The Syntax and Semantics of μ-Charts

Greg Reeve and Steve Reeves

Department of Computer Science,
University of Waikato,
New Zealand
{gregr,stever}@cs.waikato.ac.nz

Working paper 04/2004

Abstract. μ-Charts is a language for specifying the behaviour of reactive systems. The language is a simplified variant of the well-known language Statecharts that was introduced by Harel [1]. Development of the μ-Charts language is ongoing research undertaken under the auspices of the Formal Methods Laboratory of the Computer Science Department, University of Waikato [5]. This paper gives a comprehensive treatment of the syntax and semantic definition for μ-Charts.

1 Introduction

The language StateCharts was introduced by Harel in [1]. Because Statecharts are based on finite-state automata, which are easily represented in a graphical form, they were the first of many "visual" specification languages. The ability to use diagrams to represent specifications has made Statecharts a popular specification tool for software engineers. The invention of Statecharts marked the beginning of extensive research into making precise their semantics, originally given informally in [1]. Some of this research led to the creation of another similar specification language for reactive systems called μ-Charts.

The rather long pedigree of μ-Charts is a chain of formalisms starting with Statecharts, as described above, and includes Mini-Statecharts [6], an extended version of Mini-Statecharts [13], the μ-Charts of [7], the μ-Charts of [12] and the μ-Charts of [9].

More specifically μ-Charts was originally presented by Philipps and Scholz in [7, 8] where they give their first semantics for the language. Later in [12], Scholz continued to develop μ-Charts. Whilst making substantial changes to the first semantics, Scholz gives some refinement rules and a notion of implementation of charts. Subsequently a translation method from μ-Charts into the $\mathcal{Z}$ specification language was developed [11, 10, 9] in conjunction with the then ISuRF and later $\mathcal{Z}_\lambda$ research projects [5].

Previously, the most detailed description of the Z translation of μ-Charts was given in [10]. Here the authors showed both how to extend the original semantics for μ-Charts to include local variables and value-carrying signals and how the Z translation evolved to cope with this additional machinery.

This paper reflects work that has been ongoing since then. In some respects it is both a step forward and a step backwards from the last presentation of the charts to Z translation [10]. Here we present, again, the latest syntactic and semantic definition for the μ-Charts language. Unlike previously, we consider the presented translation of charts into Z not as another encoding of "the μ-Charts semantics" but as the semantics itself. Also, we give the semantic encoding or translation in its entirety rather than via examples. The semantics relates more closely (though not exactly) to the later definition by Scholz [12] rather than the original semantic definition of Phillips and Scholz. We reiterate, however, that our intention is not to compare this semantics with that, but simply to state the semantics that we consider as the current meaning for μ-Charts.

The semantics we give here do not include local variables and value-carrying signals. It is however the basis for several current projects, one of which is the

proper integration of local variables and value-carrying signals. Others include the development of a refinement calculus and the ongoing development of tool support for the language.

In section 2 we present the definitions of the text-based syntax for charts in conjunction with demonstrating their related graphical presentation. Section 3 gives the semantics of the language. The proofs of the lemmas and propositions of Section 3 are given in Appendix A.

2 The syntax of μ-Charts: the μ-chart language

This section defines the syntax that we intend for μ-Charts discussed in this paper. We present a textual syntactic definition in parallel with describing the graphical representation of the language. First we define the basic building block for μ-Charts specifications which are sequential charts. Then we introduce the syntactic operators that can be used to combine sequential charts into more sophisticated μ-charts. In particular we give the operators for allowing the composition of two charts, the decomposition of a chart, *i.e.* embedding an arbitrary chart into a state of a sequential chart, and the hiding operator that allows the restriction of the signals with which a chart is prepared to interact with its environment. We make the point that each of these are important "first-class" syntactic operators in the language despite the ability to consider them semantically as syntactic sugar.

We formally define the textual syntax of the language using a phrase structure grammar. The graphical syntax is presented using examples.

The set of all possible syntactically correct μ-charts is given by the set μ-Charts as defined by the following grammar.[1]

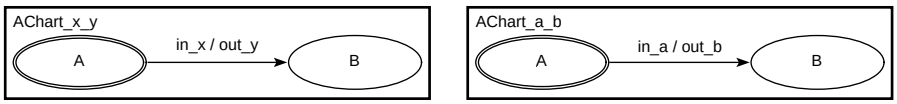
$$
\begin{array}{lcl}
\mu Charts & ::= & Charts \mid \mathbf{(}\ Charts\ \mathbf{)} \\
Charts & ::= & \mu Chart \mid \mu CompChart \\
 & \mid & \mu DecChart \mid \mu ResIChart
\end{array}
$$

In the following we give productions for the non-terminals $\mu Chart$, $\mu CompChart$, $\mu DecChart$ and $\mu ResIChart$.

2.1 Sequential μ-Charts

The smallest unit of μ-Charts is a sequential μ-chart. In general a simple sequential chart is the specification of the allowable behaviour of a named finite state automaton that takes input from its environment and instantaneously produces output accordingly. However, it is also possible to have a feedback loop in a sequential chart that causes some of the output of the chart to be instantaneously fed back and hence act as input. Here we will begin by describing simple sequential charts, *i.e.* those that do not feed back any signals. Then we describe the feed back mechanism for sequential charts in Section 2.2. A sequential chart contains the name of the automaton, a nonempty set of control states, including a special start state (denoted by a double outline in the graphical or visual form of the syntax), a feedback set, and labelled transitions between states. For example consider the chart pictured in figure 1(a).

[1] The phrase structure grammar presented here uses the bracket notation [...] to represent the optional inclusion of the term enclosed in the bracket. Also, the tokens of μ-Charts are given in bold font.



(a) (b)

Fig. 1.

This is an example of a sequential chart named $AChart\_x\_y$. It contains two control states A and B, of which A is uniquely identified as the initial state. The chart contains one transition which has the guard in_x and action out_y. Informally we could describe this transition as follows: "if the chart $AChart\_x\_y$ is in state A and the current input event contains the signal in_x then we instantaneously move to control state B outputing signal out_y".

Notice that we have also introduced a naming convention for charts. Due to the form of the name $AChart\_x\_y$ it is understood that the chart is "parameterized" on the labels x and y. Because we never use or refer to the abstraction $AChart$ itself, it is not necessary to define this mechanism in terms of usual abstraction, that is using variables, definitions and instantiations. Rather we assume the convention that, when given a chart definition such as $AChart\_x\_y$, we are free to consider the chart $AChart\_a\_b$ with the understanding that it is defined as $AChart\_x\_y$ where all occurances of x are replaced by a and similarly y by b. The chart $AChart\_a\_b$ is pictured in Figure 1(b).

To describe the attributes of charts formally, we introduce the following abstract sets.

$$[Signal, Name, State, Param]$$

Each of these sets contain labels used to identify signals, chart names, control states, variables and parameters respectively.

We also introduce the sets $Chartname$ and $\mu Signal$ whose elements are defined using the following grammar.

$$\begin{array}{lcl}
\textit{Chartname} & ::= & \textit{class-name} _ \textit{param-list} \mid \textit{class-name} \\
\mu\textit{Signal} & ::= & \textit{signal-name} _ \textit{param-list} \mid \textit{signal-name} \\
\textit{param-list} & ::= & Param\ [_\ \textit{param-list}] \\
\textit{class-name} & ::= & Name \\
\textit{signal-name} & ::= & Signal
\end{array}$$

Now, consider all possible reactive systems that we might ever want or be able to specify using a μ-chart. The set $Chartname$ is the set of labels used to name sequential charts. The set $\mu Signal$ contains labels that name (possible parameterised) signals that these reactive systems can 1) use to interact with their environment, and 2) use for internal communication between their components.[2] Similarly, $State$ is the set of labels used to name control states.

[2] Again, context permitting, we use the elements of each of these sets to refer to both the label itself and the object that the label represents.

Each sequential chart contains a set of transitions. Each transition is a member of the set $\mu Transition$ and is made up of a tuple of the form $(S_f, S_t, label)$ such that S_f represents the state from which the transition originated, S_t represents the destination state of the transition and *label* defines under what conditions the transition can be taken and the resulting action of that transition. The label of the transition is an element of the set *Transition*. We define the necessary sets as follows.

$\mu Transition$	::=	(*State* , *State* , *Transition*)
Transition	::=	*guard* / *action*
guard	::=	*signal-expr* [& *guard*]
signal-expr	::=	[-]*Signal*
action	::=	{*signal-list*}
signal-list	::=	*Signal* [, *signal-list*]

Now we can define the set $\mu Chart$ of all allowable sequential charts.

$\mu Chart$	::=	(*Chartname*, *control-states*, *initial-state*, *feedback-set*, *transitions*[, *signal-list*])
control-states	::=	{ *initial-state* [, *states-list*] }
initial-state	::=	*State*
states-list	::=	*State* [, *states-list*]
feedback-set	::=	{[*signal-list*]}
transitions	::=	{ [*transition-list*] }
transition-list	::=	$\mu Transition$ [, *transition-list*]

According to this definition sequential charts are described as a tuple that has at least the fields $(C, \Sigma, \sigma, \Psi, \delta)$ such that

- C is the name of the chart
- Σ represents the set of control states
- σ names the initial state
- Ψ represents the set of feedback signals
- δ represents the set of transitions that are defined in the chart

In fact there is an optional set in a sequential chart definition that would extend the tuple to $(C, \Sigma, \sigma, \Psi, \delta, I)$. The first five fields are as described above. The set I can be used to represent the input interface for the chart C. The reason that this set is optional is because it is often the case that the set of transitions δ has sufficient information to calculate the input interface of a chart. On the other hand, the reason that it may be required is because a chart can be defined to have a bigger input interface than those that are syntactically determinable. The reason that we can specify an input interface that has more signals than the set evident from the chart diagram will be explained later in Section **??**.

Notice that every sequential chart must have at least one state (the initial state) but can have an empty set of feedback signals and transitions.

We introduce three auxiliary syntactic functions that are often used in conjunction with the syntactic definition of charts.

$$in_s : \mu Chart \to \mathbb{P}\, \mu Signal$$
$$out_s : \mu Chart \to \mathbb{P}\, \mu Signal$$
$$\omega : Chartname \nrightarrow \mu Charts$$

For each chart $C \in \mu Charts$ there exists a unique input and output interface. As we mentioned above, in the absense of a defined input interface for a sequential chart C, we can calculate the input interface using the function in_s. The interface is the set of signals that appear in any of the transition guards of the chart C. The input interface denotes the set of signals that the specified reactive system is prepared to react to when provided as input from the environment in which the system resides. Similarly, the output interface is either defined or calculated using out_s to be the set of signals that appear in the actions of the chart C. We give the general versions of these functions in Section 2.6.

The third function ω when applied to a name, say C, returns the chart with the name C. We assume this function is always defined to have the property that for any sequential chart $\omega\ C = (C, \Sigma, \sigma, \delta)$ for appropriate Σ, σ and δ. In fact, while this function is technically necessary, in practice the name of a chart, say C, is often used where the intention is $\omega\ C$. We use this shorthand freely assuming the context makes it clear which we are referring to.

There is a clear relation between the textual form of the syntax that is presented here and the graphical presentation of sequential charts, for example Figure 1(b). The chart name is presented in the upper-left corner of the chart and each of the control state are labelled ovals *etc.*. However it is worth noting that the textual definition of charts does not suppose to encode any spatial layout of a graphical chart. Hence, for any "textual chart" there are a great number of equivalent but different looking "graphical charts". Also, the graphical presentation assumes that the input and output interfaces for a chart can be calculated from the transitions of the chart. If this is not the case, that is, if either of these interfaces are bigger than they appear, then the appropriate sets must be defined along with graphical presentation of the chart.

2.2 μ-Charts that feed back signals

The feedback set in sequential charts allows us to model the instantaneous feedback of designated output signals in sequential charts. If a signal is in the feedback set then whenever that signal is output by the chart it is also instantaneously available as input. If we consider the input / output behaviour that a sequential chart describes as a back box that has a wire representing input and a wire representing output then the feedback of signals can be considered as a loop-back wire that connects the output to the input and carries only the designated signals.

In the graphical representation of charts the feedback set is presented in a box attached to the outline that encloses the chart. Because sequential charts often have no associated feedback, *i.e.* their feedback set is empty, the graphical representation, as demonstrated in Figure 1(a), allows such simple sequential charts to be drawn without any feedback box attached. On the other hand Figure 2 is an example of a sequential chart $AChart\_a\_b$ that feeds back all of its signals, *i.e.* it has the feedback set containing in_a and out_b.

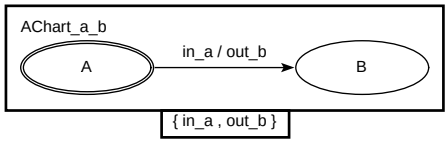


Fig. 2.

2.3 Composition of μ-charts

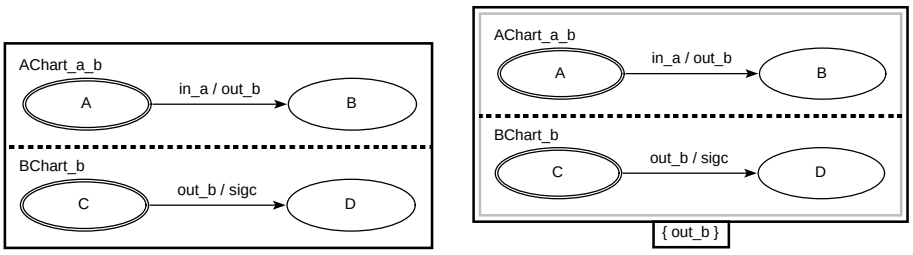
Now that we have introduced the basic building block of the language, *i.e.* sequential charts, we present some operators that can be used to combine arbitrary charts into more sophisticated μ-charts.

Given any two charts $C_1 \in \mu Charts$ and $C_2 \in \mu Charts$ and a set of signals Ψ we denote the composition of these charts in parallel by an expression of the form $C_1 \mid \Psi \mid C_2$. The set of all such charts formed using the composition operator is defined by the set $\mu CompChart$.

$$\mu CompChart \quad ::= \quad \mu Charts \mid \{\ \textit{signal-list}\ \} \mid \mu Charts$$

From this definition we can see, as one might expect, the feedback of signals is intimately coupled with the composition of charts. Hence when we compose two charts the signals that they are able to communicate on are included as part of the composition operator. The feedback set can be empty, in which case the composition would represent two charts that proceed in lock-step but do not affect one another in any way.

The graphical notation again uses two conventions for the treatment of feedback between composed charts. Consider the two composed charts in Figure 3.



(a) (b)

Fig. 3.

The chart in Figure 3(a) represents the chart $AChart\_a\_b \mid \{\} \mid BChart\_b$ where $AChart\_a\_b$ is $(AChart\_a\_b,\ \{A, B\},\ A,\ \{\},\ \{(A, B, in\_a/out\_b)\})$ and $BChart\_b$ is similarly defined. The chart of Figure 3(b) represents the chart $AChart\_a\_b \mid \{out\_b\} \mid BChart\_b$.

2.4 The decomposition of a μ-chart

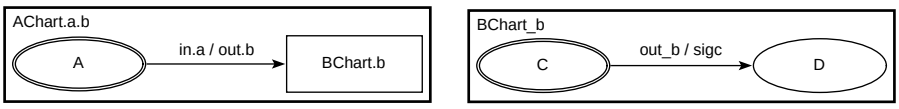
A decomposed μ-chart refers to a sequential chart that has other charts embedded into some or possibly all of its states. The purpose of decomposing a state in a sequential chart is to mimic a master / slave relationship between the charts. The sequential chart that has its states decomposed can be considered the master in the relationship because it effectively allows the decomposing chart, *i.e.* the slave, the ability to react to input from the environment only when the master is in the state that the slave decomposes. Of course a master can have several slaves. A slave, however, has only one master which is always a sequential chart. The master may itself be a slave, *i.e.* there can be more than one level of decomposition.

The syntax for decomposition is defined by the following grammar.

$$\begin{array}{lll} \mu DecChart & ::= & \textbf{Dec}\ \mu Chart\ \textbf{by}\ \{\ dec\text{-}states\ \} \\ dec\text{-}states & ::= & (\ State\ ,\ \mu Charts\ )\ [,\ dec\text{-}states\ ] \end{array}$$

In the graphical representation of a decomposed chart the decomposed states of the master are written as rectangles (rather than the usual oval state representation). The slave chart is then given in a separate chart diagram. Figure 4 presents an example of a decomposed chart given by the textual expression

$$Dec\ (AChart\_a\_b,\ \{A, BChart\_b\},\ A,\ \{(A, BChart\_b, in\_a/out\_b)\})\ by \\ (BChart\_b\ ,\ (BChart\_b,\ \{C, D\},\ C,\ \{(C, D, out\_b/sigc)\}))$$



(a) Master (b) Slave

Fig. 4.

2.5 The signal hiding operator

The final chart operator that we introduce is the hiding operator. As we discussed above the basic building blocks or parts of a μ-chart are sequential charts. A complex specification is (in general) constructed by combining several sub-charts using the operators that we have already introduced in this section. We can consider each sub-chart as sharing an observable interface with its environment. For example consider the composition (with feedback) of the sequential charts $AChart\_a\_b$ and $BChart\_b$ pictured in Figure 3(b). The "effective" environment, *i.e.* what a chart sees and allows to be seen, for the respective charts $AChart\_a\_b$, $BChart\_b$ and $AChart\_a\_b \mid \{out\_b\} \mid BChart\_b$ are all different things. It is usually the case that we wish to distinguish between the way that one part of the specification, *e.g.* $AChart\_a\_b$, interacts with its environment and the way that the overall specification, *e.g.* $AChart\_a\_b \mid \{out\_b\} \mid BChart\_b$ interacts with its environment. For instance it may be the case that in this example we want the signal out_b to be used only for internal communication between $AChart\_a\_b$ and $BChart\_b$. Hence the interface that these charts share individually must contain signal out_b, however we want to hide this signal from the observable interface of the specification $AChart\_a\_b \mid \{out\_b\} \mid BChart\_b$. This is the purpose of the hiding operator.

As we have already seen, there are in fact two interfaces for any chart, the input interface and the output interface. The hiding operator allows us to restrict either or both of these sets of signals. Restricting the input interface can be considered as filtering the input from the environment, *i.e.* the specified system does not react to the signals that are filtered. And restricting the output interface means the hidden signals are not observable as output from the specified reactive system. We will refer to both of these as signal hiding and distinguish the two using input or output hiding where necessary.

The set of syntactically correct charts that contain hiding are given by $\mu ResIChart$ which is defined as follows.

$$
\begin{array}{lll}
\mu ResIChart & ::= & {}_{[filter\text{-}signals]} \lfloor\!\lfloor\, \mu Charts \,\rfloor\!\rfloor_{[hide\text{-}signals]} \\
filter\text{-}signals & ::= & \{\, [signal\text{-}list] \,\} \\
hide\text{-}signals & ::= & \{\, [signal\text{-}list] \,\} \\
signal\text{-}list & ::= & Signal\ [,\ signal\text{-}list]
\end{array}
$$

Notice that both the set of input signals hidden from the environment (on the left) and the set of hidden output signals (right) are optional. This allows the obvious shorthand notation such that $_X[\![C]\!] = {}_X[\![C]\!]_\varnothing$ and $[\![C]\!]_Y = {}_\varnothing[\![C]\!]_Y$.

The graphical notation includes the sets of filtered and hidden signals in the box attached for feedback. The feedback box for a chart that has signals hidden can be parsed as "signals filtered from input [feedback] signals hidden from output". For example the chart $_{out\_b}[AChart\_a\_b \mid \{out\_b\} \mid BChart\_b]$ pictured in Figure 5 hides the signal *out_b* from the input interface of the chart. This means that the only way that chart *BChart_b* can be affected by the input signal *out_b* is if that signal is output from the chart *AChart_a_b*. Note however, that the signal *out_b* is still observable output from this specification.

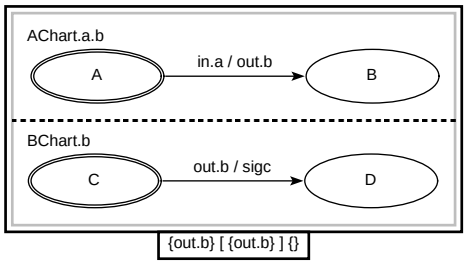


Fig. 5.

Because it is common to describe charts with no hidden signals, a shorthand notation in the graphical representation for such charts is to give just the feedback set. Hence any chart that has one set in its feedback box, say the set F, is assumed to be shorthand for the expression $\varnothing[F]\varnothing$.

2.6 Auxiliary semantic functions

As for sequential charts we often define the useful auxiliary function

$$\omega : Chartname \nrightarrow \mu Charts$$

for more complex charts. We often introduce a complex μ-Charts specification and give it a simpler name, for example the expression $C_{12} = C_1 \mid F \mid C_2$ defines the function ω such that $\omega\ C_{12} = C_1 \mid F \mid C_2$. Again we generally use the name C_{12} to refer to the chart $\omega\ C_{12}$.

Finally, we give the general definition for two of the syntactic functions, *in* and *out*, that were introduced for sequential charts (as in_s and out_s) in Section 2.1.

$$in : \mu Charts \to \mathbb{P}\, \mu Signal$$

$$\begin{aligned}
&in\ (C, \Sigma, \sigma, \delta) = in_s\ (C, \Sigma, \sigma, \delta) \\
&in\ (C, \Sigma, \sigma, \delta, I) = I \\
&in\ (C \mid feedbackset) = in\ C \\
&in\ (C_1 \mid feedbackset \mid C_2) = in\ C_1 \cup in\ C_2 \\
&in\ (Dec\ C\ by\ Slaves) = \\
&\qquad in\ C \cup (\bigcup \{(\lambda(sn, Slave) \bullet in\ Slave) \mid (sn, Slave) \in Slaves\}) \\
&in\ ({}_{hidden}[C]_) = in\ C \setminus hidden
\end{aligned}$$

$$out : \mu Charts \to \mathbb{P}\, \mu Signal$$

$$\begin{aligned}
&out\ (C, \Sigma, \sigma, \delta) = out_s\ (C, \Sigma, \sigma, \delta) \\
&out\ (C \mid feedbackset) = out\ C \\
&out\ (C_1 \mid feedbackset \mid C_2) = out\ C_1 \cup out\ C_2 \\
&out\ (Dec\ C\ by\ Slaves) = \\
&\qquad out\ C \cup (\bigcup \{(\lambda(sn, Slave) \bullet out\ Slave) \mid (sn, Slave) \in Slaves\}) \\
&out\ (_[C]_{hidden}) = out\ C \setminus hidden
\end{aligned}$$

3 Language semantics

Our formal semantics for μ-Charts is given by a method for constructing a Z model for an arbitrary chart. The kernel logic $\mathcal{Z}_C$ for Z, presented in [4], gives the Z model a meaning grounded in typed set theory. Hence, the combination of these gives us a semantics for μ-Charts grounded in typed set theory.

As we have already seen a μ-chart that specifies the allowable behaviour of a reactive system can be (and in most interesting cases is) made up by using language operators to combine simple sequential charts into more complex specifications. We present the semantic by giving an informal account of the meaning of the language constructs in parallel with describing the general method for creating the Z model and hence the formal semantics.

A simplifying assumption made in the semantics for μ-charts is that all constituent sequential sub-charts, that are combined to describe a reactive system, proceed in lock-step. When a step happens is determined by the environment in which the chart resides. That is, when the environment produces input each sequential chart makes a transition. Notice that we distinguish between an input signal and input (or an input event), which is a set of input signals.

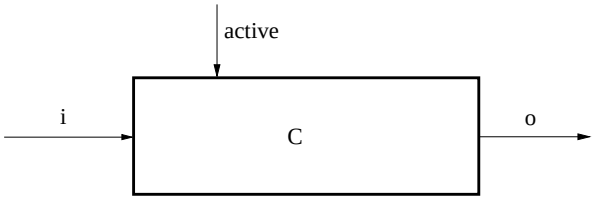
Initially, we give a step semantics for charts. This describes the behaviour of a chart in terms of the output that the chart produces in response to input from the environment assuming a given state. The step semantics essentially relates the current configuration of a chart[3] and input to a new configuration and the resulting output. This relation describes every possible step that a chart can take.

We will show later how this step semantics can be lifted to give a trace semantics that abstracts away the information about the charts configuration. This gives the meaning of charts in terms of sets of observable input / output traces that an implementation of the specification can exhibit over time. The trace semantics is given primarily to relate the account of μ-Charts refinement given here back to the previous work describing refinement. The notion of chart refinement presented here is characterized in terms of the step semantics. Hence it is not strictly necessary to have the trace semantics at all, though trace semantics helps to give a more intuitive understanding of the resulting refinement relation.

Building the Z model for charts requires two separate tasks. The first part of the process is to create a general model for each of the separate constituent sub-charts in the μ-chart being described. We call this the transition model. As one might expect given the language, this process is recursive in nature and follows exactly the structure of the chart being modeled. The overall transition model is built up by initially describing the model for each of the sequential sub-charts followed by the description of each of the new sub-charts created using the language operators. Note this implies that the final model describes the largest

[3] The term configuration is used to refer to the "state" of a chart because a μ-chart that has several sequential sub-charts can be in several states at the same time, in other words each sequential sub-chart has its own current state.

sub-chart which is itself the chart given as the overall description of the reactive system of interest. The modular nature of the language is such that the model of an arbitrary sub-chart can be consider as a "black-box". The following diagram demonstrates both the informal "circuit diagrams" that we use to motivate the semantic description and the "black-box" characteristics of the transition model for an arbitrary chart C.



The circuit diagram gives the structure of the general transition model for a chart. That is, we can consider the model of any chart with arbitrary structure as a circuit that has two inputs and one output. The chart model defines a function of output given the two inputs.

Sections 3.1 to 3.6 give both an informal description and a formal treatment that introduces the transition model for each of the language constructs. The formal treatment has two parts. The first gives a generic process for creating a Z model for an arbitrary μ-chart. The Z description of the chart can be used in the standard Z fashion to investigate the behaviour of the chart. Using available proof assistant and Z animation tools is a useful way to perform some validation of the μ-chart's behaviour. The second part of the formal treatment uses the meaning of the Z itself, via the Z logic of [4], to give a logic for reasoning about μ-charts. This logic is presented in terms of introduction and elimination rules for each of the language constructs. It can be used to reason about the model of a μ-chart. The rules are presented in a natural deduction style, closely following the work of Deutsch, Henson and Reeves [2, 3].

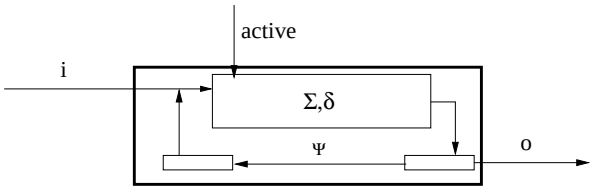
The final step in building the model for a μ-chart is to hide the machinery that is present in the transition model, that is the machinery that is necessary to model the inherent structure of charts. We call this final "top-level" model the step semantics for a chart. Section 3.7 describes this process and then investigates some of the properties that hold of the model.

3.1 Sequential charts

A sequential chart is essentially a finite state automaton that describes the set of output signals that results from reacting to a set of inputs in a given state. The general transition model of a sequential chart has two notions of state. The first being the usual automaton notion of state which is determined by the transitions that have happened due to input since the automaton was initialized. This state is denoted in a chart diagram by labeled ovals or rectangles. The second notion of state represents whether or not the sequential chart is active. The reason that a chart can be active or inactive is because it may be the slave of

another chart in the specification via decomposition. Finally, a sequential chart can instantaneously feed back output signals. This means that, during any step of the chart, fed back output is also considered as input.

The structure of the transition model of a sequential chart $(C, \Sigma, \sigma, \Psi, \delta)$ is demonstrated by the following circuit diagram.



This diagram can be separated into three logical parts as described above. The finite state automaton specification of the input/output relationship is denoted by the component labeled Σ, δ. The active / inactive state mechanism is denoted by the input labeled *active*. And the feedback mechanism is represented by the line labeled Ψ.

Consider the chart $(C, \{A, B\}, A, \{\}, \delta)$ (where δ is the appropriate transition description) pictured in Figure 6.

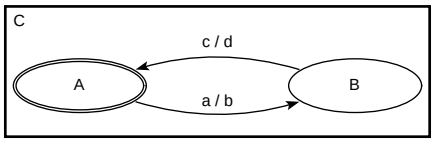


Fig. 6. A simple sequential μ-chart

We can clearly see the finite state description of the input and output behaviour from this typical sequential μ-chart diagram. Informally the behaviour that this chart captures can be described as: the chart starts in state A; if it is in state A and the signal a is input then signal b is output and the chart changes to state B; and similarly if it is in state B and c is in the input then d is output and the new state is A. Notice that we have not yet described what happens when the actions on the transitions are not satisfied. We defer this discussion until after we introduce the Z semantics.

The ability of a sub-chart to be active or inactive is necessary for charts that are embedded (*i.e.* slaves of) some other chart in a decomposition. When a chart is active it reacts to input as specified by its finite state description. When it is inactive it ignores input, remains in the same state and gives no output. In fact, no output actually means the output is the empty set of signals.

The behaviour of the feedback mechanism is uninteresting in this example because there are no signals fed back. However, in general, a sequential chart can feed back its output signals which then instantaneously act as input. This is represented in the circuit diagram by the link labeled Ψ that creates a loop

between output and input and is assumed to carry only the specified feedback signals. We discuss in more detail exactly what feedback in sequential charts means semantically in Section 3.3.

3.2 The Z transition model for sequential μ-Charts

The essence of the general transition model for sequential μ-charts is the description of each of the transitions in a sequential chart as a separate Z operation schema. Each of these operation schemas (one for each transition in the chart) are combined using Z schema disjunction to give one schema that describes the abstract transition behaviour of the sequential chart. The Z state of the model has an observation that determines the current state of the chart. The operation schema describing a transition describes how and when that state changes. We proceed by introducing the Z that is given as the transition semantics of a chart and then discussing the meaning of that Z.

For a general sequential chart $(C, \Sigma, \sigma_0, \Psi, \delta)$ we introduce the following definitions (left) that encode the chart's states, input interface and output interface in Z.

$$
\begin{array}{|l}
states_C : \mathbb{P}\, \mu_{State} \\
in_C : \mathbb{P}\, \mu Signal \\
out_C : \mathbb{P}\, \mu Signal \\
\Psi : \mathbb{P}\, \mu Signal \\
\hline
states_C = \Sigma \\
in_C = in\ C \\
out_C = out\ C
\end{array}
\qquad
Chart_C == [c_C : states_C]
$$

$$
\begin{array}{|l}
\multicolumn{1}{l}{Init_C} \\
\hline
Chart_C \\
\hline
c_C = \sigma_0
\end{array}
$$

The general state schema $Chart_C$ (top right) models the automaton state for the chart C. And the initial state of the chart is modeled by the schema $Init_C$ (bottom right).

A separate state schema is also given for each automaton state in the chart. So for all $\sigma \in \Sigma$ there exists a schema such that

$$C\sigma == [Chart_C \mid c_C = \sigma]$$

Now we give an operation schema for each transition in the chart. That is for all $(S_f, S_t, guard/action) \in \delta$ we define an operation schema that has the following structure.

$\delta_{S_f S_t}$

CS_f
CS'_t
$i_C? : \mathbb{P}\, in_C$
$active\_ : \mathbb{P}\, \mu_{State}$
$o_C! : \mathbb{P}\, out_C$

$active(C)$
$\rho(guard)$
$o_C! = action$

This general operation schema introduces a new syntactic function ρ that produces the Z predicate that models the syntactic representation of a transition's guard. If we consider a transition's guard in general as a (possibly empty) list of signal expressions separated by the symbol & then each of the elements in the list can be classified into two categories: either a positive signal expression—simply the name of a signal; or a negative signal expression—the signal name is prepended with a minus sign. A positive signal expression, say *sig* where $sig \in \mu Signal$, is denoted by the Z expression $sig \in i_C? \cup (o_C! \cap \Psi)$. A negative signal expression, say $-sig$, is denoted by the Z expression $sig \notin i_C? \cup (o_C! \cap \Psi)$. The function ρ performs this transformation on each signal expression in the list and connects the results back together using logical conjunction. If the list is empty the predicate produced by ρ would be *true*.

This general scheme for giving the Z for a transition defines the semantic function $[\![.]\!]_{Z_t}$ such that for an arbitrary transition $(S_f, S_t, guard/action)$

$$[\![(S_f, S_t, guard/action)]\!]_{Z_t} =_{def} \delta_{S_f S_t}$$

where the schema $\delta_{S_f S_t}$ results from the method described above.

Along with the schemas for each transition we also need a single schema that models the behaviour of the chart when it is inactive. Again for the general sequential chart $(C, \Sigma, \sigma_0, \Psi, \delta)$, the inactive schema is given as follows.

$Inactive_C$

$\Xi Chart_C$
$i_C? : \mathbb{P}\, in_C$
$active\_ : \mathbb{P}\, \mu_{State}$
$o_C! : \mathbb{P}\, out_C$

$\neg\ active(C)$
$o_C! = \{\}$

The transition semantic of a sequential chart can now be given by the following definition.[4]

$$[\![(C, \Sigma, \sigma_0, \Psi, \delta)]\!]_{Z_t} =_{def} \delta_C$$

[4] We use the notation $\bigvee X$ to denote the schema disjunction of all the schemas in the set X.

where

$$\delta_C == (\bigvee \{ [\![t]\!]_{Z_t} \mid t \in \delta \}) \vee Inactive_C$$

This concludes the presentation of the Z that is used to give the transition model for a sequential chart. We now consider the meaning of this Z in terms of the particular example that is pictured in Figure 6.

The following Z schemas result from giving the transition model of this chart.[5]

δ_{AB}

CA
CB'
$i_C? : \mathbb{P}\, in_C$
$active\_ : \mathbb{P}\, \mu_{State}$
$o_C! : \mathbb{P}\, out_C$

$active(C)$
$a \in i_C? \cup (o_C! \cap \Psi)$
$o_C! = \{b\}$

$Inactive_C$

$\Xi Chart_C$
$i_C? : \mathbb{P}\, in_C$
$active\_ : \mathbb{P}\, \mu_{State}$
$o_C! : \mathbb{P}\, out_C$

$\neg\ active(C)$
$o_C! = \{\}$

The transition in chart C from state A to B results in the Z schema δ_{AB}. Rather than describing the schema in terms of its syntax we proceed directly to describing its meaning that is given by the set $[\![\delta_{AB}]\!]$. In the theory $\mathcal{Z}_C$ [4] the meaning of a schema is given as a set of bindings. A binding is a mapping between the labeled observations l_i and their respective values t_i and is written as $\langle\!|\ ...l_i \Rrightarrow t_i ...\ |\!\rangle$.

Hence the meaning of the schema δ_{AB} is given by the following set comprehension.

$$[\![\delta_{AB}]\!] = \{\langle\!|\ c_C \Rrightarrow A,\ i_C? \Rrightarrow i,\ active\_ \Rrightarrow active,\ c'_C \Rrightarrow B,\ o_C! \Rrightarrow \{b\}\ |\!\rangle \mid i \subseteq in_C \wedge active \subseteq \mu_{State} \bullet C \in active \wedge a \in i\}$$

We can see from this definition that each binding in the semantic set has five labeled observations. The meaning of these are:

- c_C—the state of the chart before the transition happens, in this case the state A
- $i_C?$—the set of input signals which are offered by the environment that are in the input interface of the chart
- $active\_$—a set that denotes all currently active charts
- c'_C—the state of the chart after the transition happens, in this case state B
- $o_C!$—the output generated by this sequential chart, in this case the set containing the signal b

[5] We assume that all of the entities given in the presented Z whose definitions are omitted have the appropriate type and obvious meaning.

Hence each transition description is "parameterized" on the current state of the chart c_C, the input from the environment $i_C?$ and whether or not this chart is active. If the precondition of the schema holds, *i.e.* the chart is active and the transition is triggered, then the output contributed by this sequential chart is defined by the set $o_C!$. Note that in the schema itself the expression $(o_C! \cap \Psi)$ models the feedback mechanism for the sequential chart. In this case $\Psi = \{\}$, that is the feedback set is empty. A similar schema to δ_{AB}, that models the transition from state B to A called δ_{BA}, would also be given.

The second schema $Inactive_C$ describes the behaviour of the sequential chart when it is inactive. We can see from the set of bindings

$$\begin{aligned}&[\![Inactive_C]\!] = \\ &\quad \{(\!|\, c_C \Rrightarrow s,\ i_C? \Rrightarrow i,\ active\_ \Rrightarrow active,\ c'_C \Rrightarrow s,\ o_C! \Rrightarrow \{\}\ |\!) \mid \\ &\qquad\qquad s \in \{A, B\} \land active \subseteq \mu_{State} \land i \subseteq in_C \bullet C \notin active\}\end{aligned}$$

that the schema $Inactive_C$ faithfully models the inactive behaviour of the chart as we described above.

Now the complete transition model for chart C is defined as the disjunction of each of the individual transition schemas to be $\delta_C == \delta_{AB} \lor \delta_{BA} \lor Inactive_C$. Because of the definition of schema disjunction the set $[\![\delta_C]\!]$ contains all of the bindings from the sets $[\![\delta_{AB}]\!]$, $[\![\delta_{BA}]\!]$ and $[\![Inactive_C]\!]$.

This method of investigating the semantics of the transition model for charts, that is considering the meaning of schemas as sets of bindings, is somewhat informative in the simple example presented above. However, in general we do not want to be forced to resort to complex set definitions and manipulations to reason about the meaning of charts. Therefore, as discussed previously, we give a set of rules that characterise and allow us to reason about the Z semantic model for charts. These and other rules, presented later, give us a logic for charts.

Firstly, we formalize what it means for a binding of the transition model to satisfy a specific (syntactic) transition of the chart. Given an arbitrary transition of the form $t = (S_f, S_t, guard/action)$, from the chart C, where $[\![[\![t]\!]_{Z_t}]\!]^{\mathbb{P}\, T}$ we have,

$$\begin{aligned}Trans\ t\ z\ =_{def}\ & z.c_C = t.S_f \land \rho(t.guard)[\alpha T / z.\alpha T] \land \\ & z.c'_C = t.S_t \land z.o_C! = t.action\end{aligned}$$

The terms $t.S_f$ *etc.* are assumed to be defined in the obvious way such that $t.S_f$ gives the "from state" of a transition, $t.guard$ gives the guard component of a transition, $t.S_t$ gives the "to state" and $t.action$ returns the action component. The function ρ is as defined above.

Now we give the formal definition of the transition model for charts directly in terms of the meaning of the Z model. For the arbitrary sequential chart $(C, \Sigma, \sigma_0, \Psi, \delta)$, we have,[6]

[6] We make the assumption that whenever a meta-variable is used to define the type of the transition model for a chart, *e.g.* T, it represents the type of the schema as

$$[\![\delta_C]\!]^{\mathbb{P}\,T} =_{def} \{z \in T \mid C \notin z.active\_ \land z \mathrel{\dot\in} \Xi[c_C : \Sigma] \land z.o_C! = \{\} \lor C \in z.active\_ \land \exists t \in \delta \bullet Trans\ t\ z\}$$

From this definition we derive the following introduction and elimination rules[7].

$$\frac{z \in \delta_C \quad active\ C\ z \quad Trans\ t\ z \vdash P}{P}\ (Z_t\,\bar{}_1) \qquad \frac{z \in \delta_C \quad inactive\ C\ z}{z \mathrel{\dot\in} \Xi Chart_C}\ (Z_t\,\bar{}_2)$$

$$\frac{z \in \delta_C \quad inactive\ C\ z}{z.o_C! = \{\}}\ (Z_t\,\bar{}_3)$$

$$\frac{active\ C\ z \quad Trans\ t\ z}{z \in \delta_C}\ (Z_t\,^+_1)$$

$$\frac{inactive\ C\ z \quad z \mathrel{\dot\in} \Xi Chart_C \quad z.o_C! = \{\}}{z \in \delta_C}\ (Z_t\,^+_2)$$

where we assume the usual conditions for t and P.

Proofs for each of these rules are trivial given the definition $[\![\delta_C]\!]$. The predicates $active\ C\ z$ and $inactive\ C\ z$ are defined as,

$$active\ C\ z =_{def} C \in z.active\_$$
$$inactive\ C\ z =_{def} \neg\ active\ C\ z$$

for sequential charts.

Given these rules we can start to investigate some of the properties of the transition model. In particular, the following propositions allow us to reason about which bindings are in the transition model and which are not.

Firstly, we make the observation that the transition model given makes a distinction between two sources of input that can contribute to the signals that trigger a transition. The external input to the chart generated by the environment $i_C?$ and the fed back output from the chart itself, *i.e.* for the chart $(C, \Sigma, \sigma_0, \Psi, \delta)$ the set denoted by $o_C! \cap \Psi$. We formalize this observation in the following proposition.

Proposition 1. *For the arbitrary sequential chart* $(C, \Sigma, \sigma_0, \Psi, \delta)$, $z \in T$ *and* $x \in T$,

$$\frac{z \in \delta_C \quad x \upharpoonright T_i \doteq z \quad z.i_C? \cup fb\ z = x.i_C? \cup fb\ x}{x \in \delta_C}\ (Z_t^{\in})$$

where $[\![\delta_C]\!]^{\mathbb{P}\,T}$ *and* $T_i = T - [i_C? : \mathbb{P}\, in_C]$

given in the Z model rather than the type of the schema after normalisation. For example the type $T = [c_C, c'_C : states_C;\ i_C? : \mathbb{P}\, in_C;\ active\_ : \mathbb{P}\, \mu_{State};\ o_C! : \mathbb{P}\, out_C]$ rather than the alternative in which the observation $i_C?$ would have the normalised type $\mathbb{P}\,\mu Signal$ *etc.*.

[7] Context permitting we omit the semantic brackets $[\![.]\!]$, so $z \in \delta_C$ is shorthand for $z \in [\![\delta_C]\!]$.

The function fb is a shorthand for representing the feedback set that is applicable to a chart. For the sequential chart C and the arbitrary binding $z \in T$, as above,

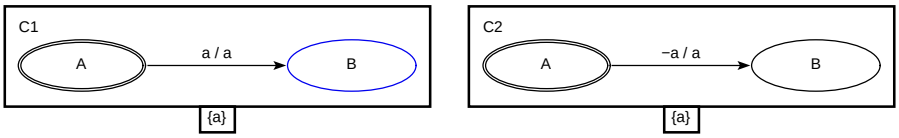
$$fb\, z \;=_{def}\; z.o_C! \cap \Psi$$

Here we proved the rule $(Z_i^{\in})$ just for sequential charts. In the following sections we prove the same rule for charts that are built using each of the chart operators. Hence these rules hold for charts in general regardless of their structure.

3.3 Feedback of signals in sequential μ-Charts

The semantics of feedback in sequential charts is encoded in the transition model as given in the previous section. In this section we give two pathological examples that demonstrate exactly the semantics of sequential charts with feedback.

The examples $C_1 = (C1, \Sigma, \sigma, \{a\}, \delta_1)$ and $C_2 = (C2, \Sigma, \sigma, \{a\}, \delta_2)$ (for appropriate Σ, σ, δ_1 and δ_2) are given in Figure 7. The only difference between these two μ-charts is that the transition in the chart C_1 is triggered by the presence of the signal a whereas the transition in chart C_2 is triggered by the absence of the signal a. Both examples output and feed back a.



(a) (b)

Fig. 7.

The encoding of feedback is present in the schemas that describe the respective transitions in these charts.

δ^1_{AB}

$C1A$
$C1B'$
$i_{C_1}? : \mathbb{P}\, in_{C_1}$
$active\_ : \mathbb{P}\, \mu_{State}$
$o_{C_1}! : \mathbb{P}\, out_{C_1}$

$active(C1)$
$a \in i_{C_1}? \cup (o_{C_1}! \cap \Psi)$
$o_{C_1}! = \{a\}$

δ^2_{AB}

$C2A$
$C2B'$
$i_{C_2}? : \mathbb{P}\, in_{C_2}$
$active\_ : \mathbb{P}\, \mu_{State}$
$o_{C_2}! : \mathbb{P}\, out_{C_2}$

$active(C2)$
$a \notin i_{C_2}? \cup (o_{C_2}! \cap \Psi)$
$o_{C_2}! = \{a\}$

Given $\Psi = \{a\}$, the respective predicates $a \in i_{C_1}? \cup (o_{C_1}! \cap \Psi)$ and $a \notin i_{C_2}? \cup (o_{C_2}! \cap \Psi)$ mean that both of the transitions' guards rely not only on the environment's input, *e.g.* $i_{C_1}?$, but also on the fed back output, *e.g.* $(o_{C_1}! \cap \{a\})$.

Now we can state and prove lemmas that describe exactly the meaning of the two examples.

Firstly, for the chart pictured in Figure 7(a), we show that the transition from state A to B happens regardless of the input from the environment. For simplicity we make the assumption that the charts are active.

Lemma 1. *Given* $\delta_{C_1} = [\![(C_1, \Sigma, \sigma, \Psi, \delta_1)]\!]_{Zt}$, $\Sigma = \{A, B\}$, $\sigma = A$, $\Psi = \{a\}$ *and* $\delta_1 = \{(A, B, a/a)\}$, *for arbitrary* $z \in T$ *and input* $i \subseteq in_C$ *we have,*

$$\frac{active\ C_1\ z \quad z \doteq \langle\!|\ c_{C_1} \Rrightarrow A i_{C_1}? \Rrightarrow i, c'_{C_1} \Rrightarrow B, o_{C_1}! \Rrightarrow \{a\}\ |\!\rangle}{z \in \delta_{C_1}}$$

where $[\![\delta_{C_1}]\!]^{\mathbb{P}\,T}$

Hence, the meaning of the chart C_1 is that the transition from state A to state B always happens in the first step of the chart, that is regardless of the input that the environment offers, and the signal a is output. This also implies that this chart has identical behaviour to a similar chart in which the transition is labeled $/a$, *i.e.* a true trigger.

Likewise, we show that the transition model for the chart C_2 pictured in Figure 7(b) does not contain any bindings that model the transition from state A to B. This is because the single transition in chart C_2 is inconsistent in the presence of feed back on the signal a under any input. Essentially, the output invalidates the trigger of the transition, hence the transition can never happen. This is expressed by the following lemma.

Lemma 2. *Given* $\delta_{C_2} = [\![(C2, \Sigma, \sigma, \Psi, \delta_2)]\!]_{Zt}$, $\Sigma = \{A, B\}$, $\sigma = A$, $\Psi = \{a\}$ *and* $\delta_2 = \{(A, B, -a/a)\}$, *for arbitrary* $z \in T$,

$$\frac{active\ C_2\ z}{z \notin \delta_{C_2}}$$

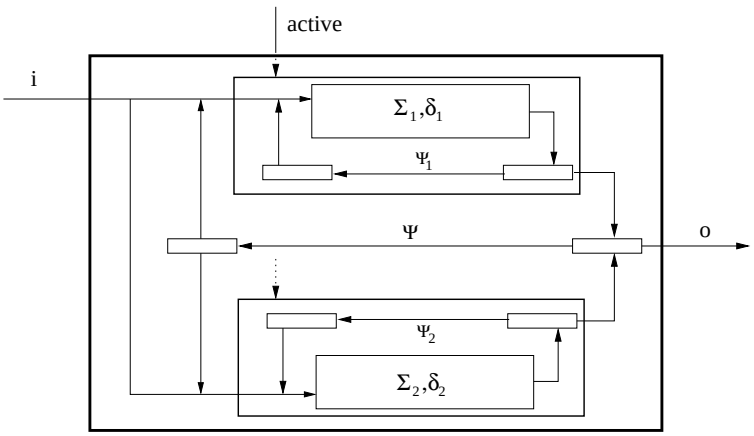
where $[\![\delta_{C_2}]\!]^{\mathbb{P}\,T}$

In fact in this simple example the transition model is the empty set of bindings because the inconsistent transition is the only candidate to be modeled. In general any inconsistent transitions can be removed from the chart without changing its behaviour.

3.4 The composition operator

The composition operator allows us to take two μ-charts C_1 and C_2 and join them together to form a new more complex chart $C_1 \mid \Psi \mid C_2$ where Ψ is a set of signals. As mentioned, we assume that the charts run separately but synchronously, *i.e.* in lock step with one another. Their only medium of communication is asynchronous via the multicast of signals. The set Ψ denotes the signals that the charts C_1 and C_2 can communicate on. The communication is asynchronous in that output is always enabled, that is a chart can always broadcast signals, however there is no guarantee that the other chart in the composition is listening, in other words ready to react on the signals broadcast. Signals persist only during one step of the chart.

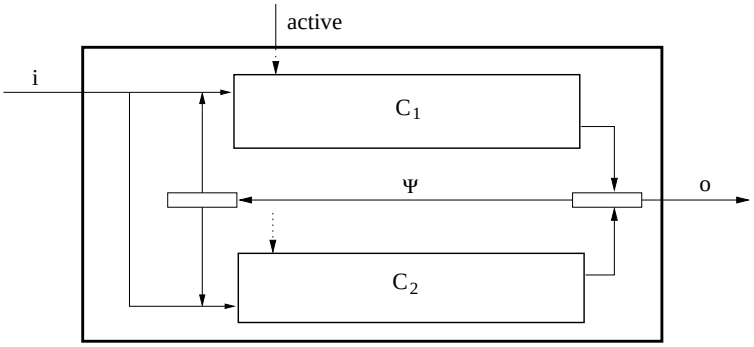
The following diagram demonstrates the structure of the composed chart $(C1, \Sigma_1, \sigma_1, \Psi_1, \delta_1) \mid \Psi \mid (C2, \Sigma_2, \sigma_2, \Psi_2, \delta_2)$.



Notice that any signals that the sets Ψ and Ψ_1 (respectively Ψ and Ψ_2) have in common will be fed back on two separate paths in this diagram. Also, the composition operator not only allows C_1 to communicate with C_2 using the signals in Ψ but effectively changes the feedback characteristics of C_1 itself. The output that C_1 treats as input via feedback is now all of the signals in $\Psi_1 \cup \Psi$.

Both charts in the composition have an equivalent active state, *i.e.* one chart is active if and only if the other is active. In the circuit diagram we use dotted lines as a shorthand to indicate that the single active input to the composition is linked to both of the active inputs from the parts of the composition.

This example of the composition operator assumes that C_1 and C_2 are sequential charts. The operator in general however is defined over arbitrary charts. Hence the general picture for a composed chart $C_1 \mid \Psi \mid C_2$, where C_1 and C_2 are any charts, is as follows.



The transition model for composed charts is constructed by recursively constructing transition models for the two constituent parts of the composition. The base case being when the parts are themselves sequential charts. The model of the parts are then combined to create the transition model for the composition.

The transition model for the composed chart $C = (C_1 \mid \Psi \mid C_2)$ contains the following Z definitions and schemas.

$$
\begin{array}{|l}
states_C : \mathbb{P}\, \mu_{State} \\
in_C : \mathbb{P}\, \mu Signal \\
out_C : \mathbb{P}\, \mu Signal \\
\Psi : \mathbb{P}\, \mu Signal \\
\hline
states_C = states_{C_1} \cup states_{C_2} \\
in_C = in_{C_1} \cup in_{C_2} \\
out_C = out_{C_1} \cup out_{C_2}
\end{array}
$$

$$
\begin{array}{|l}
\hline
\delta_C \\
\Delta Chart_C \\
i_C? : \mathbb{P}\, in_C \\
active\_ : \mathbb{P}\, \mu_{State} \\
o_C! : \mathbb{P}\, out_C \\
\hline
active(C_1) \Leftrightarrow active(C_2) \\
\exists\, i_{C_1}?, i_{C_2}?, o_{C_1}!, o_{C_2}! : \mathbb{P}\, \mu Signal \bullet \\
\quad i_{C_1}? = (i_C? \cup (o_C! \cap \Psi)) \cap in_{C_1} \wedge \\
\quad i_{C_2}? = (i_C? \cup (o_C! \cap \Psi)) \cap in_{C_2} \wedge \\
\quad o_C! = o_{C_1}! \cup o_{C_2}! \wedge \\
\quad \delta_{C_1} \wedge \delta_{C_2} \\
\hline
\end{array}
$$

$$
\begin{array}{|l}
\hline
Chart_C \\
Chart_{C_1} \\
Chart_{C_2} \\
\hline
\end{array}
$$

$$
\begin{array}{|l}
\hline
Init_C \\
Init_{C_1} \\
Init_{C_2} \\
\hline
\end{array}
$$

This Z makes the obvious assumption that any entity subscripted with C_1 comes from the transition model of the chart C_1 and similarly for C_2. Importantly, the bindings that inhabit the set $[\![\delta_C]\!]$ (*i.e.* the semantics of the schema δ_C) have a similar signature to those in both $[\![\delta_{C_1}]\!]$ and $[\![\delta_{C_2}]\!]$. This shows that the Z model is consistent with the modular nature of charts demonstrated by the informal circuit diagrams.

As with sequential charts we give the definition of the transition model for composed charts directly in terms of the meaning of the Z model. Given an arbitrary composed chart $C = C_1 \mid \Psi \mid C_2$ we have,

$$
\begin{aligned}
[\![\delta_C]\!]^{\mathbb{P}\,T} =_{def} \{z \in T \mid \exists\, z_1 \in \delta_{C_1};\ z_2 \in \delta_{C_2} \bullet\ & z \doteq z_1 \wedge z \doteq z_2 \wedge \\
& z_1.i_{C_1}? = (z.i_C? \cup (z.o_C! \cap \Psi)) \cap in_{C_1} \wedge \\
& z_2.i_{C_2}? = (z.i_C? \cup (z.o_C! \cap \Psi)) \cap in_{C_2} \wedge \\
& z.o_C! = z_1.o_{C_1}! \cup z_2.o_{C_2}! \wedge \\
& \mathit{active}\ C_1\ z \Leftrightarrow \mathit{active}\ C_2\ z\}
\end{aligned}
$$

From this definition we derive the following introduction and elimination rules for composed charts.

Given $C = C_1 \mid \Psi \mid C_2$ and assuming the usual conditions for z_1, z_2 and Q we have,

$$\frac{z \in \delta_C \quad z_1 \in \delta_{C_1}, z_2 \in \delta_{C_2}, z \doteq z_1, z \doteq z_2, P_1, P_2, P_3, P_4 \vdash Q}{Q}\ (|\_|^-)$$

where P_1 is $z_1.i_{C_1}? = (z.i_C? \cup (z.o_C! \cap \Psi)) \cap in_{C_1}$ P_2 is $z_2.i_{C_2}? = (z.i_C? \cup (z.o_C! \cap \Psi)) \cap in_{C_2}$ P_3 is $z.o_C! = z_1.o_{C_1}! \cup z_2.o_{C_2}!$, and P_4 is $\mathit{active}\ C_1\ z \Leftrightarrow \mathit{active}\ C_2\ z$, and,

$$\frac{z_1 \in \delta_{C_1} \quad z_2 \in \delta_{C_2} \quad z \doteq z_1 \quad z \doteq z_2 \quad P_1 \quad P_2 \quad P_3 \quad P_4}{z \in \delta_C}\ (|\_|^+)$$

The predicates $\mathit{active}\ C\ z$ and $\mathit{inactive}\ C\ z$ are defined for composed charts as,

$$\mathit{active}\ C\ z =_{def} \mathit{active}\ C_1\ z \vee \mathit{active}\ C_2\ z$$
$$\mathit{inactive}\ C\ z =_{def} \mathit{inactive}\ C_1\ z \wedge \mathit{inactive}\ C_2\ z$$

While it may seem counter intuitive to have the disjunction in the definition of $\mathit{active}\ C\ z$, it is so defined in order that we have the property $\mathit{inactive}\ C\ z \Leftrightarrow \neg\ (\mathit{active}\ C\ z)$ for composed charts.

We introduce a more useful form of the rule $(|\_|^+)$ as follows [8].

Proposition 2. *For arbitrary* $o_1, o_2 \in \mathbb{P}\ out_C$,

$$\frac{\begin{array}{l} z.o_C! = o_1 \cup o_2 \\ z \star \langle\!|\ i_{C_1}? \Rrightarrow (z.i_C? \cup fb\ z) \cap in_{C_1},\ o_{C_1}! \Rrightarrow o_1\ |\!\rangle \mathrel{\dot\in} \delta_{C_1} \\ z \star \langle\!|\ i_{C_2}? \Rrightarrow (z.i_C? \cup fb\ z) \cap in_{C_2},\ o_{C_2}! \Rrightarrow o_2\ |\!\rangle \mathrel{\dot\in} \delta_{C_2} \\ \mathit{active}\ C_1\ z \Leftrightarrow \mathit{active}\ C_2\ z \end{array}}{z \in \delta_C}\ (|\_|_2^+)$$

where $fb\ z =_{def} z.o_C! \cap \Psi$.

Also from $(|\_|^-)$ and $(|\_|^+)$, we can derive some other necessary introduction and elimination rules.

[8] The binding concatenation operator $\star$ is defined as in [2]. See Appendix A.3 for the proofs of the propositions in this section.

Proposition 3. *Given* $C = C_1 \mid \Psi \mid C_2$, *for arbitrary* $z \in T$,

$$\frac{z \in \delta_C \quad \mathit{inactive}\ C\ z}{z \dot{\in} \Xi\, \mathit{Chart}_C}\ (\mathit{iact}_1^-) \qquad \frac{z \in \delta_C \quad \mathit{inactive}\ C\ z}{z.o_C! = \{\}}\ (\mathit{iact}_2^-)$$

$$\frac{\mathit{inactive}\ C\ z \quad z \dot{\in} \Xi\, \mathit{Chart}_C \quad z.o_C! = \{\}}{z \in \delta_C}\ (\mathit{iact}^+)$$

where $[\![\delta_C]\!]^{\mathbb{P}\,T}$.

Notice that the rules (iact_1^-), (iact_2^-) and (iact^+) have an identical form to the respective rules $(Z_{t\,2}^-)$, $(Z_{t\,3}^-)$ and $(Z_{t\,2}^+)$ from Section 3.2. In fact, eventually we prove rules of this form for all chart regardless of their composition. Hence, we can assume they hold for all valid μ-charts.

That the basic requirement of symmetry holds of the composition operator is given by the following rule[9].

Proposition 4.

$$\frac{}{[\![\delta_{C_1 \mid \Psi \mid C_2}]\!] = [\![\delta_{C_2 \mid \Psi \mid C_1}]\!]}\ (\mid \_ \mid_{sym})$$

Like for sequential charts (see Proposition 1), we can prove the rule $(Z_t^{\in})$ for composed charts.

Proposition 5. *Given* $C = C_1 \mid \Psi \mid C_2$,

$$\frac{z \in \delta_C \quad x \upharpoonright T_i \doteq z \quad z.i_C? \cup \mathit{fb}\ z = x.i_C? \cup \mathit{fb}\ x}{x \in \delta_C}\ (Z_t^{\in})$$

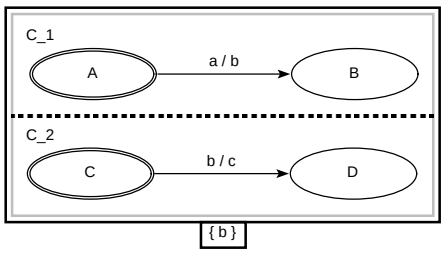
where $[\![\delta_C]\!]^{\mathbb{P}\,T}$, $T_i = T - [i_C? : \mathbb{P}\, \mathit{in}_C]$ *and* $\mathit{fb}\ z =_{\mathit{def}} z.o_C! \cap \Psi$.

The proof of Proposition 5, that is that the rule $(Z_t^{\in})$ holds for composed charts of the form $C = C_1 \mid \Psi \mid C_2$, relies only on the fact that the rule $(Z_t^{\in})$ itself holds for each part of the composition, *i.e.* C_1 and C_2. Hence, by an inductive argument, $(Z_t^{\in})$ holds regardless of the structure of C_1 and C_2.

As expected, given the instantaneous feed back of signals in charts, whenever the fed back output from one part of the composition, say C_1 triggers a transition in the other part, say C_2, there is a corresponding binding in the transition model of the composition. This binding represents a transition of the composition that is triggered by the same input as the transition in C_1 and outputs the combination of the outputs from the respective parts. That this is the case follows directly from Proposition 5.

We give a concrete example of using this rule. Consider the following μ-chart.

[9] Recall the transition model $[\![\delta_C]\!]$ is a set of bindings so the notion of equality used here is the usual set equality



Given this chart we can easily show that

$$\langle\!| \; c_{C_1} \Rrightarrow A,\; c'_{C_1} \Rrightarrow B,\; i_{C_1}? \Rrightarrow \{a\},\; active\_ \Rrightarrow \{C_1, C_2\},\; o_{C_1}! \Rrightarrow \{b\} \; |\!\rangle \in \delta_{C_1} \qquad (1)$$

and

$$\langle\!| \; c_{C_2} \Rrightarrow C,\; c'_{C_2} \Rrightarrow D, i_{C_2}? \Rrightarrow \{b\},\; active\_ \Rrightarrow \{C_1, C_2\},\; o_{C_2}! \Rrightarrow \{c\} \; |\!\rangle \in \delta_{C_2} \qquad (2)$$

Hence, from (1) and (2) by $(|\_|^+)$ we have,

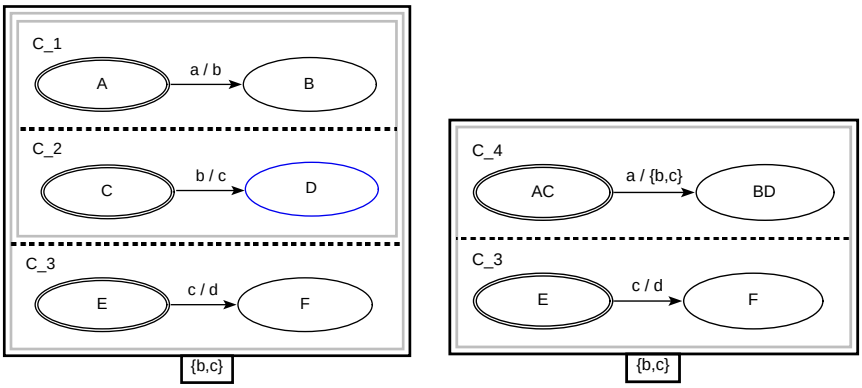
$$\begin{array}{l} \langle\!| \; c_{C_1} \Rrightarrow A,\; c_{C_2} \Rrightarrow C,\; c'_{C_1} \Rrightarrow B,\; c'_{C_2} \Rrightarrow D, \\ \quad i_C? \Rrightarrow \{a, b\},\; active\_ \Rrightarrow \{C_1, C_2\},\; o_C! \Rrightarrow \{b, c\} \; |\!\rangle \in \delta_C. \end{array} \qquad (3)$$

Now from (3) and Proposition 5, given that $\{a\} \cup \{b\} = \{a, b\} \cup \{b\}$, we can show that

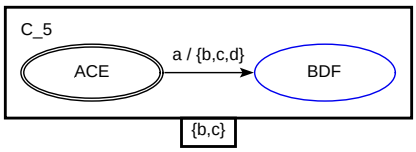
$$\begin{array}{l} \langle\!| \; c_{C_1} \Rrightarrow A,\; c_{C_2} \Rrightarrow C,\; c'_{C_1} \Rrightarrow B,\; c'_{C_2} \Rrightarrow D, \\ \quad i_C? \Rrightarrow \{a\},\; active\_ \Rrightarrow \{C_1, C_2\},\; o_C! \Rrightarrow \{b, c\} \; |\!\rangle \in \delta_C. \end{array} \qquad (4)$$

also holds. This binding realizes the behaviour of the composed chart C, that is, assuming initial states, when the environment offers the signal a, both charts C_1 and C_2 make a transition. Clearly the transition in C_1 is triggered by the input a. The transition in C_2 however, is triggered by the instantaneous feedback of the output signal b.

This example demonstrates that the transition model essentially recursively sequentializes" composed charts. By "sequentialize" we refer to the process of taking the cross product of all of the transitions in each part of the composition and creating a transition in the composition model that represents both of these transitions happening together. This process is recursive whenever the parts of the composition are not sequential μ-charts. Consider the following illustration of this process.



(a) (b)



(c)

Fig. 8.

The chart C_4 of Figure 8(b) is the sequential representation of $C_1 \mid \{a, b\} \mid C_2$. Chart C_5 is the sequential representation of $C_4 \mid \{a, b\} \mid C_3$.

That the following holds,

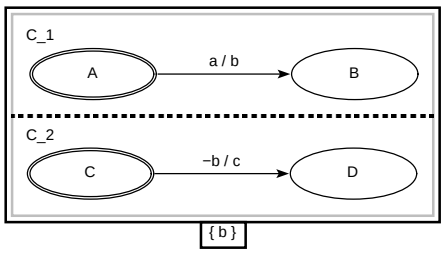
$$\langle\!| \; c_{C_1} \Rrightarrow A,\; c_{C_2} \Rrightarrow C,\; c_{C_3} \Rrightarrow E,\; c'_{C_1} \Rrightarrow B,\; c'_{C_2} \Rrightarrow D,\; c_{C_3} \Rrightarrow F, \\ i_C? \Rrightarrow \{a, b, c\},\; active\_ \Rrightarrow \{C_1, C_2, C_3\},\; o_C! \Rrightarrow \{b, c, d\} \; |\!\rangle \in \delta_C.$$

and hence,

$$\langle\!| \; c_{C_1} \Rrightarrow A,\; c_{C_2} \Rrightarrow C,\; c_{C_3} \Rrightarrow E,\; c'_{C_1} \Rrightarrow B,\; c'_{C_2} \Rrightarrow D,\; c_{C_3} \Rrightarrow F, \\ i_C? \Rrightarrow \{a\},\; active\_ \Rrightarrow \{C_1, C_2, C_3\},\; o_C! \Rrightarrow \{b, c, d\} \; |\!\rangle \in \delta_C.$$

where $C = (C_1 \mid \{b, c\} \mid C_2) \mid \{b, c\} \mid C_3$, that is the chart pictured in Figure 8(a), is trivial to show using the same method as used in the example above.

Of course, that this process works correctly, that is models our intuition for charts that contain negated signals, requires that the combination of inconsistent transitions is not represented in the transition model. Take for example the following chart.



The following lemma states, assuming that C_1 and C_2 are active, that the transition model for this chart contains no bindings that represent a transition from the configuration A, C to the configuration B, D. Hence, even though there are explicitly defined transitions in charts C_1 and C_2 respectively, the result of combining these two charts using composition is that the composed chart has no explicitly defined transitions.

Lemma 3. *Given* $C = C_1 \mid \{b\} \mid C_2$, *for arbitrary* $z \in T$ *we have,*

$$\frac{\{C_1, C_2\} \subseteq z.active\_}{z \notin \delta_C}$$

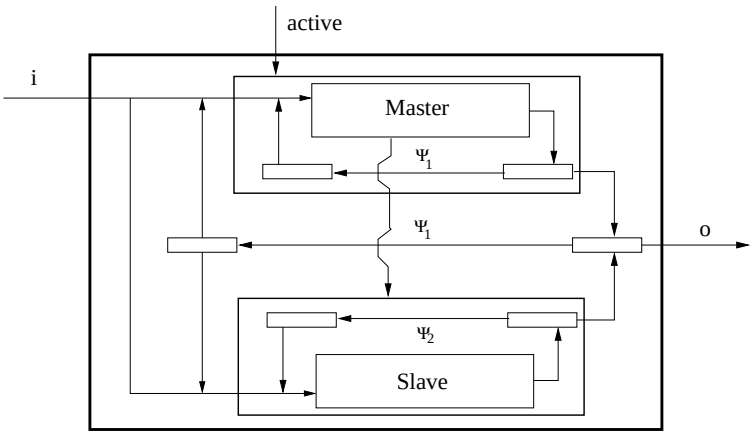
where $[\![\delta_C]\!]^{\mathbb{P}\,T}$

The fact that, in this particular example, we prove (see Appendix A.3) that there are no explicit transitions at all in the model of the composition, *i.e.* $[\![\delta_C]\!] = \{\}$, is because the example has only two candidate transitions that can combine to cause an explicit transition in the composition. Because these transitions are inconsistent, under the defined feedback, results in the empty set of bindings for the transition model.

3.5 The decomposition operator

Another of the useful structuring mechanisms of μ-charts is the decomposition operator. The decomposition of a sequential chart refers to replacing a state in the chart with another μ-chart. This creates a master / slave relationship between the sequential chart, the master, and the arbitrary chart, the slave, that replaces a state in the master.

Intuitively, we can consider the behaviour of a master and a slave in such a chart as though they are composed in parallel. Consider the following circuit diagram representing the chart $Dec\ (Master, \Sigma_1, \sigma_1, \Psi_1, \delta_1)\ by\ \{(\sigma_1, (Slave, \Sigma_2, \sigma_2, \Psi_2, \delta_2))\}$.



There are two subtle but significant differences between this diagram and the corresponding circuit diagram for the composition operator. The first being that the active state of the slave is determined by the master and not by the active input to the overall decomposition. This is because the state of the master determines whether or not the slave is active. The second difference is that the feedback signals that the master and slave share or communicate on are determined by the feedback set that is present in the definition of the master, *i.e.* denoted by the links label Ψ_1 in the diagram.

Hence when the master is in the state decomposed by the slave, that is both charts are active, then the decomposition is exactly the same as the composition of the master and slave with the feedback set equivalent to that of the master chart.

Again, the structure of the circuit diagram assumes both the master and slave are sequential charts. In general the slave can have arbitrary structure. However, unlike composition, the master must be a sequential chart.

The semantics that results from adhering exactly to this model of decomposition produces some "unexpected" behaviour (at least behaviour that may be considered somewhat counter to intuition). For example consider the decomposed chart $C = Dec\ (\omega\ C_1)\ by\ \{(C_2, \omega\ C_2)\}$ pictured in Figure 9.

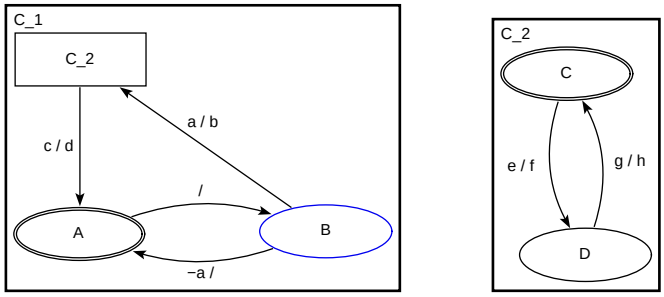
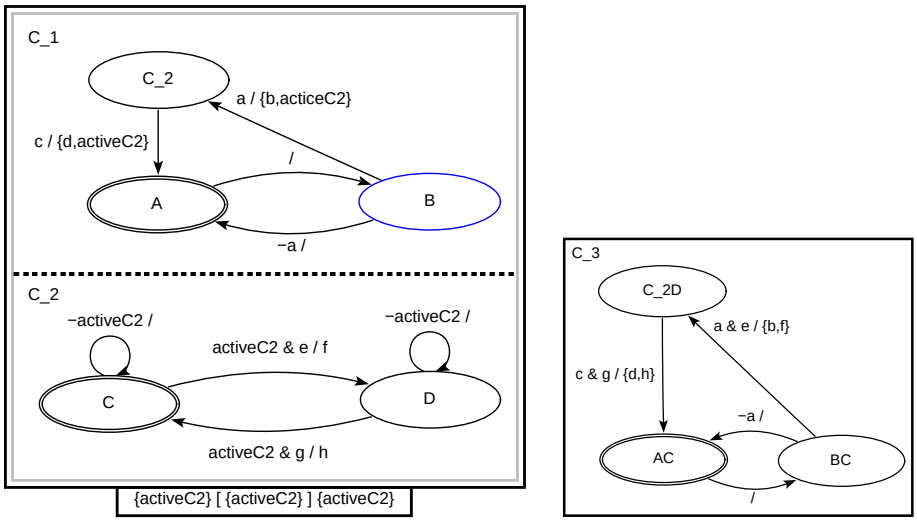


Fig. 9.

The semantics of the example decomposed chart is demonstrated by the composed chart $C' =_{\{activeC2\}} [C_1 \mid \{activeC2\} \mid C_2]_{\{activeC2\}}$ pictured in Figure 10(a). The chart C_3 (Figure 10(b)) is the sequential equivalent of C'. Hence the chart C_3 exhibits identical behaviour to the original decomposed chart C.



(a) (b)

Fig. 10.

The behaviour of the chart C_3 does not appear to capture the intention of the decomposed chart C under any standard interpretation. However, consider the following examples of complimentary interpretations that we may wish to capture using the decomposition operator. Under some situations we may wish that the slave chart C_2 is initialized each time a transition in the master enters

the slave. On the other hand we may wish that the slave remembers what state it was in when it was last exited by the master. It may be the case that the master should be idle, that is remain in the slave's state and produce no output, while the slave is free to make transitions or we may wish to allow the master to perform some task in parallel with the slave, for example counting the number of transitions the slave makes. Importantly, each of these interpretations can be implemented using explicit transitions in the original decomposed chart such that the new behavior is a refinement of the behaviour that is assigned to chart C. To make explicit in the semantic model one of these interpretations for the chart C would have the side-effect that the alternate interpretation is no longer expressible at all. Hence by giving the most general semantics we allow different interpretations to be assigned in the future without modifying the language.

In general a clear goal of any semantic model is to capture our intuition. However, we choose to keep the semantic model of charts as simple and general as possible and modify our intuition slightly. We make this choice because, not surprisingly, it is often the case that the intuition that we have is based on the example uses that have been recently encountered (or at best is limited to the ability to imagine future uses). Hence, while the initial temptation is to complicate the model so that it exactly captures our current intuition, inevitably a new example arises which counters that intuition and therefore invalidates the complicated model. The ability to make good choices about the semantics highlights one of the benefits of a formal treatment of a language. When faced with several seemingly valid choices for meaning we can often rely on the eloquence and generality of the formal model to determine the most useful meaning.

The transition model for the decomposed chart $C = Dec\ (\omega\ M)\ by\ \{(S, \omega\ S)\}$, where $\omega\ M = (M, \Sigma, \sigma, \Psi, \delta)$ and arbitrary S, contains the following Z definitions and schemas.

$$
\begin{array}{|l}
states_C : \mathbb{P}\, \mu_{State} \\
in_C : \mathbb{P}\, \mu Signal \\
out_C : \mathbb{P}\, \mu Signal \\
\hline
states_C = states_M \cup states_S \\
in_C = in_M \cup in_S \\
out_C = out_M \cup out_S
\end{array}
$$

$$
\begin{array}{|l}
\hline
Chart_C \\
Chart_M \\
Chart_S \\
\hline
\end{array}
$$

$$
\begin{array}{|l}
\hline
Init_C \\
Init_M \\
Init_S \\
\hline
\end{array}
$$

$$
\begin{array}{|l}
\hline
\delta_C \\
\Delta Chart_C \\
i_C? : \mathbb{P}\, in_C \\
active\_ : \mathbb{P}\, \mu_{State} \\
o_C! : \mathbb{P}\, out_C \\
\hline
active(S) \Leftrightarrow (active(M) \\
\qquad\qquad \wedge\ (MS \vee MS')) \\
\exists\, i_M?, i_S?, o_M!, o_S! : \mathbb{P}\, \mu Signal \bullet \\
\qquad i_M? = (i_C? \cup (o_C! \cap \Psi)) \cap in_M \wedge \\
\qquad i_S? = (i_C? \cup (o_C! \cap \Psi)) \cap in_S \wedge \\
\qquad o_C! = o_M! \cup o_S! \wedge \\
\qquad \delta_M \wedge \delta_S \\
\hline
\end{array}
$$

Again we give the definition of the transition model in terms of sets of bindings. Given the arbitrary decomposed chart $C = Dec\ (\omega\ M)\ by\ \{(S, \omega\ S)\}$, where $\omega\ M = (M, \Sigma, \sigma, \Psi, \delta)$ and arbitrary S we have,

$$\begin{aligned}[\![\delta_C]\!]^{\mathbb{P}\,T} =_{def} \{z \in T \mid \exists\, z_m \in \delta_M;\ z_s \in \delta_S \bullet\ & z \doteq z_m \wedge z \doteq z_s \wedge \\ & z.o_C! = z_m.o_M! \cup z_s.o_S! \wedge \\ & z_m.i_M = (z.i_C? \cup (z.o_C! \cap \Psi)) \cap in_M \wedge \\ & z_s.i_S = (z.i_C? \cup (z.o_C! \cap \Psi)) \cap in_S \wedge \\ & active\ S\ z \Leftrightarrow (active\ M\ z \wedge \\ & \qquad (c_M = S \vee c'_M = S)\}\end{aligned}$$

Then the introduction and elimination rules for decomposed charts are as follows. Given $C = Dec\ (\omega\ M)\ by\ \{(S, \omega\ S)\}$, where $\omega\ M = (M, \Sigma, \sigma, \Psi, \delta)$, and assuming the usual conditions for z_m, z_s and Q we have,

$$\frac{z \in \delta_C \quad z_m \in \delta_M, z_s \in \delta_S, z \doteq z_m, z \doteq z_s, P_1, P_2, P_3, P_4 \vdash Q}{Q}\ (M_S^-)$$

where P_1 is $z.o_C! = z_m.o_M! \cup z_s.o_S!$, P_2 is $z_m.i_M = (z.i_C? \cup (z.o_C! \cap \Psi)) \cap in_M$, P_3 is $z_s.i_S = (z.i_C? \cup (z.o_C! \cap \Psi)) \cap in_S$ and P_4 is $active\ S\ z \Leftrightarrow (active\ M\ z \wedge (c_M = S \vee c'_M = S))$, and,

$$\frac{z_m \in \delta_M \quad z_s \in \delta_S \quad z \doteq z_m \quad z \doteq z_s \quad P_1 \quad P_2 \quad P_3 \quad P_4}{z \in \delta_C}\ (M_S^+)$$

The proofs of the introduction and elimination rules are trivial given the definition for decomposition.

The definition of the predicates $active\ C\ z$ and $inactive\ C\ z$ for decomposed charts is the same as for composed charts.

$$active\ C\ z =_{def} active\ M\ z \vee active\ S\ z$$
$$inactive\ C\ z =_{def} inactive\ M\ z \wedge inactive\ S\ z$$

As expected, given the definition of decomposed charts, we can show that two charts that share a master / slave relationship react in the same way as if they are composed in parallel whenever the master is in the decomposed state and vice versa. This gives us another useful introduction and elimination rule. Note that proofs for all of the propositions in this section are presented in Appendix A.4.

Proposition 6. *Given* $\delta_{M_S} = [\![Dec\ (\omega\ M)\ by\ \{(S, \omega\ S)\}]\!]_{Z_t}$, *for arbitrary* S, $\omega\ M = (M, \Sigma, \sigma, \Psi, \delta)$, *and* $\delta_{M||S} = [\![\omega\ M \mid \Psi \mid \omega\ S]\!]_{Z_t}$, *then for arbitrary* $z \in T$,

$$\frac{z.c_M = S \vee z.c'_M = S \quad z \in \delta_{M_S}}{z \in \delta_{M||S}}\ (M_{S2}^-)$$

$$\frac{z.c_M = S \vee z.c'_M = S \quad z \in \delta_{M||S}}{z \in \delta_{M_S}}\ (M_{S2}^+)$$

where $[\![\delta_{M_S}]\!]^{\mathbb{P}\,T}$.

On the other hand, when the master is not in a decomposed state the slave contribute no output and does not change state, that is the slave makes no transition. Hence we derive further elimination rules as follows.

Proposition 7. *Given* $C = Dec\ (\omega\ M)$ *by* $\{(S, \omega\ S)\}$, *for arbitrary* S *and* $\omega\ M = (M, \Sigma, \sigma, \Psi, \delta)$ *and* $z \in T$,

$$\frac{z \in \delta_C \quad z.c_M \neq S \quad z.c'_M \neq S}{inactive\ S\ z}\ (M^-_{S3})$$

$$\frac{z \in \delta_C \quad z.c_M \neq S \quad z.c'_M \neq S}{z \star \langle\!|\ i_M? \Rrightarrow z.i_C? \cap in_M, o_M! \Rrightarrow z.o_C!\ |\!\rangle \in \dot{\delta}_M}\ (M^-_{S4})$$

$$\frac{z \in \delta_C \quad z.c_M \neq S \quad z.c'_M \neq S}{z \star \langle\!|\ i_S? \Rrightarrow z.i_C? \cap in_S, o_S! \Rrightarrow \{\}\ |\!\rangle \in \dot{\delta}_S}\ (M^-_{S5})$$

We split the rule (M^+_S) into three more specific rules.

Proposition 8. *Given* $C = Dec\ (\omega\ M)$ *by* $\{(S, \omega\ S)\}$, *for arbitrary* S *and* $\omega\ M = (M, \Sigma, \sigma, \Psi, \delta)$. *For arbitrary* $o_m, o_s \in \mathbb{P}\ in_C$ *and* $z \in T$,

$$\frac{\begin{array}{l} z.o_C! = o_m \cup o_s \\ z.c_M = S \\ z \star \langle\!|\ i_M? \Rrightarrow (z.i_C? \cup fb\ z) \cap in_M, o_M! \Rrightarrow o_m\ |\!\rangle \in \dot{\delta}_M \\ z \star \langle\!|\ i_S? \Rrightarrow (z.i_C? \cup fb\ z) \cap in_S, o_S! \Rrightarrow o_s\ |\!\rangle \in \dot{\delta}_S \\ active\ M\ z \Leftrightarrow active\ S\ z \end{array}}{z \in \delta_C}\ (M^+_{S3})$$

$$\frac{\begin{array}{l} z.o_C! = o_m \cup o_s \\ z.c'_M = S \\ z \star \langle\!|\ i_M? \Rrightarrow (z.i_C? \cup fb\ z) \cap in_M, o_M! \Rrightarrow o_m\ |\!\rangle \in \dot{\delta}_M \\ z \star \langle\!|\ i_S? \Rrightarrow (z.i_C? \cup fb\ z) \cap in_S, o_S! \Rrightarrow o_s\ |\!\rangle \in \dot{\delta}_S \\ active\ M\ z \Leftrightarrow active\ S\ z \end{array}}{z \in \delta_C}\ (M^+_{S4})$$

$$\frac{\begin{array}{l} z.c_M \neq S \quad z.c'_M \neq S \\ inactive\ S\ z \\ z \star \langle\!|\ i_M? \Rrightarrow (z.i_C? \cup fb\ z) \cap in_M, o_M! \Rrightarrow z.o_C!\ |\!\rangle \in \dot{\delta}_M \\ z \star \langle\!|\ i_S? \Rrightarrow (z.i_C? \cup fb\ z) \cap in_S, o_S! \Rrightarrow \{\}\ |\!\rangle \in \dot{\delta}_S \end{array}}{z \in \delta_C}\ (M^+_{S5})$$

where $[\![\delta_C]\!]^{\mathbb{P}\ T}$.

The rules $(iact^-_1)$, $(iact^-_2)$ and $(iact^+)$ hold for decomposed charts.

Proposition 9. *Given* $C = Dec\ (\omega\ M)$ *by* $\{(S, \omega\ S)\}$, *for arbitrary* S, $\omega\ M = (M, \Sigma, \sigma, \Psi, \delta)$ *and* $z \in T$,

$$\frac{z \in \delta_C \quad inactive\ C\ z}{z \dot{\in} \Xi Chart_C}\ (iact_1^-) \qquad \frac{z \in \delta_C \quad inactive\ C\ z}{z.o_C! = \{\}}\ (iact_2^-)$$

$$\frac{inactive\ C\ z \quad z \dot{\in} \Xi Chart_C \quad z.o_C! = \{\}}{z \in \delta_C}\ (iact^+)$$

where $[\![\delta_C]\!]^{\mathbb{P}\ T}$.

Finally, we show that the transition model for decomposed charts, like sequential and composed charts, does not distinguish between the source of input, that is the rule $(Z_t^{\in})$ holds for decomposed charts.

Proposition 10. *Given* $C = Dec\ (\omega\ M)$ *by* $\{(S, \omega\ S)\}$, *for arbitrary* S *and* $\omega\ M = (M, \Sigma, \sigma, \Psi, \delta)$,

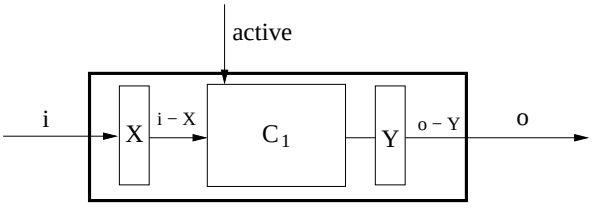
$$\frac{z \in \delta_C \quad x \upharpoonright T_i \doteq z \quad z.i_C? \cup fb\ z = x.i_C? \cup fb\ x}{x \in \delta_C}\ (Z_t^{\in})$$

where $[\![\delta_C]\!]^{\mathbb{P}\ T}$, $T_i = T - [i? : \mathbb{P}\ in_C]$ *and* $fb\ z =_{def} z.o_C! \cap \Psi$.

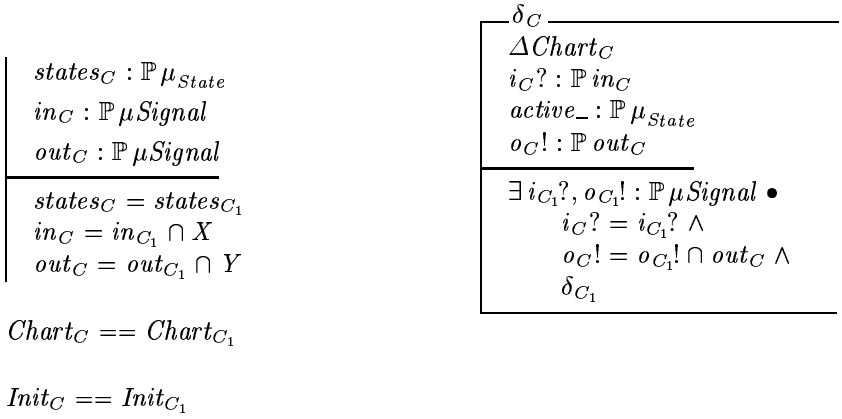
3.6 The hiding operator

The final structuring mechanism of μ-Charts is the hiding operator. As we have already mentioned, the hiding operator allows the designer limited control over the environment in which a chart resides. The hiding operator allows two conceptually different types of signal filtering to be applied to a chart. Signals can be filtered from the environment's input, that is input hiding. Effectively input hiding is the same as removing the hidden signals from the charts input interface. Hence hidden input signals do not effect the charts reaction and can be ignored when reasoning about the chart's behaviour. The second type of hiding, output hiding, allows the designer to restrict the set of output signals that the environment of the chart can see.

The following diagram demonstrates the general structure of the model for charts that hide the input signals in set X and output signals from the set Y. That is the model of a chart $C =_X [C_1]_Y$ has the follow structure.



The following Z definitions and schemas provide the transition model for the chart $C =_X [C_1]_Y$.



Notice that apart from the obvious distinction between the schema δ_C and the existing δ_{C_1}, *i.e.* the new predicate that restricts the output, there are some more subtle differences introduced using the Z type system. In particular the input observation $i_C?$ has the type $\mathbb{P}\,in_C$ rather than $\mathbb{P}\,in_{C_1}$. When reasoning about the schema δ_C this observation gets normalised so that its type, as for the input observation $i_{C_1}?$ in δ_{C_1}, is $\mathbb{P}\,\mu Signal$. However, the normalisation also adds the additional constraint $i_C? \subseteq in_C$ to the predicate part of the normalised version of the schema δ_C. Thus restricting the set of binding from δ_{C_1} to those that take input from the new input interface for the chart C and hence modeling the hiding of input.

Interestingly, the process of hiding input restricts the set of binding that is the model of the original chart, whereas, output hiding requires that we modify the bindings that inhabit the model of the original chart. This is because when we restrict the input interface of a chart we are making the specified reactive system less reactive. That is there are less signals that the environment can use to effect a reaction from the system in question. On the other hand, hiding output does not affect the reactivity of the specified system, rather it lessens the ability of system to affect its environment.

Once again, we give the definition of the transition model in terms of the meaning of the Z. Given the chart $C =_X [C_1]_Y$ we have,

$$[\![\delta_C]\!]^{\mathbb{P}\,T} =_{def} \{z \in T \mid \exists\, z_1 \in \delta_{C_1} \bullet z \doteq z_1 \wedge z.i_C? \subseteq in_C \wedge z.i_C? = z_1.i_{C_1}? \wedge z.o_C! = z_1.o_{C_1}! \cap out_C\}$$

Notice that the predicate $z.i_C? \subseteq in_C$ is not strictly necessary in the above definition because it follows from the fact that the binding z is in the type T. However, we leave it explicit to clearly demonstrate that the set of bindings is appropriately restricted by the hiding of input.

The following introduction and elimination rules follow trivially from this definition. Given $C =_X [C_1]_Y$ and assuming the usual conditions for z_1 and Q we have for arbitrary $z \in T$,

$$\frac{z \in \delta_C \quad z_1 \in \delta_{C_1}, z \doteq z_1, z.i_C? = z_1.i_{C_1}?, z.o_C! = z_1.o_{C_1}! \cap out_C \vdash Q}{Q} \; (x[]_Y^-)$$

$$\frac{z_1 \in \delta_{C_1} \quad z \doteq z_1 \quad z.i_C? = z_1.i_{C_1}? \quad z.o_C! = z_1.o_{C_1}! \cap out_C}{z \in \delta_C} \; (x[]_Y^+)$$

The predicates *active C z* and *inactive C z* are for charts with hiding defined trivially as,

$$active\ C\ z =_{def} active\ C_1\ z$$
$$inactive\ C\ z =_{def} inactive\ C_1\ z$$

The rules $(iact_1^-)$, $(iact_2^-)$ and $(iact^+)$ hold for charts that contain hiding.

Proposition 11. *Given $C =_X [C_1]_Y$, for arbitrary $z \in T$,*

$$\frac{z \in \delta_C \quad inactive\ C\ z}{z \dot{\in} \Xi Chart_C} \; (iact_1^-) \qquad \frac{z \in \delta_C \quad inactive\ C\ z}{z.o_C! = \{\}} \; (iact_2^-)$$

$$\frac{inactive\ C\ z \quad z \dot{\in} \Xi Chart_C \quad z.o_C! = \{\}}{z \in \delta_C} \; (iact^+)$$

where $[\![\delta_C]\!]^{\mathbb{P}\,T}$.

And finally the rule $(Z_t^{\in})$ holds for charts with hiding.

Proposition 12. *Given $C =_X [C_1]_Y$ and assuming that Ψ is defined to be the same as the feedback set Ψ_1 for the chart C_1,*

$$\frac{z \in \delta_C \quad x \restriction T_i \doteq z \quad z.i_C? \cup fb\ z = x.i_C? \cup fb\ x}{x \in \delta_C} \; (Z_t^{\in})$$

where $[\![\delta_C]\!]^{\mathbb{P}\,T}$, $T_i = T - [i_C? : \mathbb{P}\, in_C]$ and $fb\ z =_{def} z.o_C! \cap \Psi$.

3.7 Step semantics

In this section we give the general method for defining the step semantics for a chart.

The step semantics is no more than the transition model of the top-most subchart of a μ-chart with the active state machinery hidden. Given an arbitrary μ-chart called C, the step behaviour of C is defined by another schema which, by convention, we call $CSys$.

$$\mu Signal = in\ C$$

$$\begin{array}{|l}
\hline
CSys \\
\Delta Chart_C \\
i_C? : \mathbb{P}\, in_C \\
o_C! : \mathbb{P}\, out_C \\
\hline
\exists\, active_ : \mathbb{P}\, \mu_{State} \bullet \\
\quad active(C) \land \\
\quad \delta_C \\
\hline
\end{array}$$

The set $\mu Signal$ is defined to contain all of the signals that appear in any of the sub-charts. The schema $CSys$ hides the active state observation and specifies that the topmost chart(s) in any hierarchical structure is (are) active.

Now the definition of the step semantics for a chart is as follows:

$$[\![CSys]\!]^{\mathbb{P}\, T} =_{def} \{z \in T \mid \exists\, z_1 \in \delta_C \bullet active\ C\ z_1 \land z \doteq z_1\}$$

From this definition we derive the following introduction and elimination rules.

For arbitrary chart C, assuming the usual conditions for z_1 and Q we have,

$$\frac{z \in CSys \quad z_1 \in \delta_C, z \doteq z_1, active\ C\ z \vdash Q}{Q}\ (Z_s{}^-)$$

$$\frac{z_1 \in \delta_C \quad z \doteq z_1 \quad active\ C\ z_1}{z \in CSys}\ (Z_s{}^+)$$

Now the schema $CSys$ and its meaning describes the step semantics for μ-Charts. We will often refer to this step semantics as the *partial relations semantics*. This is because the meaning of the schema $CSys$ can be considered as a partial relation that maps the before state of a chart and input to its after state and output. It is a partial relation because the behaviour of the chart outside of the defined transitions is yet to be defined.

References

1. D. Harel. Statecharts: A visual formalism for complex systems. *Science of Computing*, pages 231–274, 1987.
2. M. Deutsch, M. C. Henson, and S. Reeves. An analysis of total correctness refinement models for partial relation semantics: part 1. *The Logic Journal of the IGPL*, 11(3):285–316, 2003.
3. M. Deutsch, M. C. Henson, and S. Reeves. Operation refinement and monotonicity in the schema calculus. In Didier Bert, Jonathan P. Bowen, Steve King, and Marina Walden, editors, *ZB 2003: Formal Specification and Development in Z and B*, volume 2651 of *Lecture Notes in Computer Science*, pages 103–126. Springer-Verlag, 2003.
4. M. C. Henson and S. Reeves. Investigating Z. *Journal of Logic and Computation*, 10(1):1–30, 2000.
5. The $\mathcal{Z}_\lambda$ web site is `http://www.cs.waikato.ac.nz/Research/fm/index.html`.
6. Dieter Nazareth, Franz Regensburger, and Peter Scholz. Mini-statecharts: A lean version of statecharts. Technical Report TUM-I9610, Technische Univerität München, 1996.
7. J. Philipps and P. Scholz. Compositional specification of embedded systems with statecharts. In M. Bidoit and M. Dauchet, editors, *TAPSOFT '97: Theory and Practice of Software Development*, number 1214 in LNCS, pages 637–651. Springer-Verlag, 1997.
8. Jan Philipps and Peter Scholz. Formal verification of statecharts with instantaneous chain reaction. In E. Brinksma, editor, *Tools and Algorithms for the Construction and Analysis of Systems*, volume 1217 of *Lecture Notes in Computer Science*, pages 224–238. Springer-Verlag, 1997.
9. Greg Reeve and Steve Reeves. μ-Charts and Z: Examples and extensions. In *Proceedings of APSEC2000*. IEEE Computer Society, 2000.
10. Greg Reeve and Steve Reeves. μ-Charts and Z: Extending the translation. Technical Report 00/11, Department of Computer Science, University of Waikato, 2000.
11. Greg Reeve and Steve Reeves. μ-Charts and Z: Hows, whys and wherefores. In W. Grieskamp, T. Santen, and B. Stoddart, editors, *Integrated Formal Methods 2000: Proceedings of the 2nd. International Workshop on Integrated Formal Methods*, LNCS 1945. Springer-Verlag, 2000.
12. P. Scholz. *Design of Reactive Systems and their Distributed Implementation with Statecharts*. PhD thesis, Institut für Informatik, Technische Universität München, August 1998. TUM-I9821.
13. Peter Scholz. An extended version of mini-statecharts. Technical Report TUM-I9628, Technische Univerität München, 1996.

A Proofs

This section presents the proofs for the various propositions and lemmas presented throughout this document.

A.1 Proof for Section 3.2: The Z transition model for sequential μ-Charts

In order to prove Proposition 1 we introduce and prove the following lemmas.

Lemma 4. *Given the arbitrary sequential chart* $(C, \Sigma, \sigma, \Psi, \delta)$ *and transition* $t = (S_f, S_t, guard/action)$*, such that* $t \in \delta$*. For the arbitrary bindings* $z \in T$ *and* $x \in T$*,*

$$\frac{\rho(guard)[\alpha T/z.\alpha T] \quad z.i_C? \cup fb\, z = x.i_C? fb\, x}{\rho(guard)[\alpha T/x.\alpha T]}$$

where $[\![t]\!]^{\mathbb{P}\, T}_{Z_t}$ *and* $T_i = T - [i_C? : \mathbb{P}\, in_C]$

Recall that the function ρ (see Section 3.2) is defined recursively over the syntactic structure of transition guards. A transition guard (as defined in Section 2.1) is a list of signal expressions (either positive or negative) separated by the symbol &. Hence we prove this lemma by an induction over the number of signals in the guard.

When the guard is empty the function ρ returns the predicate *true*. Hence lemma 4 holds trivially.

Given an arbitrary signal expression *sig_expr* and a well formed guard *sigs* containg zero or more signal expressions, assuming lemma 4 holds for the guard *sigs*, then we prove it holds for the guard *sigs* & *sig_expr*. From the definition of ρ we know that $\rho(sigs \;\&\; sig\_expr) =_{def} \rho(sigs) \wedge \rho(sig\_expr)$. Hence, using the induction hypothesis, the proof is reduce to showing lemma 4 holds where $guard = sig\_expr$. We split this into two cases: one for a positive signal expression *sig_expr*; and one for a negative signal expression.

For a positive signal expression we have,

$$\frac{\dfrac{z.i_C? \cup fb\, z = x.i_C? \cup fb\, x \qquad \dfrac{\dfrac{\rho(sig\_expr)[\alpha T/z.\alpha T]}{(sig\_expr \in i_C? \cup (o_C! \cap \Psi))[\alpha T/z.\alpha T]}\ (\rho\text{-}df)}{sig\_expr \in z.i_C? \cup (z.o_C! \cap \Psi)}}{sig\_expr \in x.i_C? \cup (x.o_C! \cap \Psi)}}{\dfrac{(sig\_expr \in i_C? \cup (o_C! \cap \Psi))[\alpha T/x.\alpha T]}{\rho(sig\_expr)[\alpha T/x.\alpha T]}\ (\rho\text{-}df)}$$

In the case where the signal expression is negative can be proved in the same way where each occurrence of $\in$ is replaced with $\notin$.

Therefore, by induction, we have shown that lemma 4 holds.

Now using Lemma 4 we can show the following lemma holds.

Lemma 5. *Given the arbitrary chart* $(C, \Sigma, \sigma, \Psi, \delta)$*, for all* $t \in \delta$*,* $z \in T$ *and* $x \in T$*,*

$$\frac{\mathit{Trans}\ t\ z \quad x \upharpoonright T_i \doteq z \quad z.i_C? \cup \mathit{fb}\ z = x.i_C? \cup \mathit{fb}\ x}{\mathit{Trans}\ t\ x}$$

where $[\![\delta_C]\!]^{\mathbb{P}\,T}$ *and* $T_i = T - [i_C? : \mathbb{P}\,\mu \mathit{Signal}]$

Proof

$$\dfrac{\dfrac{\dfrac{\dfrac{\mathit{Trans}\ t\ z}{z.c_C = t.S_f \wedge z.c'_C = t.S_t \wedge z.o_C! = t.action}\ (df) \quad x \upharpoonright T_i \doteq z}{x.c_C = t.S_f \wedge x.c'_C = t.S_t \wedge x.o_C! = t.action} \quad \begin{matrix}\zeta \\ \vdots\end{matrix}}{x.c_C = t.S_f \wedge \rho(t.guard)[\alpha\,T/x.\alpha\,T] \wedge x.c'_C = t.S_t \wedge x.o_C! = t.action}\ (\wedge^+)}{\mathit{Trans}\ t\ x}\ (df)$$

where ζ_1 is:

$$\dfrac{\dfrac{\mathit{Trans}\ t\ z}{\rho(t.guard)[\alpha\,T/z.\alpha\,T]}\ (df) \quad z.i_C? \cup \mathit{fb}\ z = x.i_C? \cup \mathit{fb}\ x}{\rho(t.guard)[\alpha\,T/x.\alpha\,T]}\ (lem4)$$

Proposition 1. *For the arbitrary sequential chart* $(C, \Sigma, \sigma_0, \Psi, \delta)$, $z \in T$ *and* $x \in T$,

$$\frac{z \in \delta_C \quad x \upharpoonright T_i \doteq z \quad z.i_C? \cup fb\, z = x.i_C? \cup fb\, x}{x \in \delta_C}\ (Z_t^{\in})$$

where $[\![\delta_C]\!]^{\mathbb{P}\,T}$ *and* $T_i = T - [i_C? : \mathbb{P}\, in_C]$

Proof

$$\dfrac{\dfrac{}{\begin{array}{c}active\ C\ z \vee \\ inactive\ C\ z\end{array}} \qquad \dfrac{\dfrac{z \in \delta_C \quad \dfrac{}{active\ C\ z}\,{}^{1}}{\exists\, t \in \delta \bullet Trans\ t\ z}\,(Z_t{}^{-}_{1}) \quad \begin{array}{c}\zeta_1 \\ \vdots\end{array}}{x \in \delta_C}\,(\exists^{-})(2) \qquad \begin{array}{c}\zeta_2 \\ \vdots\end{array}}{x \in \delta_C}\,(\vee^{-})(1)$$

ζ_1 is:

$$\dfrac{\dfrac{\dfrac{}{Trans\ t_1\ z}\,{}^{2} \quad x \upharpoonright T_i \doteq z \quad \begin{array}{c}z.i_C? \cup fb\, z = \\ x.i_C? \cup fb\, x\end{array}}{Trans\ t_1\ x}\,(lem5) \qquad \dfrac{\dfrac{}{active\ C\ z}\,{}^{1} \quad x \upharpoonright T_i \doteq z}{active\ C\ x}}{x \in \delta_C}\,(Z_t{}^{+}_{1})$$

ζ_2 is:

$$\dfrac{\dfrac{\dfrac{z \in \delta_C \quad \dfrac{}{inactive\ C\ z}\,{}^{1}}{z.o_C! = \{\}}\,(Z_t{}^{-}_{3}) \quad x \upharpoonright T_i \doteq z}{x.o_C! = \{\}} \quad \begin{array}{c}\zeta_3 \\ \vdots\end{array} \qquad \dfrac{\dfrac{}{inactive\ C\ z}\,{}^{1} \quad x \upharpoonright T_i \doteq z}{inactive\ C\ x}}{x \in \delta_C}\,(Z_t{}^{+}_{2})$$

ζ_3 is:

$$\dfrac{\dfrac{z \in \delta_C \quad \dfrac{}{inactive\ C\ z}\,{}^{1}}{z \mathrel{\dot{\in}} \Xi\, Chart_C}\,(Z_t{}^{-}_{2}) \quad x \upharpoonright T_i \doteq z}{x \mathrel{\dot{\in}} \Xi\, Chart_C}\,([c_C, c'_C] \sqsubseteq T_i)$$

A.2 Proofs for Section 3.3: Feedback of signals in sequential μ-Charts

Lemma 1. *Given* $\delta_{C_1} = [\![(C_1, \Sigma, \sigma, \Psi, \delta_1)]\!]_{Zt}$, $\Sigma = \{A, B\}$, $\sigma = A$, $\Psi = \{a\}$ *and* $\delta_1 = \{(A, B, a/a)\}$, *for arbitrary* $z \in T$ *and input* $i \subseteq in_C$ *we have,*

$$\frac{active\ C_1\ z \quad z \doteq \langle\!|\ c_{C_1} \Rrightarrow A i_{C_1}? \Rrightarrow i, c'_{C_1} \Rrightarrow B, o_{C_1}! \Rrightarrow \{a\}\ |\!\rangle}{z \in \delta_{C_1}}$$

where $[\![\delta_{C_1}]\!]^{\mathbb{P}\,T}$

Proof

$$\dfrac{active\ C_1\ z \qquad \dfrac{\dfrac{\begin{array}{c}\zeta_1\\ \vdots\\ z \doteq \langle\!|\ c_{C_1} \Rrightarrow A, i_{C_1}? \Rrightarrow i, c'_{C_1} \Rrightarrow B, o_{C_1}! \Rrightarrow \{a\}\ |\!\rangle\end{array}}{z.c_{C_1} = A \wedge z.c'_{C_1} = B \wedge z.o_{C_1}! = \{a\} \wedge \rho(a)[\alpha\, T/z.\alpha\, T]}\ (df)}{Trans\ (A, B, a/a)\ z}\ (Z_{t\ 1}^{+})}{z \in \delta_{C_1}}$$

where ζ_1 is:

$$\dfrac{\dfrac{\dfrac{}{a \in \{a\}}}{a \in z.i_{C_1}? \cup (\{a\} \cap \{a\})} \qquad \dfrac{z \doteq \langle\!|\ c_{C_1} \Rrightarrow A, i_{C_1}? \Rrightarrow i, c'_{C_1} \Rrightarrow B, o_{C_1}! \Rrightarrow \{a\}\ |\!\rangle}{z.o_{C_1}! = \{a\}}}{a \in z.i_{C_1}? \cup (z.o_{C_1}! \cap \{a\})}}{\rho(a)[\alpha\, T/z.\alpha\, T]}$$

Lemma 2. *Given* $\delta_{C_2} = [\![(C2, \Sigma, \sigma, \Psi, \delta_2)]\!]_{Zt}$, $\Sigma = \{A, B\}$, $\sigma = A$, $\Psi = \{a\}$ *and* $\delta_2 = \{(A, B, -a/a)\}$, *for arbitrary* $z \in T$,

$$\frac{active\ C_2\ z}{z \notin \delta_{C_2}}$$

where $[\![\delta_{C_2}]\!]^{\mathbb{P}\,T}$

Proof

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{z \in \delta_{C_2}}\,{}^{2} \quad active\ C\ z}{\exists\, t \in \delta_{C_2} \bullet Trans\ t\ z}\,(Z_t{}^{-}{}_1)}{\exists\, t \in \{(A, B, -a/a)\} \bullet Trans\ t\ z}\,(\delta_{C_2}\text{-}df)}{z \notin \delta_{C_2}} \qquad \dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{t \in \{(A, B, -a/a)\}}\,{}^{1} \quad \dfrac{}{Trans\ t\ z}\,{}^{1}}{Trans\ (A, B, -a/a)\ z}\,(df)}{\rho(-a)[\alpha T/z.\alpha T]}\,(\rho\text{-}df)}{a \notin z.i_C? \cup (\{a\} \cap \{a\})}}{\bot}\,(\bot^{-})(2)}{z \notin \delta_{C_2}}\,(\exists^{-})(1)$$

A.3 Proofs for Section 3.4: The composition operator

Proposition 2. *For arbitrary* $o_1, o_2 \in \mathbb{P}\, out_C$,

$$\frac{\begin{array}{l} z.o_C! = o_1 \cup o_2 \\ z \star \langle\!|\ i_{C_1}?\Rrightarrow (z.i_C? \cup fb\ z) \cap in_{C_1},\ o_{C_1}!\Rrightarrow o_1\ |\!\rangle \mathrel{\dot\in} \delta_{C_1} \\ z \star \langle\!|\ i_{C_2}?\Rrightarrow (z.i_C? \cup fb\ z) \cap in_{C_2},\ o_{C_2}!\Rrightarrow o_2\ |\!\rangle \mathrel{\dot\in} \delta_{C_2} \\ active\ C_1\ z \Leftrightarrow active\ C_2\ z \end{array}}{z \in \delta_C}\ (|\_|_2^+)$$

where $fb\ z =_{def} z.o_C! \cap \Psi$.

Proof

Assuming $z_1 = z \star \langle\!|\ i_{C_1}?\Rrightarrow (z.i_C? \cup fb\ z) \cap in_{C_1},\ o_{C_1}!\Rrightarrow o_1\ |\!\rangle$, $z_2 = z\star\langle\!|\ i_{C_2}?\Rrightarrow (z.i_C? \cup fb\ z) \cap in_{C_2},\ o_{C_2}!\Rrightarrow o_2\ |\!\rangle$, $[\![\delta_{C_1}]\!]^{\mathbb{P}\,T_1}$ and $[\![\delta_{C_2}]\!]^{\mathbb{P}\,T_2}$,

$$\frac{\begin{array}{cccccc} \zeta_1 & \zeta_2 & & & \zeta_3 \quad \zeta_4 \quad \zeta_5 & \\ \vdots & \vdots & \dfrac{}{z_1 \restriction T_1 \doteq z} & \dfrac{}{z_2 \restriction T_2 \doteq z} & \vdots \quad \vdots \quad \vdots & active\ C_1\ z \Leftrightarrow active\ C_2\ z \end{array}}{z \in \delta_C}\ (|\_|^+)$$

where ζ_1 and ζ_2 are:

$$\frac{z_1 \mathrel{\dot\in} \delta_{C_1}}{z_1 \restriction T_1 \in \delta_{C_1}}\ (T_1 \sqsubseteq T \sqcup T_1) \qquad \frac{z_2 \mathrel{\dot\in} \delta_{C_2}}{z_2 \restriction T_2 \in \delta_{C_1}}\ (T_2 \sqsubseteq T \sqcup T_2)$$

ζ_3 and ζ_4 are:

$$\frac{\dfrac{}{z_1.i_{C_1}? = (z.i_C? \cup fb\ z) \cap in_{C_1}}\ (z_1\text{-}df)}{(z_1 \restriction T_1).i_{C_1}? = (z.i_C? \cup fb\ z) \cap in_{C_1}} \qquad \frac{\dfrac{}{z_2.i_{C_2}? = (z.i_C? \cup fb\ z) \cap in_{C_2}}\ (z_2\text{-}df)}{(z_2 \restriction T_2).i_{C_2}? = (z.i_C? \cup fb\ z) \cap in_{C_2}}$$

and ζ_5 is:

$$\frac{\dfrac{}{z_1.o_{C_1}! = o_1} \quad \dfrac{}{z_2.o_{C_2}! = o_2} \quad z.o_C! = o_1 \cup o_2}{z.o_C! = z_1.o_{C_1}! \cup z_2.o_{C_2}!}$$

Proposition 3. *Given $C = C_1 \mid \Psi \mid C_2$, for arbitrary $z \in T$,*

$$\frac{z \in \delta_C \quad inactive\ C\ z}{z \mathrel{\dot\in} \Xi\, Chart_C}\ (iact_1^-) \qquad \frac{z \in \delta_C \quad inactive\ C\ z}{z.o_C! = \{\}}\ (iact_2^-)$$

$$\frac{inactive\ C\ z \quad z \mathrel{\dot\in} \Xi\, Chart_C \quad z.o_C! = \{\}}{z \in \delta_C}\ (iact^+)$$

where $[\![\delta_C]\!]^{\mathbb{P}\,T}$.

Proof

For $(iact_1^-)$ we have,

$$\dfrac{\dfrac{\dfrac{\dfrac{}{y_1 \in \delta_{C_1}}\,1 \quad \dfrac{\dfrac{}{z \doteq y_1}\,1 \quad \dfrac{inactive\ C\ z}{inactive\ C_1\ z}\ (df)}{inactive\ C_1\ y_1}\ (iact_1^-)}{y_1 \mathrel{\dot\in} \Xi\, Chart_{C_1}} \quad \dfrac{}{z \doteq y_1}\,1}{y_1 \restriction T \mathrel{\dot\in} \Xi\, Chart_{C_1}}}{z \mathrel{\dot\in} \Xi\, Chart_{C_1}} \quad \begin{matrix}\zeta\\ \vdots\\ z \mathrel{\dot\in} \Xi\, Chart_{C_2}\end{matrix}}{z \mathrel{\dot\in} (\Xi\, Chart_{C_1} \wedge \Xi\, Chart_{C_2})}\ S_\wedge^+\ (df)$$

$$\dfrac{z \in \delta_C \qquad z \mathrel{\dot\in} \Xi\, Chart_C}{z \mathrel{\dot\in} \Xi\, Chart_C}\ (|\_|^-)(1)$$

where ζ is:

$$\dfrac{\dfrac{\dfrac{\dfrac{}{y_2 \in \delta_{C_2}}\,1 \quad \dfrac{\dfrac{}{z \doteq y_2}\,1 \quad \dfrac{inactive\ C\ z}{inactive\ C_2\ z}\ (df)}{inactive\ C_2\ y_2}\ (iact_1^-)}{y_2 \mathrel{\dot\in} \Xi\, Chart_{C_2}} \quad \dfrac{}{z \doteq y_2}\,1}{y_2 \restriction T \mathrel{\dot\in} \Xi\, Chart_{C_2}}}{z \mathrel{\dot\in} \Xi\, Chart_{C_2}}$$

For $(iact_2^-)$ we have,

$$\dfrac{\dfrac{\dfrac{}{y_1 \in \delta_{C_1}}\,1 \quad \dfrac{\dfrac{}{z \doteq y_1}\,1 \quad \dfrac{inactive\ C\ z}{inactive\ C_1\ z}}{inactive\ C_1\ y_1}}{y_1.o_{C_1}! = \{\}}\,(iact_2^-) \quad \begin{matrix}\zeta_1\\ \vdots\end{matrix} \quad \dfrac{}{z.o_C! = y_1.o_{C_1}! \cup y_2.o_{C_2}!}\,1}{z \in \delta_C \qquad z.o_C! = \{\}}\,(|\_|^-)(1)}{z.o_C! = \{\}}$$

where ζ_1 is:

$$\dfrac{\dfrac{}{y_2 \in \delta_{C_2}}\,1 \quad \dfrac{\dfrac{}{z \doteq y_2}\,1 \quad \dfrac{inactive\ C\ z}{inactive\ C_2\ z}}{inactive\ C_2\ y_2}\,(iact_2^-)}{y_2.o_{C_2}! = \{\}}$$

And for $(iact^+)$, assuming $z_1 = z \star \langle\!|\ i_{C_1}?\Rrightarrow(z.i_C? \cup fb\ z) \cap in_{C_1}, o_{C_1}!\Rrightarrow\{\}\ |\!\rangle$ and $z_2 = z \star \langle\!|\ i_{C_2}?\Rrightarrow(z.i_C? \cup fb\ z) \cap in_{C_2}, o_{C_2}!\Rrightarrow\{\}\ |\!\rangle$, we have,

$$\dfrac{\dfrac{z.o_C! = \{\}}{z.o_C! = \{\} \cup \{\}} \quad \begin{matrix}\zeta_1\\ \vdots\end{matrix} \quad \dfrac{}{z_1 \mathrel{\dot\in} \delta_{C_1}} \quad \begin{matrix}\zeta_2\\ \vdots\end{matrix} \quad \dfrac{}{z_2 \mathrel{\dot\in} \delta_{C_2}} \quad \dfrac{\dfrac{\dfrac{inactive\ C\ z}{inactive\ C_1\ z \wedge inactive\ C_2\ z}\,(df)}{\neg\ active\ C_1\ z \Leftrightarrow \neg\ active\ C_2\ z}}{active\ C_1\ z \Leftrightarrow active\ C_2\ z}}{z \in \delta_C}\,(|\_|_2^+)$$

where ζ_1 is:

$$\dfrac{\dfrac{z \mathrel{\dot\in} \Xi Chart_C}{z_1 \restriction T_1 \mathrel{\dot\in} \Xi Chart_{C_1}}\,(df) \quad \dfrac{inactive\ C\ z}{inactive\ C_1\ (z_1 \restriction T_1)}\,(df) \quad \dfrac{}{(z_1 \restriction T_1).o_{C_1}! = \{\}}\,(df)}{z_1 \restriction T_1 \in \delta_{C_1}}\,(iact^+)$$

and ζ_2 is:

$$\dfrac{\dfrac{z \mathrel{\dot\in} \Xi Chart_C}{z_2 \restriction T_2 \mathrel{\dot\in} \Xi Chart_{C_2}}\,(df) \quad \dfrac{inactive\ C\ z}{inactive\ C_2\ (z_2 \restriction T_2)}\,(df) \quad \dfrac{}{(z_2 \restriction T_2).o_{C_2}! = \{\}}\,(df)}{z_2 \restriction T_2 \in \delta_{C_2}}\,(iact^+)$$

where $[\![\delta_{C_1}]\!]^{\mathbb{P}\,T_1}$, $[\![\delta_{C_2}]\!]^{\mathbb{P}\,T_2}$.

Proposition 4.

$$\dfrac{}{[\![\delta_{C_1|\Psi|C_2}]\!] = [\![\delta_{C_2|\Psi|C_1}]\!]}\,(|\_|_{sym})$$

Proof

Trivial using the introduction and elimination rules for the composition operator.

Proposition 5. *Given* $C = C_1 \mid \Psi \mid C_2$,

$$\frac{z \in \delta_C \quad x \upharpoonright T_i \doteq z \quad z.i_C? \cup fb\, z = x.i_C? \cup fb\, x}{x \in \delta_C}\ (Z_t^{\in})$$

where $[\![\delta_C]\!]^{\mathbb{P}\,T}$, $T_i = T - [i_C? : \mathbb{P}\, in_C]$ *and* $fb\, z =_{def} z.o_C! \cap \Psi$.

Proof

Assuming $x_1 = (x \star \langle\!|\ i_{C_1}? \Rrightarrow (x.i_C? \cup fb\, x) \cap in_{C_1}, o_{C_1}! \Rrightarrow o_1\ |\!\rangle) \upharpoonright T_1$ and $x_2 = (x \star \langle\!|\ i_{C_2}? \Rrightarrow (x.i_C? \cup fb\, x) \cap in_{C_2}, o_{C_2}! \Rrightarrow o_2\ |\!\rangle) \upharpoonright T_2$, we have,

$$\dfrac{\dfrac{z \in \delta_C \qquad \dfrac{\dfrac{\dfrac{}{z.o_C! = y_1.o_{C_1}! \cup y_2.o_{C_1}!}\,1 \quad \dfrac{x \upharpoonright T_i \doteq z}{z.o_C! = x.o_C!}}{x.o_C! = y_1.o_{C_1}! \cup y_2.o_{C_1}!} \quad \begin{matrix}\zeta(1)\\ \vdots\end{matrix} \quad \begin{matrix}\zeta(2)\\ \vdots\end{matrix} \quad \dfrac{}{\begin{matrix}active\ C_1\ z \Leftrightarrow\\ active\ C_2\ z\end{matrix}}\,1}{x \in \delta_C}\ (|-|_2^+)}{x \in \delta_C}\ (|-|^-)(1)$$

where $\zeta(n)$ is[10]:

$$\dfrac{\dfrac{}{y_n \in \delta_{C_n}}\,1 \quad \begin{matrix}\zeta_2(n)\\ \vdots\\ x_n \upharpoonright T_{i_n} \doteq y_n\end{matrix} \quad \begin{matrix}\zeta_1(n)\\ \vdots\\ y_n.i_{C_n}? \cup fb\, y_n = x_n.i_{C_n}? \cup fb\, x_n\end{matrix}}{\dfrac{x_n \in \delta_{C_n}}{x \star \langle\!|\ i_{C_n}? \Rrightarrow (x.i_C? \cup fb\, x) \cap in_{C_n}, o_{C_n}! \Rrightarrow o_n\ |\!\rangle \mathrel{\dot\in} \delta_{C_n}}}\ (Z_t^{\in})$$

$\zeta_1(n)$ is,

$$\dfrac{\dfrac{\dfrac{z.i_C? \cup fb\, z = x.i_C? \cup fb\, x \qquad \dfrac{}{y_n.i_{C_n}? = (z.i_C? \cup fb\, z) \cap in_{C_n}}\,1}{y_n.i_{C_n}? = (x.i_C? \cup fb\, x) \cap in_{C_n}} \qquad \dfrac{\begin{matrix}\zeta_2(n)\\ \vdots\\ x_n \upharpoonright T_{i_n} \doteq y_n\end{matrix}}{\dfrac{x_n.o_{C_n}! = y_n.o_{C_n}!}{fb\, x_n = fb\, y_n}}}{y_n.i_{C_n}? \cup fb\, y_n = ((x.i_C? \cup fb\, x) \cap in_{C_n}) \cup fb\, y_n}}{\dfrac{y_n.i_{C_n}? \cup fb\, y_n = ((x.i_C? \cup fb\, x) \cap in_{C_n}) \cup fb\, x_n}{y_n.i_{C_n}? \cup fb\, y_n = x_n.i_{C_n}? \cup fb\, x_n}\ (x_n\text{-}df)}$$

[10] Due to the identical structure of the two proofs $\zeta(1)$ and $\zeta(2)$ we give a paramaterised verion of the proof in $\zeta(n)$. The proof $\zeta(1)$ is then simply $\zeta(n)$ with all occurances of n replaced by 1 and similarly for $\zeta(2)$

and $\zeta_2(n)$ is,

$$\dfrac{\dfrac{\dfrac{}{z \doteq y_n}\,1 \quad \dfrac{\dfrac{}{x_n \doteq x \upharpoonright T_i}\,(x_n\text{-}df) \qquad x \upharpoonright T_i \doteq z}{x_n \doteq z}\,(T_n \sqcap T_i \sqsubseteq T_i \sqcap T)}{x_n \doteq y_n}\,(T_n \sqcup T = T \sqcup T_n)}{x_n \upharpoonright T_{i_n} \doteq y_n}\,(T_{i_n} \sqsubseteq T_n)$$

where $[\![\delta_{C_1}]\!]^{\mathbb{P}\,T_1}, [\![\delta_{C_2}]\!]^{\mathbb{P}\,T_2}$, $T_{i_1} = T_1 - [i_{C_1}? : \mathbb{P}\,in_{C_1}]$ and $T_{i_2} = T_2 - [i_{C_2}? : \mathbb{P}\,in_{C_2}]$.

Lemma 3. *Given $C = C_1 \mid \{b\} \mid C_2$, for arbitrary $z \in T$ we have,*

$$\frac{\{C_1, C_2\} \subseteq z.active\_}{z \notin \delta_C}$$

where $[\![\delta_C]\!]^{\mathbb{P}\,T}$

Proof

$$\dfrac{\dfrac{}{z \in \delta_C}\,{\scriptstyle 1} \qquad \dfrac{\dfrac{\dfrac{}{y_1 \in \delta_{C_1}}\,{\scriptstyle 2} \qquad \dfrac{\dfrac{}{y_1 \doteq z}\,{\scriptstyle 2} \quad \{C_1, C_2\} \subseteq z.active\_}{active\ C_1\ y_1}\ (Z_t^{-1})}{\exists\, t_1 \in \delta_1 \bullet Trans\ t_1\ y_1} \qquad \begin{array}{c}\zeta_1\\ \vdots\end{array}}{\bot}\ (\exists^-)(3)}{\bot}\ (|\_|^-)(2)}{z \notin \delta_C}\ (\bot^-)(1)$$

where ζ_1 is:

$$\dfrac{\dfrac{\dfrac{\dfrac{}{t_1 \in \delta_1 \wedge Trans\ t_1\ y_1}\,{\scriptstyle 3}}{t_1 \in \{(A, B, a/b)\} \wedge Trans\ t_1\ y_1}\ (\delta_1\text{-}df)}{t_1 = (A, B, a/b) \wedge Trans\ t_1\ y_1}\ (df\text{-}\rho) \qquad \begin{array}{c}\zeta_2\\ \vdots\end{array}}{\dfrac{y_1.o_{C_1}! = \{b\} \qquad b \notin y_1.o_{C_1}!}{\bot}}$$

ζ_2 is:

$$\dfrac{\dfrac{\dfrac{}{y_2 \in \delta_{C_2}}\,{\scriptstyle 2} \qquad \dfrac{\dfrac{}{y_2 \doteq z}\,{\scriptstyle 2} \quad \{C_1, C_2\} \subseteq z.active\_}{\{C_1, C_2\} \subseteq y_2.active\_}}{active\ C_2\ y_2}\ (Z_t^{-1})}{\exists\, t_2 \in \delta_{C_2} \bullet Trans\ t_2\ y_2} \qquad \dfrac{\begin{array}{c}\zeta_3\\ \vdots\\ b \notin z.o_C!\end{array} \qquad \dfrac{}{\begin{array}{l}z.o_C! =\\ y_1.o_{C_1}! \cup y_2.o_{C_2}!\end{array}}\,{\scriptstyle 2}}{b \notin y_1.o_{C_1}!}}{b \notin y_1.o_{C_1}!}\ (\exists^-)(4)$$

and ζ_3 is:

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{y_2.i_{C_2}? = (z.i_C? \cup (z.o_C! \cap \Psi)) \cap in_{C_2}}\,{\scriptstyle 2}}{y_2.i_{C_2}? = (z.i_C? \cup (z.o_C! \cap \{b\})) \cap \{b\}} \qquad \dfrac{\dfrac{\dfrac{\dfrac{}{t_2 \in \delta_2 \wedge Trans\ t_2\ y_2}\,{\scriptstyle 4}}{t_2 \in \{(C, D, -b/c)\} \wedge Trans\ t_2\ y_2}\ (\delta_2\text{-}df)}{Trans\ (C, D, -b/c)\ y_2}\ (df\text{-}\rho)}{b \notin y_2.i_{C_2}? \cup (y_2.o_{C_2}! \cap \{\})}}{b \notin y_2.i_{C_2}?}}{b \notin (z.i_C? \cup (z.o_C! \cap \{b\})) \cap \{b\}}}{b \notin z.o_C!}$$

A.4 Proofs for Section 3.5: The decomposition operator

Proposition 6. *Given* $\delta_{M_S} = [\![Dec\ (\omega\ M)\ by\ \{(S, \omega\ S)\}]\!]_{Z_t}$, *for arbitrary* S, $\omega\ M = (M, \Sigma, \sigma, \Psi, \delta)$, *and* $\delta_{M||S} = [\![\omega\ M \mid \Psi \mid \omega\ S]\!]_{Z_t}$, *then for arbitrary* $z \in T$,

$$\frac{z.c_M = S \vee z.c'_M = S \quad z \in \delta_{M_S}}{z \in \delta_{M||S}}\ (M^-_{S2})$$

$$\frac{z.c_M = S \vee z.c'_M = S \quad z \in \delta_{M||S}}{z \in \delta_{M_S}}\ (M^+_{S2})$$

where $[\![\delta_{M_S}]\!]^{\mathbb{P}\ T}$.

Proof

Given that the following holds,

$$\frac{z.c_M = S \vee z.c'_M = S \quad active\ S\ z \Leftrightarrow (active\ M\ z \wedge (z.c_M = S \vee z.c'_M = S))}{active\ S\ z \Leftrightarrow active\ M\ z}$$

The proof of (M^-_{S2}) follows trivially using the rules (M^-_S) and $(|\_|^+)$. Similarly given,

$$\frac{z.c_M = S \vee z.c'_M = S \quad active\ S\ z \Leftrightarrow active\ M\ z}{active\ S\ z \Leftrightarrow (active\ M\ z \wedge (z.c_M = S \vee z.c'_M = S))}$$

The rule (M^+_{S2}) holds trivially using $(|\_|^-)$ and (M^+_S).

Proposition 7. *Given* $C = Dec\ (\omega\ M)$ *by* $\{(S, \omega\ S)\}$, *for arbitrary* S *and* $\omega\ M = (M, \Sigma, \sigma, \Psi, \delta)$ *and* $z \in T$,

$$\frac{z \in \delta_C \quad z.c_M \neq S \quad z.c'_M \neq S}{inactive\ S\ z}\ (M^-_{S3})$$

$$\frac{z \in \delta_C \quad z.c_M \neq S \quad z.c'_M \neq S}{z \star \langle\!|\ i_M?{\Rrightarrow} z.i_C? \cap in_M, o_M!{\Rrightarrow} z.o_C!\ |\!\rangle \mathrel{\dot\in} \delta_M}\ (M^-_{S4})$$

$$\frac{z \in \delta_C \quad z.c_M \neq S \quad z.c'_M \neq S}{z \star \langle\!|\ i_S?{\Rrightarrow} z.i_C? \cap in_S, o_S!{\Rrightarrow} \{\}\ |\!\rangle \mathrel{\dot\in} \delta_S}\ (M^-_{S5})$$

Proof

For (M^-_{S3}) we have,

$$\dfrac{z \in \delta_C \qquad \dfrac{\dfrac{\dfrac{\dfrac{}{z_m \doteq z}\,1 \qquad \dfrac{\dfrac{}{active\ S\ z}\,2 \qquad \dfrac{}{\begin{array}{l}active\ S\ z \Leftrightarrow\\ (active\ M\ z \wedge\\ (z_m.c_M = S \vee z_m.c'_M = S))\end{array}}\,1}{\begin{array}{l}active\ M\ z \wedge\\ (z_m.c_M = S \vee z_m.c'_M = S)\end{array}}}{z_m.c_M = S \vee z_m.c'_M = S}}{z.c_M = S \vee z.c'_M = S} \qquad \dfrac{z.c_M \neq S \quad z.c'_M \neq S}{\neg\ (z.c_M = S \vee z.c'_M = S)}}{\dfrac{\dfrac{\bot}{\neg\ active\ S\ z}\,(\bot^-)(2)}{inactive\ S\ z}\,(df)}\,(M^-_S)(1)}{inactive\ S\ z}$$

For $(M_{S\,4}^{-})$, assuming $[\![\delta_M]\!]^{\mathbb{P}\,T_m}$, $T_{i_m} = T_m - [i_M? : \mathbb{P}\,in_M]$ and $z_m = (z \star \langle\!| \; i_M? \Rrightarrow z.i_C? \cap in_M,\ o_M! \Rrightarrow z.o_C! \; |\!\rangle) \upharpoonright T_m$, we have,

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\zeta_1}{\vdots}}{y.i_M? \cup fb\ y_m = (z.i_C? \cap in_M) \cup fb\ z_m} \quad \dfrac{}{z_m.i_M? = z.i_C? \cap in_M}\ (z_m\text{-}df)}{y_m.i_M? \cup fb\ y_m = z_m.i_M? \cup fb\ z_m} \quad \dfrac{}{y_m \in \delta_M}\ 1 \quad \dfrac{\dfrac{\zeta_3}{\vdots}}{z_m \upharpoonright T_{i_m} \doteq y_m}}{z \in \delta_C \qquad z_m \in \delta_M}\ (Z_t^{\in})}{z_m \in \delta_M}\ (M_S^{-})(1)}{z \star \langle\!| \; i_M? \Rrightarrow z.i_C? \cap in_M,\ o_M! \Rrightarrow z.o_C! \; |\!\rangle \mathrel{\dot\in} \delta_M}$$

where ζ_1 is:

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\zeta_3}{\vdots}}{z_m \upharpoonright T_{i_m} \doteq y_m}}{z_m.o_M! = y_m.o_M!}}{fb\ z_m = fb\ y_m} \quad \dfrac{\dfrac{\dfrac{\zeta_2}{\vdots}}{((z.i_C? \cup fb\ z) \cap in_M) \cup fb\ z_m = (z.i_C? \cap in_M) \cup fb\ z_m} \quad \dfrac{}{y_m.i_M? = (z.i_C? \cup fb\ z) \cap in_M)}\ 1}{y.i_M? \cup fb\ z_m = (z.i_C? \cap in_M) \cup fb\ z_m}}{y.i_M? \cup fb\ y_m = (z.i_C? \cap in_M) \cup fb\ z_m}$$

ζ_2 is:

$$\dfrac{\dfrac{\dfrac{}{(z.i_C? \cap in_M) \cup fb\ z = (z.i_C? \cap in_M) \cup fb\ z}}{((z.i_C? \cup fb\ z) \cap in_M) \cup fb\ z = (z.i_C? \cap in_M) \cup fb\ z} \quad \dfrac{\dfrac{\dfrac{}{z.o_C! = \mathtt{z}_m.o_M!}\ (z_m\text{-}df)}{z.o_C! \cap \Psi = \mathtt{z}_m.o_M! \cap \Psi}}{fb\ z = fb\ z_m}}{((z.i_C? \cup fb\ z) \cap in_M) \cup fb\ z_m = (z.i_C? \cap in_M) \cup fb\ z_m}$$

and ζ_3 is:

$$\dfrac{\dfrac{\dfrac{\dfrac{}{z.o_C! = y_m.o_M! \cup y_s.o_S!}\ 1 \quad \dfrac{\dfrac{z \in \delta_C \quad z.c_M \neq S \quad z.c'_M \neq S}{inactive\ S\ z}\ (M_{S\,3}^{-}) \quad \dfrac{}{z \doteq y_s}\ 1}{inactive\ S\ y_s} \quad \dfrac{}{y_s \in \delta_S}\ 1}{y_s.o_S! = \{\}}\ (iact_2^{-})}{z.o_C! = y_m.o_M!} \quad \dfrac{}{z \doteq y_m}\ 1}{\langle\!| \; o_m! \Rrightarrow z.o_C! \; |\!\rangle \doteq y_m}}{z \star \langle\!| \; o_m! \Rrightarrow z.o_C! \; |\!\rangle \doteq y_m}}{z_m \upharpoonright T_{i_m} \doteq y_m}$$

Similarly, for $(M^-_{S\,5})$, assuming $[\![\delta_S]\!]^{\mathbb{P}\,T_s}$, $[\![\Xi\,Chart_S]\!]^{\mathbb{P}\,T_\Xi}$ and $z_s = (z \star \langle\!| \; i_S ? \Rrightarrow z.i_C ? \cap in_S,\; o_S ! \Rrightarrow \{\} \; |\!\rangle) \upharpoonright T_s$, we have,

$$
\dfrac{
  \dfrac{
    \dfrac{}{z \doteq y_s}\,1
    \qquad
    \dfrac{
      \dfrac{}{y_s \in \delta_S}\,1
      \qquad
      \dfrac{
        \begin{array}{c}\zeta_1\\ \vdots \\ inactive\ S\ z\end{array}
        \qquad
        \dfrac{}{z \doteq y_s}\,1
      }{inactive\ S\ y_s}\,(iact_1^-)
    }{y_s \mathrel{\dot\in} \Xi\,Chart_S}\,(T_\Xi \sqsubseteq T \sqcap T_s)
  }{z \mathrel{\dot\in} \Xi\,Chart_S}\,(T_\Xi \sqsubseteq T \sqcap T_s)
  \qquad
  \dfrac{}{z_s.o_S! = \{\}}\,(z_s\text{-}df)
  \qquad
  \dfrac{
    \begin{array}{c}\zeta_1\\ \vdots \\ inactive\ S\ z\end{array}
  }{inactive\ S\ z_s}
}{z_s \in \delta_S}\,(iact^+)
}{
  \dfrac{z \in \delta_C \qquad z_s \in \delta_S}{z_s \in \delta_S}\,(M^-_S)(1)
}{z \star \langle\!| \; i_S ? \Rrightarrow z.i_C ? \cap in_S, o_S ! \Rrightarrow \{\} \; |\!\rangle \mathrel{\dot\in} \delta_S}
$$

where ζ_1 is:

$$
\dfrac{z \in \delta_C \quad z.c_M \neq S \quad z.c'_M \neq S}{inactive\ S\ z}\,(M^-_{S\,3})
$$

Proposition 8. *Given* $C = Dec\ (\omega\ M)$ *by* $\{(S, \omega\ S)\}$, *for arbitrary* S *and* $\omega\ M = (M, \Sigma, \sigma, \Psi, \delta)$. *For arbitrary* $o_m, o_s \in \mathbb{P}\, in_C$ *and* $z \in T$,

$$\frac{\begin{array}{l} z.o_C! = o_m \cup o_s \\ z.c_M = S \\ z \star \langle\!|\ i_M?\Rrightarrow(z.i_C? \cup fb\ z) \cap in_M, o_M!\Rrightarrow o_m\ |\!\rangle \in \dot{\delta}_M \\ z \star \langle\!|\ i_S?\Rrightarrow(z.i_C? \cup fb\ z) \cap in_S, o_S!\Rrightarrow o_s\ |\!\rangle \in \dot{\delta}_S \\ active\ M\ z \Leftrightarrow active\ S\ z \end{array}}{z \in \delta_C}\ (M^+_{S\,3})$$

$$\frac{\begin{array}{l} z.o_C! = o_m \cup o_s \\ z.c'_M = S \\ z \star \langle\!|\ i_M?\Rrightarrow(z.i_C? \cup fb\ z) \cap in_M, o_M!\Rrightarrow o_m\ |\!\rangle \in \dot{\delta}_M \\ z \star \langle\!|\ i_S?\Rrightarrow(z.i_C? \cup fb\ z) \cap in_S, o_S!\Rrightarrow o_s\ |\!\rangle \in \dot{\delta}_S \\ active\ M\ z \Leftrightarrow active\ S\ z \end{array}}{z \in \delta_C}\ (M^+_{S\,4})$$

$$\frac{\begin{array}{l} z.c_M \neq S \quad z.c'_M \neq S \\ inactive\ S\ z \\ z \star \langle\!|\ i_M?\Rrightarrow(z.i_C? \cup fb\ z) \cap in_M, o_M!\Rrightarrow z.o_C!\ |\!\rangle \in \dot{\delta}_M \\ z \star \langle\!|\ i_S?\Rrightarrow(z.i_C? \cup fb\ z) \cap in_S, o_S!\Rrightarrow \{\}\ |\!\rangle \in \dot{\delta}_S \end{array}}{z \in \delta_C}\ (M^+_{S\,5})$$

where $[\![\delta_C]\!]^{\mathbb{P}\,T}$.

Proof

For $(M^+_{S\,3})$ we have,

$$\frac{\dfrac{z.c_M = S}{z.c_M = S \vee z.c'_M = S}\ (\vee^+) \qquad \dfrac{\begin{array}{l} z.o_C! = o_m \cup o_s \\ z \star \langle\!|\ i_M?\Rrightarrow(i_C? \cup fb\ z) \cap in_M, o_M!\Rrightarrow o_m\ |\!\rangle \in \dot{\delta}_M \\ z \star \langle\!|\ i_S?\Rrightarrow(z.i_C? \cup fb\ z) \cap in_S, o_S!\Rrightarrow o_s\ |\!\rangle \in \dot{\delta}_S \\ active\ M\ z \Leftrightarrow active\ S\ z \end{array}}{z \in \delta_{M||S}}\ (|\_|^+_2)}{z \in \delta_C}\ (M^+_{S\,2})$$

The proof of $(M^+_{S\,4})$ has the same structure as the previous proof. The only difference is that the disjuction introduction $(\vee^+)$ follows from the right disjunct rather than the left.

For (M^{+}_{S5}), assuming $z_1 = z \star \langle\!\langle\ i_M?\Rrightarrow(z.i_C? \cup \mathit{fb}\ z) \cap \mathit{in}_M, o_M!\Rrightarrow z.o_C!\ \rangle\!\rangle$, $z_m = z_1 \restriction T_m$, $z_2 = z \star \langle\!\langle\ i_S?\Rrightarrow(z.i_C? \cup \mathit{fb}\ z) \cap \mathit{in}_S, o_S!\Rrightarrow\{\}\ \rangle\!\rangle$ and $z_s = z_2 \restriction T_s$, we have,

$$\dfrac{\dfrac{z_1 \mathrel{\dot\in} \delta_M}{z_m \in \delta_M} \quad \dfrac{z_2 \mathrel{\dot\in} \delta_S}{z_s \in \delta_S} \quad \dfrac{\dfrac{}{z \doteq z_1}\,(z_1\text{-}df)}{z \doteq z_m}\,(z_m\text{-}df) \quad \dfrac{\dfrac{}{z \doteq z_2}\,(z_2\text{-}df)}{z \doteq z_s}\,(z_s\text{-}df) \quad \zeta_1 \quad \zeta_2 \quad \zeta_3 \quad \zeta_4}{z \in \delta_C}\,(M^{+}_{S})$$

where ζ_1 is:

$$\dfrac{\dfrac{\dfrac{\dfrac{}{z.o_C! = z.o_C!}\,(\mathit{ref})}{z.o_C! = z.o_C! \cup \{\}}}{z.o_C! = z_1.o_M! \cup z_2.o_S!}\,(z_m \& z_s\text{-}df)}{z.o_C! = z_m.o_M! \cup z_s.o_S!}$$

ζ_2 is:

$$\dfrac{}{z_m.i_M? = (z.i_C? \cup \mathit{fb}\ z) \cap \mathit{in}_M}\,(z_m\text{-}df)$$

ζ_3 is:

$$\dfrac{}{z_s.i_S? = (z.i_C? \cup \mathit{fb}\ z) \cap \mathit{in}_S}\,(z_s\text{-}df)$$

and ζ_4 is:

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{z.c_M \neq S}{z_m.c_M \neq S}\,(z_1 \& z_m\text{-}df) \quad \dfrac{z.c'_M \neq S}{z_m.c'_M \neq S}\,(z_1 \& z_m\text{-}df)}{z_m.c_M \neq S \wedge z_m.c'_M \neq S}}{\mathit{inactive}\ M\ z \vee (z_m.c_M \neq S \wedge z_m.c'_M \neq S)}\,(\vee^{+}) \quad \mathit{inactive}\ S\ z}{\mathit{inactive}\ S\ z \wedge (\mathit{inactive}\ M\ z \vee (z_m.c_M \neq S \wedge z_m.c'_M \neq S))}}{\mathit{inactive}\ S\ z \Leftrightarrow (\mathit{inactive}\ M\ z \vee (z_m.c_M \neq S \wedge z_m.c'_M \neq S))}}{\mathit{active}\ S\ z \Leftrightarrow (\mathit{active}\ M\ z \wedge (z_m.c_M = S \vee z_m.c'_M = S))}$$

where $[\![\delta_M]\!]^{\mathbb{P}\ T_m}$ and $[\![\delta_S]\!]^{\mathbb{P}\ T_s}$.

Proposition 9. *Given* $C = Dec\ (\omega\ M)$ *by* $\{(S, \omega\ S)\}$, *for arbitrary* S, $\omega\ M = (M, \Sigma, \sigma, \Psi, \delta)$ *and* $z \in T$,

$$\frac{z \in \delta_C \quad inactive\ C\ z}{z \mathrel{\dot\in} \Xi\, Chart_C}\ (iact_1^-) \qquad \frac{z \in \delta_C \quad inactive\ C\ z}{z.o_C! = \{\}}\ (iact_2^-)$$

$$\frac{inactive\ C\ z \quad z \mathrel{\dot\in} \Xi\, Chart_C \quad z.o_C! = \{\}}{z \in \delta_C}\ (iact^+)$$

where $[\![\delta_C]\!]^{\mathbb{P}\ T}$.

Proof

For $(iact_1^-)$ we have,

$$\dfrac{\dfrac{\dfrac{}{z \doteq y_m}\,1 \quad \dfrac{inactive\ C\ z}{inactive\ M\ z}\ (inactive\text{-}df)}{\dfrac{}{y_m \in \delta_M}\,1 \quad inactive\ M\ y_m}\ (Z_{i\ 2}^-) \qquad \dfrac{y_m \mathrel{\dot\in} \Xi\, Chart_M}{y_m \restriction T \mathrel{\dot\in} \Xi\, Chart_M}\ (z \in T)}{\dfrac{}{z \doteq y_m}\,1 \quad z \mathrel{\dot\in} \Xi\, Chart_M} \qquad \begin{array}{c}\zeta\\ \vdots\\ z \mathrel{\dot\in} \Xi\, Chart_S\end{array}\ S_\wedge^+$$

$$\dfrac{\dfrac{z \in \delta_C \qquad z \mathrel{\dot\in} (\Xi\, Chart_M \wedge \Xi\, Chart_S)}{z \mathrel{\dot\in} \Xi\, Chart_C}\ (Chart_C\text{-}df)}{z \mathrel{\dot\in} \Xi\, Chart_C}\ (M_S^-)(1)$$

where ζ is:

$$\dfrac{\dfrac{\dfrac{}{z \doteq y_s}\,1 \quad \dfrac{inactive\ C\ z}{inactive\ S\ z}\ (inactive\text{-}df)}{\dfrac{}{y_s \in \delta_S}\,1 \quad inactive\ S\ y_s}\ (iact_1^-) \qquad \dfrac{y_s \mathrel{\dot\in} \Xi\, Chart_S}{y_s \restriction T \mathrel{\dot\in} \Xi\, Chart_S}\ (z \in T)}{\dfrac{}{z \doteq y_s}\,1 \quad z \mathrel{\dot\in} \Xi\, Chart_S}$$

For $(iact_2^-)$ we have,

$$\dfrac{\dfrac{\dfrac{\dfrac{}{z \doteq y_m}\,1 \quad \dfrac{inactive\ C\ z}{inactive\ M\ z}\ (inactive\text{-}df)}{\dfrac{}{y_m \in \delta_M}\,1 \quad inactive\ M\ y_m}\ (iact_2^-) \qquad \begin{array}{c}\zeta_1\\ \vdots\\ y_m.o_{C_1}! = \{\}\end{array} \qquad \dfrac{}{z.o_C! = y_m.o_{C_1}! \cup y_s.o_{C_2}!}\,1}{z \in \delta_C \qquad z.o_C! = \{\}}\ (M_S^-)(1)}{z.o_C! = \{\}}$$

where ζ_1 is:

$$\dfrac{\dfrac{}{y_s \in \delta_S}\,{}^{1} \quad \dfrac{\dfrac{}{z \doteq y_s}\,{}^{1} \quad \dfrac{inactive\ C\ z}{inactive\ S\ z}\ (inactive\text{-}df)}{inactive\ S\ y_s}}{y_m.o_{C_2}! = \{\}}\ (iact_2^-)$$

And for $(iact^+)$, assuming $z_1 = z \star \langle\!|\ i_M?\Rrightarrow(z.i_C? \cup fb\ z) \cap in_M, o_M!\Rrightarrow\{\}\ |\!\rangle$, $z_m = z_1 \restriction T_m$, $z_2 = z \star \langle\!|\ i_S?\Rrightarrow(z.i_C? \cup fb\ z) \cap in_S, o_S!\Rrightarrow\{\}\ |\!\rangle$ and $z_s = z_2 \restriction T_s$, we have,

$$\dfrac{\begin{array}{cccccccc}\zeta_2 & \zeta_3 & & & \zeta_4 & \zeta_5 & \zeta_6 & \zeta_7\\ \vdots & \vdots & & & \vdots & \vdots & \vdots & \vdots\\ z_m \in \delta_M & z_s \in \delta_S & \dfrac{}{z \doteq z_m}\ (z_m\text{-}df) & \dfrac{}{z \doteq z_s}\ (z_s\text{-}df) & & & & \end{array}}{z \in \delta_C}\ (M_S^+)$$

where ζ_2 is:

$$\dfrac{\dfrac{z \dot{\in} \Xi Chart_C}{z_m \dot{\in} \Xi Chart_M}\ (df) \quad \dfrac{\dfrac{inactive\ C\ z}{inactive\ M\ z}\ (inactive\text{-}df)}{inactive\ M\ z_m}\ (z_m\text{-}df) \quad \dfrac{}{z_m.o_M! = \{\}}\ (z_m\text{-}df)}{z_m \in \delta_M}\ (iact^+)$$

ζ_3 is:

$$\dfrac{\dfrac{z \dot{\in} \Xi Chart_C}{z_s \dot{\in} \Xi Chart_S}\ (df) \quad \dfrac{\dfrac{inactive\ C\ z}{inactive\ S\ z}\ (inactive\text{-}df)}{inactive\ S\ z_s}\ (z_s\text{-}df) \quad \dfrac{}{z_s.o_S! = \{\}}\ (z_s\text{-}df)}{z_s \in \delta_S}\ (iact^+)$$

ζ_4 is:

$$\dfrac{\dfrac{\dfrac{}{z.o_C! = \{\}}}{z.o_C! = \{\} \cup \{\}}}{z.o_C! = z_m.o_M! \cup z_s.o_S!}\ (z_m \& z_s\text{-}df)$$

ζ_5 is:

$$\dfrac{}{z_m.i_M? = (z.i_C? \cup fb\ z) \cap in_M}\ (z_m\text{-}df)$$

ζ_6 is:

$$\dfrac{}{z_s.i_S? = (z.i_C? \cup fb\ z) \cap in_S}\ (z_s\text{-}df)$$

and ζ_7 is:

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{inactive\ C\ z}{inactive\ M\ z \wedge inactive\ S\ z}\ (inactive\text{-}df)}{inactive\ M\ z \Leftrightarrow inactive\ S\ z}}{\neg\ active\ M\ z \Leftrightarrow \neg\ active\ S\ z}}{active\ M\ z \Leftrightarrow active\ S\ z}}{active\ S\ z \Leftrightarrow (active\ M\ z \wedge (z_m.c_M = S \vee z_m.c'_M = S))}$$

where $[\![\delta_M]\!]^{\mathbb{P}\ T_m}$, $[\![\delta_S]\!]^{\mathbb{P}\ T_s}$.

Proposition 10. *Given* $C = Dec\ (\omega\ M)$ *by* $\{(S, \omega\ S)\}$, *for arbitrary* S *and* $\omega\ M = (M, \Sigma, \sigma, \Psi, \delta)$,

$$\frac{z \in \delta_C \quad x \upharpoonright T_i \doteq z \quad z.i_C? \cup fb\ z = x.i_C? \cup fb\ x}{x \in \delta_C}\ (Z_t^{\in})$$

where $[\![\delta_C]\!]^{\mathbb{P}\ T}$, $T_i = T - [i? : \mathbb{P}\ in_C]$ *and* $fb\ z =_{def} z.o_C! \cap \Psi$.

Proof

The proof of this proposition is split into two cases. First consider the case when the master chart is in the state of the decomposed chart, *i.e.* $c_M = S \vee c'_M = S$. Because in this case the master and slave react exactly as though they are composed in parallel the proof is trivial given that $(Z_t^{\in})$ holds for composed charts.

$$\dfrac{z.c_M = S \vee z.c'_M = S \quad \dfrac{z \in \delta_C \quad z.c_M = S \vee z.c'_M = S}{z \in \delta_{M||S}}\ (M_{S\,2}^{-}) \quad x \upharpoonright T_i \doteq z \quad z.i_C? \cup fb\ z = x.i_C? \cup fb\ x}{x \in \delta_{M||S}}\ (Z_t^{\in})}{x \in \delta_C}\ (M_{S\,2}^{+})$$

Note that the use of the rule $(Z_t^{\in})$ in the proof of the rule itself is valid here because we have already shown that $(Z_t^{\in})$ holds for composed charts.

The second case is when the slave is not active, *i.e.* $c_M \neq S \wedge c'_M \neq S$. Assuming $[\![\delta_M]\!]^{\mathbb{P}\ T_m}$ and $[\![\delta_S]\!]^{\mathbb{P}\ T_s}$ we have,

$$\dfrac{\dfrac{z.c_M \neq S \wedge z.c'_M \neq S \quad x \upharpoonright T_i \doteq z}{x.c_M \neq S \wedge x.c'_M \neq S} \quad \begin{matrix}\zeta_1 \\ \vdots\end{matrix} \quad \begin{matrix}\zeta_4 \\ \vdots\end{matrix} \quad \dfrac{z \in \delta_C \quad z.c_M \neq S \wedge z.c'_M \neq S}{inactive\ S\ z}\ (M_{S\,3}^{-}) \quad x \upharpoonright T_i \doteq z}{inactive\ S\ x}\ (M_{S\,5}^{+})}{x \in \delta_C}$$

where ζ_1 is, assuming $z_m = (z \star \langle\!|\ i_M? \Rrightarrow (z.i_C? \cup fb\ z) \cap in_M, o_M! \Rrightarrow z.o_C!\ |\!\rangle) \upharpoonright T_m$, $x_m = (x \star \langle\!|\ i_M? \Rrightarrow (x.i_C? \cup fb\ x) \cap in_M, o_M! \Rrightarrow x.o_C!\ |\!\rangle) \upharpoonright T_m$ and $T_{i_m} = T_m - [i? : \mathbb{P}\ in_M]$:

$$\dfrac{\dfrac{\dfrac{z \in \delta_C \quad z.c_M \neq S \quad z.c'_M \neq S}{z_1 \in \delta_M}\ (M_{S\,4}^{-})}{z_m \in \delta_M} \quad \begin{matrix}\zeta_2 \\ \vdots\end{matrix} \quad \begin{matrix}\zeta_3 \\ \vdots\end{matrix}}{x_m \in \delta_M} \quad \begin{matrix}x_m \upharpoonright T_{i_m} \doteq z_m \\ \end{matrix} \quad \begin{matrix}z_m.i_M? \cup fb\ z_m = \\ x_m.i_M? \cup fb\ x_m\end{matrix}\ (Z_t^{\in})$$

$$x \star \langle\!|\ i_M? \Rrightarrow (x.i_C? \cup fb\ x) \cap in_M, o_M! \Rrightarrow x.o_C!\ |\!\rangle \in \delta_M$$

ζ_2 is:

$$\dfrac{\dfrac{x \upharpoonright T_i \doteq z}{x.o_C! = z.o_C!}}{\dfrac{x \upharpoonright T_i \doteq z \quad \langle\!|\ o_M! \Rrightarrow x.o_C!\ |\!\rangle = \langle\!|\ o_M! \Rrightarrow z.o_C!\ |\!\rangle}{(x \star \langle\!|\ o_M! \Rrightarrow x.o_C!\ |\!\rangle) \upharpoonright T_i \doteq z \star \langle\!|\ o_M! \Rrightarrow z.o_C!\ |\!\rangle}}{x_m \upharpoonright T_{i_m} \doteq z_m}\ (T \upharpoonright T_{i_m} \sqsubseteq T \upharpoonright T_i)$$

ζ_3 is:

$$\dfrac{\dfrac{\dfrac{x \upharpoonright T_i \doteq z}{x.o_C! = z.o_C!}}{x_m.o_M! = z_m.o_M!}\ (z_m \& x_m\text{-}df)}{fb\ x_m = fb\ z_m} \qquad \dfrac{\dfrac{\dfrac{z.i_C? \cup fb\ z = x.i_C? \cup fb\ x}{(z.i_C? \cup fb\ z) \cap in_M = (x.i_C? \cup fb\ x) \cap in_M}}{z_m.i_M? = x_m.i_M?}\ (z_m \& x_m\text{-}df)}{z_m.i_M? \cup fb\ z_m = x_m.i_M? \cup fb\ z_m}$$

$$z_m.i_M? \cup fb\ z_m = x_m.i_M? \cup fb\ x_m$$

and similarly ζ_4 is, assuming $z_s = (z \star \langle\!|\ i_S? \Rrightarrow (z.i_C? \cup fb\ z) \cap in_S, o_S! \Rrightarrow \{\}\ |\!\rangle) \upharpoonright T_s$, $x_s = (x \star \langle\!|\ i_S? \Rrightarrow (x.i_C? \cup fb\ x) \cap in_S, o_S! \Rrightarrow \{\}\ |\!\rangle) \upharpoonright T_s$ and $T_{i_s} = T_s - [i? : \mathbb{P}\ in_S]$:

$$\dfrac{\dfrac{\dfrac{z \in \delta_C \quad z.c_M \neq S \quad z.c'_M \neq S}{z \star \langle\!|\ i_S? \Rrightarrow (z.i_C? \cup fb\ z) \cap in_S, o_S! \Rrightarrow \{\}\ |\!\rangle \in \dot{\delta}_S}\ (M_{S\,5}^{-}) \qquad \dfrac{x \upharpoonright T_i \doteq z}{x_s \upharpoonright T_{i_s} \doteq z_s}\ \begin{matrix}\zeta_5\\ \vdots\end{matrix}}{z_s \in \delta_S}\ (Z_t^{\in})}{x_s \in \delta_S}}{x \star \langle\!|\ i_S? \Rrightarrow (x.i_C? \cup fb\ z) \cap in_S, o_S! \Rrightarrow \{\}\ |\!\rangle \in \dot{\delta}_S}$$

ζ_5 is:

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{z.i_C? \cup fb\ z = x.i_C? \cup fb\ x}{(z.i_C? \cup fb\ z) \cap in_S = (x.i_C? \cup fb\ x) \cap in_S}}{z_s.i_S? = x_s.i_S?}}{z_s.i_S? \cup (\{\} \cap \Psi_s) = x_s.i_S? \cup (\{\} \cap \Psi_s)}\ (z_s \& x_s\text{-}df)}{z_s.i_S? \cup (z_s.o_S! \cap \Psi_s) = x_s.i_S? \cup (x_s.o_S! \cap \Psi_s)}\ (fb\text{-}df)}{z_s.i_S? \cup fb\ z_s = x_s.i_S? \cup fb\ x_s}$$

A.5 Proofs for Section 3.6: The hiding operator

Proposition 11. *Given* $C =_X [C_1]_Y$, *for arbitrary* $z \in T$,

$$\frac{z \in \delta_C \quad inactive\ C\ z}{z \dot{\in} \Xi Chart_C}\ (iact_1^-) \qquad \frac{z \in \delta_C \quad inactive\ C\ z}{z.o_C! = \{\}}\ (iact_2^-)$$

$$\frac{inactive\ C\ z \quad z \dot{\in} \Xi Chart_C \quad z.o_C! = \{\}}{z \in \delta_C}\ (iact^+)$$

where $[\![\delta_C]\!]^{\mathbb{P}\ T}$.

Proof

For $(iact_1^-)$ we have,

$$\dfrac{\dfrac{\dfrac{}{y_1 \in \delta_{C_1}}\,1 \quad \dfrac{\dfrac{}{z \doteq y_1}\,1 \quad \dfrac{inactive\ C\ z}{inactive\ C_1\ z}\ (inact\text{-}df)}{inactive\ C_1\ y_1}\ (iact_1^-)}{y_1 \dot{\in} \Xi Chart_{C_1}} \qquad \dfrac{}{z \doteq y_1}\,1}{\dfrac{z \in \delta_C \qquad \dfrac{z \dot{\in} \Xi Chart_{C_1}}{z \dot{\in} \Xi Chart_C}\ (Chart_C\text{-}df)}{z \dot{\in} \Xi Chart_C}\ (_X[]^-_Y)(1)}$$

For $(iact_2^-)$ we have,

$$\dfrac{\dfrac{\dfrac{}{y_1 \in \delta_{C_1}}\,1 \quad \dfrac{\dfrac{}{z \doteq y_1}\,1 \quad \dfrac{inactive\ C\ z}{inactive\ C_1\ z}\ (inact\text{-}df)}{inactive\ C_1\ y_1}\ (iact_1^-)}{y_1.o_{C_1}! = \{\}} \qquad \dfrac{}{\begin{array}{l} z.o_C! = \\ y_1.o_{C_1}! \cap out_C \end{array}}\,1}{\dfrac{z \in \delta_C \qquad z.o_C! = \{\}}{z.o_C! = \{\}}\ (_X[]^-_Y)(1)}\ (Chart_C\text{-}df)$$

And for $(iact^+)$, assuming $z_1 = z \star \langle\!|\ i_{C_1}?{\Rrightarrow}z.i_C?, o_{C_1}{\Rrightarrow}\{\}\ |\!\rangle$, we have,

$$\dfrac{\begin{array}{c}\zeta \\ \vdots \\ z_1 \in \delta_{C_1}\end{array} \quad \dfrac{}{z \doteq z_1}\ (z_1\text{-}df) \quad \dfrac{}{z.i_C? = z_1.i_{C_1}?} \quad \dfrac{\dfrac{z.o_C! = \{\}}{z.o_C! = \{\} \cap out_C}}{z.o_C! = z_1.o_{C_1}! \cap out_C}\ (z_1\text{-}df)}{z \in \delta_C}\ (_X[]^+_Y)$$

where ζ is:

$$\dfrac{\dfrac{\dfrac{z \mathrel{\dot\in} \Xi\, Chart_C}{z_1 \mathrel{\dot\in} \Xi\, Chart_C}\,(z_1\text{-}df)}{z_1 \mathrel{\dot\in} \Xi\, Chart_{C_1}}\,(Chart_{C_1}\text{-}df) \qquad \dfrac{}{z_1.o_{C_1}! = \{\}}\,(z_1\text{-}df) \qquad \dfrac{\dfrac{inactive\ C\ z}{inactive\ C_1\ z}\,(inactive\text{-}df)}{inactive\ C_1\ z_1}\,(z_1\text{-}df)}{z_1 \in \delta_{C_1}}\,(iact^+)$$

Proposition 12. *Given* $C =_X [C_1]_Y$ *and assuming that* Ψ *is defined to be the same as the feedback set* Ψ_1 *for the chart* C_1,

$$\frac{z \in \delta_C \quad x \upharpoonright T_i \doteq z \quad z.i_C? \cup fb\, z = x.i_C? \cup fb\, x}{x \in \delta_C}\ (Z_i^{\in})$$

where $[\![\delta_C]\!]^{\mathbb{P}\,T}$, $T_i = T - [i_C? : \mathbb{P}\, in_C]$ *and* $fb\, z =_{def} z.o_C! \cap \Psi$.

Proof

Assuming $x_1 = (z_1 \upharpoonright T_{i_1}) \star \langle\!\langle\ i_{C_1}? \Rrightarrow x.i_C?\ \rangle\!\rangle$, $[\![C_1]\!]^{\mathbb{P}\,T_1}$ and $T_{i_1} = T_1 - [i? : \mathbb{P}\, in_{C_1}]$,

$$\dfrac{z \in \delta_C \qquad \dfrac{\begin{array}{cccc} \zeta_1 & \zeta_2 & & \zeta_3 \\ \vdots & \vdots & & \vdots \\ x_1 \in \delta_{C_1} & x \doteq x_1 & \dfrac{}{x.i_C? = x_1.i_{C_1}?}\ (x_1\text{-}df) & x.o_C! = x_1.o_{C_1}! \cap out_C \end{array}}{x \in \delta_C}\ (x\,[]_{\overline{Y}}^{+})}{x \in \delta_C}\ (x\,[]_{\overline{Y}})(1)$$

where ζ_1 is:

$$\dfrac{\dfrac{\dfrac{\dfrac{x \upharpoonright T_i \doteq z}{o_C! = z.o_C!}}{x.o_C! \cap \Phi = z.o_C! \cap \Phi}}{fb\, x = fb\, z} \qquad \dfrac{\dfrac{z.i_{C_1}? \cup fb\, z = x.i_C? \cup fb\, x}{z.i_{C_1}? \cup fb\, z = x_1.i_{C_1}? \cup fb\, x}\ (x_1\text{-}df) \qquad \dfrac{}{z.i_C? = z_1.i_{C_1}?}\ 1}{z_1.i_{C_1}? \cup fb\, z = x_1.i_{C_1}? \cup fb\, x}}{\dfrac{z_1.i_{C_1}? \cup fb\, z = x_1.i_{C_1}? \cup fb\, z}{z_1.i_{C_1}? = x_1.i_{C_1}?}} \qquad \dfrac{\dfrac{x_1.o_{C_1}! = z_1.o_{C_1}!}{x_1.o_{C_1}! \cap \Phi_1 = z_1.o_{C_1}! \cap \Phi_1}\ (x_1\text{-}df)}{fb\, x_1 = fb\, z_1}$$

$$\dfrac{\dfrac{}{z_1 \in \delta_{C_1}}\ 1 \qquad \dfrac{z_1.i_{C_1}? \cup fb\, z_1 = x_1.i_{C_1}? \cup fb\, x_1}{z_1.i_{C_1}? \cup fb\, z_1 = x_1.i_{C_1}? \cup fb\, x_1} \qquad \dfrac{}{x_1 \upharpoonright T_{i_1} \doteq z_1}\ (x_1\text{-}df)}{x_1 \in \delta_{C_1}}\ (Z_i^{\in})$$

ζ_2 is:

$$\dfrac{\dfrac{x \upharpoonright T_i \doteq z \qquad \dfrac{}{z \doteq z_1}\ 1}{x \upharpoonright T_i \doteq z_1}\ (T_{\sqcap} T_1 \sqsubseteq T_i)}{x \doteq z_1}\ (T \sqcap T_1 \sqsubseteq T_i \sqcap T_1) \qquad \dfrac{}{x_1 \upharpoonright T_{i_1} \doteq z_1 \upharpoonright T_{i_1}}\ (x_1\text{-}df)}{x \doteq x_1}\ (T \sqcap T_1 \sqsubseteq T_{i_1})$$

and ζ_3 is:

$$\dfrac{\dfrac{x \upharpoonright T_i \doteq z}{x.o_C! = z.o_C!} \qquad \dfrac{\dfrac{}{z.o_C! = z_1.o_{C_1}! \cap out_C}\,{}^{1} \qquad \dfrac{}{z_1.o_{C_1}! = x_1.o_{C_1}!}\,{}^{(x_1\text{-}df)}}{z.o_C! = x_1.o_{C_1}! \cap out_C}}{x.o_C! = x_1.o_{C_1}! \cap out_C}$$