Process algebra with local communication

Process Algebra with Local Communication

Muck van Weerdenburg

Department of Mathematics and Computer Science, Eindhoven University of Technology

P.O. Box 513, 5600 MB Eindhoven, The Netherlands

M.J.van.Weerdenburg@tue.nl

Abstract

In process algebras like μCRL [6] and ACP [2] communication is defined globally. In the context of component based architectures one wishes to define subcomponents of a system separately, including communication within that subcomponent. In this document we define a process algebra that has a local communication function that facilitates component based architectures.

1 Introduction

In modelling systems, component based architectures are a natural way of separating different parts (and sub-parts) of a system and specifying how these parts relate to each other by means of, for example, communication. A way of specifying such systems is by the use of process algebras, which are formally defined and facilitate verification. In process algebras like μCRL [6] and ACP [2] one can put components in parallel and define communication globally. Parallel operators interleave actions *or* let them communicate, if they are allowed to by the global communication function. However, because in the context of component based architectures one wishes to define subcomponents of a system separately, including communication within that subcomponent, globally defined communication requires one to keep in mind which actions and communications are used in other components to be able to define proper communication of one of the components.

To support hierarchical modelling, we define a process algebra (*LoCo*) that has a (local) communication operator instead of the global communication function. When introducing such an operator, however, one has to reconsider the semantics of parallel operators. With global communication in ACP, and the like, one has the auxiliary parallel operator |, which forces communication on the first actions of the process parameters. If communication between a pair of actions was not defined, deadlock was the result. But one does not expect a local communication operator to have a similar effect on an operator somewhere within the process that is its parameter. Instead one wishes it to behave like any other operator, that is affecting the actions of a process parameter (independent of the operators that "produced" such an action).

A solution to this situation are multiactions. Multiactions are collections of actions that happen simultaneously, but independently (i.e. they do not communicate). So instead of letting | force communication, it will now *synchronise* (multi)actions, in which communication can be applied by a communication operator. Such multiactions have been used before to add synchrony to CCS [5] or BPA [7].

Note that in this document we will not consider the use of data, but addition of data is straightforward.

We first describe the components of our algebra informally. The basic elements of processes are multiactions. Multiactions are bags of actions that execute together. We write a multiaction of actions a, b and c as $\langle a, b, c \rangle$ (or $\langle b, a, c \rangle$ as order has no meaning in bags). Often we write multiactions

that consist of only one action as that action alone (i.e. a instead of $\langle a \rangle$). We can combine such multi-actions with the common operators $\cdot$ and $+$ to form a sequence of multiactions or a nondeterministic choice between multiactions, respectively. To denote inaction or deadlock we write δ.

For parallel composition we have the merge $\parallel$ which interleaves and/or synchronises multiactions. For example, $(a + b) \parallel (c \cdot d)$ has the same behaviour as $(a + b) \cdot c \cdot d + c \cdot ((a + b) \cdot d + d \cdot (a + b) + (\langle a, d \rangle + \langle b, d \rangle)) + (\langle a, c \rangle + \langle b, c \rangle) \cdot d$. A new communication operator Γ allows explicit specification of (two or more) actions that communicate with each other (e.g. $\Gamma_{\{a|b \to c\}}(P)$, which means that a and b communicate to c in process P), besides just being synchronised. To limit the behaviour of a process, it has been common to define which actions are not allowed. However, as the number of multiactions we want to prohibit can increase exponentially with the number of parallel processes, we added a restriction operator ∇ that specifies precisely which multiactions are allowed, by a set V of action sequences (e.g. $V = \{a\}$ or $V = \{b|c|c, d, c|e\}$). If one wishes that in the parallel composition of a,b and c action a does not execute synchronised with another action and b and c must synchronise, one can write $\nabla_{\{a,b|c\}}(a \parallel b \parallel c)$, which behaves as $a \cdot \langle b, c \rangle + \langle b, c \rangle \cdot a$.

The blocking operator ∂_H (commonly referred to as encapsulation operator) prohibits actions in its set parameter H from executing (e.g. $\partial_{\{a\}}(a + b \cdot \langle a, c \rangle)$, which behaves as $b \cdot \delta$), the hiding operator τ_I makes actions in I invisible (e.g. $\tau_{\{a\}}(\langle a, b \rangle)$ becomes $\langle b \rangle$) and the renaming operator ρ renames actions (e.g. $\rho_{\{a \to b\}}(a)$ becomes b). The special case of the empty multiaction $\langle \rangle$ is called a silent step, which we often write as τ. Finally we have process variables with which we can write equations as $X = a \cdot X$ to denote the process that can do infinitely many a's.

In the rest of this document we formally define the syntax and semantics of *LoCo* and introduce an axiomatisation, which we will show to be sound and complete. We also include some examples and a set of alphabet axioms.

2   Syntax

For the description of the syntax of *LoCo* we use a BNF like notation. We use $\mid$ to separate alternatives, $[\,]$ to specify an optional part and $(\,)^*$ for zero or more occurrences of an expression. As terminals we use sets, meaning that any element of that set is allowed at that position.

Definition 2.1.   Let $\mathcal{N}_A$ be the set of action names and X the set of process variables. Our process algebra *LoCo* has the following syntax. (A_M describes (multi)actions, N "communication action names", V sets of N (for ∇), C sets of communication relations (for Γ), IH sets of action names (to be hidden with τ or blocked with ∂), R renaming functions (for ρ), T_P *LoCo* terms and E *LoCo* expressions.)

$$
\begin{array}{rcl}
A_M & ::= & \mathcal{N}_A \mid \langle [\mathcal{N}_A \, (,\mathcal{N}_A)^*] \rangle \\
N & ::= & \mathcal{N}_A \, (\mid \mathcal{N}_A)^* \\
N^+ & ::= & \mathcal{N}_A \mid \mathcal{N}_A \, (\mid \mathcal{N}_A)^* \\
V & ::= & \{N \, (,N)^*\} \\
C & ::= & \{N^+ \to \mathcal{N}_A \, (,N^+ \to \mathcal{N}_A)^*\} \\
IH & ::= & \{\mathcal{N}_A \, (,\mathcal{N}_A)^*\} \\
R & ::= & \{\mathcal{N}_A \to \mathcal{N}_A \, (,\mathcal{N}_A \to \mathcal{N}_A)^*\} \\
T_P & ::= & A_M \mid \delta \mid \tau \mid T_P + T_P \mid T_P \cdot T_P \mid T_P \parallel T_P \mid T_P \,\underline{\parallel}\, T_P \mid T_P | T_P \mid X \mid \\
& & (T_P) \mid \nabla_V(T_P) \mid \Gamma_C(T_P) \mid \partial_{IH}(T_P) \mid \tau_{IH}(T_P) \mid \rho_R(T_P) \\
E & ::= & X = T_P \mid E, E
\end{array}
$$

As not all (infix) operators are necessarily enclosed within parentheses (e.g. we may write $a \cdot b + c$), terms can be interpreted in different ways (e.g. $(a \cdot b) + c$ or $a \cdot (b + c)$). To overcome this problem we give all infix operators a binding strength with the meaning that if, for example, $\cdot$ binds stronger

than $+$ we interpret $a \cdot b + c$ as $(a \cdot b) + c$. The order of the operators (in decreasing strength) is as follows: $\cdot, \mid, \parallel\!\!\!\lfloor\,, \parallel, +$. We assume all infix operators are right associative.

Instead of writing the sequence of terms $t_1, t_2, \ldots, t_n$ (e.g. the actions in a multiaction) we often write $\overrightarrow{t}$. The set of all actions A is defined by $\mathcal{N}_A$, and the set of all multiactions $\mathbb{A}$ is defined by $\{\langle \overrightarrow{a} \rangle \mid \overrightarrow{a} \in \overrightarrow{A}\}$. We also write α instead of the action $\langle \overrightarrow{a} \rangle$.

To be able to reason about terms of our language, we introduce some sets and notations. The set V_P, with elements $x, y, \ldots$, consists of process variables. For the set of *LoCo* terms (described by) T_P we have elements $t, u, \ldots$ and process-closed terms $p, q, \ldots$ in T_{pc} (terms that do not have any process variables in them).

Note that this syntax allows one to write sets (R and C) that can contain elements with the same left hand side (e.g. $\{a \rightarrow b, a \rightarrow c\}$). This should not be possible as the meaning of these sets are meant to be functions. Therefore we put the restriction on this syntax that in the sets described by R and C no left hand side of an element may be the same as the left hand side of another.

Another restriction on C is that left hand sides must be disjoint (i.e. $\{a\mid b \rightarrow c, d\mid b \rightarrow e\}$ is not allowed as b occurs in both left hand sides). This is to ensure unicity of the communication (see the semantics later on).

Now we have defined our syntax, we will have a look at some examples of *LoCo* processes. Process $M = (coin \parallel button) \cdot product \cdot M$ models a simple vendor machine that waits for a user to insert a coin and press a button (in any order) and then gives a product. This process has the same behaviour as $M' = (coin \cdot button + button \cdot coin + \langle coin, button \rangle) \cdot product \cdot M'$. Another example is $\nabla_{\{a,b\}}(\tau_{\{s_{ab}\}}(\Gamma_{\{s_a\mid s_b \rightarrow s_{ab}\}}(A \parallel B)))$, with $A = a \cdot s_a$ and $B = s_b \cdot b$, which models two separate processes A and B that have to synchronise such that a happens before b. We come back to this last process in the examples in Section 8.

3 Operational Semantics

Now we have a formal syntax, we give meaning to terms by the following semantics.

Definition 3.1. Let T and T' be some sets. We call $F \subseteq T \times T'$ a *function* from T to T' if, and only if, $\forall_{t \in T}(\forall_{t'_1, t'_2 \in T'}(\langle t, t'_1 \rangle, \langle t, t'_2 \rangle \in F \Rightarrow t'_1 = t'_2))$ holds. The subset of functions in the powerset of $T \times T'$ is written as $T \rightarrow T'$ (i.e. $T \rightarrow T' \subseteq \mathcal{P}(T \times T')$, with $F \in T \rightarrow T'$ if, and only if, F is a function from T to T').

Note that we often write a colon (:) instead of $\in$.

Definition 3.2. Let T and T' be sets and $F : T \rightarrow T'$. The *domain* of F, notation $dom(F)$, is defined as follows:

$$dom(F) = \{t \mid \exists_{t' \in T'}(\langle t, t' \rangle \in F)\}$$

Definition 3.3. Let T and T' be sets and $F : T \rightarrow T'$. Also, let $t \in dom(F)$. We define the *function application*, notation $F(t)$ (i.e. F applied to t), as follows:

$$F(t) = t' \quad \text{with } \langle t, t' \rangle \in F$$

Definition 3.4. Let T and T' be sets with $T \subseteq T'$ and $F : T \rightarrow T'$. The *domain extension* F^+ of F is defined by:

$$F^+ = F \cup \{\langle t, t \rangle \mid t \in T \wedge t \notin dom(F)\}$$

We describe the semantics of *LoCo* by looking at which actions can be executed by a process and what (process) the result of such an action is. This is expressed in one of the following definitions, but first we need to express that the order of actions within a multiaction is irrelevant (i.e. $\langle \overrightarrow{a}, \overrightarrow{b} \rangle$ and $\langle \overrightarrow{b}, \overrightarrow{a} \rangle$ are the same). We do this by interpreting them as bags of actions.

Definition 3.5. Let S be a set. A *bag* B of S is a function $S \to \mathbb{N}$, with the meaning that an element $s \in S$ occurs $B(s)$ times in the bag B. We write a bag as $[s_1, s_2, \ldots, s_n]$, such that an element $s \in S$ occurs $B(s)$ times exactly. We write $\mathbb{B}(S)$ to denote the set of all bags of S (i.e. $B \in \mathbb{B}(S)$).

Definition 3.6. Let S be a set and let B_1 and B_2 be bags of S (i.e. $B_1, B_2 \in \mathbb{B}(S)$). The *joining* operator $\oplus_S : \mathbb{B}(S) \times \mathbb{B}(S) \to \mathbb{B}(S)$ is defined by $(B_1 \oplus_S B_2)(s) = B_1(s) + B_2(s)$ for all $s \in S$.

Definition 3.7. Let S be some set and $B, C \in \mathbb{B}(S)$, $T \subseteq S$ and $s \in S$. We introduce the *intersection* $\cap : \mathbb{B}(S) \times \mathcal{P}(S) \to \mathcal{P}(S)$, the *bag inclusion* $\subseteq : \mathbb{B}(S) \times \mathbb{B}(S) \to B$, the *element test* $\in : S \times \mathbb{B}(S) \to B$ and the *size function* $|\,| : \mathbb{B}(S) \to \mathbb{N}$ as follows:

$$
\begin{array}{llll}
s \in B & = & B(s) > 0 & \\
[\,] \cap T & = & \emptyset & \\
([s] \oplus_S B) \cap T & = & B \cap T & \text{if } s \notin T \\
([s] \oplus_S B) \cap T & = & \{s\} \cup (B \cap T) & \text{if } s \in T \\
B \subseteq C & = & \forall_{s \in S}(B(s) \leq C(s)) & \\
|B| & = & \sum_{s \in S} B(s) &
\end{array}
$$

Note that a bag $B \in \mathbb{B}(S)$ can be considered to be of the following structure: $B = [\,]$ or $B = [s] \oplus_S C$, with $s \in S$ and $C \in \mathbb{B}(S)$. We usually write $\oplus$ instead of $\oplus_S$, if this cannot lead to any confusion. In this document we only consider finite bags.

Definition 3.8. We call $\boldsymbol{A} = \mathcal{N}_A$ the set of semantic actions (i.e. $a \in \boldsymbol{A}$).

Let t be some syntactic term. The *interpretation* of t will be written as $[\![t]\!]$. The specific value of $[\![t]\!]$ will be separately defined for all (needed) forms of t.

Definition 3.9. Let α be a multiaction, with $\alpha = \langle a_1, a_2, \ldots, a_n \rangle$. The interpretation of a multiaction $[\![\alpha]\!]$ is defined by $[\![\langle a_1, a_2, \ldots, a_n \rangle]\!] = [a_1, a_2, \ldots, a_n]$.

The following three definitions give a generic way to interpret the sets of the operators $\nabla, \partial, \Gamma, \tau, \rho$, described by V, IH, C, R in our syntax.

Definition 3.10. Let S be some syntactic set of terms (i.e. a collection of syntactic terms $t_1, \ldots, t_n$ separated by commas and enclosed in { and }). The interpretation of S (as a true set) is defined as follows:

$$[\![\{t_1, \ldots, t_n\}]\!] = \{[\![t_1]\!], \ldots, [\![t_n]\!]\}$$

Definition 3.11. Let $t_1, t_2, \ldots, t_n$ be some terms. The interpretation of $t_1|t_2|\ldots|t_n$ is defined as follows:

$$[\![t_1|t_2|\ldots|t_n]\!] = [[\![t_1]\!], [\![t_2]\!], \ldots, [\![t_n]\!]]$$

Definition 3.12. Let t and t' be some syntactic terms. The interpretation of $\to$ is defined as follows:

$$[\![t \to t']\!] = \langle [\![t]\!], [\![t']\!] \rangle$$

In the semantics we use certain functions which are defined below. These functions are for instance needed to determine whether or not ∇_V will (dis)allow an action or whether or not an action is hidden by τ_I.

Definition 3.13. Let S be a function $\mathcal{N}_A \to \mathcal{N}_A$ and let $a \in \boldsymbol{A}$. Also, let $m \in \mathbb{B}(\boldsymbol{A})$. The *function mapping* operator $\bullet : (\mathcal{N}_A \to \mathcal{N}_A) \times \mathbb{B}(\boldsymbol{A}) \to \mathbb{B}(\boldsymbol{A})$ is defined as follows:

$$
\begin{aligned}
S \bullet [] &= [] \\
S \bullet ([a] \oplus m) &= [S^+(a)] \oplus (S \bullet m)
\end{aligned}
$$

Definition 3.14. Let I be a set of action names (i.e. $I \subseteq \mathcal{N}_A$) and let $a \in \boldsymbol{A}$. Also, let $m \in \mathbb{B}(\boldsymbol{A})$. The *hiding* function $\theta : \mathbb{B}(\boldsymbol{A}) \times \mathcal{P}(\mathcal{N}_A) \to \mathbb{B}(\boldsymbol{A})$ is defined as follows:

$$
\begin{aligned}
\theta([], I) &= [] & \\
\theta([a] \oplus m, I) &= \theta(m, I) & \text{if } a \in I \\
\theta([a] \oplus m, I) &= [a] \oplus \theta(m, I) & \text{if } a \notin I
\end{aligned}
$$

For the communication operator we need a somewhat more complex definition. We introduce a communication function that takes the interpretation of a multiaction and finds all occurrences of left hand sides in the C parameter of the operator and replaces those occurrences with the corresponding right hand side.

Definition 3.15. Let $N_\mathbb{B} = \{b \mid b \in \mathbb{B}(\mathcal{N}_A) \wedge 1 < |b|\}$, $a \in \boldsymbol{A}$, $b \in N_\mathbb{B}$ and $m, n, o \in \mathbb{B}(\boldsymbol{A})$. Also let $C : N_\mathbb{B} \to \mathcal{N}_A$ with $\forall_{\langle b,a \rangle, \langle c,a \rangle \in C}(b \neq c \Rightarrow \forall_{n \in b}(n \notin c))$. The *communication* function $\gamma : \mathbb{B}(\boldsymbol{A}) \times (N_\mathbb{B} \to \mathcal{N}_A) \to \mathbb{B}(\boldsymbol{A})$ is defined by the following definition:

$$
\begin{aligned}
\gamma(m \oplus n, C) &= [a] \oplus \gamma(n, C) & \text{if } \langle m, a \rangle \in C \\
\gamma(m, C) &= m & \text{if } \langle m, a \rangle \notin C
\end{aligned}
$$

Note that the extra condition on C is required to make γ a true function (i.e. $\gamma(m, C)$ is a unique bag). This is also a restriction on the syntax that was given earlier.

Definition 3.16. Let X be a process variable and t a term (possibly containing X). A *recursive specification* is a set of equations of the form $X = t$.

Let p and q be terms that do not contain any free variables and let $m \in \mathbb{B}(\boldsymbol{A})$. The relation $p \xrightarrow{m} q$ states that the process described by p can make a *transition* by executing a multiaction, with interpretation m, and will behave as the process described by q after it. The predicate $p \xrightarrow{m} \checkmark$ states that the process described by p can *terminate* by executing a multiaction with interpretation m.

Definition 3.17. Let E be a recursive specification. We define *process semantics* $Sem_{LoCo}(E) = \langle T_P, \mathbb{B}(\boldsymbol{A}), \longrightarrow, \longrightarrow \checkmark \rangle$, with a *transition* predicate $\longrightarrow$, taking two processes (from T_P) and an interpretation of an action (from $\mathbb{B}(\boldsymbol{A})$) as parameters, and a *termination* predicate $\longrightarrow \checkmark$, taking one process and an interpretation of an action as parameters, inductively by the following rules.

$$
\overline{a \xrightarrow{[a]} \checkmark} \qquad\qquad \overline{\alpha \xrightarrow{[\![\alpha]\!]} \checkmark} \qquad\qquad \overline{\tau \xrightarrow{[]} \checkmark}
$$

$$
\frac{t \xrightarrow{m} \checkmark}{\substack{t + u \xrightarrow{m} \checkmark \\ u + t \xrightarrow{m} \checkmark}} \qquad\qquad \frac{t \xrightarrow{m} t'}{\substack{t + u \xrightarrow{m} t' \\ u + t \xrightarrow{m} t'}}
$$

$$
\frac{t \xrightarrow{m} \checkmark}{t \cdot u \xrightarrow{m} u} \qquad\qquad \frac{t \xrightarrow{m} t'}{t \cdot u \xrightarrow{m} t' \cdot u}
$$

$$\frac{t \overset{m}{\longrightarrow} \checkmark}{\begin{array}{c} t \parallel u \overset{m}{\longrightarrow} u \\ u \parallel t \overset{m}{\longrightarrow} u \end{array}} \qquad \frac{t \overset{m}{\longrightarrow} t'}{\begin{array}{c} t \parallel u \overset{m}{\longrightarrow} t' \parallel u \\ u \parallel t \overset{m}{\longrightarrow} u \parallel t' \end{array}} \qquad \frac{t \overset{m}{\longrightarrow} \checkmark, u \overset{n}{\longrightarrow} \checkmark}{t \parallel u \overset{m \oplus n}{\longrightarrow} \checkmark}$$

$$\frac{t \overset{m}{\longrightarrow} t', u \overset{n}{\longrightarrow} u'}{t \parallel u \overset{m \oplus n}{\longrightarrow} t' \parallel u'} \qquad \frac{t \overset{m}{\longrightarrow} \checkmark, u \overset{n}{\longrightarrow} u'}{\begin{array}{c} t \parallel u \overset{m \oplus n}{\longrightarrow} u' \\ u \parallel t \overset{m \oplus n}{\longrightarrow} u' \end{array}}$$

$$\frac{t \overset{m}{\longrightarrow} \checkmark}{t \mathbin{\underline{\parallel}} u \overset{m}{\longrightarrow} u} \qquad \frac{t \overset{m}{\longrightarrow} t'}{t \mathbin{\underline{\parallel}} u \overset{m}{\longrightarrow} t' \parallel u}$$

$$\frac{t \overset{m}{\longrightarrow} \checkmark, u \overset{n}{\longrightarrow} \checkmark}{t | u \overset{m \oplus n}{\longrightarrow} \checkmark} \qquad \frac{t \overset{m}{\longrightarrow} \checkmark, u \overset{n}{\longrightarrow} u'}{\begin{array}{c} t|u \overset{m \oplus n}{\longrightarrow} u' \\ u|t \overset{m \oplus n}{\longrightarrow} u' \end{array}} \qquad \frac{t \overset{m}{\longrightarrow} t', u \overset{n}{\longrightarrow} u'}{t|u \overset{m \oplus n}{\longrightarrow} t' \parallel u'}$$

$$\frac{t \overset{m}{\longrightarrow} \checkmark}{\nabla_V(t) \overset{m}{\longrightarrow} \checkmark} \, m \in [\![V]\!] \cup \{[]\} \qquad \frac{t \overset{m}{\longrightarrow} t'}{\nabla_V(t) \overset{m}{\longrightarrow} \nabla_V(t')} \, m \in [\![V]\!] \cup \{[]\}$$

$$\frac{t \overset{m}{\longrightarrow} \checkmark}{\Gamma_C(t) \overset{\gamma(m, [\![C]\!])}{\longrightarrow} \checkmark} \qquad \frac{t \overset{m}{\longrightarrow} t'}{\Gamma_C(t) \overset{\gamma(m, [\![C]\!])}{\longrightarrow} \Gamma_C(t')}$$

$$\frac{t \overset{m}{\longrightarrow} \checkmark}{\partial_H(t) \overset{m}{\longrightarrow} \checkmark} \, m \cap [\![H]\!] = \emptyset \qquad \frac{t \overset{m}{\longrightarrow} t'}{\partial_H(t) \overset{m}{\longrightarrow} \partial_H(t')} \, m \cap [\![H]\!] = \emptyset$$

$$\frac{t \overset{m}{\longrightarrow} \checkmark}{\tau_I(t) \overset{\theta(m, [\![I]\!])}{\longrightarrow} \checkmark} \qquad \frac{t \overset{m}{\longrightarrow} t'}{\tau_I(t) \overset{\theta(m, [\![I]\!])}{\longrightarrow} \tau_I(t')}$$

$$\frac{t \overset{m}{\longrightarrow} \checkmark}{\rho_R(t) \overset{[\![R]\!] \bullet m}{\longrightarrow} \checkmark} \qquad \frac{t \overset{m}{\longrightarrow} t'}{\rho_R(t) \overset{[\![R]\!] \bullet m}{\longrightarrow} \rho_R(t')}$$

$$\frac{t \overset{m}{\longrightarrow} \checkmark}{X \overset{m}{\longrightarrow} \checkmark} \, X = t \in E \qquad \frac{t \overset{m}{\longrightarrow} t'}{X \overset{m}{\longrightarrow} t'} \, X = t \in E$$

Table 1: *LoCo* Semantics

To be able to compare and calculate with processes, we need to know when two processes are equal (i.e. have the same behaviour). We use the equality given by the following definition.

Definition 3.18.   Let $Sem_{LoCo}(E) = \langle T_P, \mathbb{B}(\boldsymbol{A}), \longrightarrow, \longrightarrow \checkmark \rangle$, with E a set of process expressions and $\longrightarrow$ and $\longrightarrow \checkmark$ the transition relations on processes T_P with multiactions $\mathbb{B}(\boldsymbol{A})$, be a process semantics. Also, let t, t', u and u' be process terms, in which process variables may occur only if they are in E, and $m \in \mathbb{B}(\boldsymbol{A})$. A *bisimulation* is a relation B on processes such that if tBu:

- for all t' and m, $t \overset{m}{\longrightarrow} t'$ means that there exists a u' with $u \overset{m}{\longrightarrow} u'$ and $t'Bu'$

- for all u' and m, $u \overset{m}{\longrightarrow} u'$ means that there exists a t' with $t \overset{m}{\longrightarrow} t'$ and $t'Bu'$

- for all m, $t \xrightarrow{m} \checkmark$ means that $u \xrightarrow{m} \checkmark$

- for all m, $u \xrightarrow{m} \checkmark$ means that $t \xrightarrow{m} \checkmark$

We write the union of all bisimulation relations B as $\underleftrightarrow{}$. To state that two processes p and q are *bisimilar* we write $LoCo \vDash p \underleftrightarrow{} q$ (or just $p \underleftrightarrow{} q$).

As the rules in Table 1 are in the *path* format [1], we have that bisimulation $\underleftrightarrow{}$ is a congruence with respect to all operators.

4   Axioms

To be able to calculate (more) easily, we introduce the following axiomatisation for the semantics given in the previous section, which is proved to be sound and complete in the next two sections. (Note that this axiomatisation does not include the use of process variables and thus completeness only considers the semantics of process-closed terms.)

Definition 4.1.   Let $t, u \in T_P$. We can *derive* t to u, notation $LoCo \vdash t \doteq u$, when this follows from the rules and axioms below. Usually we just write $t \doteq u$. The following rules hold for $\doteq$ (with $t, u, v \in T_P$, $x \in V_P$, $\star$ a unary process operator and $\circ$ a binary process operator).

$$\frac{}{t \doteq t}$$

$$\frac{t \doteq u}{u \doteq t} \qquad\qquad \frac{t \doteq u, u \doteq v}{t \doteq v}$$

$$\frac{t \doteq u}{\star(t) \doteq \star(u)} \qquad\qquad \frac{t \doteq t', u \doteq u'}{t \circ u \doteq t' \circ u'}$$

$$\frac{t \doteq u}{t[v/x] \doteq u[v/x]} \qquad \frac{\forall_{x \in V_P}(\forall_{v \in T_{pc}}(t[v/x] \doteq u[v/x]))}{t \doteq u}$$

Our axiomatisation is the following:

MA1	$a \doteq \langle a \rangle$	*VD*	$\nabla_V(\delta) \doteq \delta$
MA2	$\tau \doteq \langle \rangle$	*V1*	$\nabla_V(\alpha) \doteq \alpha \quad if \; [\![\alpha]\!] \in [\![V]\!] \cup \{[]\}$
MA3	$\langle \overrightarrow{a}, \overrightarrow{b} \rangle \doteq \langle \overrightarrow{b}, \overrightarrow{a} \rangle$	*V2*	$\nabla_V(\alpha) \doteq \delta \quad if \; [\![\alpha]\!] \notin [\![V]\!] \cup \{[]\}$
		V3	$\nabla_V(x + y) \doteq \nabla_V(x) + \nabla_V(y)$
A1	$x + y \doteq y + x$	*V4*	$\nabla_V(x \cdot y) \doteq \nabla_V(x) \cdot \nabla_V(y)$
A2	$x + (y + z) \doteq (x + y) + z$		
A3	$x + x \doteq x$	*DD*	$\partial_H(\delta) \doteq \delta$
A4	$(x + y) \cdot z \doteq x \cdot z + y \cdot z$	*D1*	$\partial_H(\alpha) \doteq \alpha \quad if \; [\![\alpha]\!] \cap [\![H]\!] = \emptyset$
A5	$(x \cdot y) \cdot z \doteq x \cdot (y \cdot z)$	*D2*	$\partial_H(\alpha) \doteq \delta \quad if \; [\![\alpha]\!] \cap [\![H]\!] \neq \emptyset$
A6	$x + \delta \doteq x$	*D3*	$\partial_H(x + y) \doteq \partial_H(x) + \partial_H(y)$
A7	$\delta \cdot x \doteq \delta$	*D4*	$\partial_H(x \cdot y) \doteq \partial_H(x) \cdot \partial_H(y)$
CM1	$x \parallel y \doteq x \parallel\!\!\!\!\lfloor\; y + y \parallel\!\!\!\!\lfloor\; x + x\vert y$	*TID*	$\tau_I(\delta) \doteq \delta$
CM2	$\boldsymbol{\alpha} \parallel\!\!\!\!\lfloor\; x \doteq \boldsymbol{\alpha} \cdot x$	*TI1*	$\tau_I(\alpha) \doteq \beta \quad with \; [\![\beta]\!] = \theta([\![\alpha]\!], I)$
CM3	$\boldsymbol{\alpha} \cdot x \parallel\!\!\!\!\lfloor\; y \doteq \boldsymbol{\alpha} \cdot (x \parallel y)$	*TI3*	$\tau_I(x + y) \doteq \tau_I(x) + \tau_I(y)$

$CM4$	$(x + y) \parallel z \doteq x \parallel z + y \parallel z$	$TI4$	$\tau_I(x \cdot y) \doteq \tau_I(x) \cdot \tau_I(y)$
$CM5$	$\boldsymbol{\alpha} \cdot x \vert \boldsymbol{\beta} \doteq (\boldsymbol{\alpha} \vert \boldsymbol{\beta}) \cdot x$		
$CM6$	$\boldsymbol{\alpha} \vert \boldsymbol{\beta} \cdot x \doteq (\boldsymbol{\alpha} \vert \boldsymbol{\beta}) \cdot x$	RD	$\rho_R(\delta) \doteq \delta$
$CM7$	$\boldsymbol{\alpha} \cdot x \vert \boldsymbol{\beta} \cdot y \doteq (\boldsymbol{\alpha} \vert \boldsymbol{\beta}) \cdot (x \parallel y)$	$R1$	$\rho_R(\alpha) \doteq \beta \quad with \; [\![\beta]\!] = [\![R]\!] \bullet [\![\alpha]\!]$
$CM8$	$(x + y) \vert z \doteq x \vert z + y \vert z$	$R3$	$\rho_R(x + y) \doteq \rho_R(x) + \rho_R(y)$
$CM9$	$x \vert (y + z) \doteq x \vert y + x \vert z$	$R4$	$\rho_R(x \cdot y) \doteq \rho_R(x) \cdot \rho_R(y)$
$CM10$	$\langle \overrightarrow{a} \rangle \vert \langle \overrightarrow{b} \rangle \doteq \langle \overrightarrow{a}, \overrightarrow{b} \rangle$		
		GD	$\Gamma_C(\delta) \doteq \delta$
$CD1$	$\delta \vert \boldsymbol{\alpha} \doteq \delta$	$G1$	$\Gamma_C(\alpha) \doteq \beta \quad with \; [\![\beta]\!] = \gamma([\![\alpha]\!], [\![C]\!])$
$CD2$	$\boldsymbol{\alpha} \vert \delta \doteq \delta$	$G3$	$\Gamma_C(x + y) \doteq \Gamma_C(x) + \Gamma_C(y)$
		$G4$	$\Gamma_C(x \cdot y) \doteq \Gamma_C(x) \cdot \Gamma_C(y)$

With $a, b \in A$, $\alpha, \beta \in \mathbb{A}$, $\boldsymbol{\alpha}, \boldsymbol{\beta} \in \mathbb{A} \cup \{\delta\}$, $x, y \in V_P$ and $p, q \in T_{pc}$.

Table 2: *LoCo* Axioms

When we do wish to use recursive processes, we will (at least) need some extension to the $\doteq$ defined above and also RDP^- and RSP [2] like principles. We informally define them as follows.

Definition 4.2. We extend $\doteq$ with the following rule:

$$\frac{X = t \in E}{X \doteq t}$$

Definition 4.3. We call a specification *guarded* if every occurrence of a process variable on the right hand side of an equation is preceded by at least one action (possibly after substitution).

Definition 4.4. A *solution* of a specification $X = t$ is a process p, such that $p \doteq t[p/X]$ holds, where $t[p/X]$ is t with all occurrences of X replaced by p.

Definition 4.5. The *Restricted Recursive Definition Principle* (RDP) states that every guarded specification has a solution.

Definition 4.6. The *Recursive Specification Principle* (RSP) states that every guarded specification has at most one solution.

5   Soundness of the axioms

The soundness of the axiomatisation in the previous section is stated in the following theorem.

Theorem 5.1. Let $p, q \in T_{pc}$. The axiomatisation of *LoCo* is sound (i.e.   *LoCo* $\vdash p \doteq q \Rightarrow$ *LoCo* $\vDash p \underline{\leftrightarrow} q$).

Proof 5.1. In Appendix A the soundness of the axioms given above has been proved (one axiom per subsection). The structure of each of these proofs is the same, namely:

- A relation R is given for the axiom. For an axiom $t \doteq u$, the relation R is the minimal relation that satisfies $tRu \wedge pRp \wedge (qRr \Leftrightarrow rRq) \wedge E$, for all $p, q, r \in T_{pc}$ and for all instantiations of process variables in t and u. E is *true*, unless it is explicitly defined otherwise in the subsection.

- For each conjunct of the definition of R it is shown that the properties that define a bisimulation hold.

It is clear that if this is done, R is proven to be a bisimulation relation.

To shorten the proofs and get rid of trivial facts, that would otherwise be repeated in every subsection, we do not define R explicitly (as it is always of the form given above) or prove the conjuncts pRp and $qRr \Leftrightarrow rRq$ (and E, if it is *true*). The proofs of these conjuncts are trivial and therefore not given at all.

When we say a process cannot terminate in any of the subsections, we mean that there is no m for which $\xrightarrow{m} \checkmark$ holds for that process. (So it could very well be that the process can indeed terminate, but not in one "step".) $\qquad\square$

Note that we should also prove the given rules of $\doteq$, but as these are quite straightforward (with, for example, the symmetry of $\underleftrightarrow{\ }$ and the fact that $\underleftrightarrow{\ }$ is a congruence) they are not given.

6 Completeness

To prove that our axiomatisation is complete (i.e. that whenever two processes p and q are bisimilar we know that $p \doteq q$) we introduce basic terms and show that any (process-closed) term can be rewritten to such a basic term. By doing this, the actual completeness proof only has to consider these basic terms.

Note that we often say that something holds "by induction", with which we mean to say that it follows from the (implicit) induction hypothesis.

6.1 Basic terms

Definition 6.1. Let $p, q, r \in T_{pc}$ and let $\alpha \in \mathbb{A} \cup \{\delta\}$. A *basic term p* is a term having one of the following forms:

- α

- $\alpha \cdot q$, with q a basic term

- $q + r$, with q and r basic terms

Definition 6.2. We call the operators that can occur in basic terms $(\cdot, +)$ *basic operators*.

Theorem 6.3. Let $p, q \in T_{pc}$. For each p, which uses only basic operators, there exists a basic term q with $p \doteq q$.

Proof 6.3. We use induction on the structure of p:

- $p = a$, which means that $p = a \doteq \langle a \rangle$.

- $p = \alpha$, which already is a basic term.

- $p = \delta$, which already is a basic term.

- $p = \tau$, which means that $p = \tau \doteq \langle \rangle$.

- $p = q + r$, which means that by induction we have basic terms q' and r' with $q \doteq q' \wedge r \doteq r'$, and $p = q + r \doteq q' + r'$.

- $p = q \cdot r$, which means that by induction we have basic terms q' and r' with $q \doteq q' \wedge r \doteq r'$. We now prove that for each basic term s there is a basic term t with $t \doteq s \cdot r$, which means that there is a basic term p' with $p = q \cdot r \doteq q' \cdot r \doteq p'$. By the structure of s:

    - α, which means that $s \cdot r = \alpha \cdot r \doteq \alpha \cdot r'$, or

- $\alpha \cdot s'$, with s' a basic term, which means that $s \cdot r = (\alpha \cdot s') \cdot r \doteq \alpha \cdot (s' \cdot r) \doteq \alpha \cdot u$, with basic term $u \doteq s' \cdot r$ by induction, or
- $s' + s''$, with s' and s'' basic terms, which means that $s \cdot r = (s' + s'') \cdot r \doteq s' \cdot r + s'' \cdot r \doteq u + u'$, with basic terms $u \doteq s' \cdot r$ and $u' \doteq s'' \cdot r$ by induction.

$\square$

Lemma 6.4. Let $p, q \in T_{pc}$. If there are only basic operators in p and q, then there is a term $r \in T_{pc}$ without non-basic operators, such that $r \doteq p \circ q$ (for some non-basic binary operator $\circ$).

Proof 6.4.   By Theorem 6.3 we know that there are basic terms p' and q' with $p \doteq p'$ and $q \doteq q'$. We prove, by induction on the number of symbols in p' and q', that there is a term r' without non-basic operators, such that $r' \doteq p' \circ q'$ and thus $r' \doteq p \circ q$.

- $p' \parallel\!\!\!\!\perp q'$, and p' is of structure

  - α, which means that $r \doteq p' \parallel\!\!\!\!\perp q' = \alpha \parallel\!\!\!\!\perp q' \doteq \alpha \cdot q'$, or
  - $\alpha \cdot s$, with s a basic term, which means that $r \doteq p' \parallel\!\!\!\!\perp q' = (\alpha \cdot s) \parallel\!\!\!\!\perp q' \doteq \alpha \cdot (s \parallel q') = \alpha \cdot s'$, with $s' \doteq s \parallel q'$ a term without non-basic operators by induction, or
  - $s + s'$, with s and s' basic terms, which means that $r \doteq p' \parallel\!\!\!\!\perp q' = (s + s') \parallel\!\!\!\!\perp q' \doteq s \parallel\!\!\!\!\perp q' + s' \parallel\!\!\!\!\perp q' = t + t'$, with $t \doteq s \parallel\!\!\!\!\perp q'$ and $t' \doteq s' \parallel\!\!\!\!\perp q'$ terms without non-basic operators by induction.

- $p' | q'$, and

  - $p' = \alpha \wedge q' = \alpha'$, which means that $r \doteq p' | q' = \alpha | \alpha'$, with α and α' actions or δ, which means that $\alpha | \alpha'$ is an action or δ
  - $p' = \alpha \cdot s \wedge q' = \alpha'$, with s a basic term, which means that $r \doteq p' | q' = (\alpha \cdot s) | \alpha' \doteq (\alpha | \alpha') \cdot s$, with α and α' actions or δ, which means that $\alpha | \alpha'$ is an action or δ
  - $p' = \alpha \wedge q' = \alpha' \cdot s$, with s a basic term, which is symmetrical to the previous case
  - $p' = \alpha \cdot s \wedge q' = \alpha' \cdot s'$, with s and s' basic terms, which means that $r \doteq p' | q' = (\alpha \cdot s) | (\alpha' \cdot s') \doteq (\alpha | \alpha') \cdot (s \parallel s') = (\alpha | \alpha') \cdot t$, with α and α' actions or δ, which means that $\alpha | \alpha'$ is an action or δ, and $t \doteq s \parallel s'$ a term without non-basic operators by induction
  - $p' = s + s'$, with basic terms s and s', which means that $r \doteq p' | q' = (s + s') | q' \doteq s | q' + s' | q' = t + t'$, with $t \doteq s | q'$ and $t' \doteq s' | q'$ terms without non-basic operators by induction
  - $q' = s + s'$, with basic terms s and s', which is symmetrical to the previous case.

- $p' \parallel q'$, which means that $r \doteq p' \parallel q' \doteq p' \parallel\!\!\!\!\perp q' + q' \parallel\!\!\!\!\perp p' + p' | q' = s + t + u$, with $s \doteq p' \parallel\!\!\!\!\perp q'$, $t \doteq q' \parallel\!\!\!\!\perp p'$ and $u \doteq p' | q'$ terms without non-basic operators by the previous cases.

- $\nabla_V(p')$, and p' is of structure

  - α, which means that $r \doteq \nabla_V(p') \doteq \nabla_V(\alpha)$, with $\nabla_V(\alpha)$ an action or δ, or
  - $\alpha \cdot s$, with s a basic term, which means that $r \doteq \nabla_V(p') = \nabla_V(\alpha \cdot s) \doteq \nabla_V(\alpha) \cdot \nabla_V(s) = \nabla_V(\alpha) \cdot s'$, with $\nabla_V(\alpha)$ an action or δ and $s' \doteq \nabla_V(s)$ a term without non-basic operators by induction, or
  - $s + s'$, with s and s' basic terms, which means that $r \doteq \nabla_V(p') = \nabla_V(s + s') \doteq \nabla_V(s) + \nabla_V(s') = t + t'$, with $t \doteq \nabla_V(s)$ and $t' \doteq \nabla_V(s')$ terms without non-basic operators by induction.

- $\tau_I(p')$, which is similar to the previous case.

- $\rho_R(p')$, which is similar to the previous case.

- $\Gamma_C(p')$, which is similar to the previous case.

- $\partial_H(p')$, which is similar to the previous case.

$\square$

Theorem 6.5. Let $p, q \in T_{pc}$. The *elimination theorem* states that all non-basic operators ($\|$, $\|\!\|$, $|$, ∇_V, Γ_C, τ_I and ρ_R) can be eliminated. That is, for each term p there exists a term q with $p \doteq q$ and q does only contain basic operators.

Proof 6.5. With induction on the number of symbols in p:

- p is an action, deadlock or tau, which means there are no non-basic operators in p.

- $p = q \circ r$, with $\circ$ a basic operator, which means that we have terms q' and r' with $q \doteq q'$ and $r \doteq r'$ which contain no non-basic operators by induction and thus $p = q \circ r \doteq q' \circ r'$ does not as well.

- $p = \star(q)$, with $\star$ a non-basic unary operator, which means that we have a term q' with $q \doteq q'$ which contains no non-basic operators by induction and thus $p = \star(q) \doteq \star(q') \doteq r$, with r a term without non-basic operators by Lemma 6.4.

- $p = q \circ r$, with $\circ$ a non-basic binary operator, which means that we have terms q' and r' with $q \doteq q'$ and $r \doteq r'$ which contain no non-basic operators by induction and thus $p = q \circ r \doteq q' \circ r' \doteq s$, with s a term without non-basic operators by Lemma 6.4.

$\square$

Theorem 6.6. Let $p, q \in T_{pr}$. For each p there exists a basic term q with $p \doteq q$.

Proof 6.6. By Theorem 6.5 we know there exists a r, with $p \doteq r$, in which no non-basic operators occur. And Theorem 6.3 states that for such a term there is a basic term q with $r \doteq q$. Thus, as $p \doteq r \doteq q$, there is a basic term for each process-closed term p. $\square$

6.2 Completeness

Before we can prove completeness we need the following definitions and lemmas. Note that we use standard predicate calculus (with $\equiv, \Rightarrow, \wedge, \vee, \neg, \forall, \exists$) to define, derive and prove. This means we consider $p \leftrightarrow q$ and $p \doteq q$ to be predicates on p and q and will write derivations like $p \doteq \alpha + \alpha \equiv p \doteq \alpha$, for some $p \in T_{pc}$.

Definition 6.7. Let $p, q \in T_{pc}$. The *bisimulation inclusion* $\rightarrowtail$ of a bisimulation $\leftrightarrow$ is defined by $p \rightarrowtail q \equiv p + q \leftrightarrow q$.

Lemma 6.8. We can formulate the definition of $\rightarrowtail$ as follows:

$$p \rightarrowtail q \equiv \forall_{\alpha \in \mathbb{B}(A)}((p \xrightarrow{\alpha} \checkmark \Rightarrow q \xrightarrow{\alpha} \checkmark) \wedge \forall_{p'}(p \xrightarrow{\alpha} p' \Rightarrow \exists_{q'}(q \xrightarrow{\alpha} q' \wedge p' \leftrightarrow q')))$$

Proof 6.8. With the definition of $\leftrightarrow$ in the more formal form of $p \leftrightarrow q \equiv \forall_{\alpha \in \mathbb{B}(A)}((p \xrightarrow{\alpha} \checkmark \Rightarrow q \xrightarrow{\alpha} \checkmark) \wedge (q \xrightarrow{\alpha} \checkmark \Rightarrow p \xrightarrow{\alpha} \checkmark)) \wedge \forall_{p'}(p \xrightarrow{\alpha} p' \Rightarrow \exists_{q'}(q \xrightarrow{\alpha} q' \wedge p' \leftrightarrow q')) \wedge \forall_{q'}(q \xrightarrow{\alpha} q' \Rightarrow \exists_{p'}(p \xrightarrow{\alpha} p' \wedge p' \leftrightarrow q')))$ the proof is as follows:

$$
\begin{aligned}
& p \underrightarrow{\rightarrow} q \\
\equiv\ & p + q \underline{\leftrightarrow} q \\
\equiv\ & \forall_{\alpha \in \mathbb{B}(A)}((p + q \xrightarrow{\alpha} \checkmark \Rightarrow q \xrightarrow{\alpha} \checkmark) \wedge (q \xrightarrow{\alpha} \checkmark \Rightarrow p + q \xrightarrow{\alpha} \checkmark) \wedge \\
& \quad\quad \forall_{p'}(p + q \xrightarrow{\alpha} p' \Rightarrow \exists_{q'}(q \xrightarrow{\alpha} q' \wedge p' \underline{\leftrightarrow} q')) \wedge \forall_{q'}(q \xrightarrow{\alpha} q' \Rightarrow \exists_{p'}(p + q \xrightarrow{\alpha} p' \wedge p' \underline{\leftrightarrow} q'))) \\
\equiv\ & \forall_{\alpha \in \mathbb{B}(A)}(((p \xrightarrow{\alpha} \checkmark \Rightarrow q \xrightarrow{\alpha} \checkmark) \wedge (q \xrightarrow{\alpha} \checkmark \Rightarrow q \xrightarrow{\alpha} \checkmark)) \wedge true \wedge \\
& \quad\quad \forall_{p'}((p \xrightarrow{\alpha} p' \Rightarrow \exists_{q'}(q \xrightarrow{\alpha} q' \wedge p' \underline{\leftrightarrow} q')) \wedge (q \xrightarrow{\alpha} p' \Rightarrow \exists_{q'}(q \xrightarrow{\alpha} q' \wedge p' \underline{\leftrightarrow} q'))) \wedge true) \\
\equiv\ & \forall_{\alpha \in \mathbb{B}(A)}(((p \xrightarrow{\alpha} \checkmark \Rightarrow q \xrightarrow{\alpha} \checkmark) \wedge true) \wedge \forall_{p'}((p \xrightarrow{\alpha} p' \Rightarrow \exists_{q'}(q \xrightarrow{\alpha} q' \wedge p' \underline{\leftrightarrow} q')) \wedge true)) \\
\equiv\ & \forall_{\alpha \in \mathbb{B}(A)}((p \xrightarrow{\alpha} \checkmark \Rightarrow q \xrightarrow{\alpha} \checkmark) \wedge \forall_{p'}(p \xrightarrow{\alpha} p' \Rightarrow \exists_{q'}(q \xrightarrow{\alpha} q' \wedge p' \underline{\leftrightarrow} q')))
\end{aligned}
$$

$\square$

Definition 6.9. Let $p, q \in T_{pc}$. The *axiomatic inclusion* $\dot{\leqslant}$ of an axiomatic relation $\doteq$ is defined by $p \dot{\leqslant} q \equiv p + q \doteq q$.

Lemma 6.10. An axiomatic inclusion $\dot{\leqslant}$ is sound with respect to a bisimulation inclusion $\underrightarrow{\rightarrow}$ if $\doteq$ is sound with respect to $\underline{\leftrightarrow}$ (i.e. $\forall_{p,q}(p \doteq q \Rightarrow p \underline{\leftrightarrow} q) \Rightarrow \forall_{p,q}(p \dot{\leqslant} q \Rightarrow p \underrightarrow{\rightarrow} q)$).

Proof 6.10. Trivial. $\square$

Lemma 6.11. Let $p, q \in T_{pc}$. The relation $\dot{\leqslant}$ is antisymmetric:

$$
p \doteq q \equiv p \dot{\leqslant} q \wedge q \dot{\leqslant} p
$$

Proof 6.11.

$$
\begin{array}{ll}
& p \dot{\leqslant} q \wedge q \dot{\leqslant} p \quad\quad\quad\quad\quad\quad & p \doteq q \\
\equiv\ & p + q \doteq q \wedge q + p \doteq p & \equiv\ p \doteq q \wedge p \doteq q \\
\equiv\ & q \doteq p + q \wedge p + q \doteq p & \equiv\ p \doteq q \wedge p + p \doteq q \wedge q \doteq p \\
\Rightarrow\ & p \doteq q & \equiv\ p \doteq q \wedge p + q \doteq q \wedge q + q \doteq p \\
& & \equiv\ p \doteq q \wedge p \dot{\leqslant} q \wedge q + p \doteq p \\
& & \Rightarrow\ p \dot{\leqslant} q \wedge q \dot{\leqslant} p
\end{array}
$$

$\square$

Lemma 6.12. Let $p, q, r \in T_{pc}$. The axiomatic inclusion distributes over the alternative composition as follows:

$$
p + q \dot{\leqslant} r \equiv p \dot{\leqslant} r \wedge q \dot{\leqslant} r
$$

Proof 6.12.

$$
\begin{array}{ll}
& p + q \dot{\leqslant} r \quad\quad\quad\quad\quad\quad\quad\quad\quad & p \dot{\leqslant} r \wedge q \dot{\leqslant} r \\
\equiv\ & p + q + r \doteq r & \equiv\ p + r \doteq r \wedge q + r \doteq r \\
\equiv\ & p + q + r \doteq r \wedge p + q + r \doteq r & \Rightarrow\ p + q + r \doteq r \\
\equiv\ & p + q + r \doteq r \wedge p + q + r \doteq r \wedge p + q + q + r \doteq r & \equiv\ p + q \dot{\leqslant} r \\
\equiv\ & p + q + r \doteq r \wedge p + p + q + r \doteq r \wedge q + p + q + r \doteq r \\
\Rightarrow\ & p + r \doteq r \wedge q + r \doteq r \\
\equiv\ & p \dot{\leqslant} r \wedge q \dot{\leqslant} r
\end{array}
$$

$\square$

We write $\# : T_{pc} \to \mathbb{N}$ for the number of (process) symbols in a term. Its definition is quite straightforward and will therefore not be given in an other way than saying that every symbol (operator or (in)action) is counted. Note that this does not include data operators or constants.

Lemma 6.13. Let $p, q, q', r, s \in T_{pc}$ and A be complete and have equality and quantifier elimination. We can split p in smaller (or equally large) pieces if $p \underrightarrow{\rightarrow} q + q'$, as expressed by the following:

$$p \xrightarrow{} q + q' \Rightarrow \exists_{r,s}(p \doteq r + s \wedge \#(r) \leq \#(p) \wedge \#(s) \leq \#(p) \wedge r \xrightarrow{} q \wedge s \xrightarrow{} q')$$

Proof 6.13. With induction on the structure of p:

- $p = \alpha$, which means that

  - $p = \delta$ and thus $p = \delta \doteq \delta + \delta$, or

  - $p \xrightarrow{[\![\alpha]\!]} \checkmark$ and thus $q + q' \xrightarrow{[\![\alpha]\!]} \checkmark$ and $q \xrightarrow{[\![\alpha]\!]} \checkmark \vee q' \xrightarrow{[\![\alpha]\!]} \checkmark$; if $q \xrightarrow{[\![\alpha]\!]} \checkmark$ then we say $r = \alpha$ and otherwise $r = \delta$ and the same for q' and s; then $p = \alpha \doteq r + s$ (with trivially $r \xrightarrow{} q$ and $s \xrightarrow{} q'$)

- $p = \alpha \cdot r$, which is similar to the previous case

- $p = r + r'$

$$
\begin{aligned}
 & p \xrightarrow{} q + q' \\
\equiv\ & r + r' \xrightarrow{} q + q' \\
\equiv\ & r \xrightarrow{} q + q' \wedge r' \xrightarrow{} q + q' \\
\equiv\ & \exists_{s,t}(r \doteq s + t \wedge \#(s) \leq \#(r) \wedge \#(t) \leq \#(r) \wedge s \xrightarrow{} q \wedge t \xrightarrow{} q') \wedge \\
 & \exists_{s',t'}(r' \doteq s' + t' \wedge \#(s') \leq \#(r') \wedge \#(t') \leq \#(r') \wedge s' \xrightarrow{} q \wedge t' \xrightarrow{} q') \\
\equiv\ & \exists_{s,s',t,t'}(r \doteq s + t \wedge \#(s) \leq \#(r) \wedge \#(t) \leq \#(r) \wedge s \xrightarrow{} q \wedge t \xrightarrow{} q' \wedge \\
 & \quad r' \doteq s' + t' \wedge \#(s') \leq \#(r') \wedge \#(t') \leq \#(r') \wedge s' \xrightarrow{} q \wedge t' \xrightarrow{} q') \\
\Rightarrow\ & \exists_{s,s',t,t'}(r + r' \doteq s + t + s' + t' \wedge \#(s + s') \leq \#(r + r') \wedge \#(t + t') \leq \#(r + r') \wedge \\
 & \quad s + s' \xrightarrow{} q \wedge t + t' \xrightarrow{} q') \\
\Rightarrow\ & \exists_{s,t}(p \doteq s + t \wedge \#(s) \leq \#(p) \wedge \#(t) \leq \#(p) \wedge s \xrightarrow{} q \wedge t \xrightarrow{} q')
\end{aligned}
$$

$\square$

Lemma 6.14. Let $\alpha, \beta \in \mathbb{A}$ and $m \in \mathbb{B}(\boldsymbol{A})$. The following holds:

$$\alpha \xrightarrow{m} \checkmark \wedge \beta \xrightarrow{m} \checkmark \Rightarrow \alpha \doteq \beta$$

Proof 6.14. With induction on α:

- $\alpha = \langle \rangle$, which means that $m = [\![\alpha]\!] = [\![\langle \rangle]\!] = []$ and thus $[\![\beta]\!] = []$ which implies $\beta = \langle \rangle$, or

- $\alpha = \langle a, \overrightarrow{b} \rangle$, with $a \in A$ and $\langle \overrightarrow{b} \rangle \in \mathbb{A}$, which means that $m = [\![\alpha]\!] = [\![\langle a, \overrightarrow{b} \rangle]\!] = [a, \overrightarrow{b}]$ and $[\![\beta]\!] = [a, \overrightarrow{b}]$ and thus $\beta = \langle \overrightarrow{c}, a, \overrightarrow{d} \rangle$, with $\langle \overrightarrow{c} \rangle, \langle \overrightarrow{d} \rangle \in \mathbb{A}$ and $[\overrightarrow{b}] = [\overrightarrow{c}, \overrightarrow{d}]$, which means that $\beta = \langle \overrightarrow{c}, a, \overrightarrow{d} \rangle \doteq \langle a, \overrightarrow{d}, \overrightarrow{c} \rangle \doteq \langle a \rangle | \langle \overrightarrow{d}, \overrightarrow{c} \rangle \doteq \langle a \rangle | \langle \overrightarrow{b} \rangle \doteq \langle a, \overrightarrow{b} \rangle = \alpha$, with $\langle \overrightarrow{b} \rangle \doteq \langle \overrightarrow{d}, \overrightarrow{c} \rangle$ by induction.

$\square$

Lemma 6.15. Let $\alpha \in \mathbb{A}$ and $\beta \in \mathbb{A} \cup \{\delta\}$. The following holds:

$$\alpha \xrightarrow{} \beta \Rightarrow \alpha \doteq \beta$$

Proof 6.15. Because $\alpha \in \mathbb{A}$ we have $\alpha \xrightarrow{[\![\alpha]\!]} \checkmark$ and if $\alpha \xrightarrow{} \beta$ also $\beta \xrightarrow{[\![\alpha]\!]} \checkmark$ (by Lemma 6.8). Then $\alpha \doteq \beta$ follows from Lemma 6.14. $\square$

Lemma 6.16. Let $\alpha \in \mathbb{A}$, $\beta \in \mathbb{A} \cup \{\delta\}$ and $p, q \in T_{pc}$. The following holds:

$$\alpha \cdot p \xrightarrow{} \beta \cdot q \Rightarrow \alpha \doteq \beta \wedge p \underline{\leftrightarrow} q$$

Proof 6.16. Because $\alpha \in \mathbb{A}$ we have $\alpha \cdot p \xrightarrow{\llbracket \alpha \rrbracket} p$ and if $\alpha \underset{\rightarrow}{\rightarrow} \beta$ also $\beta \cdot q \xrightarrow{\llbracket \alpha \rrbracket} q \wedge p \underline{\leftrightarrow} q$ (by Lemma 6.8). Then, because also $\alpha \xrightarrow{\llbracket \alpha \rrbracket} \checkmark$ and $\beta \xrightarrow{\llbracket \alpha \rrbracket} \checkmark$, $\alpha \doteq \beta$ follows from Lemma 6.14. $\qquad \square$

Theorem 6.17. Let $p, q \in T_{pc}$. *LoCo* is complete (i.e. $LoCo \vDash p \underline{\leftrightarrow} q \Rightarrow LoCo \vdash p \doteq q$).

Proof 6.17. Given $p \underline{\leftrightarrow} q$, we need to show that $p \doteq q$. As there are basic terms p' and q' with $p \doteq p' \wedge q \doteq q'$ (and, because $\doteq$ is sound, $p \underline{\leftrightarrow} p' \wedge q \underline{\leftrightarrow} q'$), it is sufficient to show that $p' \doteq q'$. We do this by proving that $p' \underset{\rightarrow}{\rightarrow} q' \Rightarrow p' \dot{\leqslant} q'$ and $q' \underset{\rightarrow}{\rightarrow} p' \Rightarrow q' \dot{\leqslant} p'$. From this obviously follows $p \underline{\leftrightarrow} q \equiv p' \underline{\leftrightarrow} q' \equiv p' \underset{\rightarrow}{\rightarrow} q' \wedge q' \underset{\rightarrow}{\rightarrow} p' \Rightarrow p' \dot{\leqslant} q' \wedge q' \dot{\leqslant} p' \equiv p' \doteq q'$.

We now prove $p' \underset{\rightarrow}{\rightarrow} q' \Rightarrow p' \dot{\leqslant} q'$ with induction on the number of symbols in p' and q':

- $p' = \alpha$, which means that $p' = \delta$ and thus $p' = \delta \dot{\leqslant} q'$ *or* $p' \neq \delta$ and then by induction on q'

  - $q' = \beta$, which means that $\alpha \dot{\leqslant} \beta$ because of $p' \underset{\rightarrow}{\rightarrow} q'$ (by Lemma 6.15), or

  - $q' = \beta \cdot r$, with basic term r, which is not possible as q' should be able to terminate, or

  - $q' = r + r'$, with basic terms r and r', which means that, by Lemma 6.13, $\exists_{s,s'}(p' \doteq s + s' \wedge \#(s) \leq \#(p') \wedge \#(s') \leq \#(p') \wedge s \underset{\rightarrow}{\rightarrow} r \wedge s' \underset{\rightarrow}{\rightarrow} r')$ and by induction $\exists_{s,s'}(p' \doteq s + s' \wedge s \dot{\leqslant} r \wedge s' \dot{\leqslant} r') \Rightarrow \exists_{s,s'}(p' \doteq s + s' \wedge s \dot{\leqslant} r + r' \wedge s' \dot{\leqslant} r + r') \Rightarrow \exists_{s,s'}(p' \doteq s + s' \wedge s + s' \dot{\leqslant} r + r') \Rightarrow p' \dot{\leqslant} r + r' = q'$.

- $p' = \alpha \cdot r$, with basic term r, which means that $p' = \delta \cdot r$ and thus $p' = \delta \cdot r \doteq \delta \dot{\leqslant} q'$ *or* $p' \neq \delta \cdot r$ and then by induction on q'

  - $q' = \beta$, which is not possible as q' can only terminate (and p' cannot), or

  - $q' = \beta \cdot s$, which means that $\alpha \doteq \beta \wedge r \underline{\leftrightarrow} s$ because of $p' \underset{\rightarrow}{\rightarrow} q'$ (by Lemma 6.16) and thus $r \dot{\leqslant} s \wedge s \dot{\leqslant} r$ by induction (with $r \underline{\leftrightarrow} s \equiv r \underset{\rightarrow}{\rightarrow} s \wedge s \underset{\rightarrow}{\rightarrow} r$) and $\alpha \cdot r \doteq \alpha \cdot (s + r) \doteq \beta \cdot (r + s) \doteq \beta \cdot s$, or

  - $q' = s + s'$, with basic terms s and s', which means that, by Lemma 6.13, $\exists_{t,t'}(p' \doteq t + t' \wedge \#(t) \leq \#(p') \wedge \#(t') \leq \#(p') \wedge t \underset{\rightarrow}{\rightarrow} s \wedge t' \underset{\rightarrow}{\rightarrow} s')$ and by induction $\exists_{t,t'}(p' \doteq t + t' \wedge t \dot{\leqslant} s \wedge t' \dot{\leqslant} s') \Rightarrow \exists_{t,t'}(p' \doteq t + t' \wedge t \dot{\leqslant} s + s' \wedge t' \dot{\leqslant} s + s') \Rightarrow \exists_{t,t'}(p' \doteq t + t' \wedge t + t' \dot{\leqslant} s + s') \Rightarrow p' \dot{\leqslant} s + s' = q'$.

- $p' = r + r'$, with basic terms r and r', which means that $p' \underset{\rightarrow}{\rightarrow} q' \equiv r + r' \underset{\rightarrow}{\rightarrow} q' \equiv r \underset{\rightarrow}{\rightarrow} q' \wedge r' \underset{\rightarrow}{\rightarrow} q' \overset{IH}{\equiv} r \dot{\leqslant} q' \wedge r' \dot{\leqslant} q' \equiv r + r' \dot{\leqslant} q' \equiv p' \dot{\leqslant} q'$.

The proof of $q' \underset{\rightarrow}{\rightarrow} p' \Rightarrow q' \dot{\leqslant} p'$ is symmetrical to the proof above. $\qquad \square$

7   Abstraction

If we want τ (or $\langle \rangle$) to be a "real" *silent step*, we want to be able to remove τ where its presence can not be determined. Our current definition of bisimulation therefore no longer suits us and we therefore introduce another form of bisimulation.

We use rooted branching bisimulation $\underline{\leftrightarrow}_{rb}$ and, as our semantics fits the "RBB cool" format [3], we know $\underline{\leftrightarrow}_{rb}$ is a congruence. To be able to give a nice definition we introduce a termination predicate $\downarrow$ and assume that $\checkmark$ is a state (but not an allowed process constant) and that it is the only state for which $\downarrow$ holds (i.e. $x \xrightarrow{m} \checkmark$ now just means that x can make a transition m to $\checkmark$ and $\checkmark$ terminates).

Definition 7.1. Let $Sem_{LoCo}(E) = \langle T_P, \mathbb{B}(\boldsymbol{A}), \longrightarrow, \longrightarrow \checkmark \rangle$, with E a set of process expressions and $\longrightarrow$ and $\longrightarrow \checkmark$ the transition relations on processes T_P with multiactions $\mathbb{B}(\boldsymbol{A})$, be a process semantics. Also, let t, t', u and u' be process terms in which process variables may occur only if they are in E and $m \in \mathbb{B}(\boldsymbol{A})$. *Branching bisimulation* $\underline{\leftrightarrow}_b$ is the union of all relations B such that if tBu (with $\xrightarrow{m^*}$ zero or more m-transitions):

- for all t' and m, $t \overset{m}{\longrightarrow} t'$ means that $m = [] \wedge t'Bu$ or there exist u' and u'' with $u \overset{[]^*}{\longrightarrow} u'' \overset{m}{\longrightarrow} u' \wedge tBu'' \wedge t'Bu'$

- for all u' and m, $u \overset{m}{\longrightarrow} u'$ means that $m = [] \wedge tBu'$ or there exist t' and t'' with $t \overset{[]^*}{\longrightarrow} t'' \overset{m}{\longrightarrow} t' \wedge t''Bu \wedge t'Bu'$

- $t \downarrow$ means that there exists a u' with $u \overset{[]^*}{\longrightarrow} u' \downarrow \wedge tBu'$

- $u \downarrow$ means that there exists a u' with $t \overset{[]^*}{\longrightarrow} t' \downarrow \wedge t'Bu$

Definition 7.2. Let $Sem_{LoCo}(E) = \langle T_P, \mathbb{B}(\boldsymbol{A}), \longrightarrow, \longrightarrow \checkmark \rangle$, with E a set of process expressions and $\longrightarrow$ and $\longrightarrow \checkmark$ the transition relations on processes T_P with multiactions $\mathbb{B}(\boldsymbol{A})$, be a process semantics. Also, let t, t', u and u' be process terms in which process variables may occur only if they are in E and $m \in \mathbb{B}(\boldsymbol{A})$. *Rooted branching bisimulation* $\underline{\leftrightarrow}_{rb}$ is defined by $t \underline{\leftrightarrow}_{rb} u$ if, and only if, $t \underline{\leftrightarrow}_b u \wedge rooted(t, u)$, with $rooted(t, u)$ defined as follows:

- if $t \overset{m}{\longrightarrow} t'$, then $u \overset{m}{\longrightarrow} u' \wedge t' \underline{\leftrightarrow}_b u'$

- if $u \overset{m}{\longrightarrow} u'$, then $t \overset{m}{\longrightarrow} t' \wedge t' \underline{\leftrightarrow}_b u'$

- if $t \downarrow$, then $u \downarrow$

- if $u \downarrow$, then $t \downarrow$

Now we have this new form of equivalence, we also need a matching (i.e. sound and complete) axiomatisation. Fortunately, the axioms given before are still sound, but to make the axiomatisation (relatively) complete again (i.e. to have axioms that reflect the behaviour of τ) we believe it is sufficient to add the following two axioms, as in [4]. (Note that x, y and z cannot be $\checkmark$).

$$T1 \quad x \cdot \tau \doteq x$$
$$T2 \quad x \cdot (\tau \cdot (y + z) + y) \doteq x \cdot (y + z)$$

In Appendix B we have proved the soundness of the axioms $T1$ and $T2$. The other axioms of $LoCo$, which have already been proven to be sound with respect to $\underline{\leftrightarrow}$, do not need to be proven again, as $\underline{\leftrightarrow} \subset \underline{\leftrightarrow}_{rb}$ holds. The structure of the proofs of $T1$ and $T2$ is the same as before, but instead of proving R to be a bisimulation we have proved R to be a branching bisimulation and $rooted(t, u)$ (for axiom $t \doteq u$).

When working with abstraction, one might encounter processes of the following form: $X = i \cdot X + Y$, where one wishes to hide i. This process can in fact do an infinite amount of i's, but, after hiding, these actions should not be observable. One usually wishes to have some form of fairness in which an infinite sequence of τ's is not possible. To express this we introduce a fairness rule.

In current algebras, like ACP, there exist rules as "KFAR" and "CFAR" [2]. Although these rules basically express the same fairness as we wish to express and fit in our own language, they need that the (recursive) specification is linearised (before the application of the abstraction operator τ_I).

Instead we wish to see if it is possible to first apply the abstraction and afterwards apply some form of fairness rule to the (possibly, or even probably) reduced specification. The proposition of the basic idea follows. It is clear that if this rule is suitable, we can extend it to a more general rule (as "CFAR").

Proposition 7.3. The *fairness rule* we propose is the following:

$$\frac{Y \doteq \tau_{\{i\}}(Y),\, X' \doteq i \cdot X' + Y,\, X \doteq \tau_{\{i\}}(X')}{\tau \cdot X \doteq \tau \cdot Y}$$

This rule basically follows from KFAR, as shown in the following derivation:

$$\frac{Y \doteq \tau_{\{i\}}(Y), X' \doteq i \cdot X' + Y, X \doteq \tau_{\{i\}}(X')}{Y \doteq \tau_{\{i\}}(Y), \tau \cdot \tau_{\{i\}}(X') \doteq \tau \cdot \tau_{\{i\}}(Y), X \doteq \tau_{\{i\}}(X')}$$
$$\tau \cdot X \doteq \tau \cdot Y$$

One might wonder why the X is defined as an abstraction of X' and not just as $X = \tau \cdot X + Y$. The problem with this equation is that, for example, $X = \tau \cdot a$, $Y = \delta$ is a solution, but with the abstraction of X' this is not the case. This does mean, however, that our wish to be able to simply apply a fair abstraction rule to an abstracted specification is not possible, unless we know that the variable in question satisfies the conditions of the above rule. Fortunately, this will be the case in practice, as the abstracted specification is an abstraction of some specification. It should therefore be save to use this rule as the following, if one knows the occurring τ is a result of abstraction:

$$\frac{X \doteq \tau X + Y}{\tau X \doteq \tau Y}$$

8   Examples

8.1   Synchronising 1

Assume we have the following two processes A and B:

$$A = a \cdot s_a \ , \quad B = s_b \cdot b$$

If these processes are running in parallel and we wish that they synchronise on the s actions, such that a happens before b, we can write the following process:

$$S = \nabla_{\{a,b\}}(\tau_{\{s_{ab}\}}(\Gamma_{\{s_a|s_b \rightarrow s_{ab}\}}(A \parallel B)))$$

By applying the axioms we can show that this process indeed does what we wished it did.

$$
\begin{aligned}
S \ &= \ \nabla_{\{a,b\}}(\tau_{\{s_{ab}\}}(\Gamma_{\{s_a|s_b \rightarrow s_{ab}\}}(A \parallel B))) \\
&\doteq \ \nabla_{\{a,b\}}(\tau_{\{s_{ab}\}}(\Gamma_{\{s_a|s_b \rightarrow s_{ab}\}}(A \,\|\!\|\, B + B \,\|\!\|\, A + A|B))) \\
&\doteq \ \nabla_{\{a,b\}}(\tau_{\{s_{ab}\}}(\Gamma_{\{s_a|s_b \rightarrow s_{ab}\}}(a \cdot (s_a \parallel B) + s_b \cdot (b \parallel A) + \langle a, s_b \rangle \cdot (s_a \parallel b)))) \\
&\doteq \ \nabla_{\{a,b\}}(\tau_{\{s_{ab}\}}(\Gamma_{\{s_a|s_b \rightarrow s_{ab}\}}(a \cdot (s_a \parallel B)))) + \\
&\quad\ \ \nabla_{\{a,b\}}(\tau_{\{s_{ab}\}}(\Gamma_{\{s_a|s_b \rightarrow s_{ab}\}}(s_b \cdot (b \parallel A)))) + \\
&\quad\ \ \nabla_{\{a,b\}}(\tau_{\{s_{ab}\}}(\Gamma_{\{s_a|s_b \rightarrow s_{ab}\}}(\langle a, s_b \rangle \cdot (s_a \parallel b)))) \\
&\doteq \ a \cdot \nabla_{\{a,b\}}(\tau_{\{s_{ab}\}}(\Gamma_{\{s_a|s_b \rightarrow s_{ab}\}}(s_a \parallel B))) + \delta + \delta \\
&\doteq \ a \cdot \nabla_{\{a,b\}}(\tau_{\{s_{ab}\}}(\Gamma_{\{s_a|s_b \rightarrow s_{ab}\}}(s_a \,\|\!\|\, B + B \,\|\!\|\, s_a + \langle s_a, s_b \rangle \cdot b))) \\
&\doteq \ a \cdot \nabla_{\{a,b\}}(\tau_{\{s_{ab}\}}(\Gamma_{\{s_a|s_b \rightarrow s_{ab}\}}(s_a \cdot B + s_b \cdot (b \parallel s_a) + \langle s_a, s_b \rangle \cdot b))) \\
&\doteq \ a \cdot (\nabla_{\{a,b\}}(\tau_{\{s_{ab}\}}(\Gamma_{\{s_a|s_b \rightarrow s_{ab}\}}(s_a \cdot B))) + \\
&\quad\ \ \ \ \ \nabla_{\{a,b\}}(\tau_{\{s_{ab}\}}(\Gamma_{\{s_a|s_b \rightarrow s_{ab}\}}(s_b \cdot (b \parallel s_a)))) + \\
&\quad\ \ \ \ \ \nabla_{\{a,b\}}(\tau_{\{s_{ab}\}}(\Gamma_{\{s_a|s_b \rightarrow s_{ab}\}}(\langle s_a, s_b \rangle \cdot b)))) \\
&\doteq \ a \cdot (\delta + \delta + \tau \cdot \nabla_{\{a,b\}}(\tau_{\{s_{ab}\}}(\Gamma_{\{s_a|s_b \rightarrow s_{ab}\}}(b)))) \\
&\doteq \ a \cdot \tau \cdot b \\
&\doteq \ a \cdot b
\end{aligned}
$$

8.2   Synchronising 2

Assume we have three processes in parallel and we wish that they synchronise on an action *tick*. For simplicity we use the following definition for the processes:
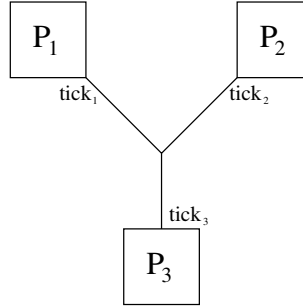
$$P_i = tick_i \cdot P_i$$

Figure 1: Synchronising processes.

When putting three of these processes in parallel, we get the following behaviour:

$$
\begin{aligned}
P^3 \;=\;\; & P_0 \parallel P_1 \parallel P_2 \\
\doteq\;\; & P_0 \parallel (P_1 \mathbin{\rlap{\parallel}{\,\_}} P_2 + P_2 \mathbin{\rlap{\parallel}{\,\_}} P_1 + P_1|P_2) \\
\doteq\;\; & P_0 \parallel (\langle\, tick_1\, \rangle \cdot (P_1 \parallel P_2) + (\langle\, tick_2\, \rangle \cdot (P_1 \parallel P_2) + \langle\, tick_1,\, tick_2\, \rangle \cdot (P_1 \parallel P_2))) \\
\doteq\;\; & P_0 \mathbin{\rlap{\parallel}{\,\_}} (\langle\, tick_1\, \rangle \cdot (P_1 \parallel P_2) + (\langle\, tick_2\, \rangle \cdot (P_1 \parallel P_2) + \langle\, tick_1,\, tick_2\, \rangle \cdot (P_1 \parallel P_2)))+ \\
& (\langle\, tick_1\, \rangle \cdot (P_1 \parallel P_2) + (\langle\, tick_2\, \rangle \cdot (P_1 \parallel P_2) + \langle\, tick_1,\, tick_2\, \rangle \cdot (P_1 \parallel P_2)) \mathbin{\rlap{\parallel}{\,\_}} P_0+ \\
& P_0|(\langle\, tick_1\, \rangle \cdot (P_1 \parallel P_2) + (\langle\, tick_2\, \rangle \cdot (P_1 \parallel P_2) + \langle\, tick_1,\, tick_2\, \rangle \cdot (P_1 \parallel P_2)) \\
\doteq\;\; & \langle\, tick_0\, \rangle \cdot (P_0 \parallel (\langle\, tick_1\rangle \cdot (P_1 \parallel P_2) + (\langle\, tick_2\, \rangle \cdot (P_1 \parallel P_2) + \langle\, tick_1,\, tick_2\rangle \cdot (P_1 \parallel P_2)))+ \\
& \langle\, tick_1\rangle \cdot (P_0 \parallel P_1 \parallel P_2) + \langle\, tick_2\rangle \cdot (P_0 \parallel P_1 \parallel P_2) + \langle\, tick_1,\, tick_2\rangle \cdot (P_0 \parallel P_1 \parallel P_2)+ \\
& \langle\, tick_0,\, tick_1\rangle \cdot (P_0 \parallel P_1 \parallel P_2) + \langle\, tick_0,\, tick_2\rangle \cdot (P_0 \parallel P_1 \parallel P_2) + \langle\, tick_0,\, tick_1,\, tick_2\rangle \cdot (P_0 \parallel P_1 \parallel P_2) \\
\doteq\;\; & \langle\, tick_0\rangle \cdot P^3 + \langle\, tick_1\rangle \cdot P^3 + \langle\, tick_2\rangle \cdot P^3+ \\
& \langle\, tick_0,\, tick_1\rangle \cdot P^3 + \langle\, tick_0,\, tick_2\rangle \cdot P^3 + \langle\, tick_1,\, tick_2\rangle \cdot P^3+ \\
& \langle\, tick_0,\, tick_1,\, tick_2\rangle \cdot P^3 \\
\doteq\;\; & (\langle\, tick_0\rangle + \langle\, tick_1\rangle + \langle\, tick_2\rangle+ \\
& \langle\, tick_0,\, tick_1\rangle + \langle\, tick_0,\, tick_2\rangle + \langle\, tick_1,\, tick_2\rangle+ \\
& \langle\, tick_0,\, tick_1,\, tick_2\rangle) \cdot P^3
\end{aligned}
$$

As we can see, and as could have been expected, the three processes in parallel can do their actions separately or synchronously with one or two of the other processes. We stated before that we wish to synchronise them all, so we will put a ∇_V, with $V = \{\, tick_0|\, tick_1|\, tick_2\}$, around P^3 so only this multiaction in V may happen.

$$
\begin{aligned}
P_V^3 \;=\;\; & \nabla_V(P^3) \\
\doteq\;\; & \langle\, tick_0,\, tick_1,\, tick_2\rangle \cdot \nabla_V(P^3) \\
\doteq\;\; & \langle\, tick_0,\, tick_1,\, tick_2\rangle \cdot P_V^3
\end{aligned}
$$

9 Alphabet axioms

As the linearisation of parallel expressions has the tendency to significantly increase the amount of memory needed to store resulting expressions and often such expressions are enclosed by for instance restriction operators, one wishes to be able to remove as much of the (sub)expressions as possible before elimination of parallel operators. To help this process we give a list of alphabet axioms, which can be used to easily *push* certain operators deeper into an expression to (partially) effectuate its behaviour and possibly limiting the increase in memory needed for linearisation.

Definition 9.1. Let $p, p' \in T_{pc}$ and $m \in \mathbb{B}(\boldsymbol{A})$. We define the alphabet $\alpha_\nu(p)$ of a process p by $(p \xrightarrow{m} \checkmark \Rightarrow \mu(m) \in \alpha_\nu(p)) \wedge (p \xrightarrow{m} p' \Rightarrow \{\mu(m)\} \cup \alpha_\nu(p') \subseteq \alpha_\nu(p))$.

Note that the above definition is in terms of the semantics. This suits us when proving soundness, but during linearisation one probably wants an axiomatic definition. Although we do not give such a definition here, we do want to note that in practice it is probably well acceptable to use an over-approximation for α_ν (i.e. some $\widetilde{\alpha_\nu}$ with $\forall_t(\alpha_\nu(t) \subseteq \widetilde{\alpha_\nu}(t)))$, as if the axioms presented below hold for such an over-approximation, they will certainly hold for α_ν.

Definition 9.2. Let $a, b \in \mathcal{N}_A$, $v, w \in \mathbb{B}(\mathcal{N}_A)$ and $V \subseteq \mathbb{B}(\mathcal{N}_A)$. We define the set $\mathcal{N}(V)$ of actions in a set of bags V as follows:

$$\mathcal{N}(V) = \{a \mid a \in v \wedge v \in V\}$$

Definition 9.3. Let $a, b \in \mathbb{B}(\mathcal{N}_A)$ and $V \subseteq \mathbb{B}(\mathcal{N}_A)$. We define the set $\Downarrow(V)$ of subbags of bags in set V by $\Downarrow(V) = \{b \mid a \in V \wedge b \subseteq a\}$.

Definition 9.4. Let S, T be sets and $F : S \to T$. Also, let $s \in S$ and $t \in T$. We define the $dom(F)$ and $rng(F)$ as follows:

$$\begin{array}{rcl} dom(F) & = & \{s \mid \exists_{t \in T}(\langle s, t \rangle \in F)\} \\ rng(F) & = & \{t \mid \exists_{s \in S}(\langle s, t \rangle \in F)\} \end{array}$$

Note we write $S \cap T$ and $S \cup T$ while we actually mean a (syntactic) set U such that $[\![U]\!] = [\![S]\!] \cap [\![T]\!]$ or $[\![U]\!] = [\![S]\!] \cup [\![T]\!]$ respectively.

With these definitions we have the following alphabet axioms:

VA1	$\nabla_V(x) \doteq x$	$if \; \alpha_\nu(x) \subseteq [\![V]\!]$
VA2	$\nabla_V(x) \doteq \delta$	$if \; [\![V]\!] \cap \alpha_\nu(x) = \emptyset$
VA3	$\nabla_V(\nabla_{V'}(x)) \doteq \nabla_{V \cap V'}(x)$	
VA4	$\nabla_V(x \parallel y) \doteq \nabla_V(x \parallel \nabla_{V'}(y))$	$if \; \Downarrow([\![V]\!]) \subseteq [\![V']\!]$

CA1	$\Gamma_C(x) \doteq x$	$if \; dom([\![C]\!]) \cap \Downarrow(\alpha_\nu(x)) = \emptyset$
CA2	$\Gamma_C(\Gamma_{C'}(x)) \doteq \Gamma_{C \cup C'}(x)$	$if \; \mathcal{N}(dom([\![C]\!])) \cap \mathcal{N}(dom([\![C']\!])) = \emptyset \; \wedge \; \mathcal{N}(dom([\![C]\!])) \cap rng([\![C']\!]) = \emptyset$
CA3	$\Gamma_C(x \parallel y) \doteq x \parallel \Gamma_C(y)$	$if \; \Downarrow(dom([\![C]\!])) \cap \Downarrow(\alpha_\nu(x)) = \emptyset$
CA4	$\Gamma_C(x \parallel y) \doteq \Gamma_C(x \parallel \Gamma_C(y))$	$if \; \mathcal{N}(dom([\![C]\!])) \cap rng([\![C]\!]) = \emptyset$

DA1	$\partial_H(x) \doteq x$	$if \; [\![H]\!] \cap \mathcal{N}(\alpha_\nu(x)) = \emptyset$
DA2	$\partial_H(x) \doteq \delta$	$if \; \forall_{v \in \alpha_\nu(x)}(v \cap [\![H]\!] \neq \emptyset)$
DA3	$\partial_H(\partial_{H'}(x)) \doteq \partial_{H \cup H'}(x)$	
DA4	$\partial_H(x \parallel y) \doteq \partial_H(x) \parallel \partial_H(y)$	

TA1	$\tau_I(x) \doteq x$	$if \; [\![I]\!] \cap \mathcal{N}(\alpha_\nu(x)) = \emptyset$
TA3	$\tau_I(\tau_{I'}(x)) \doteq \tau_{I \cup I'}(x)$	
TA4	$\tau_I(x \parallel y) \doteq \tau_I(x) \parallel \tau_I(y)$	

RA1	$\rho_R(x) \doteq x$	$if \; dom([\![R]\!]) \cap \mathcal{N}(\alpha_\nu(x)) = \emptyset$
RA2	$\rho_R(\rho_{R'}(x)) \doteq \rho_{R \cup R'}(x)$	$if \; dom([\![R]\!]) \cap dom([\![R']\!]) = \emptyset \wedge dom([\![R]\!]) \cap rng([\![R']\!]) = \emptyset$
RA3	$\rho_R(\rho_{R'}(x)) \doteq \rho_{R''}(x)$	$if \; [\![R'']\!] = \{\langle a, b \rangle \mid (\langle a, b \rangle \in [\![R]\!] \wedge a \notin (dom([\![R']\!]) \cup rng([\![R']\!]))) \vee$
		$(\langle c, b \rangle \in [\![R]\!] \wedge \langle a, c \rangle \in [\![R']\!]) \vee$
		$(\langle a, b \rangle \in [\![R']\!] \wedge b \notin dom([\![R]\!]))\}$
RA4	$\rho_R(x \parallel y) \doteq \rho_R(x) \parallel \rho_R(y)$	

VC1	$\nabla_V(\Gamma_C(x)) \doteq \nabla_V(\Gamma_C(\nabla_{V'}(x)))$	$if \; [\![V']\!] = \{v \oplus w \mid \gamma(v, [\![C]\!]) \oplus w \in [\![V]\!]\}$
VC2	$\Gamma_C(\nabla_V(x)) \doteq \nabla_V(x)$	$if \; dom(C) \cap \Downarrow(V) = \emptyset$

$VD1 \quad \nabla_V(\partial_H(x)) \doteq \partial_H(\nabla_V(x))$

$VD2 \quad \nabla_V(\partial_H(x)) \doteq \nabla_{V'}(x) \quad \text{if } \llbracket V' \rrbracket = \{v \mid v \in \llbracket V \rrbracket \wedge \mathcal{N}(\{v\}) \cap \llbracket H \rrbracket = \emptyset\}$

$VD3 \quad \partial_H(\nabla_V(x)) \doteq \nabla_{V'}(x) \quad \text{if } \llbracket V' \rrbracket = \{v \mid v \in \llbracket V \rrbracket \wedge \mathcal{N}(\{v\}) \cap \llbracket H \rrbracket = \emptyset\}$

$VT \quad \nabla_V(\tau_I(x)) \doteq \tau_I(\nabla_{V'}(x)) \quad \text{if } \llbracket V' \rrbracket = \{v \mid \theta(v, \llbracket I \rrbracket) \in \llbracket V \rrbracket\}$

$VR \quad \nabla_V(\rho_R(x)) \doteq \rho_R(\nabla_{V'}(x)) \quad \text{if } \llbracket V' \rrbracket = \{v \mid (\llbracket R \rrbracket \bullet v) \in \llbracket V \rrbracket\}$

$CD1 \quad \partial_H(\Gamma_C(x)) \doteq \Gamma_C(\partial_H(x)) \quad \text{if } (\mathcal{N}(dom(C)) \cup rng(C)) \cap H = \emptyset$

$CD2 \quad \Gamma_C(\partial_H(x)) \doteq \partial_H(x) \quad \text{if } \mathcal{N}(dom(C)) \subseteq H$

$CT1 \quad \tau_I(\Gamma_C(x)) \doteq \Gamma_C(\tau_I(x)) \quad \text{if } (\mathcal{N}(dom(C)) \cup rng(C)) \cap I = \emptyset$

$CT2 \quad \Gamma_C(\tau_I(x)) \doteq \tau_I(x) \quad \text{if } \mathcal{N}(dom(C)) \subseteq I$

$CR1 \quad \rho_R(\Gamma_C(x)) \doteq \Gamma_C(\rho_R(x)) \quad \text{if } dom(R) \cap rng(C) = dom(R) \cap \mathcal{N}(dom(C)) = rng(R) \cap \mathcal{N}(dom(C)) = \emptyset$

$CR2 \quad \Gamma_C(\rho_R(x)) \doteq \rho_R(x) \quad \text{if } \mathcal{N}(dom(C)) \subseteq dom(R) \wedge \mathcal{N}(dom(C)) \cap rng(R) = \emptyset$

$DT \quad \partial_H(\tau_I(x)) \doteq \tau_I(\partial_H(x)) \quad \text{if } \llbracket I \rrbracket \cap \llbracket H \rrbracket = \emptyset$

$DR \quad \partial_H(\rho_R(x)) \doteq \rho_R(\partial_{H'}(x)) \quad \text{if } \llbracket H' \rrbracket = \{v \mid (\llbracket R \rrbracket \bullet v) \in \llbracket H \rrbracket\}$

$TR \quad \tau_I(\rho_R(x)) \doteq \rho_R(\tau_{I'}(x)) \quad \text{if } \llbracket I \rrbracket = \{\llbracket R \rrbracket^+(a) \mid a \in \llbracket I' \rrbracket\}$

Table 3: *LoCo* Alphabet Axioms

Theorem 9.5. All alphabet axioms are sound, i.e. for all axioms $t \doteq u$ and condition C in Table 3 it holds that for all processes p and q in T_{pc}, where p and q are instantiations of t and u, respectively, such that C holds, $C \wedge LoCo \vdash p \doteq q \Rightarrow LoCo \vDash p \underline{\leftrightarrow} q$

Proof 9.5. To prove the soundness of these alphabet axioms, we use a different approach as before, except for the axioms containing a parallel operator. As the operators used in these axioms are all defined in a similar way, namely:

$$\frac{x \xrightarrow{m} \checkmark}{O(x) \xrightarrow{f(m)} \checkmark} \, C(m) \qquad \frac{x \xrightarrow{m} x'}{O(x) \xrightarrow{f(m)} O(x')} \, C(m)$$

If we now wish to prove an axiom $O_1(\ldots O_n(x) \ldots) \doteq O_{n+1}(\ldots O_{n'}(x) \ldots)$, we get the following:

- $O_1(\ldots O_n(x) \ldots) \xrightarrow{m} \checkmark$, which means that $O_2(\ldots O_n(x) \ldots) \xrightarrow{m_1} \checkmark \wedge C_1(m_1) \wedge m = f_1(m_1)$ $\wedge \ldots \wedge x \xrightarrow{m_n} \checkmark \wedge C_n(m_n) \wedge m_{n-1} = f_n(m_n)$ and therefore $C_{n+1}(m_n) \wedge \ldots \wedge C_{n'}(f_{n'-1}(\ldots f_{n+1}(m_n) \ldots))$ and $m = f'_n(\ldots f_{n+1}(m_n) \ldots)$ and $O_{n+1}(\ldots O_{n'}(x) \ldots) \xrightarrow{m} \checkmark$

- Etc.

So what really has to be proven is that for each m, such that $x \xrightarrow{m} \checkmark \vee x \xrightarrow{m} x'$, it holds that $C_n(m) \wedge \ldots \wedge C_1(f_2(\ldots f_n(m))) \equiv C_{n'}(m) \wedge \ldots \wedge C_{n+1}(f_{n+2}(\ldots f_{n'}(m)))$ and $C_n(m) \wedge \ldots \wedge C_1(f_2(\ldots f_n(m))) \Rightarrow f_1(\ldots f_n(m)) = f_{n+1}(\ldots f_{n'}(m))$.

The proofs for the alphabet axioms are given in Appendix C. $\hfill \square$

A   Soundness proofs

The structure of each of these proofs is the same, namely:

- A relation R is given for the axiom. For an axiom $t \doteq u$, the relation R is the minimal relation that satisfies $tRu \wedge pRp \wedge (qRr \Leftrightarrow rRq) \wedge E$, for all $p, q, r \in T_{pc}$ and for all instantiations of process variables in t and u. E is *true*, unless it is explicitly defined otherwise in the subsection.

- For each conjunct of the definition of R it is shown that the properties that define a bisimulation hold.

It is clear that if this is done, R is proven to be a bisimulation relation.

To shorten the proofs and get rid of trivial facts, that would otherwise be repeated in every subsection, we do not define R explicitly (as it is always of the form given above) or prove the conjuncts pRp and $qRr \Leftrightarrow rRq$ (and E, if it is *true*). The proofs of these conjuncts are trivial and therefore not given at all.

When we say a process cannot terminate in any of the subsections, we mean that there is no m for which $\xrightarrow{m} \checkmark$ holds for that process. (So it could very well be that the process can indeed terminate, but not in one "step".)

A.1 $MA1$ $a \doteq \langle a \rangle$

- $a \xrightarrow{m} \checkmark$, which means that $m = [a]$ and $\langle a \rangle \xrightarrow{[a]} \checkmark$

- $\langle a \rangle \xrightarrow{m} \checkmark$, which is the same as the previous case

- $a \xrightarrow{m} p$, which is not possible

- $\langle a \rangle \xrightarrow{m} p$, which is not possible

A.2 $MA2$ $\tau \doteq \langle \rangle$

This proof is similar to that of axiom $MA2$ with $m = []$.

A.3 $MA3$ $\langle \overrightarrow{a}, \overrightarrow{b} \rangle \doteq \langle \overrightarrow{b}, \overrightarrow{a} \rangle$

Both $\langle \overrightarrow{a}, \overrightarrow{b} \rangle$ and $\langle \overrightarrow{b}, \overrightarrow{a} \rangle$ cannot do any transition. They can only terminate with $[\![\langle \overrightarrow{a}, \overrightarrow{b} \rangle]\!]$ and $[\![\langle \overrightarrow{b}, \overrightarrow{a} \rangle]\!]$ resp., which are equal.

A.4 $A1$ $x + y \doteq y + x$

- $p + q \xrightarrow{m} \checkmark$, which means that

    - $p \xrightarrow{m} \checkmark$, which means that $q + p \xrightarrow{m} \checkmark$, or
    - $q \xrightarrow{m} \checkmark$, which means that $q + p \xrightarrow{m} \checkmark$

- $q + p \xrightarrow{m} \checkmark$, symmetrical to the prove above

- $p + q \xrightarrow{m} r$, which means that

    - $p \xrightarrow{m} r$, which means that $q + p \xrightarrow{m} r$, with rRr, or
    - $q \xrightarrow{m} r$, which means that $q + p \xrightarrow{m} r$, with rRr

- $q + p \xrightarrow{m} r$, symmetrical to the prove above

A.5   A2 $x + (y + z) \doteq (x + y) + z$

- $p + (q + r) \xrightarrow{m} \checkmark$, which means that

  - $p \xrightarrow{m} \checkmark$, which means that $p + q \xrightarrow{m} \checkmark$ and $(p + q) + r \xrightarrow{m} \checkmark$, or
  - $q + r \xrightarrow{m} \checkmark$, which means that
    - $q \xrightarrow{m} \checkmark$, which means that $p + q \xrightarrow{m} \checkmark$ and $(p + q) + r \xrightarrow{m} \checkmark$, or
    - $r \xrightarrow{m} \checkmark$, which means that $(p + q) + p \xrightarrow{m} \checkmark$

- $(p + q) + r \xrightarrow{m} \checkmark$, symmetrical to the prove above

- $p + (q + r) \xrightarrow{m} r'$, which means that

  - $p \xrightarrow{m} r'$, which means that $p + q \xrightarrow{m} r'$ and $(p + q) + r \xrightarrow{m} r'$, with $r'Rr'$, or
  - $q + r \xrightarrow{m} r'$, which means that
    - $q \xrightarrow{m} r'$, which means that $p + q \xrightarrow{m} r'$ and $(p + q) + r \xrightarrow{m} r'$, with $r'Rr'$, or
    - $r \xrightarrow{m} r'$, which means that $(p + q) + r \xrightarrow{m} r'$, with $r'Rr'$

- $(p + q) + r \xrightarrow{m} r'$, symmetrical to the prove above

A.6   A3 $x + x \doteq x$

- $p + p \xrightarrow{m} \checkmark$, which means that $p \xrightarrow{m} \checkmark$

- $p \xrightarrow{m} \checkmark$, which means that $p + p \xrightarrow{m} \checkmark$

- $p + p \xrightarrow{m} p'$, which means that $p \xrightarrow{m} p'$, with $p'Rp'$

- $p \xrightarrow{m} p'$, which means that $p + p \xrightarrow{m} p'$, with $p'Rp'$

A.7   A4 $(x + y) \cdot z \doteq x \cdot z + y \cdot z$

- $(p + q) \cdot r \xrightarrow{m} \checkmark$, which is not possible

- $p \cdot r + q \cdot r \xrightarrow{m} \checkmark$, which means that

  - $p \cdot r \xrightarrow{m} \checkmark$, which is not possible, or
  - $q \cdot r \xrightarrow{m} \checkmark$, which is not possible

- $(p + q) \cdot r \xrightarrow{m} r'$, which means that

  - $p + q \xrightarrow{m} \checkmark \wedge r' = r$, which means that
    - $p \xrightarrow{m} \checkmark$, which means that $p \cdot r \xrightarrow{m} r$ and $p \cdot r + q \cdot r \xrightarrow{m} r$, with $r'Rr$, or
    - $q \xrightarrow{m} \checkmark$, which means that $q \cdot r \xrightarrow{m} r$ and $p \cdot r + q \cdot r \xrightarrow{m} r$, with $r'Rr$, or
  - $p + q \xrightarrow{m} q' \wedge r' = q' \cdot r$ which means that
    - $p \xrightarrow{m} q'$, which means that $p \cdot r \xrightarrow{m} q' \cdot r$ and $p \cdot r + q \cdot r \xrightarrow{m} q' \cdot r$, with $r'Rq' \cdot r$, or
    - $q \xrightarrow{m} q'$, which means that $q \cdot r \xrightarrow{m} q' \cdot r$ and $p \cdot r + q \cdot r \xrightarrow{m} q' \cdot r$, with $r'Rq' \cdot r$

- $p \cdot r + q \cdot r \xrightarrow{m} r'$, which means that

  - $p \cdot r \xrightarrow{m} r'$, which means that
    - $p \xrightarrow{m} \checkmark \wedge r' = r$, which means that $p + q \xrightarrow{m} \checkmark$ and $(p + q) \cdot r \xrightarrow{m} r$, with $r'Rr$, or

- $p\xrightarrow{m}p' \wedge r' = p' \cdot r$, which means that $p + q\xrightarrow{m}p'$ and $(p+q) \cdot r\xrightarrow{m}p' \cdot r$, with $r'Rp' \cdot r$, or

- $q \cdot r\xrightarrow{m}r'$, which means that

  - $q\xrightarrow{m}\checkmark \wedge r' = r$, which means that $p + q\xrightarrow{m}\checkmark$ and $(p+q) \cdot r\xrightarrow{m}r$, with $r'Rr$, or
  - $q\xrightarrow{m}q' \wedge r' = q' \cdot r$, which means that $p + q\xrightarrow{m}q'$ and $(p+q) \cdot r\xrightarrow{m}q' \cdot r$, with $r'Rq' \cdot r$

A.8 $A5$ $(x \cdot y) \cdot z \doteq x \cdot (y \cdot z)$

- $(p \cdot q) \cdot r\xrightarrow{m}\checkmark$, which is not possible

- $p \cdot (q \cdot r)\xrightarrow{m}\checkmark$, which is not possible

- $(p \cdot q) \cdot r\xrightarrow{m}r'$, which means that

  - $p \cdot q\xrightarrow{m}\checkmark$, which is not possible
  - $p \cdot q\xrightarrow{m}q' \wedge r' = q' \cdot r$, which means that
    - $p\xrightarrow{m}\checkmark \wedge q' = q$, which means that $p \cdot (q \cdot r)\xrightarrow{m}q \cdot r$, with $r'Rq \cdot r$, or
    - $p\xrightarrow{m}p' \wedge q' = p' \cdot q$, which means that $p \cdot (q \cdot r)\xrightarrow{m}p' \cdot (q \cdot r)$, with $r'R(p' \cdot q) \cdot r$

A.9 $A6$ $x + \delta \doteq x$

- $p + \delta\xrightarrow{m}\checkmark$, which means that $p\xrightarrow{m}\checkmark$

- $p\xrightarrow{m}\checkmark$, which means that $p + \delta\xrightarrow{m}\checkmark$

- $p + \delta\xrightarrow{m}p'$, which means that $p\xrightarrow{m}p'$, with $p'Rp'$

- $p\xrightarrow{m}p'$, which means that $p + \delta\xrightarrow{m}p'$, with $p'Rp'$

A.10 $A7$ $\delta \cdot x \doteq \delta$

Both $\delta \cdot p$ and δ cannot terminate or do any transition.

A.11 $CM1$ $x \parallel y \doteq x \,\|\!\|\, y + y \,\|\!\|\, x + x|y$

We define E as $p \parallel qRq \parallel p$ (for all $p, q \in T_{pc}$).

- $p \parallel q\xrightarrow{m}\checkmark$, which means that $p\xrightarrow{n}\checkmark \wedge q\xrightarrow{n'}\checkmark \wedge m = n \oplus n'$ and $p|q\xrightarrow{m}\checkmark$ and $p \,\|\!\|\, q + q \,\|\!\|\, p + p|q\xrightarrow{m}\checkmark$

- $p \,\|\!\|\, q + q \,\|\!\|\, p + p|q\xrightarrow{m}\checkmark$, which means that

  - $p \,\|\!\|\, q\xrightarrow{m}\checkmark$, which is not possible, or
  - $q \,\|\!\|\, p\xrightarrow{m}\checkmark$, which is not possible, or
  - $p|q\xrightarrow{m}\checkmark$, which means that $p\xrightarrow{n}\checkmark \wedge q\xrightarrow{n'}\checkmark \wedge m = n \oplus n'$ and $p \parallel q\xrightarrow{m}\checkmark$

- $p \parallel q\xrightarrow{m}r$, which means that

  - $p\xrightarrow{m}\checkmark \wedge r = q$, which means that $p \,\|\!\|\, q\xrightarrow{m}q$ and $p \,\|\!\|\, q + q \,\|\!\|\, p + p|q\xrightarrow{m}q$, with rRq, or
  - $q\xrightarrow{m}\checkmark \wedge r = p$, which means that $q \,\|\!\|\, p\xrightarrow{m}p$ and $p \,\|\!\|\, q + q \,\|\!\|\, p + p|q\xrightarrow{m}p$, with rRp, or

- $p\xrightarrow{m}p' \wedge r = p' \parallel q$, which means that $p \parallel\!\!\!\parallel q \xrightarrow{m} p' \parallel q$ and $p \parallel\!\!\!\parallel q + q \parallel\!\!\!\parallel p + p|q \xrightarrow{m} p' \parallel q$, with $rRp' \parallel q$, or

  - $q\xrightarrow{m}q' \wedge r = p \parallel q'$, which means that $q \parallel\!\!\!\parallel p \xrightarrow{m} q' \parallel p$ and $p \parallel\!\!\!\parallel q + q \parallel\!\!\!\parallel p + p|q \xrightarrow{m} q' \parallel p$, with $rRq' \parallel p$

- $p \parallel\!\!\!\parallel q + q \parallel\!\!\!\parallel p + p|q \xrightarrow{m}r$, which means that

  - $p \parallel\!\!\!\parallel q \xrightarrow{m}r$, which means that
    - $p\xrightarrow{m} \checkmark \wedge r = q$, which means that $p \parallel q \xrightarrow{m}q$, with rRq, or
    - $p\xrightarrow{m}p' \wedge r = p' \parallel q$, which means that $p \parallel q \xrightarrow{m}p' \parallel q$, with $rRp' \parallel q$, or
  - $q \parallel\!\!\!\parallel p \xrightarrow{m}r$, which means that
    - $q\xrightarrow{m} \checkmark \wedge r = p$, which means that $p \parallel q \xrightarrow{m}p$, with rRp, or
    - $q\xrightarrow{m}q' \wedge r = q' \parallel p$, which means that $p \parallel q \xrightarrow{m}p \parallel q'$, with $rRp \parallel q'$, or
  - $p|q \xrightarrow{m}r$, which means that
    - $p\xrightarrow{n}p' \wedge q\xrightarrow{n'}q' \wedge m = n \oplus n' \wedge r = p' \parallel q'$, which means that $p \parallel q \xrightarrow{m}p' \parallel q'$, with $rRp' \parallel q'$, or
    - $p\xrightarrow{n} \checkmark \wedge q\xrightarrow{n'}q' \wedge m = n \oplus n' \wedge r = q'$, which means that $p \parallel q \xrightarrow{m}q'$, with rRq', or
    - $p\xrightarrow{n}p' \wedge q\xrightarrow{n'} \checkmark \wedge m = n \oplus n' \wedge r = p'$, which means that $p \parallel q \xrightarrow{m}p'$, with rRp'

- $p \parallel q \xrightarrow{m} \checkmark$, which means that $p\xrightarrow{n} \checkmark \wedge q\xrightarrow{n'} \checkmark \wedge m = n \oplus n'$ and $q \parallel p \xrightarrow{m} \checkmark$

- $q \parallel p \xrightarrow{m} \checkmark$, which is symmetrical to the proof above

- $p \parallel q \xrightarrow{m}r$, which means that

  - $p\xrightarrow{m} \checkmark \wedge r = q$, which means that $q \parallel p \xrightarrow{m}q$, with rRq, or
  - $q\xrightarrow{m} \checkmark \wedge r = p$, which means that $q \parallel p \xrightarrow{m}p$, with rRp, or
  - $p\xrightarrow{m}p' \wedge r = p' \parallel q$, which means that $q \parallel p \xrightarrow{m}q \parallel p'$, with $rRq \parallel p'$, or
  - $q\xrightarrow{m}q' \wedge r = p \parallel q'$, which means that $q \parallel p \xrightarrow{m}q' \parallel p$, with $rRq' \parallel p$, or
  - $p\xrightarrow{n} \checkmark \wedge q\xrightarrow{n'}q' \wedge m = n \oplus n' \wedge r = q'$, which means that $q \parallel p \xrightarrow{m}q'$, with rRq', or
  - $p\xrightarrow{n}p' \wedge q\xrightarrow{n'} \checkmark \wedge m = n \oplus n' \wedge r = p'$, which means that $q \parallel p \xrightarrow{m}p'$, with rRp', or
  - $p\xrightarrow{n}p' \wedge q\xrightarrow{n'}q' \wedge m = n \oplus n' \wedge r = p' \parallel q'$, which means that $q \parallel p \xrightarrow{m}q' \parallel p'$, with $rRq' \parallel p'$, or

- $q \parallel p \xrightarrow{m}r$, which is symmetrical to the proof above

A.12   *CM2 $\boldsymbol{\alpha} \parallel\!\!\!\parallel x \doteq \boldsymbol{\alpha}{\cdot}x$*

- Both $\boldsymbol{\alpha} \parallel\!\!\!\parallel p$ and $\boldsymbol{\alpha}{\cdot}p$ can not terminate

- $\boldsymbol{\alpha} \parallel\!\!\!\parallel p \xrightarrow{m}p'$, which means that $\boldsymbol{\alpha} \xrightarrow{m} \checkmark \wedge p' = p$ and $\boldsymbol{\alpha}{\cdot}p \xrightarrow{m}p$, with $p'Rp$

- $\boldsymbol{\alpha}{\cdot}p \xrightarrow{m}p'$, which means that $\boldsymbol{\alpha} \xrightarrow{m} \checkmark \wedge p' = p$ and $\boldsymbol{\alpha} \parallel\!\!\!\parallel p \xrightarrow{m}p$, with $p'Rp$

A.13 $CM3$ $\boldsymbol{\alpha}{\cdot}x \,\|\!\!\|\, y \doteq \boldsymbol{\alpha}{\cdot}(x \,\|\, y)$

- Both $\boldsymbol{\alpha}{\cdot}p \,\|\!\!\|\, q$ and $\boldsymbol{\alpha}{\cdot}(p \,\|\, q)$ can not terminate

- $\boldsymbol{\alpha}{\cdot}p \,\|\!\!\|\, q \xrightarrow{m} q'$, which means that $\boldsymbol{\alpha}{\cdot}p \xrightarrow{m} p' \wedge q' = p' \,\|\, q$ and $\boldsymbol{\alpha} \xrightarrow{m} \checkmark \wedge p' = p$ and $\boldsymbol{\alpha}{\cdot}(p \,\|\, q) \xrightarrow{m} p \,\|\, q$, with $q' R p \,\|\, q$

- $\boldsymbol{\alpha}{\cdot}(p \,\|\, q) \xrightarrow{m} q'$, which means that $\boldsymbol{\alpha} \xrightarrow{m} \checkmark \wedge q' = p \,\|\, q$ and $\boldsymbol{\alpha}{\cdot}p \xrightarrow{m} p$ and $\boldsymbol{\alpha}{\cdot}p \,\|\!\!\|\, q \xrightarrow{m} p \,\|\, q$, with $q' R p \,\|\, q$

A.14 $CM4$ $(x + y) \,\|\!\!\|\, z \doteq x \,\|\!\!\|\, z + y \,\|\!\!\|\, z$

- $(p + q) \,\|\!\!\|\, r$ can not terminate
- $p \,\|\!\!\|\, r + q \,\|\!\!\|\, r$ can not terminate as
  - $p \,\|\!\!\|\, r$ can not terminate and
  - $q \,\|\!\!\|\, r$ can not terminate
- $(p + q) \,\|\!\!\|\, r \xrightarrow{m} r'$, which means that
  - $p + q \xrightarrow{m} \checkmark \wedge r' = r$, which means that
    - $p \xrightarrow{m} \checkmark$, which means that $p \,\|\!\!\|\, r \xrightarrow{m} r$ and $p \,\|\!\!\|\, r + q \,\|\!\!\|\, r \xrightarrow{m} r$, with $r' R r$, or
    - $q \xrightarrow{m} \checkmark$, which means that $q \,\|\!\!\|\, r \xrightarrow{m} r$ and $p \,\|\!\!\|\, r + q \,\|\!\!\|\, r \xrightarrow{m} r$, with $r' R r$, or
  - $p + q \xrightarrow{m} q' \wedge r' = q' \,\|\, r$, which means that
    - $p \xrightarrow{m} q'$, which means that $p \,\|\!\!\|\, r \xrightarrow{m} q' \,\|\, r$ and $p \,\|\!\!\|\, r + q \,\|\!\!\|\, r \xrightarrow{m} q' \,\|\, r$, with $r' R q' \,\|\, r$, or
    - $q \xrightarrow{m} q'$, which means that $q \,\|\!\!\|\, r \xrightarrow{m} q' \,\|\, r$ and $p \,\|\!\!\|\, r + q \,\|\!\!\|\, r \xrightarrow{m} q' \,\|\, r$, with $r' R q' \,\|\, r$
- $p \,\|\!\!\|\, r + q \,\|\!\!\|\, r \xrightarrow{m} r'$, which means that
  - $p \,\|\!\!\|\, r \xrightarrow{m} r'$, which means that
    - $p \xrightarrow{m} \checkmark \wedge r' = r$, which means that $p + q \xrightarrow{m} \checkmark$ and $(p + q) \,\|\!\!\|\, r \xrightarrow{m} r$, with $r' R r$, or
    - $p \xrightarrow{m} p' \wedge r' = p' \,\|\, r$, which means that $p + q \xrightarrow{m} p'$ and $(p + q) \,\|\!\!\|\, r \xrightarrow{m} p' \,\|\, r$, with $r' R p' \,\|\, r$, or
  - $q \,\|\!\!\|\, r \xrightarrow{m} r'$, which means that
    - $q \xrightarrow{m} \checkmark \wedge r' = r$, which means that $p + q \xrightarrow{m} \checkmark$ and $(p + q) \,\|\!\!\|\, r \xrightarrow{m} r$, with $r' R r$, or
    - $q \xrightarrow{m} q' \wedge r' = q' \,\|\, r$, which means that $p + q \xrightarrow{m} q'$ and $(p+q) \,\|\!\!\|\, r \xrightarrow{m} q' \,\|\, r$, with $r' R q' \,\|\, r$

A.15 $CM5$ $\boldsymbol{\alpha}{\cdot}x | \boldsymbol{\beta} \doteq (\boldsymbol{\alpha}|\boldsymbol{\beta}) \cdot x$

- Both $\boldsymbol{\alpha}{\cdot}p|\boldsymbol{\beta}$ and $(\boldsymbol{\alpha}|\boldsymbol{\beta}) \cdot p$ cannot terminate

- $\boldsymbol{\alpha}{\cdot}p|\boldsymbol{\beta} \xrightarrow{m} p'$, which means that $\boldsymbol{\alpha}{\cdot}p \xrightarrow{n} p' \wedge \boldsymbol{\beta} \xrightarrow{n'} \checkmark \wedge m = n \oplus n'$ and $\boldsymbol{\alpha} \xrightarrow{n} \checkmark \wedge p' = p$ and $\boldsymbol{\alpha}|\boldsymbol{\beta} \xrightarrow{m} \checkmark$ and $(\boldsymbol{\alpha}|\boldsymbol{\beta}) \cdot p \xrightarrow{m} p$, with $p' R p$

- $(\boldsymbol{\alpha}|\boldsymbol{\beta}) \cdot p \xrightarrow{m} p'$, which means that $\boldsymbol{\alpha}|\boldsymbol{\beta} \xrightarrow{m} \checkmark \wedge p' = p$ and $\boldsymbol{\alpha} \xrightarrow{n} \checkmark \wedge \boldsymbol{\beta} \xrightarrow{n'} \checkmark \wedge m = n \oplus n'$ and $\boldsymbol{\alpha}{\cdot}p \xrightarrow{n} p$ and $\boldsymbol{\alpha}{\cdot}p|\boldsymbol{\beta} \xrightarrow{m} p$, with $p' R p$

A.16 $CM6$ $\boldsymbol{\alpha}|\boldsymbol{\beta}{\cdot}x \doteq (\boldsymbol{\alpha}|\boldsymbol{\beta}) \cdot x$

Proof is symmetrical to the proof of axiom $CM5$.

A.17   *CM7 $\boldsymbol{\alpha}\cdot x|\boldsymbol{\beta}\cdot y \doteq (\boldsymbol{\alpha}|\boldsymbol{\beta})\cdot(x\parallel y)$*

- Both $\boldsymbol{\alpha}\cdot p|\boldsymbol{\beta}\cdot q$ and $(\boldsymbol{\alpha}|\boldsymbol{\beta})\cdot(p\parallel q)$ cannot terminate

- $\boldsymbol{\alpha}\cdot p|\boldsymbol{\beta}\cdot q \xrightarrow{m} r$, which means that $\boldsymbol{\alpha}\cdot p \xrightarrow{n} p' \wedge \boldsymbol{\beta}\cdot q \xrightarrow{n'} q' \wedge m = n \oplus n' \wedge r = p' \parallel q'$ and $\boldsymbol{\alpha} \xrightarrow{n} \checkmark \wedge p' = p$ and $\boldsymbol{\beta} \xrightarrow{n'} \checkmark \wedge q' = q$ and $\boldsymbol{\alpha}|\boldsymbol{\beta} \xrightarrow{m} \checkmark$ and $(\boldsymbol{\alpha}|\boldsymbol{\beta})\cdot(p\parallel q) \xrightarrow{m} p \parallel q$, with $rRp \parallel q$

- $(\boldsymbol{\alpha}|\boldsymbol{\beta})\cdot(p\parallel q) \xrightarrow{m} r$, which means that $\boldsymbol{\alpha}|\boldsymbol{\beta} \xrightarrow{m} \checkmark \wedge r = p \parallel q$ and $\boldsymbol{\alpha} \xrightarrow{n} \checkmark \wedge \boldsymbol{\beta} \xrightarrow{n'} \checkmark \wedge m = n \oplus n'$ and $\boldsymbol{\alpha}\cdot p \xrightarrow{n} p$ and $\boldsymbol{\beta}\cdot q \xrightarrow{n'} q$ and $\boldsymbol{\alpha}\cdot p|\boldsymbol{\beta}\cdot q \xrightarrow{m} p \parallel q$, with $rRp \parallel q$

A.18   *CM8 $(x+y)|z \doteq x|z + y|z$*

- $(p+q)|r \xrightarrow{m} \checkmark$, which means $p + q \xrightarrow{n} \checkmark \wedge r \xrightarrow{n'} \checkmark \wedge m = n \oplus n'$ and
    - $p \xrightarrow{n} \checkmark$, which means $p|r \xrightarrow{m} \checkmark$ and $p|r + q|r \xrightarrow{m} \checkmark$, or
    - $q \xrightarrow{n} \checkmark$, which means $q|r \xrightarrow{m} \checkmark$ and $p|r + q|r \xrightarrow{m} \checkmark$

- $p|r + q|r \xrightarrow{m} \checkmark$, which means that
    - $p|r \xrightarrow{m} \checkmark$, which means $p \xrightarrow{n} \checkmark \wedge r \xrightarrow{n'} \checkmark \wedge m = n \oplus n'$ and $p + q \xrightarrow{n} \checkmark$ and $(p+q)|r \xrightarrow{m} \checkmark$, or
    - $q|r \xrightarrow{m} \checkmark$, which means $q \xrightarrow{n} \checkmark \wedge r \xrightarrow{n'} \checkmark \wedge m = n \oplus n'$ and $p + q \xrightarrow{n} \checkmark$ and $(p+q)|r \xrightarrow{m} \checkmark$

- $(p+q)|r \xrightarrow{m} r'$, which means that
    - $p + q \xrightarrow{n} \checkmark \wedge r \xrightarrow{n'} q' \wedge m = n \oplus n' \wedge r' = q'$, which means that
        - $p \xrightarrow{n} \checkmark$, which means that $p|r \xrightarrow{m} q'$ and $p|r + q|r \xrightarrow{m} q'$, which $r'Rq'$, or
        - $q \xrightarrow{n} \checkmark$, which means that $q|r \xrightarrow{m} q'$ and $p|r + q|r \xrightarrow{m} q'$, which $r'Rq'$, or
    - $p + q \xrightarrow{n} q' \wedge r \xrightarrow{n'} \checkmark \wedge m = n \oplus n' \wedge r' = q'$, which means that
        - $p \xrightarrow{n} p' \wedge q' = p'$, which means that $p|r \xrightarrow{m} p'$ and $p|r + q|r \xrightarrow{m} p'$, which $r'Rp'$, or
        - $q \xrightarrow{n} p' \wedge q' = p'$, which means that $q|r \xrightarrow{m} p'$ and $p|r + q|r \xrightarrow{m} p'$, which $r'Rp'$, or
    - $p + q \xrightarrow{n} p' \wedge r \xrightarrow{n'} q' \wedge m = n \oplus n' \wedge r' = p' \parallel q'$, which means that
        - $p \xrightarrow{n} p'' \wedge p' = p''$, which means that $p|r \xrightarrow{m} p'' \parallel r'$ and $p|r + q|r \xrightarrow{m} p' \parallel r'$, which $r'Rp'' \parallel q'$, or
        - $q \xrightarrow{n} p'' \wedge p' = p''$, which means that $q|r \xrightarrow{m} p'' \parallel r'$ and $p|r + q|r \xrightarrow{m} p' \parallel r'$, which $r'Rp'' \parallel q'$

- $p|r + q|r \xrightarrow{m} r'$, which means that
    - $p|r \xrightarrow{m} p' \wedge r' = p'$, which means that
        - $p \xrightarrow{n} \checkmark \wedge r \xrightarrow{n'} q' \wedge m = n \oplus n' \wedge p' = q'$, which means $p + q \xrightarrow{n} \checkmark$ and $(p+q)|r \xrightarrow{m} q'$, with $r'Rq'$, or
        - $p \xrightarrow{n} q' \wedge r \xrightarrow{n'} \checkmark \wedge m = n \oplus n' \wedge p' = q'$, which means $p + q \xrightarrow{n} q'$ and $(p+q)|r \xrightarrow{m} q'$, with $r'Rq'$, or
        - $p \xrightarrow{n} p'' \wedge r \xrightarrow{n'} q' \wedge m = n \oplus n' \wedge p' = p'' \parallel q'$, which means $p + q \xrightarrow{n} p''$ and $(p+q)|r \xrightarrow{m} p'' \parallel q'$, with $r'Rp'' \parallel q'$
    - $q|r \xrightarrow{m} p' \wedge r' = p'$, which is symmetrical to the previous

A.19   *CM*9 $x|(y+z) \doteq x|y + x|z$

Proof is symmetrical to the proof of *CM*8.

A.20   *CM*10 $\langle \overrightarrow{a} \rangle | \langle \overrightarrow{b} \rangle \doteq \langle \overrightarrow{a}, \overrightarrow{b} \rangle$

- $\langle \overrightarrow{a} \rangle | \langle \overrightarrow{b} \rangle \xrightarrow{m} \checkmark \wedge m = [\![\langle \overrightarrow{a} \rangle]\!] \oplus [\![\langle \overrightarrow{b} \rangle]\!] = [\![\langle \overrightarrow{a}, \overrightarrow{b} \rangle]\!]$ and $\langle \overrightarrow{a}, \overrightarrow{b} \rangle \xrightarrow{m} \checkmark$.

- Both $\langle \overrightarrow{a} \rangle | \langle \overrightarrow{b} \rangle$ and $\langle \overrightarrow{a}, \overrightarrow{b} \rangle$ cannot do any transition.

A.21   *CD*1 $\delta | \boldsymbol{\alpha} \doteq \delta$

Both $\delta | \boldsymbol{\alpha}$ and δ cannot terminate or make any transition.

A.22   *CD*2 $\boldsymbol{\alpha} | \delta \doteq \delta$

Both $\boldsymbol{\alpha} | \delta$ and δ cannot terminate or make any transition.

A.23   *VD* $\nabla_V(\delta) \doteq \delta$

Both $\nabla_V(\delta)$ and δ cannot terminate or make any transition.

A.24   *V*1 $\nabla_V(\alpha) \doteq \alpha$ *if* $[\![\alpha]\!] \in [\![V]\!] \cup \{[]\}$

Assuming $[\![\alpha]\!] \in [\![V]\!] \cup \{[]\}$:

- $\nabla_V(\alpha) \xrightarrow{m} \checkmark$, which means that $\alpha \xrightarrow{m} \checkmark$

- $\alpha \xrightarrow{m} \checkmark$, which means that $\nabla_V(\alpha) \xrightarrow{m} \checkmark$

Both $\nabla_V(\alpha)$ and α cannot do any transition.

A.25   *V*2 $\nabla_V(\alpha) \doteq \delta$ *if* $[\![\alpha]\!] \notin [\![V]\!] \cup \{[]\}$

Assuming $[\![\alpha]\!] \notin [\![V]\!] \cup \{[]\}$, both $\nabla_V(\alpha)$ and δ cannot terminate or make any transition.

A.26   *V*3 $\nabla_V(x+y) \doteq \nabla_V(x) + \nabla_V(y)$

- $\nabla_V(p+q) \xrightarrow{m} \checkmark$, which means that $p+q \xrightarrow{m} \checkmark \wedge m \in [\![V]\!] \cup \{[]\}$ and

    - $p \xrightarrow{m} \checkmark$, which means that $\nabla_V(p) \xrightarrow{m} \checkmark$ and $\nabla_V(p) + \nabla_V(q) \xrightarrow{m} \checkmark$, or
    - $q \xrightarrow{m} \checkmark$, which means that $\nabla_V(q) \xrightarrow{m} \checkmark$ and $\nabla_V(p) + \nabla_V(q) \xrightarrow{m} \checkmark$

- $\nabla_V(p) + \nabla_V(q) \xrightarrow{m} \checkmark$, which means that

    - $\nabla_V(p) \xrightarrow{m} \checkmark$, which means that $p \xrightarrow{m} \checkmark \wedge m \in [\![V]\!] \cup \{[]\}$ and $p+q \xrightarrow{m} \checkmark$ and $\nabla_V(p+q) \xrightarrow{m} \checkmark$, or
    - $\nabla_V(q) \xrightarrow{m} \checkmark$, which means that $q \xrightarrow{m} \checkmark \wedge m \in [\![V]\!] \cup \{[]\}$ and $p+q \xrightarrow{m} \checkmark$ and $\nabla_V(p+q) \xrightarrow{m} \checkmark$

- $\nabla_V(p+q) \xrightarrow{m} r$, which means that $p+q \xrightarrow{m} r' \wedge m \in [\![V]\!] \cup \{[]\} \wedge r = z'$ and

    - $p \xrightarrow{m} p' \wedge r' = p'$, which means that $\nabla_V(p) \xrightarrow{m} p'$ and $\nabla_V(p) + \nabla_V(q) \xrightarrow{m} p'$, with rRp', or
    - $q \xrightarrow{m} q' \wedge r' = q'$, which means that $\nabla_V(q) \xrightarrow{m} q'$ and $\nabla_V(p) + \nabla_V(q) \xrightarrow{m} q'$, with rRq'

- $\nabla_V(p) + \nabla_V(q) \overset{m}{\longrightarrow} r$, which means that

  - $\nabla_V(p) \overset{m}{\longrightarrow} p' \wedge r = p'$, which means that $p \overset{m}{\longrightarrow} p'' \wedge m \in \llbracket V \rrbracket \cup \{[]\} \wedge p' = p''$ and $p + q \overset{m}{\longrightarrow} p''$ and $\nabla_V(p + q) \overset{m}{\longrightarrow} p''$, with rRp'', or

  - $\nabla_V(q) \overset{m}{\longrightarrow} q' \wedge r = q'$, which means that $q \overset{m}{\longrightarrow} q'' \wedge m \in \llbracket V \rrbracket \cup \{[]\} \wedge q' = q''$ and $p + q \overset{m}{\longrightarrow} q''$ and $\nabla_V(p + q) \overset{m}{\longrightarrow} q''$, with rRq''

A.27 $V4$ $\nabla_V(x \cdot y) \doteq \nabla_V(x) \cdot \nabla_V(y)$

Both $\nabla_V(p \cdot q)$ and $\nabla_V(p) \cdot \nabla_V(q)$ cannot terminate.

- $\nabla_V(p \cdot q \overset{m}{\longrightarrow} r)$, which means that $p \cdot q \overset{m}{\longrightarrow} r' \wedge m \in \llbracket V \rrbracket \cup \{[]\} \wedge r = \nabla_V(r')$ and

  - $p \overset{m}{\longrightarrow} \checkmark \wedge r' = q$, which means that $\nabla_V(p) \overset{m}{\longrightarrow} \checkmark$ and $\nabla_V(p) \cdot \nabla_V(q) \overset{m}{\longrightarrow} \nabla_V(q)$, with $rR\nabla_V(q)$, or

  - $p \overset{m}{\longrightarrow} p' \wedge r' = p' \cdot q$, which means that $\nabla_V(p) \overset{m}{\longrightarrow} \nabla_V(p')$ and $\nabla_V(p) \cdot \nabla_V(q) \overset{m}{\longrightarrow} \nabla_V(p') \cdot \nabla_V(q)$, with $rR\nabla_V(p' \cdot q)$

- $\nabla_V(p) \cdot \nabla_V(q) \overset{m}{\longrightarrow} r$, which means that

  - $\nabla_V(p) \overset{m}{\longrightarrow} \checkmark \wedge r = \nabla_V(q)$, which means that $p \overset{m}{\longrightarrow} \checkmark \wedge m \in \llbracket V \rrbracket \cup \{[]\}$ and $p \cdot q \overset{m}{\longrightarrow} q$ and $\nabla_V(p \cdot q) \overset{m}{\longrightarrow} \nabla_V(q)$, with $rR\nabla_V(q)$, or

  - $\nabla_V(p) \overset{m}{\longrightarrow} p' \wedge r = p' \cdot \nabla_V(q)$, which means that $p \overset{m}{\longrightarrow} p'' \wedge m \in \llbracket V \rrbracket \cup \{[]\} \wedge p' = \nabla_V(p'')$ and $p \cdot q \overset{m}{\longrightarrow} p'' \cdot q$ and $\nabla_V(p \cdot q) \overset{m}{\longrightarrow} \nabla_V(p'' \cdot q)$, with $rR\nabla_V(p'' \cdot q)$

A.28 DD $\partial_H(\delta) \doteq \delta$

Both $\partial_H(\delta)$ and δ cannot terminate or make any transition.

A.29 $D1$ $\partial_H(\alpha) \doteq \alpha$ *if* $\llbracket \alpha \rrbracket \cap \llbracket H \rrbracket = \emptyset$

Both $\partial_H(\alpha)$ and α can only terminate (with $\overset{\llbracket \alpha \rrbracket}{\longrightarrow}$).

A.30 $D2$ $\partial_H(\alpha) \doteq \delta$ *if* $\llbracket \alpha \rrbracket \cap \llbracket H \rrbracket \neq \emptyset$

Both $\partial_H(\alpha)$ and δ cannot terminate or make any transition.

A.31 $D3$ $\partial_H(x + y) \doteq \partial_H(x) + \partial_H(y)$

Proof is similar to the proof of axiom $V3$.

A.32 $D4$ $\partial_H(x \cdot y) \doteq \partial_H(x) \cdot \partial_H(y)$

Proof is similar to the proof of axiom $V4$.

A.33 TID $\tau_I(\delta) \doteq \delta$

Both $\tau_I(\delta)$ and δ cannot terminate or make any transition.

A.34   *TI1* $\tau_I(\alpha) \doteq \beta$ *with* $[\![\beta]\!] = \theta([\![\alpha]\!], I)$

Both $\tau_I(\alpha)$ and β can only terminate (with $\xrightarrow{\theta([\![\alpha]\!], I)}$).

A.35   *TI3* $\tau_I(x + y) \doteq \tau_I(x) + \tau_I(y)$

Proof is similar to the proof of axiom $V3$.

A.36   *TI4* $\tau_I(x \cdot y) \doteq \tau_I(x) \cdot \tau_I(y)$

Proof is similar to the proof of axiom $V4$.

A.37   *RD* $\rho_R(\delta) \doteq \delta$

Both $\rho_R(\delta)$ and δ cannot terminate or make any transition.

A.38   *R1* $\rho_R(\alpha) \doteq \beta$ *with* $[\![\beta]\!] = [\![R]\!] \bullet [\![\alpha]\!]$

Both $\rho_R(\alpha)$ and β can only terminate (with $\xrightarrow{[\![R]\!] \bullet [\![\alpha]\!]}$).

A.39   *R3* $\rho_R(x + y) \doteq \rho_R(x) + \rho_R(y)$

Proof is similar to the proof of axiom $V3$.

A.40   *R4* $\rho_R(x \cdot y) \doteq \rho_R(x) \cdot \rho_R(y)$

Proof is similar to the proof of axiom $V4$.

A.41   *GD* $\Gamma_C(\delta) \doteq \delta$

Both $\rho_R(\delta)$ and δ cannot terminate or make any transition.

A.42   *G1* $\Gamma_C(\alpha) \doteq \beta$ *with* $[\![\beta]\!] = \gamma([\![\alpha]\!], [\![C]\!])$

Process $\Gamma_C(\alpha)$ can only terminate with $\gamma([\![\alpha]\!], [\![C]\!])$ and β only with $[\![\beta]\!]$, which are equal according to the condition of the axiom.

A.43   *G3* $\Gamma_C(x + y) \doteq \Gamma_C(x) + \Gamma_C(y)$

Proof is similar to the proof of axiom $V3$.

A.44   *G4* $\Gamma_C(x \cdot y) \doteq \Gamma_C(x) \cdot \Gamma_C(y)$

Proof is similar to the proof of axiom $V4$.

B   Soundness proofs of τ axioms

The structure of the following proofs is the same as before, except for the fact that we prove R to be a branching bisimulation and $rooted(t, u)$ (for axiom $t \doteq u$).

B.1 $T1$ $x \cdot \tau \doteq x$

We define E as $\tau R\checkmark$.

- $p \cdot \tau R p$

    - $p \cdot \tau \downarrow$, which is not possible
    - $p \downarrow$, which means that $p = \checkmark$, which is not possible
    - $p \cdot \tau \overset{m}{\longrightarrow} p'$, which means that
        - $p \overset{m}{\longrightarrow} \checkmark \wedge p' = \tau$, with $p'R\checkmark$, or
        - $p \overset{m}{\longrightarrow} p'' \wedge p' = p'' \cdot \tau$, with $p'Rp''$, or
    - $p \overset{m}{\longrightarrow} p'$, which means that $p \cdot \tau \overset{m}{\longrightarrow} p' \cdot \tau$, with $p'Rp' \cdot \tau$

- $rooted(p \cdot \tau, p)$, which is similar to the previous case

- $\tau R\checkmark$

    - $\tau \downarrow$, which is not possible
    - $\checkmark \downarrow$, which means that $\tau \overset{[]}{\longrightarrow} \checkmark \downarrow$
    - $\tau \overset{m}{\longrightarrow} p$, which means that $p = \checkmark \wedge m = []$, with $pR\checkmark$
    - $\checkmark \overset{m}{\longrightarrow} p$, which is not possible

B.2 $T2$ $x \cdot (\tau \cdot (y + z) + y) \doteq x \cdot (y + z)$

We define E as $\tau \cdot (q + r) + q R q + r$ for all $q, r \in T_{pc}$.

- $p \cdot (\tau \cdot (q + r) + q) R p \cdot (q + r)$

    - $p \cdot (\tau \cdot (q + r) + q) \downarrow$, which is not possible
    - $p \cdot (q + r) \downarrow$, which is not possible
    - $p \cdot (\tau \cdot (q + r) + q) \overset{m}{\longrightarrow} r'$, which means that
        - $p \overset{m}{\longrightarrow} \checkmark \wedge r' = \tau \cdot (q + r) + q$, which means that $p \cdot (q + r) \overset{m}{\longrightarrow} q + r$, with $r'Rq + r$, or
        - $p \overset{m}{\longrightarrow} p' \wedge r' = p' \cdot (\tau \cdot (q + r) + q)$, which means that $p \cdot (q + r) \overset{m}{\longrightarrow} p' \cdot (q + r)$, with $r'Rp' \cdot (q + r)$, or
    - $p \cdot (q + r) \overset{m}{\longrightarrow} r'$, which means that
        - $p \overset{m}{\longrightarrow} \checkmark \wedge r' = q + r$, which means that $p \cdot (\tau \cdot (q + r) + q) \overset{m}{\longrightarrow} \tau \cdot (q + r) + q$, with $r'R\tau \cdot (q + r) + q$, or
        - $p \overset{m}{\longrightarrow} p' \wedge r' = p' \cdot (q + r)$, which means that $p \cdot (\tau \cdot (q + r) + q) \overset{m}{\longrightarrow} p' \cdot (\tau \cdot (q + r) + q)$, with $r'Rp' \cdot (\tau \cdot (q + r) + q)$, or

- $rooted(p \cdot (\tau \cdot (q + r) + q), p \cdot (q + r))$, which is similar to the previous case

- $\tau \cdot (q + r) + q R q + r$

    - $\tau \cdot (q + r) + q \downarrow$, which is not possible
    - $q + r \downarrow$, which is not possible
    - $\tau \cdot (q + r) + q \overset{m}{\longrightarrow} p$, which means that
        - $\tau \cdot (q + r) \overset{m}{\longrightarrow} p$, which means that
            - $\tau \overset{m}{\longrightarrow} \checkmark \wedge p = q + r$, which means that $m = []$ and $pRq + r$, or
            - $\tau \overset{m}{\longrightarrow} p'$, which is not possible, or
        - $q \overset{m}{\longrightarrow} p$, which means that $q + r \overset{m}{\longrightarrow} p$, with pRx
    - $q + r \overset{m}{\longrightarrow} p$, which means that $\tau \cdot (q + r) \overset{[]}{\longrightarrow} q + r \overset{m}{\longrightarrow} p$, with pRp

C   Soundness proofs of alphabet axioms

To prove the soundness of these alphabet axioms, we use a different approach as before, except for the axioms containing a parallel operator. As the operators used in these axioms are all defined in a similar way, namely:

$$\frac{x \xrightarrow{m} \checkmark}{O(x) \xrightarrow{f(m)} \checkmark} C(m) \qquad \frac{x \xrightarrow{m} x'}{O(x) \xrightarrow{f(m)} O(x')} C(m)$$

If we now wish to prove an axiom $O_1(\dots O_n(x)\dots) \doteq O_{n+1}(\dots O_{n'}(x)\dots)$, we get the following:

- $O_1(\dots O_n(x)\dots) \xrightarrow{m} \checkmark$, which means that $O_2(\dots O_n(x)\dots) \xrightarrow{m_1} \checkmark \wedge C_1(m_1) \wedge m = f_1(m_1)$ $\wedge \dots \wedge x \xrightarrow{m_n} \checkmark \wedge C_n(m_n) \wedge m_{n-1} = f_n(m_n)$ and therefore $C_{n+1}(m_n) \wedge \dots \wedge C_{n'}(f_{n'-1}(\dots f_{n+1}(m_n)\dots))$ and $m = f'_n(\dots f_{n+1}(m_n)\dots)$ and $O_{n+1}(\dots O_{n'}(x)\dots) \xrightarrow{m} \checkmark$

- Etc.

So what really has to be proven is that for each m, such that $x \xrightarrow{m} \checkmark \vee x \xrightarrow{m} x'$, it holds that $C_n(m) \wedge \dots \wedge C_1(f_2(\dots f_n(m))) \equiv C_{n'}(m) \wedge \dots \wedge C_{n+1}(f_{n+2}(\dots f_{n'}(m)))$ and $C_n(m) \wedge \dots \wedge C_1(f_2(\dots f_n(m))) \Rightarrow f_1(\dots f_n(m)) = f_{n+1}(\dots f_{n'}(m))$.

Lemma C.1. Let $v \in \mathbb{B}(\mathcal{N}_A)$. Also, let $S \subseteq \mathcal{N}_A$. The following holds:

$$v \cap S = \mathcal{N}(\{v\}) \cap S$$

Proof C.1.   Induction on the structure of v. Case $v = []$:

$$[] \cap S$$
$$= \quad \{ \text{ def. } \cap \}$$
$$\emptyset$$
$$= \quad \{ \text{ calculus } \}$$
$$\emptyset \cap S$$
$$= \quad \{ \text{ def. } \mathcal{N} \}$$
$$\mathcal{N}(\{[]\}) \cap S$$

Case $v = [a] \oplus w$:

$$([a] \oplus w) \cap S$$
$$= \quad \{ \text{ def. } \cap \}$$
$$\begin{array}{ll} w \cap S & \text{if } a \notin S \\ \{a\} \cup (w \cap S) & \text{if } a \in S \end{array}$$
$$= \quad \{ \text{ induction hypothesis } \}$$
$$\begin{array}{ll} \mathcal{N}(\{w\}) \cap S & \text{if } a \notin S \\ \{a\} \cup (\mathcal{N}(\{w\}) \cap S) & \text{if } a \in S \end{array}$$
$$= \quad \{ \text{ calculus } \}$$
$$\begin{array}{ll} (\{a\} \cup \mathcal{N}(\{w\})) \cap S & \text{if } a \notin S \\ (\{a\} \cup \mathcal{N}(\{w\})) \cap (\{a\} \cup S) & \text{if } a \in S \end{array}$$
$$= \quad \{ \text{ calculus } \}$$
$$(\{a\} \cup \mathcal{N}(\{w\})) \cap S$$
$$= \quad \{ \text{ def. } \mathcal{N} \}$$
$$\mathcal{N}(\{[a] \oplus w\}) \cap S$$

□

Lemma C.2. Let $N_{\mathbb{B}} = \{n \mid n \in \mathbb{B}(\mathcal{N}_A) \wedge 1 \leq |n|\}$ and $C : N_{\mathbb{B}} \to (\mathcal{N}_A \cup \{\tau\})$. Also, let $\mathcal{N}(dom(C)) \cap rng(C) = \emptyset$ and $m, n \in \mathbb{B}(\boldsymbol{A})$. The following holds:

$$\gamma(m \oplus n, C) = \gamma(m \oplus \gamma(n, C), C)$$

Proof C.2. Induction on the length of n and by the cases of γ. Case $|n| = 0$:

$$\gamma(m \oplus n, C)$$
$$= \quad \{ n = [] \}$$
$$\gamma(m \oplus [], C)$$
$$= \quad \{ \text{def. } \gamma \}$$
$$\gamma(m \oplus \gamma([], C), C)$$
$$= \quad \{ n = [] \}$$
$$\gamma(m \oplus \gamma(n, C), C)$$

Case $|n| > 0$ and $\exists_{n',o}(n = n' \oplus o \wedge \exists_{\langle b,a \rangle \in C}(b = \mu(n') \wedge \exists_{\overrightarrow{d} \in \overrightarrow{D}}(\chi(n', \overrightarrow{d}))))$:

$$\gamma(m \oplus n, C)$$
$$= \quad \{ \text{take } n' \text{ and } o \text{ with } n = n' \oplus o \wedge \exists_{\langle b,a \rangle \in C}(b = \mu(n') \wedge \exists_{\overrightarrow{d} \in \overrightarrow{D}}(\chi(n', \overrightarrow{d}))) \}$$
$$\gamma(m \oplus n' \oplus o, C)$$
$$= \quad \{ \text{take } \langle b, a \rangle \in C \text{ with } b = \mu(n') \wedge \exists_{\overrightarrow{d} \in \overrightarrow{D}}(\chi(n', \overrightarrow{d}))), \text{ def. } \gamma \}$$
$$[a(\overrightarrow{d})] \oplus \gamma(m \oplus o, C)$$
$$= \quad \{ \text{induction hypothesis} \}$$
$$[a(\overrightarrow{d})] \oplus \gamma(m \oplus \gamma(o, C), C)$$
$$= \quad \{ \mathcal{N}(dom(C)) \cap rng(C) = \emptyset, a \in rng(C) \}$$
$$\gamma(m \oplus [a(\overrightarrow{d})] \oplus \gamma(o, C), C)$$
$$= \quad \{ \text{def. } \gamma \text{ and } a \}$$
$$\gamma(m \oplus \gamma(n' \oplus o, C), C)$$
$$= \quad \{ \text{def. } n' \text{ and } o \}$$
$$\gamma(m \oplus \gamma(n, C), C)$$

Case $|n| > 0$ and $\exists_{n',o}(n = n' \oplus o \wedge \exists_{\langle b,\tau \rangle \in C}(b = \mu(n') \wedge \exists_{\overrightarrow{d} \in \overrightarrow{D}}(\chi(n', \overrightarrow{d}))))$, which is similar to the previous case.
Case $|n| > 0$ and $\neg\exists_{n',o}(n = n' \oplus o \wedge \exists_{c \in C}((c = \langle b, a \rangle \vee c = \langle b, \tau \rangle) \wedge b = \mu(n') \wedge \exists_{\overrightarrow{d} \in \overrightarrow{D}}(\chi(n', \overrightarrow{d}))))$:

$$\gamma(m \oplus n, C)$$
$$= \quad \{ \neg\exists_{n',o}(n = n' \oplus o \wedge \exists_{c \in C}((c = \langle b, a \rangle \vee c = \langle b, \tau \rangle) \wedge b = \mu(n') \wedge \exists_{\overrightarrow{d} \in \overrightarrow{D}}(\chi(n', \overrightarrow{d})))), \text{ def. } \gamma \}$$
$$\gamma(m \oplus \gamma(n, C), C)$$

□

C.1   *VA1* $\nabla_V(x) \doteq x$ *if* $\alpha_\nu(x) \subseteq \llbracket V \rrbracket$

$$\mu(m) \in \llbracket V \rrbracket$$

$\equiv$ $\{ \alpha_\nu(x) \subseteq \llbracket V \rrbracket \}$

$$\mu(m) \in \alpha_\nu(x) \vee \mu(m) \in \llbracket V \rrbracket$$

$\equiv$ $\{ \text{def. } \alpha_\nu \}$

$$true \vee \mu(m) \in \llbracket V \rrbracket$$

$\equiv$ $\{ \text{identity } \vee \}$

$$true$$

C.2   *VA2* $\nabla_V(x) \doteq \delta$ *if* $\llbracket V \rrbracket \cap \alpha_\nu(x) = \emptyset$

$$\mu(m) \in \llbracket V \rrbracket$$

$\equiv$ $\{ \text{def. } \alpha_\nu \}$

$$\mu(m) \in \llbracket V \rrbracket \wedge \mu(m) \in \alpha_\nu(x)$$

$\equiv$ $\{ \text{calculus} \}$

$$\mu(m) \in (\llbracket V \rrbracket \cap \alpha_\nu(x))$$

$\equiv$ $\{ \llbracket V \rrbracket \cap \alpha_\nu(x) = \emptyset \}$

$$false$$

C.3   *VA3* $\nabla_V(\nabla_{V'}(x)) \doteq \nabla_{V \cap V'}(x)$

$$\mu(m) \in \llbracket V \rrbracket \wedge \mu(m) \in \llbracket V' \rrbracket$$

$\equiv$ $\{ \text{calculus} \}$

$$\mu(m) \in (\llbracket V \rrbracket \cap \llbracket V' \rrbracket)$$

$\equiv$ $\{ \text{def. } \cap \}$

$$\mu(m) \in \llbracket V \cap V' \rrbracket$$

C.4   *VA4* $\nabla_V(x \parallel y) \doteq \nabla_V(x \parallel \nabla_{V'}(y))$ *if* $\Downarrow(V) \subseteq V'$

We define E as $\nabla_V(p)R\nabla_V(\nabla_{V'}(p))$ (for all $p \in T_{pc}$). The proof for this part is the same as the proof of axiom *VA3*, as $V \subseteq \Downarrow(V)$ and thus $V \cap V' = V$.

- $\nabla_V(p \parallel q) \xrightarrow{m} \checkmark$, which means that $p \parallel q \xrightarrow{m} \checkmark \wedge \mu(m) \in \llbracket V \rrbracket \cup \{[]\}$ and $p \xrightarrow{n} \checkmark \wedge q \xrightarrow{o} \checkmark \wedge m = n \oplus o$ and $o \in \Downarrow (V)$ and $\nabla_{V'}(q) \xrightarrow{o} \checkmark$ and therefore $p \parallel \nabla_{V'}(q) \xrightarrow{m} \checkmark$ and $\nabla_V(p \parallel \nabla_{V'}(q)) \xrightarrow{m} \checkmark$

- $\nabla_V(p \parallel \nabla_{V'}(q)) \xrightarrow{m} \checkmark$, which means that $p \parallel \nabla_{V'}(q) \xrightarrow{m} \checkmark \wedge \mu(m) \in \llbracket V \rrbracket \cup \{[]\}$ and $p \xrightarrow{n} \checkmark \wedge \nabla_{V'}(q) \xrightarrow{o} \checkmark \wedge m = n \oplus o$ and $q \xrightarrow{o} \checkmark \wedge \mu(o) \in \llbracket V' \rrbracket \cup \{[]\}$ and therefore $p \parallel q \xrightarrow{m} \checkmark$ and $\nabla_V(p \parallel q) \xrightarrow{m} \checkmark$

- $\nabla_V(p \parallel q) \xrightarrow{m} p'$, which means that $p \parallel q \xrightarrow{m} p'' \wedge p' = \nabla_V(p'') \wedge \mu(m) \in \llbracket V \rrbracket \cup \{[]\}$ and

    - $p \xrightarrow{m} \checkmark \wedge p'' = q$, which means that $p \parallel \nabla_{V'}(q) \xrightarrow{m} \nabla_{V'}(q)$ and $\nabla_V(p \parallel \nabla_{V'}(q)) \xrightarrow{m} \nabla_V(\nabla_{V'}(q))$, with $p'R\nabla_V(\nabla_{V'}(q))$, or

    - $q \xrightarrow{m} \checkmark \wedge p'' = p$, which means that $\nabla_{V'}(q) \xrightarrow{m} \checkmark$ and $p \parallel \nabla_{V'}(q) \xrightarrow{m} p$ and $\nabla_V(p \parallel \nabla_{V'}(q)) \xrightarrow{m} \nabla_V(p)$, with $p'R\nabla_V(p)$, or

- $p \xrightarrow{m} p''' \wedge p'' = p''' \parallel q$, which means that $p \parallel \nabla_{V'}(q) \xrightarrow{m} p''' \parallel \nabla_{V'}(q)$ and $\nabla_V(p \parallel \nabla_{V'}(q)) \xrightarrow{m} \nabla_V(p''' \parallel \nabla_{V'}(q))$, with $p' R \nabla_V(p''' \parallel \nabla_{V'}(q))$, or

- $q \xrightarrow{m} q' \wedge p'' = p \parallel q'$, which means that $\nabla_{V'}(q) \xrightarrow{m} \nabla_{V'}(q')$ and $p \parallel \nabla_{V'}(q) \xrightarrow{m} p \parallel \nabla_{V'}(q')$ and $\nabla_V(p \parallel \nabla_{V'}(q)) \xrightarrow{m} \nabla_V(p \parallel \nabla_{V'}(q'))$, with $p' R \nabla_V(p \parallel \nabla_{V'}(q'))$, or

- $p \xrightarrow{n} p''' \wedge q \xrightarrow{o} q' \wedge m = n \oplus o \wedge p'' = p''' \parallel q'$, which means that $\nabla_{V'}(q) \xrightarrow{o} \nabla_{V'}(q')$ and $p \parallel \nabla_{V'}(q) \xrightarrow{m} p''' \parallel \nabla_{V'}(q')$ and $\nabla_V(p \parallel \nabla_{V'}(q)) \xrightarrow{m} \nabla_V(p''' \parallel \nabla_{V'}(q'))$, with $p' R \nabla_V(p''' \parallel \nabla_{V'}(q'))$, or

- $p \xrightarrow{n} \checkmark \wedge q \xrightarrow{o} q' \wedge m = n \oplus o \wedge p'' = q'$, which means that $\nabla_{V'}(q) \xrightarrow{o} \nabla_{V'}(q')$ and $p \parallel \nabla_{V'}(q) \xrightarrow{m} \nabla_{V'}(q')$ and $\nabla_V(p \parallel \nabla_{V'}(q)) \xrightarrow{m} \nabla_V(\nabla_{V'}(q'))$, with $p' R \nabla_V(\nabla_{V'}(q'))$, or

- $p \xrightarrow{n} p''' \wedge q \xrightarrow{o} \checkmark \wedge m = n \oplus o \wedge p'' = p'''$, which means that $\nabla_{V'}(q) \xrightarrow{o} \checkmark$ and $p \parallel \nabla_{V'}(q) \xrightarrow{m} p'''$ and $\nabla_V(p \parallel \nabla_{V'}(q)) \xrightarrow{m} \nabla_V(p''')$, with $p' R \nabla_V(p''')$

- $\nabla_V(p \parallel \nabla_{V'}(q)) \xrightarrow{m} p'$, which means that $p \parallel \nabla_{V'}(q) \xrightarrow{m} p'' \wedge p' = \nabla_V(p'') \wedge \mu(m) \in \llbracket V \rrbracket \cup \{[]\}$ and

  - $p \xrightarrow{m} \checkmark \wedge p'' = \nabla_{V'}(q)$, which means that $p \parallel q \xrightarrow{m} q$ and $\nabla_V(p \parallel q) \xrightarrow{m} \nabla_V(q)$, with $p' R \nabla_V(q)$, or

  - $\nabla_{V'}(q) \xrightarrow{m} \checkmark \wedge p'' = p$, which means that $q \xrightarrow{m} \checkmark \wedge \mu(m) \in \llbracket V' \rrbracket \cup \{[]\}$ and $p \parallel q \xrightarrow{m} p$ and $\nabla_V(p \parallel q) \xrightarrow{m} \nabla_V(p)$, with $p' R \nabla_V(p)$, or

  - $p \xrightarrow{m} p''' \wedge p'' = p''' \parallel \nabla_{V'}(q)$, which means that $p \parallel q \xrightarrow{m} p''' \parallel q$ and $\nabla_V(p \parallel q) \xrightarrow{m} \nabla_V(p''' \parallel q)$, with $p' R \nabla_V(p''' \parallel q)$, or

  - $\nabla_{V'}(q) \xrightarrow{m} q' \wedge p'' = p \parallel q'$, which means that $q \xrightarrow{m} q'' \wedge q' = \nabla_{V'}(q'') \wedge \mu(m) \in \llbracket V' \rrbracket \cup \{[]\}$ and $p \parallel q \xrightarrow{m} p \parallel q''$ and $\nabla_V(p \parallel q) \xrightarrow{m} \nabla_V(p \parallel q'')$, with $p' R \nabla_V(p \parallel q'')$, or

  - $p \xrightarrow{n} p''' \wedge \nabla_{V'}(q) \xrightarrow{o} q' \wedge m = n \oplus o \wedge p'' = p''' \parallel q'$, which means that $q \xrightarrow{o} q'' \wedge q' = \nabla_{V'}(q'') \wedge \mu(o) \in \llbracket V' \rrbracket \cup \{[]\}$ and $p \parallel q \xrightarrow{m} p''' \parallel q''$ and $\nabla_V(p \parallel q) \xrightarrow{m} \nabla_V(p''' \parallel q'')$, with $p' R \nabla_V(p''' \parallel q'')$, or

  - $p \xrightarrow{n} \checkmark \wedge \nabla_{V'}(q) \xrightarrow{o} q' \wedge m = n \oplus o \wedge p'' = q'$, which means that $q \xrightarrow{o} q'' \wedge q' = \nabla_{V'}(q'') \wedge \mu(o) \in \llbracket V' \rrbracket \cup \{[]\}$ and $p \parallel q \xrightarrow{m} q''$ and $\nabla_V(p \parallel q) \xrightarrow{m} \nabla_V(q'')$, with $p' R \nabla_V(q'')$, or

  - $p \xrightarrow{n} p''' \wedge \nabla_{V'}(q) \xrightarrow{o} \checkmark \wedge m = n \oplus o \wedge p'' = p'''$, which means that $q \xrightarrow{o} \checkmark \wedge \mu(o) \in \llbracket V' \rrbracket \cup \{[]\}$ and $p \parallel q \xrightarrow{m} p'''$ and $\nabla_V(p \parallel q) \xrightarrow{m} \nabla_V(p''')$, with $p' R \nabla_V(p''')$

C.5 CA1 $\Gamma_C(x) \doteq x$ if $dom(\llbracket C \rrbracket) \cap \Downarrow(\alpha_\nu(x)) = \emptyset$

$\quad \gamma(m, \llbracket C \rrbracket) = m$

$\equiv \quad \{ \text{def. } \gamma \}$

$\quad \neg \exists_{n,o}(m = n \oplus o \wedge \exists_{c \in C}((c = \langle b, a \rangle \vee c = \langle b, \tau \rangle) \wedge b = \mu(n) \wedge \exists_{\vec{d} \in \vec{D}}(\chi(n, \vec{d}))))$

$\equiv \quad \{ \mu(n) \in \Downarrow(\alpha_\nu(x)) \}$

$\quad \neg \exists_{n,o}(m = n \oplus o \wedge false)$

$\equiv \quad \{ \text{calculus} \}$

$\quad \neg false$

$\equiv \quad \{ \text{calculus} \}$

$\quad true$

C.6   *CA2* $\Gamma_C(\Gamma_{C'}(x)) \doteq \Gamma_{C \cup C'}(x)$ *if* $\mathcal{N}(dom(\llbracket C \rrbracket)) \cap \mathcal{N}(dom(\llbracket C' \rrbracket)) = \emptyset \wedge$
$\mathcal{N}(dom(\llbracket C \rrbracket)) \cap rng(\llbracket C' \rrbracket) = \emptyset$

Induction on the length of m and by the cases of γ (with C'). Case $|m| = 0$:

$$\gamma(\gamma(m, \llbracket C' \rrbracket), \llbracket C \rrbracket) = \gamma(m, \llbracket C \cup C' \rrbracket)$$
$$\equiv \quad \{ |m| = 0 \}$$
$$\gamma(\gamma([], \llbracket C' \rrbracket), \llbracket C \rrbracket) = \gamma([], \llbracket C \cup C' \rrbracket)$$
$$\equiv \quad \{ \text{def. } \gamma \}$$
$$[] = []$$
$$\equiv \quad \{ \text{refl. } = \}$$
$$true$$

Case $|m| > 0$ and $\exists_{n,o}(m = n \oplus o \wedge \exists_{\langle b,a \rangle \in C'}(b = \mu(n) \wedge \exists_{\overrightarrow{d} \in \overrightarrow{D}}(\chi(n, \overrightarrow{d}))))$:

$$\gamma(\gamma(m, \llbracket C' \rrbracket), \llbracket C \rrbracket) = \gamma(m, \llbracket C \cup C' \rrbracket)$$
$$\equiv \quad \{ \text{take } n \text{ and } o \text{ with } m = n \oplus o \wedge \exists_{\langle b,a \rangle \in C'}(b = \mu(n) \wedge \exists_{\overrightarrow{d} \in \overrightarrow{D}}(\chi(n, \overrightarrow{d}))) \}$$
$$\gamma(\gamma(n \oplus o, \llbracket C' \rrbracket), \llbracket C \rrbracket) = \gamma(n \oplus o, \llbracket C \cup C' \rrbracket)$$
$$\equiv \quad \{ \text{def. } \gamma, \text{def. } n \text{ and } o \}$$
$$\gamma([a(\overrightarrow{d})] \oplus \gamma(o, \llbracket C' \rrbracket), \llbracket C \rrbracket) = [a(\overrightarrow{d})] \oplus \gamma(o, \llbracket C \cup C' \rrbracket)$$
$$\equiv \quad \{ a \in rng(C), \mathcal{N}(dom(\llbracket C \rrbracket)) \cap rng(\llbracket C' \rrbracket) = \emptyset \}$$
$$[a(\overrightarrow{d})] \oplus \gamma(\gamma(o, \llbracket C' \rrbracket), \llbracket C \rrbracket) = [a(\overrightarrow{d})] \oplus \gamma(o, \llbracket C \cup C' \rrbracket)$$
$$\equiv \quad \{ \text{induction hypothesis} \}$$
$$[a(\overrightarrow{d})] \oplus \gamma(o, \llbracket C \cup C' \rrbracket) = [a(\overrightarrow{d})] \oplus \gamma(o, \llbracket C \cup C' \rrbracket)$$
$$\equiv \quad \{ \text{refl. } = \}$$
$$true$$

Case $|m| > 0$ and $\exists_{n,o}(m = n \oplus o \wedge \exists_{\langle b,\tau \rangle \in C'}(b = \mu(n) \wedge \exists_{\overrightarrow{d} \in \overrightarrow{D}}(\chi(n, \overrightarrow{d}))))$, which is similar to the previous case.
Case $|m| > 0$ and $\neg \exists_{n,o}(m = n \oplus o \wedge \exists_{c \in C'}((c = \langle b,a \rangle \vee c = \langle b,\tau \rangle) \wedge b = \mu(n) \wedge \exists_{\overrightarrow{d} \in \overrightarrow{D}}(\chi(n, \overrightarrow{d}))))$:

$$\gamma(\gamma(m, \llbracket C' \rrbracket), \llbracket C \rrbracket) = \gamma(m, \llbracket C \cup C' \rrbracket)$$
$$\equiv \quad \{ \neg \exists_{n,o}(m = n \oplus o \wedge \exists_{c \in C'}((c = \langle b,a \rangle \vee c = \langle b,\tau \rangle) \wedge b = \mu(n) \wedge \exists_{\overrightarrow{d} \in \overrightarrow{D}}(\chi(n, \overrightarrow{d}))), \text{def. } \gamma \}$$
$$\gamma(m, \llbracket C \rrbracket) = \gamma(m, \llbracket C \cup C' \rrbracket)$$

Again induction on m (with C). Case $\exists_{n,o}(m = n \oplus o \wedge \exists_{\langle b,a \rangle \in C}(b = \mu(n) \wedge \exists_{\overrightarrow{d} \in \overrightarrow{D}}(\chi(n, \overrightarrow{d}))))$:

$$\gamma(m, \llbracket C \rrbracket) = \gamma(m, \llbracket C \cup C' \rrbracket)$$
$$\equiv \quad \{ \text{take } n \text{ and } o \text{ with } m = n \oplus o \wedge \exists_{\langle b,a \rangle \in C}(b = \mu(n) \wedge \exists_{\overrightarrow{d} \in \overrightarrow{D}}(\chi(n, \overrightarrow{d}))) \}$$
$$\gamma(n \oplus o, \llbracket C \rrbracket) = \gamma(n \oplus o, \llbracket C \cup C' \rrbracket)$$
$$\equiv \quad \{ \text{def. } \gamma, \text{def. } n \text{ and } o, c \in \llbracket C \rrbracket \Rightarrow c \in \llbracket C \cup C' \rrbracket \}$$
$$[a(\overrightarrow{d})] \oplus \gamma(o, \llbracket C \rrbracket) = [a(\overrightarrow{d})] \oplus \gamma(o, \llbracket C \cup C' \rrbracket)$$
$$\equiv \quad \{ \text{induction hypothesis} \}$$
$$[a(\overrightarrow{d})] \oplus \gamma(o, \llbracket C \cup C' \rrbracket) = [a(\overrightarrow{d})] \oplus \gamma(o, \llbracket C \cup C' \rrbracket)$$
$$\equiv \quad \{ \text{refl. } = \}$$
$$true$$

Case $\exists_{n,o}(m = n \oplus o \wedge \exists_{\langle b,\tau \rangle \in C}(b = \mu(n) \wedge \exists_{\vec{d} \in \vec{D}}(\chi(n, \vec{d}))))$, which is similar to the previous case.

$$\gamma(m, \llbracket C \rrbracket) = \gamma(m, \llbracket C \cup C' \rrbracket)$$
$$\equiv \quad \{ \neg\exists_{n,o}(m = n \oplus o \wedge \exists_{c \in C}((c = \langle b,a \rangle \vee c = \langle b, \tau \rangle) \wedge b = \mu(n) \wedge \exists_{\vec{d} \in \vec{D}}(\chi(n, \vec{d})))),$$
$$\neg\exists_{n,o}(m = n \oplus o \wedge \exists_{c \in C'}((c = \langle b,a \rangle \vee c = \langle b, \tau \rangle) \wedge b = \mu(n) \wedge \exists_{\vec{d} \in \vec{D}}(\chi(n, \vec{d})))), \text{ def. } \gamma \}$$
$$m = m$$
$$\equiv \quad \{ \text{ refl. } = \}$$
$$true$$

C.7   *CA3* $\Gamma_C(x \parallel y) \doteq x \parallel \Gamma_C(y)$ *if* $\Downarrow(dom(\llbracket C \rrbracket)) \cap \Downarrow(\alpha_\nu(x)) = \emptyset$

- $\Gamma_C(p \parallel q) \xrightarrow{m} \checkmark$, which means that $p \parallel q \xrightarrow{n} \checkmark \wedge m = \gamma(n, \llbracket C \rrbracket)$ and $p \xrightarrow{o} \checkmark \wedge q \xrightarrow{o'} \checkmark \wedge n = o \oplus o'$ and therefore $\Gamma_C(q) \xrightarrow{\gamma(o', \llbracket C \rrbracket)} \checkmark$ and $p \parallel \Gamma_C(q) \xrightarrow{o \oplus \gamma(o', \llbracket C \rrbracket)} \checkmark$, with $m = \gamma(n, \llbracket C \rrbracket) = \gamma(o \oplus o', \llbracket C \rrbracket) = o \oplus \gamma(o', \llbracket C \rrbracket)$ as $\Downarrow(\{o\}) \subseteq \Downarrow(\alpha_\nu(p))$ and thus $\Downarrow(dom(\llbracket C \rrbracket)) \cap \Downarrow(\{o\} = \emptyset$

- $p \parallel \Gamma_C(q) \xrightarrow{m} \checkmark$, which means that $p \xrightarrow{n} \checkmark \wedge \Gamma_C(q) \xrightarrow{o} \checkmark \wedge m = n \oplus o$ and $q \xrightarrow{o'} \checkmark \wedge o' = \gamma(o, \llbracket C \rrbracket)$ and therefore $p \parallel q \xrightarrow{n \oplus o'} \checkmark$ and $\Gamma_C(p \parallel q) \xrightarrow{\gamma(n \oplus o', \llbracket C \rrbracket)} \checkmark$, with $m = n \oplus o = n \oplus \gamma(o', \llbracket C \rrbracket) = \gamma(n \oplus o', \llbracket C \rrbracket)$ as $\Downarrow(\{n\}) \subseteq \Downarrow(\alpha_\nu(p))$

- $\Gamma_C(p \parallel q) \xrightarrow{m} p'$, which means that $p \parallel q \xrightarrow{n} p'' \wedge m = \gamma(n, \llbracket C \rrbracket) \wedge p' = \Gamma_C(p'')$ and

  - $p \xrightarrow{n} \checkmark \wedge p'' = q$ and therefore $p \parallel \Gamma_C(q) \xrightarrow{n} \Gamma_C(q)$, with $p' R \Gamma_C(q)$ and $m = \gamma(n, \llbracket C \rrbracket) = n$ as $\Downarrow(\{n\}) \subseteq \Downarrow(\alpha_\nu(p))$, or

  - $q \xrightarrow{n} \checkmark \wedge p'' = p$ and therefore $\Gamma_C(q) \xrightarrow{\gamma(n, \llbracket C \rrbracket)} \checkmark$ and $p \parallel \Gamma_C(q) \xrightarrow{\gamma(n, \llbracket C \rrbracket)} p$, with $p' R p$, or

  - $p \xrightarrow{n} p''' \wedge p'' = p''' \parallel q$ and therefore $p \parallel \Gamma_C(q) \xrightarrow{n} p''' \parallel \Gamma_C(q)$, with $p' R p''' \parallel \Gamma_C(q)$ and $m = \gamma(n, \llbracket C \rrbracket) = n$ as $\Downarrow(\{n\}) \subseteq \Downarrow(\alpha_\nu(p))$, or

  - $q \xrightarrow{n} q' \wedge p'' = p \parallel q'$ and therefore $\Gamma_C(q) \xrightarrow{\gamma(n, \llbracket C \rrbracket)} \Gamma_C(q')$ and $p \parallel \Gamma_C(q) \xrightarrow{\gamma(n, \llbracket C \rrbracket)} p \parallel \Gamma_C(q')$, with $p' R p \parallel \Gamma_C(q')$, or

  - $p \xrightarrow{o} p''' \wedge q \xrightarrow{o'} q' \wedge n = o \oplus o' \wedge p'' = p''' \parallel q'$ and therefore $\Gamma_C(q) \xrightarrow{\gamma(o', \llbracket C \rrbracket)} \Gamma_C(q')$ and $p \parallel \Gamma_C(q) \xrightarrow{o \oplus \gamma(o', \llbracket C \rrbracket)} p''' \parallel \Gamma_C(q')$, with $p' R p''' \parallel \Gamma_C(q')$ and $m = \gamma(n, repC) = \gamma(o \oplus o', \llbracket C \rrbracket) = o \oplus \gamma(o', \llbracket C \rrbracket)$ as $\Downarrow(\{o\}) \subseteq \Downarrow(\alpha_\nu(p))$, or

  - $p \xrightarrow{o} \checkmark \wedge q \xrightarrow{o'} q' \wedge n = o \oplus o' \wedge p'' = q'$ and therefore $\Gamma_C(q) \xrightarrow{\gamma(o', \llbracket C \rrbracket)} \Gamma_C(q')$ and $p \parallel \Gamma_C(q) \xrightarrow{o \oplus \gamma(o', \llbracket C \rrbracket)} \Gamma_C(q')$, with $p' R \Gamma_C(q')$ and $m = \gamma(n, repC) = \gamma(o \oplus o', \llbracket C \rrbracket) = o \oplus \gamma(o', \llbracket C \rrbracket)$ as $\Downarrow(\{o\}) \subseteq \Downarrow(\alpha_\nu(p))$, or

  - $p \xrightarrow{o} p''' \wedge q \xrightarrow{o'} \checkmark \wedge n = o \oplus o' \wedge p'' = p'''$ and therefore $\Gamma_C(q) \xrightarrow{\gamma(o', \llbracket C \rrbracket)} \checkmark$ and $p \parallel \Gamma_C(q) \xrightarrow{o \oplus \gamma(o', \llbracket C \rrbracket)} p'''$, with $p' R p'''$ and $m = \gamma(n, \llbracket C \rrbracket) = \gamma(o \oplus o', \llbracket C \rrbracket) = o \oplus \gamma(o', \llbracket C \rrbracket)$ as $\Downarrow(\{o\}) \subseteq \Downarrow(\alpha_\nu(p))$

- $p \parallel \Gamma_C(q) \xrightarrow{m} p'$, which is similar to the previous case

C.8   *CA4* $\Gamma_C(x \parallel y) \doteq \Gamma_C(x \parallel \Gamma_C(y))$ *if* $\mathcal{N}(dom(\llbracket C \rrbracket)) \cap rng(\llbracket C \rrbracket) = \emptyset$

With (implicit use of) Lemma C.2, the soundness proof of the axiom becomes similar to the proof of axiom *VA4*.

C.9   *DA1* $\partial_H(x) \doteq x$ *if* $[\![H]\!] \cap \mathcal{N}(\alpha_\nu(x)) = \emptyset$

$\qquad \mu(m) \cap [\![H]\!] = \emptyset$
$\equiv \qquad \{ \text{ Lemma C.1 } \}$
$\qquad \mathcal{N}(\{\mu(m)\}) \cap [\![H]\!] = \emptyset$
$\equiv \qquad \{ \ \{\mu(m)\} \subseteq \alpha_\nu(x),\ [\![H]\!] \cap \mathcal{N}(\alpha_\nu(x)) = \emptyset \ \}$
$\qquad true$

C.10   *DA2* $\partial_H(x) \doteq \delta$ *if* $\forall_{v \in \alpha_\nu(x)}(v \cap [\![H]\!] \neq \emptyset)$

$\qquad \mu(m) \cap [\![H]\!] = \emptyset$
$\equiv \qquad \{ \text{ def. } \alpha_\nu,\ \forall_{v \in \alpha_\nu(x)}(v \cap [\![H]\!] \neq \emptyset) \ \}$
$\qquad false$

C.11   *DA3* $\partial_H(\partial_{H'}(x)) \doteq \partial_{H \cup H'}(x)$

$\qquad \mu(m) \cap [\![H]\!] = \emptyset \wedge \mu(m) \cap [\![H']\!] = \emptyset$
$\equiv \qquad \{ \text{ set calculus } \}$
$\qquad (\mu(m) \cap [\![H]\!]) \cup (\mu(m) \cap [\![H']\!]) = \emptyset$
$\equiv \qquad \{ \text{ set calculus } \}$
$\qquad \mu(m) \cap ([\![H]\!] \cup [\![H']\!]) = \emptyset$
$\equiv \qquad \{ \text{ def. } \cup \}$
$\qquad \mu(m) \cap ([\![H \cup H']\!]) = \emptyset$

C.12   *DA4* $\partial_H(x \parallel y) \doteq \partial_H(x) \parallel \partial_H(y)$

- $\partial_H(p \parallel q) \xrightarrow{m} \checkmark$, which means that $p \parallel q \xrightarrow{m} \checkmark \wedge \mu(m) \cap [\![H]\!] = \emptyset$ and $p \xrightarrow{n} \checkmark \wedge q \xrightarrow{o} \checkmark \wedge m = n \oplus o$ and therefore $\partial_H(p) \xrightarrow{n} \checkmark \wedge \partial_H(q) \xrightarrow{o} \checkmark$ and $\partial_H(p) \parallel \partial_H(q) \xrightarrow{m} \checkmark$

- $\partial_H(p) \parallel \partial_H(q) \xrightarrow{m} \checkmark$, which means that $\partial_H(p) \xrightarrow{n} \checkmark \wedge \partial_H(q) \xrightarrow{o} \checkmark \wedge m = n \oplus o$ and $p \xrightarrow{n} \checkmark \wedge q \xrightarrow{o} \checkmark \wedge \mu(n) \cap [\![H]\!] = \emptyset \wedge \mu(o) \cap [\![H]\!] = \emptyset$ and therefore $p \parallel q \xrightarrow{m} \checkmark$ and $\partial_H(p \parallel q) \xrightarrow{m} \checkmark$

- $\partial_H(p \parallel q) \xrightarrow{m} p'$, which means that $p \parallel q \xrightarrow{m} p'' \wedge p' = \partial_H(p'') \wedge \mu(m) \cap [\![H]\!] = \emptyset$ and

  - $p \xrightarrow{m} \checkmark \wedge p'' = q$ and therefore $\partial_H(p) \xrightarrow{m} \checkmark$ and $\partial_H(p) \parallel \partial_H(q) \xrightarrow{m} \partial_H(q)$, with $p' R \partial_H(q)$, or

  - $q \xrightarrow{m} \checkmark \wedge p'' = p$ and therefore $\partial_H(q) \xrightarrow{m} \checkmark$ and $\partial_H(p) \parallel \partial_H(q) \xrightarrow{m} \partial_H(p)$, with $p' R \partial_H(p)$, or

  - $p \xrightarrow{m} p''' \wedge p'' = p''' \parallel q$ and therefore $\partial_H(p) \xrightarrow{m} \partial_H(p''')$ and $\partial_H(p) \parallel \partial_H(q) \xrightarrow{m} \partial_H(p''') \parallel \partial_H(q)$, with $p' R \partial_H(p''') \parallel \partial_H(q)$, or

  - $q \xrightarrow{m} q' \wedge p'' = p \parallel q'$ and therefore $\partial_H(q) \xrightarrow{m} \partial_H(q')$ and $\partial_H(p) \parallel \partial_H(q) \xrightarrow{m} \partial_H(p) \parallel \partial_H(q')$, with $p' R \partial_H(p) \parallel \partial_H(q')$, or

  - $p \xrightarrow{n} p''' \wedge q \xrightarrow{o} q' \wedge m = n \oplus o \wedge p'' = p''' \parallel q'$ and therefore $\partial_H(p) \xrightarrow{n} \partial_H(p''') \wedge \partial_H(q) \xrightarrow{o} \partial_H(q')$ and $\partial_H(p) \parallel \partial_H(q) \xrightarrow{m} \partial_H(p''') \parallel \partial_H(q')$, with $p' R \partial_H(p''') \parallel \partial_H(q')$, or

  - $p \xrightarrow{n} \checkmark \wedge q \xrightarrow{o} q' \wedge m = n \oplus o \wedge p'' = q'$ and therefore $\partial_H(p) \xrightarrow{n} \checkmark \wedge \partial_H(q) \xrightarrow{o} \partial_H(q')$ and $\partial_H(p) \parallel \partial_H(q) \xrightarrow{m} \partial_H(q')$, with $p' R \partial_H(q')$, or

- $p \xrightarrow{n} p''' \wedge q \xrightarrow{o} \checkmark \wedge m = n \oplus o \wedge p'' = p'''$ and therefore $\partial_H(p) \xrightarrow{n} \partial_H(p''') \wedge \partial_H(q) \xrightarrow{o} \checkmark$ and $\partial_H(p) \parallel \partial_H(q) \xrightarrow{m} \partial_H(p'')$, with $p' R \partial_H(p''')$

- $\partial_H(p) \parallel \partial_H(q) \xrightarrow{m} p'$, which is similar to the previous case

C.13   *TA1* $\tau_I(x) \doteq x$ *if* $[\![I]\!] \cap \mathcal{N}(\alpha_\nu(x)) = \emptyset$

We prove $\theta(m, [\![I]\!]) = m$ with induction on the structure of m. Case $m = []$:

$$\theta([], [\![I]\!])$$
$$= \quad \{ \text{ def. } \theta \}$$
$$[]$$

Case $m = [a(\vec{d})] \oplus n$:

$$\theta([a(\vec{d})] \oplus n, [\![I]\!])$$
$$= \quad \{ \text{ def. } \theta, a \notin [\![I]\!] \}$$
$$[a(\vec{d})] \oplus \theta(n, [\![I]\!])$$
$$= \quad \{ \text{ induction hypothesis } \}$$
$$[a(\vec{d})] \oplus n$$

C.14   *TA3* $\tau_I(\tau_{I'}(x)) \doteq \tau_{I \cup I'}(x)$

We prove $\theta(\theta(m, [\![I']\!]), [\![I]\!]) = \theta(m, [\![I \cup I']\!])$ with induction on the structure of m. Case $m = []$:

$$\theta(\theta([], [\![I']\!]), [\![I]\!])$$
$$= \quad \{ \text{ def. } \theta \}$$
$$[]$$
$$= \quad \{ \text{ def. } \theta \}$$
$$\theta([], [\![I \cup I']\!])$$

Case $m = [a(\vec{d})] \oplus n$:

$$\theta(\theta([a(\vec{d})] \oplus n, [\![I']\!]), [\![I]\!])$$

$= \quad \{ \text{ def. } \theta \}$

$\theta(\theta(n, [\![I']\!]), [\![I]\!])$	*if* $a \in [\![I']\!]$
$\theta([a(\vec{d})] \oplus \theta(n, [\![I']\!]), [\![I]\!])$	*if* $a \notin [\![I']\!]$

$= \quad \{ \text{ def. } \theta \}$

$\theta(\theta(n, [\![I']\!]), [\![I]\!])$	*if* $a \in [\![I']\!]$
$\theta(\theta(n, [\![I']\!]), [\![I]\!])$	*if* $a \notin [\![I']\!] \wedge a \in [\![I]\!]$
$[a(\vec{d})] \oplus \theta(\theta(n, [\![I']\!]), [\![I]\!])$	*if* $a \notin [\![I']\!] \wedge a \notin [\![I]\!]$

$= \quad \{ \text{ calculus } \}$

$\theta(\theta(n, [\![I']\!]), [\![I]\!])$	*if* $a \in [\![I']\!] \vee a \in [\![I]\!]$
$[a(\vec{d})] \oplus \theta(\theta(n, [\![I']\!]), [\![I]\!])$	*if* $a \notin [\![I']\!] \wedge a \notin [\![I]\!]$

$= \quad \{ \text{ calculus } \}$

$\theta(\theta(n, [\![I']\!]), [\![I]\!])$	*if* $a \in [\![I \cup I']\!]$
$[a(\vec{d})] \oplus \theta(\theta(n, [\![I']\!]), [\![I]\!])$	*if* $a \notin [\![I \cup I']\!]$

$= \quad \{ \text{ induction hypothesis } \}$

$$\begin{array}{ll}
\theta(n, [\![I \cup I']\!]) & if\ a \in [\![I \cup I']\!] \\
[a(\overrightarrow{d}\,)] \oplus \theta(n, [\![I \cup I']\!]) & if\ a \notin [\![I \cup I']\!]
\end{array}$$
$$= \quad \{\ def.\ \theta\ \}$$
$$\theta([a(\overrightarrow{d}\,)] \oplus n, [\![I \cup I']\!])$$

C.15 $TA4$ $\tau_I(x \parallel y) \doteq \tau_I(x) \parallel \tau_I(y)$

This proof is similar to the proof of axiom $DA4$.

C.16 $RA1$ $\rho_R(x) \doteq x$ if $dom([\![R]\!]) \cap \mathcal{N}(\alpha_\nu(x)) = \emptyset$

We prove $[\![R]\!] \bullet m = m$ with induction on the structure of m. Case $m = [\,]$:

$$[\![R]\!] \bullet [\,]$$
$$= \quad \{\ def.\ \bullet\ \}$$
$$[\,]$$

Case $m = [a(\overrightarrow{d}\,)] \oplus n$:

$$[\![R]\!] \bullet ([a(\overrightarrow{d}\,)] \oplus n)$$
$$= \quad \{\ def.\ \bullet\ \}$$
$$[[\![R]\!]^+(a)(\overrightarrow{d}\,)] \oplus ([\![R]\!] \bullet n)$$
$$= \quad \{\ induction\ hypothesis\ \}$$
$$[[\![R]\!]^+(a)(\overrightarrow{d}\,)] \oplus n$$
$$= \quad \{\ a \notin dom([\![R]\!])\ \}$$
$$[a(\overrightarrow{d}\,)] \oplus n$$

C.17 $RA2$ $\rho_R(\rho_{R'}(x)) \doteq \rho_{R \cup R'}(x)$ if $dom([\![R]\!]) \cap dom([\![R']\!]) = \emptyset \wedge dom([\![R]\!]) \cap rng([\![R']\!]) = \emptyset$

This is a special case of axiom $RA3$, as we have the following:

$$\{\langle a, b\rangle \mid (\langle a, b\rangle \in [\![R]\!] \wedge a \notin (dom([\![R']\!]) \cup rng([\![R']\!]))) \vee (\langle c, b\rangle \in [\![R]\!] \wedge \langle a, c\rangle \in [\![R']\!]) \vee$$
$$(\langle a, b\rangle \in [\![R']\!] \wedge b \notin dom([\![R]\!]))\}$$
$$\equiv \quad \{\ dom([\![R]\!]) \cap dom([\![R']\!]) = \emptyset \wedge dom([\![R]\!]) \cap rng([\![R']\!]) = \emptyset\ \}$$
$$\{\langle a, b\rangle \mid (\langle a, b\rangle \in [\![R]\!] \wedge true) \vee false \vee (\langle a, b\rangle \in [\![R']\!] \wedge true)\}$$
$$\equiv \quad \{\ calculus\ \}$$
$$\{\langle a, b\rangle \mid \langle a, b\rangle \in [\![R]\!] \vee \langle a, b\rangle \in [\![R']\!]\}$$
$$\equiv \quad \{\ calculus\ \}$$
$$[\![R \cup R']\!]$$

C.18 $RA3$ $\rho_R(\rho_{R'}(x)) \doteq \rho_{R''}(x)$ if $[\![R'']\!] = \{\langle a, b\rangle \mid (\langle a, b\rangle \in [\![R]\!] \wedge a \notin (dom([\![R']\!]) \cup rng([\![R']\!]))) \vee (\langle c, b\rangle \in [\![R]\!] \wedge \langle a, c\rangle \in [\![R']\!]) \vee (\langle a, b\rangle \in [\![R']\!] \wedge b \notin dom([\![R]\!]))\}$

We prove $[\![R]\!] \bullet ([\![R']\!] \bullet m) = [\![R'']\!] \bullet m$ with induction on the structure of m. Case $m = [\,]$:

$$\llbracket R \rrbracket \bullet (\llbracket R' \rrbracket \bullet [])$$
$$= \quad \{ \text{ def. } \bullet \}$$
$$[]$$
$$= \quad \{ \text{ def. } \bullet \}$$
$$\llbracket R'' \rrbracket \bullet []$$

Case $m = [a(\overrightarrow{d})] \oplus n$:

$$\llbracket R \rrbracket \bullet (\llbracket R' \rrbracket \bullet ([a(\overrightarrow{d})] \oplus n))$$
$$= \quad \{ \text{ def. } \bullet \}$$
$$[\llbracket R \rrbracket^+(\llbracket R' \rrbracket^+(a))(\overrightarrow{d})] \oplus (\llbracket R \rrbracket \bullet (\llbracket R' \rrbracket \bullet n))$$
$$= \quad \{ \text{ induction hypothesis } \}$$
$$[\llbracket R \rrbracket^+(\llbracket R' \rrbracket^+(a))(\overrightarrow{d})] \oplus (\llbracket R'' \rrbracket \bullet n)$$
$$= \quad \{ \text{ case analysis } \llbracket R' \rrbracket^+(a) \}$$
$$\begin{array}{ll} [\llbracket R \rrbracket^+(a)(\overrightarrow{d})] \oplus (\llbracket R'' \rrbracket \bullet n) & \textit{if } a \notin dom(R') \\ [\llbracket R \rrbracket^+(\llbracket R' \rrbracket^+(a))(\overrightarrow{d})] \oplus (\llbracket R'' \rrbracket \bullet n) & \textit{if } a \in dom(R') \end{array}$$
$$= \quad \{ \text{ case analysis } \llbracket R \rrbracket^+(\llbracket R' \rrbracket^+(a)) \}$$
$$\begin{array}{ll} [\llbracket R \rrbracket^+(a)(\overrightarrow{d})] \oplus (\llbracket R'' \rrbracket \bullet n) & \textit{if } a \notin dom(R') \\ [\llbracket R' \rrbracket^+(a)(\overrightarrow{d})] \oplus (\llbracket R'' \rrbracket \bullet n) & \textit{if } a \in dom(R') \wedge \llbracket R' \rrbracket^+(a) \notin dom(R) \\ [\llbracket R \rrbracket^+(\llbracket R' \rrbracket^+(a))(\overrightarrow{d})] \oplus (\llbracket R'' \rrbracket \bullet n) & \textit{if } a \in dom(R') \wedge \llbracket R' \rrbracket^+(a) \in dom(R) \end{array}$$
$$= \quad \{ \text{ def. } R'' \}$$
$$\begin{array}{ll} [\llbracket R'' \rrbracket^+(a)(\overrightarrow{d})] \oplus (\llbracket R'' \rrbracket \bullet n) & \textit{if } a \notin dom(R') \\ [\llbracket R'' \rrbracket^+(a)(\overrightarrow{d})] \oplus (\llbracket R'' \rrbracket \bullet n) & \textit{if } a \in dom(R') \wedge \llbracket R' \rrbracket^+(a) \notin dom(R) \\ [\llbracket R'' \rrbracket^+(a)(\overrightarrow{d})] \oplus (\llbracket R'' \rrbracket \bullet n) & \textit{if } a \in dom(R') \wedge \llbracket R' \rrbracket^+(a) \in dom(R) \end{array}$$
$$= \quad \{ \text{ case elimination } \}$$
$$[\llbracket R'' \rrbracket^+(a)(\overrightarrow{d})] \oplus (\llbracket R'' \rrbracket \bullet n)$$
$$= \quad \{ \text{ def. } \bullet \}$$
$$\llbracket R'' \rrbracket \bullet ([a(\overrightarrow{d})] \oplus n)$$

C.19   *RA4* $\rho_R(x \parallel y) \doteq \rho_R(x) \parallel \rho_R(y)$

This proof is similar to the proof of axiom *DA4*.

C.20   *VC1* $\nabla_V(\Gamma_C(x)) \doteq \nabla_V(\gamma_C(\nabla_{V'}(x)))$ *if* $\llbracket V' \rrbracket = \{v \oplus w \mid \gamma(v, \llbracket C \rrbracket) \oplus w \in \llbracket V \rrbracket\}$

We prove $\mu(\gamma(m, \llbracket C \rrbracket)) \in \llbracket V \rrbracket \cup \{[]\} \Rightarrow \mu([]) \in \llbracket V' \rrbracket \cup \{[]\}$ (as $(P \equiv Q \wedge P) \equiv (P \Rightarrow Q)$) by induction on the length of m and by the cases of γ. Case $|m| = 0$:

$$\mu(\gamma([], \llbracket C \rrbracket)) \in \llbracket V \rrbracket \cup \{[]\}$$
$$\Rightarrow \quad \{ \text{ right zero } \Rightarrow \}$$
$$\textit{true}$$
$$\Rightarrow \quad \{ \mu([]) = [] \}$$
$$\mu([]) \in \llbracket V' \rrbracket \cup \{[]\}$$

Case $|m| > 0$ and $\exists_{n,o}(m = n \oplus o \wedge \exists_{\langle b,a \rangle \in C}(b = \mu(n) \wedge \exists_{\overrightarrow{d} \in \overrightarrow{D}}(\chi(n, \overrightarrow{d}))))$:

$\mu(\gamma(m, [\![C]\!])) \in [\![V]\!] \cup \{[]\}$

$\equiv$ { take n and o with $m = n \oplus o \wedge \exists_{\langle b,a \rangle \in C}(b = \mu(n) \wedge \exists_{\overrightarrow{d} \in \overrightarrow{\mathcal{D}}}(\chi(n, \overrightarrow{d})))) $ }

$\mu(\gamma(n \oplus o, [\![C]\!])) \in [\![V]\!] \cup \{[]\}$

$\equiv$ { take $\langle b,a \rangle \in C$ with $b = \mu(n) \wedge \exists_{\overrightarrow{d} \in \overrightarrow{\mathcal{D}}}(\chi(n, \overrightarrow{d})))) $, def. γ }

$\mu([a(\overrightarrow{d})] \oplus \gamma(o, [\![C]\!])) \in [\![V]\!] \cup \{[]\}$

$\equiv$ { def. μ }

$[a] \oplus \mu(\gamma(o, [\![C]\!])) \in [\![V]\!] \cup \{[]\}$

$\equiv$ { calculus }

$[a] \oplus \mu(\gamma(o, [\![C]\!])) \in \{v \mid v \in [\![V]\!]\}$

$\equiv$ { calculus }

$[a] \oplus \mu(\gamma(o, [\![C]\!])) \in \{[a] \oplus v \mid [a] \oplus v \in [\![V]\!]\}$

$\equiv$ { calculus }

$\mu(\gamma(o, [\![C]\!])) \in \{v \mid [a] \oplus v \in [\![V]\!]\}$

$\equiv$ { idem. $\wedge$ }

$\mu(\gamma(o, [\![C]\!])) \in \{v \mid [a] \oplus v \in [\![V]\!]\} \wedge \mu(\gamma(o, [\![C]\!])) \in \{v \mid [a] \oplus v \in [\![V]\!]\}$

$\Rightarrow$ { induction hypothesis }

$\mu(\gamma(o, [\![C]\!])) \in \{v \mid [a] \oplus v \in [\![V]\!]\} \wedge \mu(o) \in \{v \oplus w \mid \gamma(v, [\![C]\!]) \oplus w \in \{v' \mid [a] \oplus v' \in [\![V]\!]\}\} \cup \{[]\}$

$\equiv$ { calculus }

$(\mu(\gamma(o, [\![C]\!])) \in \{v \mid [a] \oplus v \in [\![V]\!]\} \wedge \mu(o) \in \{v \oplus w \mid \gamma(v, [\![C]\!]) \oplus w \in \{v' \mid [a] \oplus v' \in [\![V]\!]\}\}) \vee$
$(\mu(\gamma(o, [\![C]\!])) \in \{v \mid [a] \oplus v \in [\![V]\!]\} \wedge \mu(o) \in \{[]\})$

$\equiv$ { calculus, $\mu(o) \in \{[]\} \equiv o = []$ }

$(\mu(\gamma(o, [\![C]\!])) \in \{v \mid [a] \oplus v \in [\![V]\!]\} \wedge \mu(o) \in \{v \oplus w \mid \gamma(v, [\![C]\!]) \oplus w \in \{v' \mid [a] \oplus v' \in [\![V]\!]\}\}) \vee$
$(\mu(\gamma([], [\![C]\!])) \in \{v \mid [a] \oplus v \in [\![V]\!]\} \wedge o = [])$

$\equiv$ { calculus, $\mu(\gamma([], [\![C]\!])) = []$ }

$(\mu(\gamma(o, [\![C]\!])) \in \{v \mid [a] \oplus v \in [\![V]\!]\} \wedge \mu(o) \in \{v \oplus w \mid \gamma(v, [\![C]\!]) \oplus w \in \{v' \mid [a] \oplus v' \in [\![V]\!]\}\}) \vee$
$([a] \in [\![V]\!] \wedge o = [])$

$\equiv$ { calculus }

$(\mu(\gamma(o, [\![C]\!])) \in \{v \mid [a] \oplus v \in [\![V]\!]\} \wedge \mu(o) \in \{v \oplus w \mid \gamma(v, [\![C]\!]) \oplus w \in \{v' \mid [a] \oplus v' \in [\![V]\!]\}\}) \vee$
$(\mu(o) \in \{v' \mid [a] \oplus v' \in [\![V]\!]\} \wedge o = [])$

$\equiv$ { calculus }

$(\mu(\gamma(o, [\![C]\!])) \in \{v \mid [a] \oplus v \in [\![V]\!]\} \wedge \mu(o) \in \{v \oplus w \mid \gamma(v, [\![C]\!]) \oplus w \in \{v' \mid [a] \oplus v' \in [\![V]\!]\}\}) \vee$
$(\mu(o) \in \{v \oplus w \mid \gamma(v, [\![C]\!]) \oplus w \in \{v' \mid [a] \oplus v' \in [\![V]\!]\}\} \wedge o = [])$

$\equiv$ { calculus }

$(\mu(\gamma(o, [\![C]\!])) \in \{v \mid [a] \oplus v \in [\![V]\!]\} \vee (\mu(o) \in \{v \oplus w \mid \gamma(v, [\![C]\!]) \oplus w \in \{v' \mid [a] \oplus v' \in [\![V]\!]\}\} \wedge o = [])) \wedge$
$(\mu(o) \in \{v \oplus w \mid \gamma(v, [\![C]\!]) \oplus w \in \{v' \mid [a] \oplus v' \in [\![V]\!]\}\} \vee$
$(\mu(o) \in \{v \oplus w \mid \gamma(v, [\![C]\!]) \oplus w \in \{v' \mid [a] \oplus v' \in [\![V]\!]\}\} \wedge o = []))$

$\Rightarrow$ { weakening }

$\mu(o) \in \{v \oplus w \mid \gamma(v, [\![C]\!]) \oplus w \in \{v' \mid [a] \oplus v' \in [\![V]\!]\}\} \vee$
$(\mu(o) \in \{v \oplus w \mid \gamma(v, [\![C]\!]) \oplus w \in \{v' \mid [a] \oplus v' \in [\![V]\!]\}\} \wedge o = [])$

$\equiv$ { $P \vee (P \wedge Q) \equiv P$ }

$$\mu(o) \in \{v \oplus w \mid \gamma(v, \llbracket C \rrbracket) \oplus w \in \{v' \mid [a] \oplus v' \in \llbracket V \rrbracket\}\}$$
$\equiv$ { calculus }
$$\mu(o) \in \{v \oplus w \mid [a] \oplus \gamma(v, \llbracket C \rrbracket) \oplus w \in \{[a] \oplus v' \mid [a] \oplus v' \in \llbracket V \rrbracket\}\}$$
$\Rightarrow$ { calculus, def. γ and a }
$$\mu(o) \in \{v \oplus w \mid \gamma(\mu(n) \oplus v, \llbracket C \rrbracket) \oplus w \in \{v' \mid v' \in \llbracket V \rrbracket\}\}$$
$\equiv$ { calculus }
$$\mu(n) \oplus \mu(o) \in \{\mu(n) \oplus v \oplus w \mid \gamma(\mu(n) \oplus v, \llbracket C \rrbracket) \oplus w \in \llbracket V \rrbracket\}$$
$\Rightarrow$ { calculus, def. μ }
$$\mu(n \oplus o) \in \{v \oplus w \mid \gamma(v, \llbracket C \rrbracket) \oplus w \in \llbracket V \rrbracket\}$$
$\equiv$ { def. n, o and V' }
$$\mu(m) \in \llbracket V' \rrbracket$$
$\Rightarrow$ { weakening }
$$\mu(m) \in \llbracket V' \rrbracket \cup \{[]\}$$

Case $|m| > 0$ and $\exists_{n,o}(m = n \oplus o \wedge \exists_{\langle b, \tau \rangle \in C}(b = \mu(n) \wedge \exists_{\overrightarrow{d} \in \overrightarrow{D}}(\chi(n, \overrightarrow{d}))))$, which is similar to the previous case.

Case $|m| > 0$ and $\neg\exists_{n,o}(m = n \oplus o \wedge \exists_{c \in C}((c = \langle b, a \rangle \vee c = \langle b, \tau \rangle) \wedge b = \mu(n) \wedge \exists_{\overrightarrow{d} \in \overrightarrow{D}}(\chi(n, \overrightarrow{d}))))$:

$$\mu(\gamma(m, \llbracket C \rrbracket)) \in \llbracket V \rrbracket \cup \{[]\}$$
$\equiv$ { $\neg\exists_{n,o}(m = n \oplus o \wedge \exists_{c \in C}((c = \langle b, a \rangle \vee c = \langle b, \tau \rangle) \wedge b = \mu(n) \wedge \exists_{\overrightarrow{d} \in \overrightarrow{D}}(\chi(n, \overrightarrow{d}))))$, def. γ }
$$\mu(m) \in \llbracket V \rrbracket \cup \{[]\}$$
$\Rightarrow$ { $\llbracket V \rrbracket \subseteq \llbracket V' \rrbracket$ }
$$\mu(m) \in \llbracket V' \rrbracket \cup \{[]\}$$

C.21   VC2 $\Gamma_C(\nabla_V(x)) \doteq \nabla_V(x)$ if $dom(C) \cap \Downarrow(V) = \emptyset$

$$\mu(m) \in \llbracket V \rrbracket$$
$\Rightarrow$ { def. $\Downarrow$ }
$$\Downarrow(\{\mu(m)\}) \subseteq \Downarrow(V)$$
$\Rightarrow$ { $dom(C) \cap \Downarrow(V) = \emptyset$ }
$$dom(C) \cap \Downarrow(\{\mu(m)\}) = \emptyset$$
$\Rightarrow$ { calculus }
$$\neg\exists_{n,o}(m = n \oplus o \wedge \exists_{c \in C}((c = \langle b, a \rangle \vee c = \langle b, \tau \rangle) \wedge b = \mu(n) \wedge \exists_{\overrightarrow{d} \in \overrightarrow{D}}(\chi(n, \overrightarrow{d}))))$$
$\Rightarrow$ { def. γ }
$$\gamma(m, \llbracket C \rrbracket) = m$$

C.22   VD1 $\nabla_V(\partial_H(x)) \doteq \partial_H(\nabla_V(x))$

$$\mu(m) \cap \llbracket H \rrbracket = \emptyset \wedge \mu(m) \in \llbracket V \rrbracket \cup \{[]\}$$
$\equiv$ { comm. $\wedge$ }
$$\mu(m) \in \llbracket V \rrbracket \cup \{[]\} \wedge \mu(m) \cap \llbracket H \rrbracket = \emptyset$$

C.23 $VD2$ $\nabla_V(\partial_H(x)) \doteq \nabla_{V'}(x)$ if $[\![V']\!] = \{v \mid v \in [\![V]\!] \wedge \mathcal{N}(\{v\}) \cap [\![H]\!] = \emptyset\}$

$\quad\quad \mu(m) \cap [\![H]\!] = \emptyset \wedge \mu(m) \in [\![V]\!] \cup \{[]\}$
$\equiv\quad\quad \{ \text{ Lemma C.1 } \}$
$\quad\quad \mathcal{N}(\{\mu(m)\}) \cap [\![H]\!] = \emptyset \wedge \mu(m) \in [\![V]\!] \cup \{[]\}$
$\equiv\quad\quad \{ \text{ calculus } \}$
$\quad\quad \mu(m) \in \{v \mid v \in [\![V]\!] \wedge \mathcal{N}(\{v\}) \cap [\![H]\!] = \emptyset\} \cup \{[]\}$
$\equiv\quad\quad \{ \text{ def. } V' \}$
$\quad\quad \mu(m) \in [\![V']\!] \cup \{[]\}$

C.24 $VD3$ $\partial_H(\nabla_V(x)) \doteq \nabla_{V'}(x)$ if $[\![V']\!] = \{v \mid v \in [\![V]\!] \wedge \mathcal{N}(\{v\}) \cap [\![H]\!] = \emptyset\}$

This proof is similar to the proof of axiom $VD2$.

C.25 VT $\nabla_V(\tau_I(x)) \doteq \tau_I(\nabla_{V'}(x))$ if $[\![V']\!] = \{v \mid \theta(v, [\![I]\!]) \in [\![V]\!]\}$

$\quad\quad \mu(\theta(m, [\![I]\!])) \in [\![V]\!] \cup \{[]\}$
$\equiv\quad\quad \{ \text{ calculus } \}$
$\quad\quad \mu(\theta(m, [\![I]\!])) \in \{v \mid v \in [\![V]\!]\} \cup \{[]\}$
$\equiv\quad\quad \{ \text{ calculus } \}$
$\quad\quad \mu(\theta(m, [\![I]\!])) \in \{\theta(v, [\![I]\!]) \mid \theta(v, [\![I]\!]) \in [\![V]\!]\} \cup \{[]\}$
$\equiv\quad\quad \{ \text{ calculus } \}$
$\quad\quad \mu(m) \in \{v \mid \theta(v, [\![I]\!]) \in [\![V]\!]\} \cup \{[]\}$
$\equiv\quad\quad \{ \text{ def. } V' \}$
$\quad\quad \mu(m) \in [\![V']\!] \cup \{[]\}$

C.26 VR $\nabla_V(\rho_R(x)) \doteq \rho_R(\nabla_{V'}(x))$ if $[\![V']\!] = \{v \mid ([\![R]\!] \bullet v) \in [\![V]\!]\}$

$\quad\quad \mu([\![R]\!] \bullet m) \in [\![V]\!] \cup \{[]\}$
$\equiv\quad\quad \{ \text{ calculus } \}$
$\quad\quad \mu([\![R]\!] \bullet m) \in \{v \mid v \in [\![V]\!]\} \cup \{[]\}$
$\equiv\quad\quad \{ \text{ calculus } \}$
$\quad\quad \mu([\![R]\!] \bullet m) \in \{[\![R]\!] \bullet v \mid ([\![R]\!] \bullet v) \in [\![V]\!]\} \cup \{[]\}$
$\equiv\quad\quad \{ \text{ calculus } \}$
$\quad\quad \mu(m) \in \{v \mid ([\![R]\!] \bullet v) \in [\![V]\!]\} \cup \{[]\}$
$\equiv\quad\quad \{ \text{ def. } V' \}$
$\quad\quad \mu(m) \in [\![V']\!] \cup \{[]\}$

C.27 $CD1$ $\partial_H(\Gamma_C(x)) \doteq \Gamma_C(\partial_H(x))$ if $(\mathcal{N}(dom(C)) \cup rng(C)) \cap H = \emptyset$

We prove $\mu(\gamma(m, [\![C]\!])) \cap H = \emptyset \equiv \mu(m) \cap H = \emptyset$ by induction on the length of m and by the cases of γ. Case $|m| = 0$:

$\quad\quad \mu(\gamma([], [\![C]\!])) \cap H = \emptyset$
$\equiv\quad\quad \{ \text{ def. } \gamma \}$
$\quad\quad \mu([]) \cap H = \emptyset$

Case $|m| > 0$ and $\exists_{n,o}(m = n \oplus o \wedge \exists_{\langle b,a \rangle \in C}(b = \mu(n) \wedge \exists_{\vec{d} \in \vec{D}}(\chi(n, \vec{d}))))$:

$$\mu(\gamma(m, [\![C]\!])) \cap H = \emptyset$$
$\equiv$ { take n and o with $m = n \oplus o \wedge \exists_{\langle b,a \rangle \in C}(b = \mu(n) \wedge \exists_{\vec{d} \in \vec{D}}(\chi(n, \vec{d})))$ }
$$\mu(\gamma(n \oplus o, [\![C]\!])) \cap H = \emptyset$$
$\equiv$ { take $\langle b, a \rangle \in C$ with $b = \mu(n) \wedge \exists_{\vec{d} \in \vec{D}}(\chi(n, \vec{d})))$, def. γ }
$$\mu([a(\vec{d})] \oplus \gamma(o, [\![C]\!])) \cap H = \emptyset$$
$\equiv$ { Lemma C.1 }
$$\mathcal{N}(\{\mu([a(\vec{d})] \oplus \gamma(o, [\![C]\!]))\}) \cap H = \emptyset$$
$\equiv$ { def. $\mathcal{N}$ }
$$(\{a\} \cup \mathcal{N}(\{\mu(\gamma(o, [\![C]\!]))\})) \cap H = \emptyset$$
$\equiv$ { Lemma C.1, calculus }
$$\{a\} \cap H = \emptyset \wedge \mu(\gamma(o, [\![C]\!])) \cap H = \emptyset$$
$\equiv$ { induction hypothesis }
$$\{a\} \cap H = \emptyset \wedge \mu(o) \cap H = \emptyset$$
$\equiv$ { $a \in rng(C)$ }
$$\mu(o) \cap H = \emptyset$$
$\equiv$ { $\mu(n) \in dom(C)$ }
$$\mu(n) \cap H = \emptyset \wedge \mu(o) \cap H = \emptyset$$
$\equiv$ { Lemma C.1, calculus }
$$\mu(n \oplus o) \cap H = \emptyset$$
$\equiv$ { def. n and o }
$$\mu(m) \cap H = \emptyset$$

Case $|m| > 0$ and $\exists_{n,o}(m = n \oplus o \wedge \exists_{\langle b,\tau \rangle \in C}(b = \mu(n) \wedge \exists_{\vec{d} \in \vec{D}}(\chi(n, \vec{d}))))$, which is similar to the previous case.

Case $|m| > 0$ and $\neg\exists_{n,o}(m = n \oplus o \wedge \exists_{c \in C}((c = \langle b,a \rangle \vee c = \langle b,\tau \rangle) \wedge b = \mu(n) \wedge \exists_{\vec{d} \in \vec{D}}(\chi(n, \vec{d}))))$:

$$\mu(\gamma(m, [\![C]\!])) \cap H = \emptyset$$
$\equiv$ { $\neg\exists_{n,o}(m = n \oplus o \wedge \exists_{c \in C}((c = \langle b,a \rangle \vee c = \langle b,\tau \rangle) \wedge b = \mu(n) \wedge \exists_{\vec{d} \in \vec{D}}(\chi(n, \vec{d})))$, def. γ }
$$\mu(m) \cap H = \emptyset$$

C.28   *CD2* $\Gamma_C(\partial_H(x)) \doteq \partial_H(x)$ *if* $\mathcal{N}(dom(C)) \subseteq H$

$$\mu(m) \cap H = \emptyset$$
$\equiv$ { Lemma C.1 }
$$\mathcal{N}(\{\mu(m)\}) \cap H = \emptyset$$
$\Rightarrow$ { $\mathcal{N}(dom(C)) \subseteq H$ }
$$\mathcal{N}(\{\mu(m)\}) \cap \mathcal{N}(dom(C)) = \emptyset$$
$\Rightarrow$ { $\neg\exists_{n,o}(m = n \oplus o \wedge \exists_{c \in C}((c = \langle b,a \rangle \vee c = \langle b,\tau \rangle) \wedge b = \mu(n) \wedge \exists_{\vec{d} \in \vec{D}}(\chi(n, \vec{d})))$, def. γ }
$$\gamma(m, [\![C]\!]) = m$$

C.29 $CT1$ $\tau_I(\Gamma_C(x)) \doteq \Gamma_C(\tau_I(x))$ *if* $(\mathcal{N}(dom(C)) \cup rng(C)) \cap I = \emptyset$

We prove $\theta(\gamma(m, [\![C]\!]), [\![I]\!]) = \gamma(\theta(m, [\![I]\!]), [\![C]\!])$ by induction on the length of m and by the cases of γ.
Case $|m| = 0$:

$$\theta(\gamma([], [\![C]\!]), [\![I]\!])$$
$=$ $\{$ def. γ and θ $\}$
$$[]$$
$=$ $\{$ def. γ and θ $\}$
$$\gamma(\theta([], [\![I]\!]), [\![C]\!])$$

Case $|m| > 0$ and $\exists_{n,o}(m = n \oplus o \wedge \exists_{\langle b,a \rangle \in C}(b = \mu(n) \wedge \exists_{\overrightarrow{d} \in \overrightarrow{D}}(\chi(n, \overrightarrow{d}))))$:

$$\theta(\gamma(m, [\![C]\!]), [\![I]\!])$$
$=$ $\{$ take n and o with $m = n \oplus o \wedge \exists_{\langle b,a \rangle \in C}(b = \mu(n) \wedge \exists_{\overrightarrow{d} \in \overrightarrow{D}}(\chi(n, \overrightarrow{d}))) \}$
$$\theta(\gamma(n \oplus o, [\![C]\!]), [\![I]\!])$$
$=$ $\{$ take $\langle b, a \rangle \in C$ with $b = \mu(n) \wedge \exists_{\overrightarrow{d} \in \overrightarrow{D}}(\chi(n, \overrightarrow{d})))$, def. γ $\}$
$$\theta([a(\overrightarrow{d})] \oplus \gamma(o, [\![C]\!]), [\![I]\!])$$
$=$ $\{$ $a \in rng(C)$ $\}$
$$[a(\overrightarrow{d})] \oplus \theta(\gamma(o, [\![C]\!]), [\![I]\!])$$
$=$ $\{$ induction hypothesis $\}$
$$[a(\overrightarrow{d})] \oplus \gamma(\theta(o, [\![I]\!]), [\![C]\!])$$
$=$ $\{$ def. a and γ $\}$
$$\gamma(n \oplus \theta(o, [\![I]\!]), [\![C]\!])$$
$=$ $\{$ $n \in dom(C)$ $\}$
$$\gamma(\theta(n \oplus o, [\![I]\!]), [\![C]\!])$$
$=$ $\{$ def. n and o $\}$
$$\gamma(\theta(m, [\![I]\!]), [\![C]\!])$$

Case $|m| > 0$ and $\exists_{n,o}(m = n \oplus o \wedge \exists_{\langle b,\tau \rangle \in C}(b = \mu(n) \wedge \exists_{\overrightarrow{d} \in \overrightarrow{D}}(\chi(n, \overrightarrow{d}))))$, which is similar to the previous case.
Case $|m| > 0$ and $\neg\exists_{n,o}(m = n \oplus o \wedge \exists_{c \in C}((c = \langle b,a \rangle \vee c = \langle b,\tau \rangle) \wedge b = \mu(n) \wedge \exists_{\overrightarrow{d} \in \overrightarrow{D}}(\chi(n, \overrightarrow{d}))))$:

$$\theta(\gamma(m, [\![C]\!]), [\![I]\!])$$
$=$ $\{$ $\neg\exists_{n,o}(m = n \oplus o \wedge \exists_{c \in C}((c = \langle b,a \rangle \vee c = \langle b,\tau \rangle) \wedge b = \mu(n) \wedge \exists_{\overrightarrow{d} \in \overrightarrow{D}}(\chi(n, \overrightarrow{d}))))$, def. γ $\}$
$$\theta(m, [\![I]\!])$$
$=$ $\{$ $\theta(m, [\![I]\!]) \subseteq m$, def. γ $\}$
$$\gamma(\theta(m, [\![I]\!]), [\![C]\!])$$

C.30 $CT2$ $\Gamma_C(\tau_I(x)) \doteq \tau_I(x)$ *if* $\mathcal{N}(dom(C)) \subseteq I$

$$\gamma(\theta(m, [\![I]\!]), [\![C]\!]) = \theta(m, [\![I]\!])$$
$\equiv$ $\{$ def. γ $\}$
$$\neg\exists_{n,o}(\theta(m, [\![I]\!]) = n \oplus o \wedge \exists_{c \in C}((c = \langle b,a \rangle \vee c = \langle b,\tau \rangle) \wedge b = \mu(n) \wedge \exists_{\overrightarrow{d} \in \overrightarrow{D}}(\chi(n, \overrightarrow{d}))))$$
$\equiv$ $\{$ $\theta(m, [\![I]\!]) \cap [\![I]\!] = \emptyset$ $\}$
$$true$$

C.31   **CR1** $\rho_R(\Gamma_C(x)) \doteq \Gamma_C(\rho_R(x))$ if $dom(R) \cap rng(C) = \emptyset \wedge dom(R) \cap \mathcal{N}(dom(C)) = \emptyset \wedge rng(R) \cap \mathcal{N}(dom(C)) = \emptyset$

We prove $[\![R]\!] \bullet \gamma(m, [\![C]\!]) = \gamma([\![R]\!] \bullet m, [\![C]\!])$ by induction on the length of m and by the cases of γ.
Case $|m| = 0$:

$$[\![R]\!] \bullet \gamma([], [\![C]\!])$$
$$= \quad \{ \text{ def. } \gamma \text{ and } \bullet \}$$
$$[]$$
$$= \quad \{ \text{ def. } \gamma \text{ and } \bullet \}$$
$$\gamma([\![R]\!] \bullet [], [\![C]\!])$$

Case $|m| > 0$ and $\exists_{n,o}(m = n \oplus o \wedge \exists_{\langle b,a \rangle \in C}(b = \mu(n) \wedge \exists_{\overrightarrow{d} \in \overrightarrow{D}}(\chi(n, \overrightarrow{d}))))$:

$$[\![R]\!] \bullet \gamma(m, [\![C]\!])$$
$$= \quad \{ \text{ take } n \text{ and } o \text{ with } m = n \oplus o \wedge \exists_{\langle b,a \rangle \in C}(b = \mu(n) \wedge \exists_{\overrightarrow{d} \in \overrightarrow{D}}(\chi(n, \overrightarrow{d}))) \}$$
$$[\![R]\!] \bullet \gamma(n \oplus o, [\![C]\!])$$
$$= \quad \{ \text{ take } \langle b,a \rangle \in C \text{ with } b = \mu(n) \wedge \exists_{\overrightarrow{d} \in \overrightarrow{D}}(\chi(n, \overrightarrow{d}))), \text{ def. } \gamma \}$$
$$[\![R]\!] \bullet ([a(\overrightarrow{d})] \oplus \gamma(o, [\![C]\!]))$$
$$= \quad \{ a \in rng(C) \}$$
$$[a(\overrightarrow{d})] \oplus ([\![R]\!] \bullet \gamma(o, [\![C]\!]))$$
$$= \quad \{ \text{ induction hypothesis } \}$$
$$[a(\overrightarrow{d})] \oplus \gamma([\![R]\!] \bullet o, [\![C]\!])$$
$$= \quad \{ \text{ def. } a \text{ and } \gamma \}$$
$$\gamma(n \oplus ([\![R]\!] \bullet o), [\![C]\!])$$
$$= \quad \{ n \in dom(C) \}$$
$$\gamma([\![R]\!] \bullet (n \oplus o), [\![C]\!])$$
$$= \quad \{ \text{ def. } n \text{ and } o \}$$
$$\gamma([\![R]\!] \bullet m, [\![C]\!])$$

Case $|m| > 0$ and $\exists_{n,o}(m = n \oplus o \wedge \exists_{\langle b,\tau \rangle \in C}(b = \mu(n) \wedge \exists_{\overrightarrow{d} \in \overrightarrow{D}}(\chi(n, \overrightarrow{d}))))$, which is similar to the previous case.
Case $|m| > 0$ and $\neg\exists_{n,o}(m = n \oplus o \wedge \exists_{c \in C}((c = \langle b,a \rangle \vee c = \langle b,\tau \rangle) \wedge b = \mu(n) \wedge \exists_{\overrightarrow{d} \in \overrightarrow{D}}(\chi(n, \overrightarrow{d}))))$:

$$[\![R]\!] \bullet \gamma(m, [\![C]\!])$$
$$= \quad \{ |m| > 0 \Rightarrow \exists_{a, \overrightarrow{d}, n}(m = [a(\overrightarrow{d})] \oplus n), \text{ take such } a, \overrightarrow{d} \text{ and } n \}$$
$$[\![R]\!] \bullet \gamma([a(\overrightarrow{d})] \oplus n, [\![C]\!])$$
$$= \quad \{ \neg\exists_{n,o}(m = n \oplus o \wedge \exists_{c \in C}((c = \langle b,a \rangle \vee c = \langle b,\tau \rangle) \wedge b = \mu(n) \wedge \exists_{\overrightarrow{d} \in \overrightarrow{D}}(\chi(n, \overrightarrow{d}))), \text{ def. } \gamma \}$$
$$[\![R]\!] \bullet ([a(\overrightarrow{d})] \oplus \gamma(n, [\![C]\!]))$$
$$= \quad \{ \text{ def. } \bullet \}$$
$$[[\![R]\!]^+(a)(\overrightarrow{d})] \oplus ([\![R]\!] \bullet \gamma(n, [\![C]\!]))$$
$$= \quad \{ \text{ induction hypothesis } \}$$
$$[[\![R]\!]^+(a)(\overrightarrow{d})] \oplus \gamma([\![R]\!] \bullet n, [\![C]\!])$$
$$= \quad \{ \neg\exists_{n,o}(m = n \oplus o \wedge \exists_{c \in C}((c = \langle b,a \rangle \vee c = \langle b,\tau \rangle) \wedge b = \mu(n) \wedge \exists_{\overrightarrow{d} \in \overrightarrow{D}}(\chi(n, \overrightarrow{d}))),$$
$$\quad rng(R) \cap \mathcal{N}(dom(C)) = \emptyset, \text{ def. } \gamma \}$$

$$\gamma([\![R]\!]^+(a)(\overrightarrow{d})] \oplus ([\![R]\!] \bullet n), [\![C]\!])$$
$$=\quad \{ \text{ def. } \bullet \}$$
$$\gamma([\![R]\!] \bullet ([a(\overrightarrow{d})] \oplus n), [\![C]\!])$$
$$=\quad \{ \text{ def. } a, \overrightarrow{d} \text{ and } n \}$$
$$\gamma([\![R]\!] \bullet m, [\![C]\!])$$

C.32   *CR2* $\Gamma_C(\rho_R(x)) \doteq \rho_R(x)$ *if* $\mathcal{N}(dom(C)) \subseteq dom(R) \wedge \mathcal{N}(dom(C)) \cap rng(R) = \emptyset$

We proof $\gamma([\![R]\!] \bullet m, [\![C]\!]) = [\![R]\!] \bullet m$ with induction on the structure of m. Case $m = []$

$$\gamma([\![R]\!] \bullet [], [\![C]\!])$$
$$=\quad \{ \text{ def. } \bullet \text{ and } \gamma \}$$
$$[]$$
$$=\quad \{ \text{ def. } \bullet \}$$
$$[\![R]\!] \bullet []$$

Case $m = [a(\overrightarrow{d})] \oplus n$:

$$\gamma([\![R]\!] \bullet ([a(\overrightarrow{d})] \oplus n), [\![C]\!])$$
$$=\quad \{ \text{ def. } bullet \}$$
$$\gamma([\![R]\!]^+(a)(\overrightarrow{d})] \oplus ([\![R]\!] \bullet n), [\![C]\!])$$
$$=\quad \{ [\![R]\!]^+(a) \in rng(R) \vee [\![R]\!]^+(a) \notin dom(R), \text{ def. } gamma \}$$
$$[[\![R]\!]^+(a)(\overrightarrow{d})] \oplus \gamma([\![R]\!] \bullet n, [\![C]\!])$$
$$=\quad \{ \text{ induction hypothesis } \}$$
$$[[\![R]\!]^+(a)(\overrightarrow{d})] \oplus ([\![R]\!] \bullet n)$$
$$=\quad \{ \text{ def. } \bullet \}$$
$$[\![R]\!] \bullet ([a(\overrightarrow{d})] \oplus n)$$

C.33   *DT* $\partial_H(\tau_I(x)) \doteq \tau_I(\partial_H(x))$ *if* $[\![I]\!] \cap [\![H]\!] = \emptyset$

We proof $\mu(\theta(m, [\![I]\!])) \cap [\![H]\!] = \emptyset \equiv \mu(m) \cap [\![H']\!] = \emptyset$ with induction on the structure of m. Case $m = []$

$$\mu(\theta([], [\![I]\!])) \cap [\![H]\!] = \emptyset$$
$$\equiv\quad \{ \text{ def. } \theta \}$$
$$\mu([]) \cap [\![H']\!] = \emptyset$$

Case $m = [a(\overrightarrow{d})] \oplus n$:

$$\mu(\theta([a(\overrightarrow{d})] \oplus n, [\![I]\!])) \cap [\![H]\!] = \emptyset$$
$$\equiv\quad \{ \text{ def. } \theta \}$$
$$\begin{array}{ll} \mu(\theta(n, [\![I]\!])) \cap [\![H]\!] = \emptyset & \text{if } a \in [\![I]\!] \\ \mu([a(\overrightarrow{d})] \oplus \theta(n, [\![I]\!])) \cap [\![H]\!] = \emptyset & \text{if } a \notin [\![I]\!] \end{array}$$
$$\equiv\quad \{ \text{ def. } \mu, \text{ Lemma C.1 and def. } \mathcal{N} \}$$
$$\begin{array}{ll} \mu(\theta(n, [\![I]\!])) \cap [\![H]\!] = \emptyset & \text{if } a \in [\![I]\!] \\ \mu([a(\overrightarrow{d})]) \cap [\![H]\!] = \emptyset \wedge \mu(\theta(n, [\![I]\!])) \cap [\![H]\!] = \emptyset & \text{if } a \notin [\![I]\!] \end{array}$$
$$\equiv\quad \{ \text{ induction hypothesis } \}$$

$$
\begin{aligned}
&\mu(n) \cap [\![H]\!] = \emptyset && \text{if } a \in [\![I]\!] \\
&\mu([a(\overrightarrow{d})]) \cap [\![H]\!] = \emptyset \wedge \mu(n) \cap [\![H]\!] = \emptyset && \text{if } a \notin [\![I]\!] \\
=\quad & \{\, a \in [\![I]\!] \Rightarrow \mu([a(\overrightarrow{d})]) \cap [\![H]\!] = \emptyset \,\} \\
&\mu([a(\overrightarrow{d})]) \cap [\![H]\!] = \emptyset \wedge \mu(n) \cap [\![H]\!] = \emptyset && \text{if } a \in [\![I]\!] \\
&\mu([a(\overrightarrow{d})]) \cap [\![H]\!] = \emptyset \wedge \mu(n) \cap [\![H]\!] = \emptyset && \text{if } a \notin [\![I]\!] \\
=\quad & \{\, \text{case elimination} \,\} \\
&\mu([a(\overrightarrow{d})]) \cap [\![H]\!] = \emptyset \wedge \mu(n) \cap [\![H]\!] = \emptyset \\
=\quad & \{\, \text{def. } \mu, \text{ Lemma C.1 and def. } \mathcal{N} \,\} \\
&\mu([a(\overrightarrow{d})] \oplus n) \cap [\![H']\!] = \emptyset
\end{aligned}
$$

C.34   DR $\partial_H(\rho_R(x)) \doteq \rho_R(\partial_{H'}(x))$ if $[\![H']\!] = \{v \mid ([\![R]\!] \bullet v) \in [\![H]\!]\}$

We proof $\mu([\![R]\!] \bullet m) \cap [\![H]\!] = \emptyset \equiv \mu(m) \cap [\![H']\!] = \emptyset$ with induction on the structure of m. Case $m = []$

$$
\begin{aligned}
&\mu([\![R]\!] \bullet []) \cap [\![H]\!] = \emptyset \\
\equiv\quad & \{\, \text{def. } \bullet, \text{ Lemma C.1 and def. } \mathcal{N} \,\} \\
&\emptyset = \emptyset \\
\equiv\quad & \{\, \text{def. } \mathcal{N} \text{ and Lemma C.1} \,\} \\
&\mu([]) \cap [\![H']\!] = \emptyset
\end{aligned}
$$

Case $m = [a(\overrightarrow{d})] \oplus n$:

$$
\begin{aligned}
&\mu([\![R]\!] \bullet ([a(\overrightarrow{d})] \oplus n)) \cap [\![H]\!] = \emptyset \\
\equiv\quad & \{\, \text{def. } \bullet \,\} \\
&\mu([[\![R]\!]^+(a)(\overrightarrow{d})] \oplus ([\![R]\!] \bullet n)) \cap [\![H]\!] = \emptyset \\
\equiv\quad & \{\, \text{def. } \bullet, \text{ Lemma C.1 and def. } \mathcal{N} \,\} \\
&\mu([[\![R]\!]^+(a)(\overrightarrow{d})]) \cap [\![H]\!] = \emptyset \wedge \mu([\![R]\!] \bullet n) \cap [\![H]\!] = \emptyset \\
\equiv\quad & \{\, \text{induction hypothesis} \,\} \\
&\mu([[\![R]\!]^+(a)(\overrightarrow{d})]) \cap [\![H]\!] = \emptyset \wedge \mu(n) \cap [\![H']\!] = \emptyset \\
\equiv\quad & \{\, \text{def. } \mu \text{ and } \mathcal{N}, \text{ calculus} \,\} \\
&[\![R]\!]^+(a) \notin [\![H]\!] \wedge \mu(n) \cap [\![H']\!] = \emptyset \\
\equiv\quad & \{\, \text{def. } H' \,\} \\
&a \notin [\![H']\!] \wedge \mu(n) \cap [\![H']\!] = \emptyset \\
\equiv\quad & \{\, \text{def. } \mu \text{ and } \mathcal{N}, \text{ calculus} \,\} \\
&\mu([a(\overrightarrow{d})]) \cap [\![H']\!] = \emptyset \wedge \mu(n) \cap [\![H']\!] = \emptyset \\
\equiv\quad & \{\, \text{Lemma C.1 and def. } \mathcal{N} \,\} \\
&\mu([a(\overrightarrow{d})] \oplus n) \cap [\![H']\!] = \emptyset
\end{aligned}
$$

C.35   TR $\tau_I(\rho_R(x)) \doteq \rho_R(\tau_{I'}(x))$ if $[\![I]\!] = \{[\![R]\!]^+(a) \mid a \in [\![I']\!]\}$

We proof $\theta([\![R]\!] \bullet m, [\![I]\!]) = [\![R]\!] \bullet \theta(m, [\![I']\!])$ with induction on the structure of m. Case $m = []$

$$
\begin{aligned}
&\theta([\![R]\!] \bullet [], [\![I]\!]) \\
=\quad & \{\, \text{def. } \bullet \text{ and } \theta \,\}
\end{aligned}
$$

$$[]$$
$$= \quad \{ \text{ def. } \bullet \text{ and } \theta \}$$
$$[\![R]\!] \bullet \theta([], [\![I']\!])$$

Case $m = [a(\overrightarrow{d})] \oplus n$:

$$\theta([\![R]\!] \bullet ([a(\overrightarrow{d})] \oplus n), [\![I]\!])$$
$$= \quad \{ \text{ def. } \bullet \}$$
$$\theta([[\![R]\!]^+(a)(\overrightarrow{d})] \oplus ([\![R]\!] \bullet n), [\![I]\!])$$
$$= \quad \{ \text{ def. } \theta \}$$

$$\begin{array}{ll} \theta([\![R]\!] \bullet n, [\![I]\!]) & \text{if } [\![R]\!]^+(a) \in [\![I]\!] \\ [[\![R]\!]^+(a)(\overrightarrow{d})] \oplus \theta([\![R]\!] \bullet n, [\![I]\!]) & \text{if } [\![R]\!]^+(a) \notin [\![I]\!] \end{array}$$

$$= \quad \{ \text{ induction hypothesis } \}$$

$$\begin{array}{ll} [\![R]\!] \bullet \theta(n, [\![I]\!]) & \text{if } [\![R]\!]^+(a) \in [\![I]\!] \\ [[\![R]\!]^+(a)(\overrightarrow{d})] \oplus ([\![R]\!] \bullet \theta(n, [\![I]\!])) & \text{if } [\![R]\!]^+(a) \notin [\![I]\!] \end{array}$$

$$= \quad \{ [\![R]\!]^+(a) \in [\![I]\!] \equiv a \in [\![I']\!], \text{ def, } \bullet \}$$

$$\begin{array}{ll} [\![R]\!] \bullet \theta(n, [\![I]\!]) & \text{if } a \in [\![I']\!] \\ [\![R]\!] \bullet ([a(\overrightarrow{d})] \oplus \theta(n, [\![I]\!])) & \text{if } [\![R]\!]^+(a) \notin [\![I]\!] \end{array}$$

$$= \quad \{ \text{ def. } \theta, [\![R]\!]^+(a) \notin [\![I]\!] \equiv a \notin [\![I']\!] \}$$

$$\begin{array}{ll} [\![R]\!] \bullet \theta(n, [\![I]\!]) & \text{if } a \in [\![I']\!] \\ [\![R]\!] \bullet \theta([a(\overrightarrow{d})] \oplus n, [\![I]\!]) & \text{if } a \notin [\![I']\!] \end{array}$$

$$= \quad \{ \text{ def. } \theta \}$$
$$[\![R]\!] \bullet \theta([a(\overrightarrow{d})] \oplus n, [\![I']\!])$$

References

[1] J. Baeten and C. Verhoef. A congruence theorem for structured operational semantics with predicates. In *International Conference on Concurrency Theory*, pages 477–492, 1993.

[2] J. Baeten and W. Weijland. *Process Algebra*, volume 18 of *Cambridge Tracts in Theoretical Computer Science*. Cambrdige University Press, 1990.

[3] B. Bloom. Structural operational semantics for weak bisimulations. *Theoretical Computer Science*, 146(1-2):25–68, 1995.

[4] J. Groote and S. Luttik. A complete axiomatisation of branching bisimulation for process algebras with alternative quantification over data. Technical Report SEN-R9830, Centrum voor Wiskunde en Informatica (CWI), ISSN 1386-369X, 1998.

[5] R. Milner. Calculi for synchrony and asynchrony. *Theoretical Computer Science*, 25:267–310, 1983.

[6] M. Reniers, J. Groote, M. van der Zwaag, and J. van Wamel. Completeness of timed μCRL. *Fundamenta Informaticae*, 50(3):361–402, 2002.

[7] W. Weijland. *Synchrony and asynchrony in process algebra*. 1989.