z_v}$.

3. otherwise, for $v \in \mathcal{V}_\Omega$,

$$Z_v \text{ is a statechart such that } \Omega_v \Rightarrow_{(sym,v)} Z_v, \text{ and}$$

$B_Z = \{sym\} \cup \bigcup_{v \in \mathcal{V}_\Omega} B_{Z_v}$;
$\rho_Z = \{(sym, sym \times \mathcal{V}_\Omega)\} \cup \bigcup_{v \in \mathcal{V}_\Omega} \rho_{Z_v}$;
$in_Z = \{(t, (sym, v)) : t \in U_{\Omega_v} - U_\Omega\} \cup \bigcup_{v \in \mathcal{V}_\Omega} in_{Z_v}$;
$out_Z = \{(t, sym)) : t \in W_\Omega\} \cup \bigcup_{v \in \mathcal{V}_\Omega} out_{Z_v}$;
$\delta_Z = \{(sym, (sym, \overline{v}))\} \cup \bigcup_{v \in \mathcal{V}_\Omega} \delta_{Z_v}$, with $\overline{v} \in \mathcal{V}_\Omega$ s.t. $\overline{v} \cap i(U_\Omega) \neq \emptyset$;
$\phi_Z = \{(sym, OR)\} \cup \bigcup_{v \in \mathcal{V}_\Omega} \phi_{Z_v}$.

For Z the statechart of Fig.1, there are two statecharts Z' and Z'' such that $\mathcal{TS}(Z) \Rightarrow_{sym} Z'$ and $\mathcal{TS}(Z) \Rightarrow_{sym} Z''$: Z' is isomorphic to Z (with ref. to Ex.2, $U_{\Omega_{v_2}} = \{h\}$), and Z'' is represented in Fig.2 (with ref. to Ex.2, $U_{\Omega_{v_2}} = \{a\}$).

Theorem 9. For a (complete) statechart Z with downward reachable states,

$$\mathcal{TS}(Z) \Rightarrow_{sym} Z' \text{ implies } \mathcal{TS}(Z) = \mathcal{TS}(Z').$$

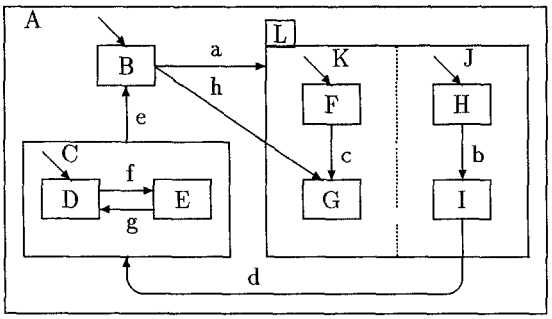

Fig. 2. A statechart built from the transition structure of the one in Fig.1.

Proof. (Sketch) Let Z be a (possibly incomplete) statechart with $\mathcal{TS}(Z) = \Omega$ and let Z' be such that $\Omega \Rightarrow_{sym} Z'$. That Z' is a statechart can be easily checked. We prove, by induction on the definition of relation $\Rightarrow_{sym}$, that, if $\Omega' = \mathcal{TS}(Z)$, then Ω and Ω' are componentwise equal but possibly unequal with respect to component U. (Actually, if Z is complete, then by construction, also Z' is, and $U_\Omega = U_{\Omega'} = \{imp\}$, thus proving the thesis).
(Base) That $\Omega = \Omega'$ immediately follows from construction.
(Inductive step) Assume that $\mathcal{V}_\Omega$ is not a singleton and let Z'_v, for $v \in \mathcal{V}_\Omega$, be the statechart induced by state (sym, v) of Z'. If $\Omega_{Z'_v} = \mathcal{TS}(Z'_v)$, then, by induction hypothesis, we have that Ω_v and $\Omega_{Z'_v}$ are componentwise equal (but possibly unequal with respect to component U). By construction Z' is a statechart having or-state root, and the transition structure of Z' can be obtained from the ones of Z'_v (for $v \in \mathcal{V}_\Omega$) as follows: $T_{\Omega'} = \bigcup_{v\in\mathcal{V}_\Omega} T_{\Omega_v} \cup W_\Omega$; $K_{\Omega'} = \bigcup_{v\in\mathcal{V}_\Omega} K_{\Omega_v} \cup \bigcup_{v,v'\in\mathcal{V}_\Omega} I_{\Omega_v} \cap O_{\Omega_{v'}}$; $O_{\Omega'} = W_\Omega \cup \bigcup_{v\in\mathcal{V}_\Omega} O_{\Omega_v} - \bigcup_{v,v'\in\mathcal{V}_\Omega} I_{\Omega_v} \cap O_{\Omega_{v'}}$; $W_{\Omega'} = W_\Omega$; $I_{\Omega'} = \bigcup_{v\in\mathcal{V}_\Omega} I_{\Omega_v} - \bigcup_{v,v'\in\mathcal{V}_\Omega} I_{\Omega_v} \cap O_{\Omega_{v'}}$; $\rightarrow_{\Omega'} = \bigcup_{v\in\mathcal{V}_\Omega} \rightarrow_{\Omega_v} \cup \{(t,t') : t \in I_{\Omega'}, t' \in W_{\Omega'}\}$; $\rhd_{\Omega'} = \bigcup_{v\in\mathcal{V}_\Omega} \rhd_{\Omega_v} \cup \{(t,t') : t \in K_{\Omega'}, t' \in W_{\Omega'}\}$. Now, by exploiting Def.6, it can be easily shown that $T_{\Omega'} = T_\Omega$, $K_{\Omega'} = K_\Omega$ $O_{\Omega'} = O_\Omega$, $I_{\Omega'} = I_\Omega$, $\rightarrow_{\Omega'} = \rightarrow_\Omega$ and $\rhd_{\Omega'} = \rhd_\Omega$. (Note that component U_Ω is not referenced in Def.6). Assume now that $\mathcal{V}_\Omega$ is a singleton. It can be proved that $\mathcal{A}_\Omega$ is not a singleton. Let Z'_v, for $v \in \mathcal{A}_\Omega$, be the statechart induced by state (sym, v) of Z'. By induction hypothesis, we have that Ω_v and $\Omega_{Z'_v}$ are componentwise equal (but possibly unequal with respect to component U). Now, Z' is a statechart having and-state root and the transition structure of Z' can be obtained from the ones of Z'_v (for $v \in \mathcal{A}_\Omega$) as follows: $T_{\Omega'} = \bigcup_{v\in\mathcal{A}_\Omega} T_{\Omega_v} \cup W_\Omega$; $K_{\Omega'} = \bigcup_{v\in\mathcal{A}_\Omega} K_{\Omega_v}$; $O_{\Omega'} = \bigcup_{v\in\mathcal{A}_\Omega} O_{\Omega_v} \cup W_\Omega$; $W_{\Omega'} = W_\Omega$; $I_{\Omega'} = I_{\Omega_v}$ (for all $v \in \mathcal{A}_\Omega$); $\rightarrow_{\Omega'} = \bigcup_{v\in\mathcal{A}_\Omega} \rightarrow_{\Omega_v} \cup \{(t,t') : t \in I_{\Omega'}, t' \in W_{\Omega'}\}$; $\rhd_{\Omega'} = \bigcup_{v\in\mathcal{V}_\Omega} \rhd_{\Omega_v} \cup \{(t,t') : t \in K_{\Omega_v}, t' \in T_{\Omega_{v'}}$, for some $v \neq v'$ and there is $h \in I_{\Omega'}$ s.t. $h \rightarrow^\star_{\Omega_v} t, h \rightarrow^\star_{\Omega'_v} t'\}$[3]. Now, by exploiting Def.7, it can be easily shown that $T_{\Omega'} = T_\Omega$, $K_{\Omega'} = K_\Omega$ $O_{\Omega'} = O_\Omega$, $\rightarrow_{\Omega'} = \rightarrow_\Omega$ and $\rhd_{\Omega'} = \rhd_\Omega$. (Equalities above can be proved by exploiting only points 1 and 3 of Def.7 where U_Ω is not referenced). □

[3] $\rightarrow^\star$ denotes the transitive closure of $\rightarrow$.

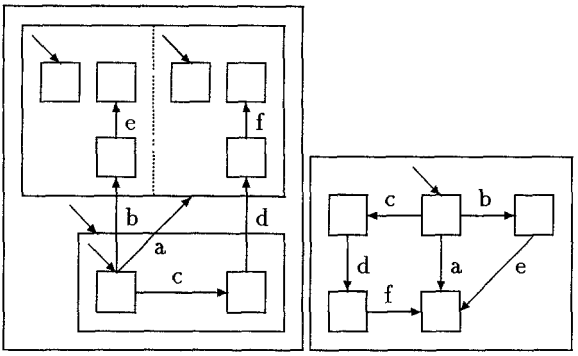

Fig. 3. A statechart Z (on the letf-hand side) and a statechart Z' such that $\mathcal{TS}(Z) \Rightarrow_{sym} Z'$.

In general, two statecharts Z and Z' such that $\mathcal{TS}(Z) \Rightarrow_{sym} Z'$ are not isomorphic. For instance, consider the statecharts Z and Z' of Fig.3. The statechart Z has an and-state though, apparently, no pair of transitions can be performed simultaneously. Since $\mathcal{TS}(Z)$ is an encoding of the behaviour properties of Z, the structural details of Z which do not affect behaviour cannot be recovered in Z'. This suggests that, in general, the algorithm of Def.8 can be exploited for simplifying a statechart by removing its irrelevant structural details.

4 Characterizing Transition Structures for Well Structured Statecharts

In the previous section we have shown that a given statechart can be represented by its transition structure. In general, a quadruple $(T, \rightarrow, \triangleright, \chi)$ (we call it *structure*), where $\rightarrow$ and $\triangleright$ are relations over set T and χ is a function on T, describes a statechart if there is a statechart Z s.t. $\mathcal{TS}(Z) = (T, \rightarrow, \triangleright, \chi)$. We are able to characterize the class of structures which describe *well structured statecharts*. A statechart Z is *well structured* iff

$$in_Z(t), out_Z(t) \in \rho_Z(\mathcal{L}_Z(t)), \text{ for all } t \in T_Z. \tag{1}$$

Transitions of well structured statecharts preserve the hierarchy on states (i.e., no transition crosses borders of boxes depicting states). Well structured statecharts is a proper restriction of the formalism which has been adopted frequently in literature to enjoy a simpler definition of semantics (e.g. see [8]). With reference to Fig.1, the represented statechart is not well structured since transitions a and

d do not satisfy constraint (1). Structures describing well structured statecharts can be axiomatically defined, thus allowing to precisely state the relationship among causality, parallelism and interruptions (i.e., hierarchy) in the formalism.

Definition 10. A 4-tuple $(T, \rightarrow, \rhd, \chi)$, where T is a finite set, $\rightarrow, \rhd \subseteq T \times T$ are anti-reflexive relations and $\chi : T \rightarrow 2^P \times 2^P$, for some finite set P, is a *good structure* iff:

1. if $t \rightarrow^\star t'$, then $t \not\rhd t'$[4];
2. there exists t s.t. $t \rightarrow^\star t'$, for all $t' \in T - \{t\}$, and $t'' \not\rightarrow t$, for all $t'' \in T$;
3. if $t \rightarrow t'$ and $t \rhd t''$, then $t' \rhd t''$;
4. if $t \rhd t'$ and $t' \not\rhd t$, then $h \rightarrow t$ and $h \not\rhd t'$ implies $h \rightarrow t'$;
5. if $t \rhd t'$, $t' \not\rhd t$, $t' \rhd t''$ and $t'' \not\rhd t'$, then $t \rhd t''$;
6. if $t \rhd t'$, $t' \not\rhd t$ and $t' \diamond t''$ [5], then $t \diamond t''$;
7. if $t \rhd t'$, $t' \not\rhd t$ and $t \diamond t''$, then $t'' \rhd t'$;
8. if $t \rightarrow t'$ and $t \diamond t''$, then $t' \diamond t''$;
9. if $t \rightarrow t'$, $h \rightarrow h'$, $t' \diamond h'$, $t \not\diamond h'$ and $h \not\diamond t'$, then $t \rightarrow h'$ and $h \rightarrow t'$;
10. if $t \rightarrow t'$, $h \rightarrow h'$, $t' \diamond h'$, $t \not\diamond h'$ and $h \diamond t'$, then $t \rightarrow g_1 \rightarrow \ldots \rightarrow g_n \rightarrow h$ and $g_i \diamond t'$;
11. if $t^{\rightarrow} \cap t'^{\rightarrow} \neq \emptyset$, then $t \rhd h$, for all $h \in t'^{\rightarrow} - t^{\rightarrow}$ and $t' \rhd h$, for all $h \in t^{\rightarrow} - t'^{\rightarrow}$;
12. if $t \rightarrow h, h', h''$, then $h \rhd h'$ and $h'' \not\rhd h'$ implies $h \rhd h''$;

The properties at points 2, 5 and 6 are satisfied also by the intere class of Statecharts. Good structures are a dual representation, with respect to the classical one, of well structured statechart. It is possible to shift from one representation to the other by exploiting the map $\mathcal{TS}$ and the relation $\Rightarrow_{sym}$.

Theorem 11. A structure Ω is a good structure iff there exists a well structured (downward reachable) statechart such that $\Omega = \mathcal{TS}(Z)$.

Proof. (Sketch) ($\Rightarrow$) It can be shown that the procedure of Def.8 is well defined when it is applied to good structures. In particular, it can be shown that if Ω is a good structure, and $\mathcal{V}_\Omega$ is a singleton, then $\mathcal{A}_\Omega$ is not a singleton, and, if $v \in \mathcal{V}_\Omega$ or $v \in \mathcal{A}_\Omega$, then Ω_v is also a good structure. So, the procedure always terminates and, for each good structure Ω, there exists a statechart Z such that $\Omega \Rightarrow_{sym} Z$. By Th.9, we have that $\Omega = \mathcal{TS}(Z)$. Moreover, it is possible to show, by induction on the definition of relation $\Rightarrow_{sym}$, that Z is well structured.
($\Leftarrow$) It is easy to check that $\mathcal{TS}(Z)$ satisfies the constraints of Def.10. □

5 Morphisms of Transition Structures and Transformations of Statecharts

In this section we show that the investigation of transition structure morphisms is useful for investigating transformations of statecharts which preserve behaviour.

[4] $\rightarrow^\star$ denotes the transitive closure of $\rightarrow$.

[5] $t \diamond t'$ is a short writing for $t \rhd t'$ and $t' \rhd t$.

A morphism between two transition structures is a map from the transition set of the former to the transition set of the latter which preserves the initial transition, sequentiality and independence relations, and transition labels. We are interested in morphisms which preserve behaviour in the sense that a behaviour corresponding to the former transition structure can be suitably mapped into a behaviour corresponding to the latter and viceversa. Behaviour preserving morphisms induce classes of behaviour preserving transformations over statecharts.

Definition 12. A (partial) map $\omega : T_\Omega \rightarrow T_{\Omega'}$ is a *morphism of transition structures* Ω and Ω', iff:

1. $\omega(imp) = imp$;
2. $t \rightarrow_\Omega t'$ implies $\omega(t) \rightarrow_{\Omega'} \omega(t')$ (whenever $\omega(t)$ and $\omega(t')$ are defined);
3. $t \rhd_\Omega t'$ implies $\omega(t) \rhd_{\Omega'} \omega(t')$ (whenever $\omega(t)$ and $\omega(t')$ are defined);
4. $\chi_{\Omega'}(\omega(t)) = \chi_\Omega(t)$ (whenever $\omega(t)$ is defined).

Unless otherwise said, a morphism is assumed to be a total map. Obviously, isomorphisms[6] of transition structures preserve behaviour. For instance, if statecharts Z, Z' and Z'' are such that $\mathcal{TS}(Z) \Rightarrow_{sym} Z'$ and $\mathcal{TS}(Z) \Rightarrow_{sym} Z''$, then their transition structures are isomorphic. In general, isomorphisms induces transformations over statecharts which allow to add (resp.: remove) irrelevant structural details to (resp.: from) a statechart or to suitably change its defaults (and sources of transitions accordingly). Actually, we are interested in morphisms less strict than isomorphisms. For instance, consider the problem of transforming a statechart into a well structured equivalent statechart. Sometimes, a statechart can be transformed into a well structured statechart provided that a number of states and transitions are duplicated (the corresponding transition structures fail to be isomorphic). As an example, see Fig.4 where a transformation of the statechart of Fig.1 is depicted. The transition a of Fig.4 does not violate the constraint 1 but it does in Fig.1. The transformation requires the duplication of transitions b and d. Unfortunately, morphisms as defined in Def.12 do not preserve behaviour, since maximality of steps is not preserved and, so, additional constraints must be considered.

Definition 13. The morphism ω of transition structures Ω and Ω' is called *strong* when,

$$\omega(t^{\rightarrow_\Omega}) = \omega(t)^{\rightarrow_{\Omega'}} \text{ for all } t \in T_\Omega \text{ and} \tag{2}$$

$$t \rhd_\Omega t' \text{ for all } t, t' \in T_\Omega \text{ s.t. } \omega(t) \rhd_{\Omega'} \omega(t') \text{ and } t \|_\Omega t'^{7}. \tag{3}$$

Let us consider the statecharts of Fig.4 and the one of Fig.1. The map ω, from the transitions of the former to the ones of the latter such that $\omega(b) = \omega(b') = b$, $\omega(d) = \omega(d') = c$, and acting as the identity elsewhere, is a strong morphism.

[6] A morphism ω is an *isomorphism* iff it is a bijection and ω^{-1} is also a morphism.

[7] For a statechart Z, the concurrency relation $\|_Z$ can be derived from $\Omega = \mathcal{TS}(Z)$ (it suffices to rephrase Def.3 by exploiting relations $\rightarrow_\Omega$ and $\rhd_\Omega$). So, we write $\|_\Omega$ instead of $\|_Z$.

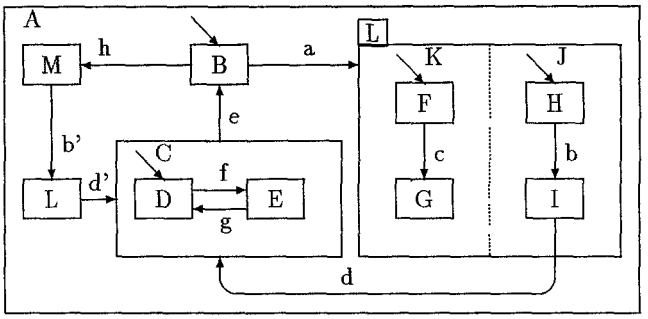

Fig. 4. A statechart admitting a strong morphism from its transition structure to the one of the statechart of Fig.1.

Partial strong morphisms preserve behaviour provided that other constraints are satisfied. In particular, we consider partial morphisms which *preserve causality, choice and communication* in the sense that if a morphism is defined over a transition t, then it is defined over all the transitions causing t, over all the transitions which are concurrent and mutually exclusive with respect to t, and over all the transitions communicating signals enabling t.

Definition 14. A partial morphism ω of transition structures Ω and Ω' *preserves causes* if, for $t, t' \in T_\Omega$ $\omega(t)$ defined and $t' \rightarrow_\Omega t$ implies $\omega(t')$ is defined;
ω *preserves choices* if, for $t, t', t'' \in T_\Omega$, that $t \rightarrow_\Omega t'$, $t \rightarrow_\Omega t''$, $\omega(t)$ and $\omega(t')$ are defined and $\omega(t'')$ is not defined implies $t'' \rhd_\Omega t'$;
ω *preserves communication* iff for all $t, t' \in T_\Omega$, $\omega(t)$ undefined and $\omega(t')$ defined implies $Act(t) \cap Ev(t') = \emptyset$.

When a morphism of transition structures Ω and Ω' is partial, we compare the part of the transition structure over which the morphism is defined, with Ω'. Partial morphisms can be exploited for proving the correctness of some refinement techniques. As an example, the statechart of Fig.1 is a refinement of both the statecharts of Fig.5 and the statechart on the left-hand side of figure Fig.5 is a refinement of the one on the right-hand side. The map ω from the set of transitions of the statechart in Fig.1 to the set of the one on the left-hand side (resp.: right-hand side) of Fig.5 such that ω is not defined in f and g (resp.: in f, g and c) and ω is the identity elsewhere, is a strong morphism which preserves causes and choices (in our examples we have not considered transition labels and so we omit the property of preserving communication). A similar morphism can be defined between the transition structures of the two statecharts of Fig.5.

Theorem 15. Let ω be a strong morphism of transition structures Ω and Ω' (resp.: a strong partial morphism which preserve causes, choices and communication), then $\mathcal{S}_0.\mathcal{S}_1 \ldots \mathcal{S}_n$ is a behaviour from a configuration $(imp^{\rightarrow_{\Omega'}}, Env)$ iff there exists a behaviour $\mathcal{S}'_0.\mathcal{S}'_1 \ldots \mathcal{S}'_n$ from $(imp^{\rightarrow_{\Omega}}, Env)$ such that

$$\mathcal{S}_0.\mathcal{S}_1 \ldots \mathcal{S}_n = \omega(\mathcal{S}'_0).\omega(\mathcal{S}'_1) \ldots \omega(\mathcal{S}'_n).$$

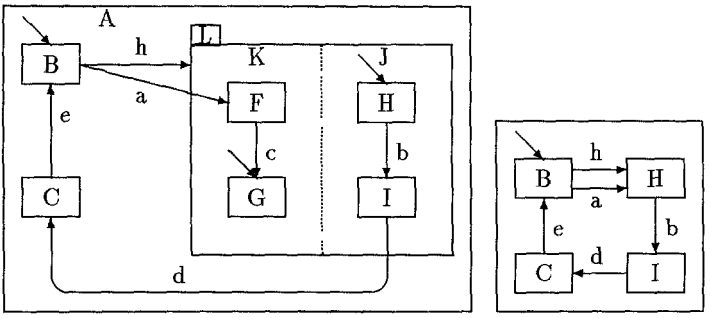

Fig. 5. Two statecharts of which the statechart in Fig.1 is a refinement.

Note that a strong morphism preserves behaviours (in the sense of Th.15) even when the classical semantics is enforced. That is not the case when partial morphisms are considered. For a discussion on the problem we refer to [6].

References

1. von der Beek, M.: A Comparison of Statecharts Variants, LNCS 863, Springer, Berlin, 1994, pp. 128–148.
2. Harel, D.: Statecharts: A Visual Formalism for Complex Systems, Science of Computer Programming 8 (1987), pp. 231–274.
3. Harel, D., Pnueli, A., Schmidt, J., P., Sherman, R.: On the Formal Semantics of Statecharts, Proc. 2nd IEEE Symposium on Logic in Computer Science, IEEE CS Press, New York, 1987, pp. 54–64.
4. Huizing, C., Gerth, R., de Roever, W.P.: Modelling Statechart Behaviour in a Fully Abstract Way, LNCS 299, Springer, Berlin, 1988, pp. 271–294.
5. Peron, A.: Synchronous and Asynchronous Models for Statecharts, Dipartimento di Informatica, Università di Pisa, PhD Thesis, TD 21/93, 1993.
6. Peron, A., Maggiolo-Schettini, A.: Transitions as Interrupts: A New Semantics for Timed Statecharts, LNCS 789, Springer, 1994, Berlin, pp. 806–821.
7. Pnueli, A., Shalev, M.: What is in a Step: On the Semantics of Statecharts, LNCS 525, Springer, 1991, Berlin, pp. 244–464.
8. Uselton, A.C., Smolka S.A.: A Process Algebraic Semantics for Statecharts via State Refinement, Proceedings Working Conference on Programming, Concepts, Methods and Calculi (PROCOMET'94), San Miniato, Italy, 1994, pp. 184–200.