Calculation of Communication with Open Terms

in GenSpect Process Algebra

(Draft)

Muck van Weerdenburg

M.J.v.Weerdenburg@student.tue.nl

We recall the definition of the communication function γ of [1].

Definition 0.1. Let $m \in \mathbb{B}(\boldsymbol{A})$ and $a \in \mathcal{N}_A$. Also, let $\overrightarrow{d}, \overrightarrow{e} \in \overrightarrow{D_\mathcal{M}}$. The function $\chi : \mathbb{B}(\boldsymbol{A}) \times \overrightarrow{D_\mathcal{M}} \to B$ is true if, and only if, all actions of the multiaction parameter have the given data vector as parameter, i.e. χ is defined as follows:

$$\begin{array}{lcll} \chi([], \overrightarrow{d}) & = & t & \\ \chi([a(\overrightarrow{e})] \oplus m, \overrightarrow{d}) & = & \chi(m, \overrightarrow{d}) & \textit{if } \overrightarrow{d} = \overrightarrow{e} \\ \chi([a(\overrightarrow{e})] \oplus m, \overrightarrow{d}) & = & f & \textit{if } \overrightarrow{d} \neq \overrightarrow{e} \end{array}$$

Definition 0.2. Let $N_\mathbb{B} = \{n \mid n \in \mathbb{B}(\mathcal{N}_A) \wedge 1 < |n|\}$, $a(\overrightarrow{d}) \in \boldsymbol{A}$, $b \in N_\mathbb{B}$ and $m, n, o \in \mathbb{B}(\boldsymbol{A})$. Also let $C : N_\mathbb{B} \to (\mathcal{N}_A \cup \{\tau\})$ with $\forall_{\langle b,a\rangle,\langle c,a\rangle \in C}(\forall_{n \in b}(n \notin c))$. The *communication* function $\gamma : \mathbb{B}(\boldsymbol{A}) \times (N_\mathbb{B} \to (\mathcal{N}_A \cup \{\tau\})) \to \mathbb{B}(\boldsymbol{A})$ is defined by the following definition:

$$\begin{array}{lcll} \gamma(m \oplus n, C) & = & [a(\overrightarrow{d})] \oplus \gamma(n, C) & \exists_{\langle b,a\rangle \in C}(b = \mu(m) \wedge \chi(m, \overrightarrow{d})) \\ \gamma(m \oplus n, C) & = & \gamma(n, C) & \exists_{\langle b,\tau\rangle \in C}(b = \mu(m) \wedge \chi(m, \overrightarrow{d})) \\ \gamma(m, C) & = & m & \neg\exists_{n,o}(m = n \oplus o \wedge \exists_{c \in C}((c = \langle b, a\rangle \vee c = \langle b, \tau\rangle) \wedge b = \mu(n) \wedge \\ & & & \qquad \exists_{\overrightarrow{d} \in \overrightarrow{D}}(\chi(n, \overrightarrow{d})))) \end{array}$$

When working with open terms one encounters the problem that we may not be able to calculate the value of $\chi(m, \overrightarrow{d})$. As we wish to calculate the possible communications of a certain multiaction, given some communication function, the result will have to be a set of tuples containing a multiaction resulting from communication and a condition, with terms $\chi(m, \overrightarrow{d})$, indicating what must hold for this communication to be possible.

But first we reformulate γ to γ' as follows, because Definition 0.2 is not really suitable from a implementation point of view. Note that we somewhat ignore the possibility of right hand sides that are τ, but this is not directly relevant for the algorithms. If one desires, one can consider $[\tau(\overrightarrow{d})]$ to be equal to $[]$ to make things fit.

Definition 0.3. Let $N_\mathbb{B} = \{n \mid n \in \mathbb{B}(\mathcal{N}_A) \wedge 1 < |n|\}$, $a(\overrightarrow{d}) \in \boldsymbol{A}$, $b \in N_\mathbb{B}$ and $m, n, o \in \mathbb{B}(\boldsymbol{A})$. Also let $C : N_\mathbb{B} \to (\mathcal{N}_A \cup \{\tau\})$ with $\forall_{\langle b,a\rangle,\langle c,a\rangle \in C}(\forall_{n \in b}(n \notin c))$. The *communication* function $\gamma : \mathbb{B}(\boldsymbol{A}) \times (N_\mathbb{B} \to (\mathcal{N}_A \cup \{\tau\})) \to \mathbb{B}(\boldsymbol{A})$ is defined by the following definition:

$$\begin{array}{lcll} \gamma'([], C) & = & [] & \\ \gamma'([a(\overrightarrow{d})] \oplus m, C) & = & [a(\overrightarrow{d})] \oplus \gamma'(m, C) & \neg\exists_{n,o,\langle b,c\rangle \in C}(m = n \oplus o \wedge b = \mu([a(\overrightarrow{d})] \oplus n) \wedge \chi(n, \overrightarrow{d})) \\ \gamma'([a(\overrightarrow{d})] \oplus m, C) & = & [c(\overrightarrow{d})] \oplus \gamma'(o, C) & \exists_{n,\langle b,c\rangle \in C}(m = n \oplus o \wedge b = \mu([a(\overrightarrow{d})] \oplus n) \wedge \chi(n, \overrightarrow{d})) \end{array}$$

Lemma 0.4. Definition 0.2 and 0.3 define equivalent functions. That is, $\gamma(m, C) = \gamma'(m, C)$, for all m and C.

Proof 0.4. The defining equations of γ' are complete, so we only need to show that γ' is sound (with respect to γ). We do this by induction on m.

Case []:

$$\begin{array}{rl} & \gamma'([], C) \\ = & [] \\ = & \gamma([], C) \end{array}$$

Case $[a(\overrightarrow{d})] \oplus m$. We do case distinction on the possibility of $a(\overrightarrow{d})$ to participate in a communication. Case $\exists_{n,\langle b,a\rangle \in C}(m = n \oplus o \wedge b = \mu([a(\overrightarrow{d})] \oplus n) \wedge \chi(n, \overrightarrow{d}))$:

$$\begin{array}{rl} & \gamma'([a(\overrightarrow{d})] \oplus m, C) \\ = & [a(\overrightarrow{d})] \oplus \gamma'(o, C) \\ = & [c(\overrightarrow{d})] \oplus \gamma(o, C) \\ = & \gamma(([a(\overrightarrow{d})] \oplus n) \oplus o, C) \\ = & \gamma([a(\overrightarrow{d})] \oplus (n \oplus o), C) \\ = & \gamma([a(\overrightarrow{d})] \oplus m, C) \end{array}$$

Case $\neg\exists_{n,o,\langle b,a\rangle \in C}(m = n \oplus o \wedge b = \mu([a(\overrightarrow{d})] \oplus n) \wedge \chi(n, \overrightarrow{d}))$, with m' and m'' such that $\gamma(m, C) = m'' \oplus \gamma(m', C)$ and $\gamma(m', C) = m'$:

$$\begin{array}{rl} & \gamma'([a(\overrightarrow{d})] \oplus m, C) \\ = & [a(\overrightarrow{d})] \oplus \gamma'(m, C) \\ = & [a(\overrightarrow{d})] \oplus \gamma(m, C) \\ = & [a(\overrightarrow{d})] \oplus m'' \oplus \gamma(m', C) \\ = & [a(\overrightarrow{d})] \oplus m'' \oplus m' \\ = & m'' \oplus [a(\overrightarrow{d})] \oplus m' \\ = & m'' \oplus \gamma([a(\overrightarrow{d})] \oplus m', C) \\ = & \gamma([a(\overrightarrow{d})] \oplus m, C) \end{array}$$

□

Taking as basis the new definition, we now define the function we are really interested in. That is, the communication function on open terms. We use the set $T_{\mathbb{B}}$ of (open) boolean terms and assume that expression depending on action arguments $\overrightarrow{d}$ are such terms.

Definition 0.5. Let $\mathbb{B}(\boldsymbol{A'})$ be the set of bags of actions with open data parameters. The extension of the communication operator over open data terms $\overline{\gamma}(m, C) : \mathbb{B}(\boldsymbol{A'}) \times (N_{\mathbb{B}} \to (\mathcal{N}_{\mathcal{A}} \cup \{\tau\})) \to \mathcal{P}(\mathbb{B}(\boldsymbol{A'}) \times T_{\mathbb{B}})$ is defined as follows.

$$\begin{array}{rcl} \overline{\gamma}([], C) & = & \{\langle [], \mathit{true}\rangle\} \\ \overline{\gamma}([a(\overrightarrow{d})] \oplus m, C) & = & \{\langle r, e\rangle \mid \exists_{n,o,\langle b,c\rangle \in C, \langle r',e'\rangle \in \overline{\gamma}(o,c)}(m = n \oplus o \wedge b = \mu([a(\overrightarrow{d})] \oplus n) \wedge \\ & & \qquad (e = \chi(n, \overrightarrow{d}) \wedge e') \wedge r = [c(\overrightarrow{d})] \oplus r')\} \ \cup \\ & & \{\langle [a(\overrightarrow{d})] \oplus r, e \wedge \neg\exists_{n,o,\langle b,c\rangle \in C}(m = n \oplus o \wedge b = \mu([a(\overrightarrow{d})] \oplus n) \wedge \\ & & \qquad \chi(n, \overrightarrow{d}))\rangle \mid \langle r, e\rangle \in \overline{\gamma}(m, C)\} \end{array}$$

Theorem 0.6. Let $m \in \mathbb{B}(\boldsymbol{A'})$ and σ an assignment of variables to closed terms. Then the following holds:

$$\forall_{\langle r,e\rangle\in\overline{\gamma}(m,C)}(e\sigma \ \Rightarrow\ r\sigma = \gamma(m,C))$$

Note that we can rewrite $\neg\exists_{n,o,\langle b,c\rangle\in C}(m = n\oplus o \wedge b = \mu([a(\overrightarrow{d})]\oplus n)\wedge\chi(n,\overrightarrow{d}))$ as follows.

$$\begin{array}{ll} & \neg\exists_{n,o,\langle b,c\rangle\in C}(m = n\oplus o \wedge b = \mu([a(\overrightarrow{d})]\oplus n)\wedge\chi(n,\overrightarrow{d}))\\ \equiv & \forall_{n,o,\langle b,c\rangle\in C}(\neg(m = n\oplus o \wedge b = \mu([a(\overrightarrow{d})]\oplus n)\wedge\chi(n,\overrightarrow{d})))\\ \equiv & \forall_{n,o,\langle b,c\rangle\in C}(\neg(m = n\oplus o \wedge b = \mu([a(\overrightarrow{d})]\oplus n))\vee\neg\chi(n,\overrightarrow{d})))\\ \equiv & \forall_{n,o,\langle b,c\rangle\in C}((m = n\oplus o \wedge b = \mu([a(\overrightarrow{d})]\oplus n))\Rightarrow\neg\chi(n,\overrightarrow{d})))\end{array}$$

Definition 0.7.

$$\begin{array}{lll}\overline{\gamma}'([],C) & = & \{\langle [], true\rangle\}\\ \overline{\gamma}'([a(\overrightarrow{d})]\oplus m,C) & = & \{\langle r,e\rangle \mid \exists_{n,o,\langle b,c\rangle\in C,\langle r',e'\rangle\in\overline{\gamma}(o,c)}(m = n\oplus o\wedge b = \mu([a(\overrightarrow{d})]\oplus n)\wedge\\ & & \qquad (e = \chi(n,\overrightarrow{d})\wedge e')\wedge r = [c(\overrightarrow{d})]\oplus r')\}\end{array}$$

Lemma 0.8.

$$\overline{\gamma}([a(\overrightarrow{d})]\oplus m,C) = \overline{\gamma}'([a(\overrightarrow{d})]\oplus m,C)\cup\{\langle [a(\overrightarrow{d})]\oplus r, e\wedge\forall_{\langle r',e'\rangle\in\overline{\gamma}'([a(\overrightarrow{d})]\oplus m,C)}(\neg e')\rangle \mid \langle r,e\rangle\in\overline{\gamma}(m,C)\}$$

We now concentrate on $\overline{\gamma}'$.

Definition 0.9.

$$\begin{array}{lll}\phi(m,\overrightarrow{d},w,n,C) & = & \{\langle r,e\rangle \mid \exists_{o,o',\langle b,c\rangle\in C,\langle r',e'\rangle\in\overline{\gamma}(o'\oplus w,C)}(n = o\oplus o'\wedge b = \mu(m\oplus o)\wedge\\ & & \qquad (e = \chi(o,\overrightarrow{d})\wedge e')\wedge r = [c(\overrightarrow{d})]\oplus r')\}\end{array}$$

Lemma 0.10. $\overline{\gamma}'([a(\overrightarrow{d})]\oplus m,C) = \phi([a(\overrightarrow{d})],\overrightarrow{d},[],m,C)$

And finally with ϕ:

$$\begin{array}{ll} & \phi(m,\overrightarrow{d},w,[],C)\\ = & \{\langle r,e\rangle \mid \exists_{o,o',\langle b,c\rangle\in C,\langle r',e'\rangle\in\overline{\gamma}(o'\oplus w,C)}([] = o\oplus o'\wedge b = \mu(m\oplus o)\wedge\\ & \qquad (e = \chi(o,\overrightarrow{d})\wedge e')\wedge r = [c(\overrightarrow{d})]\oplus r')\}\\ = & \{\langle r,e\rangle \mid \exists_{\langle b,c\rangle\in C,\langle r',e'\rangle\in\overline{\gamma}([]\oplus w,C)}(b = \mu(m\oplus [])\wedge\\ & \qquad (e = \chi([],\overrightarrow{d})\wedge e')\wedge r = [c(\overrightarrow{d})]\oplus r')\}\\ = & \{\langle r,e\rangle \mid \exists_{\langle b,c\rangle\in C,\langle r',e'\rangle\in\overline{\gamma}(w,C)}(b = \mu(m)\wedge e = e'\wedge r = [c(\overrightarrow{d})]\oplus r')\}\\ = & \{\langle [c(\overrightarrow{d})]\oplus r', e'\rangle \mid \exists_{\langle b,c\rangle\in C}(b = \mu(m))\wedge\langle r',e'\rangle\in\overline{\gamma}(w,C)\}\end{array}$$

$$\begin{array}{ll} & \phi(m,\overrightarrow{d},w,[a(\overrightarrow{f})]\oplus n,C)\\ = & \{\langle r,e\rangle \mid \exists_{o,o',\langle b,a\rangle\in C,\langle r',e'\rangle\in\overline{\gamma}(o',C)}([a(\overrightarrow{f})]\oplus n = o\oplus o'\wedge b = \mu(m\oplus o)\wedge\\ & \qquad (e = \chi(o,\overrightarrow{d})\wedge e')\wedge r = [c(\overrightarrow{d})]\oplus r')\}\end{array}$$

Here $a(\overrightarrow{d})$ can be in o or in o'. Assume it is in o.

$$\begin{array}{ll} & \{\langle r,e\rangle \mid \exists_{o,o',\langle b,a\rangle\in C,\langle r',e'\rangle\in\overline{\gamma}(o'\oplus w,C)}([a(\overrightarrow{f})]\oplus n = o\oplus o'\wedge b = \mu(m\oplus o)\wedge\\ & \qquad (e = \chi(o,\overrightarrow{d})\wedge e')\wedge r = [c(\overrightarrow{d})]\oplus r')\}\\ = & \{\langle r,e\rangle \mid \exists_{o,o',\langle b,a\rangle\in C,\langle r',e'\rangle\in\overline{\gamma}(o'\oplus w,C)}(n = (o\ominus[a(\overrightarrow{f})])\oplus o'\wedge b = \mu(m\oplus[a(\overrightarrow{f})]\oplus(o\ominus[a(\overrightarrow{f})]))\wedge\\ & \qquad (e = \chi([a(\overrightarrow{f})]\oplus(o\ominus[a(\overrightarrow{f})]),\overrightarrow{d})\wedge e')\wedge r = [c(\overrightarrow{d})]\oplus r')\}\\ = & \{\langle r,e\rangle \mid \exists_{o'',o',\langle b,a\rangle\in C,\langle r',e'\rangle\in\overline{\gamma}(o'\oplus w,C)}(n = o''\oplus o'\wedge b = \mu(m\oplus[a(\overrightarrow{f})]\oplus o'')\wedge\\ & \qquad (e = (\overrightarrow{f} = \overrightarrow{d})\wedge\chi(o'',\overrightarrow{d})\wedge e')\wedge r = [c(\overrightarrow{d})]\oplus r')\}\\ = & \{\langle r,e\wedge(\overrightarrow{f} = \overrightarrow{d})\rangle \mid \exists_{o'',o',\langle b,a\rangle\in C,\langle r',e'\rangle\in\overline{\gamma}(o'\oplus w,C)}(n = o''\oplus o'\wedge b = \mu(m\oplus[a(\overrightarrow{f})]\oplus o'')\wedge\\ & \qquad (e = \chi(o'',\overrightarrow{d})\wedge e')\wedge r = [c(\overrightarrow{d})]\oplus r')\}\\ = & \{\langle r,e\wedge(\overrightarrow{f} = \overrightarrow{d})\rangle \mid \langle r,e\rangle\in\phi(m\oplus[a(\overrightarrow{f})],\overrightarrow{d},w,n,C)\}\end{array}$$

Now assume it is in o'.

$$\begin{array}{rl} & \{\langle r,e\rangle \mid \exists_{o,o',\langle b,a\rangle\in C,\langle r',e'\rangle\in\overline{\gamma}(o'\oplus w,C)}([a(\overrightarrow{f})]\oplus n = o\oplus o' \wedge b=\mu(m\oplus o)\wedge \\ & \quad (e=\chi(o,\overrightarrow{d})\wedge e')\wedge r=[c(\overrightarrow{d})]\oplus r')\} \\ = & \{\langle r,e\rangle \mid \exists_{o,o',\langle b,a\rangle\in C,\langle r',e'\rangle\in\overline{\gamma}([a(\overrightarrow{f})]\oplus(o'\ominus[a(\overrightarrow{f})])\oplus w,C)}(n = o\oplus (o'\ominus [a(\overrightarrow{f})]) \wedge b=\mu(m\oplus o)\wedge \\ & \quad (e=\chi(o,\overrightarrow{d})\wedge e')\wedge r=[c(\overrightarrow{d})]\oplus r')\} \\ = & \{\langle r,e\rangle \mid \exists_{o,o',\langle b,a\rangle\in C,\langle r',e'\rangle\in\overline{\gamma}([a(\overrightarrow{f})]\oplus o'\oplus w,C)}(n = o\oplus o') \wedge b=\mu(m\oplus o)\wedge \\ & \quad (e=\chi(o,\overrightarrow{d})\wedge e')\wedge r=[c(\overrightarrow{d})]\oplus r')\} \\ = & \phi(m,\overrightarrow{d},w\oplus[a(\overrightarrow{f})],n,C) \end{array}$$

To conclude, we write an algorithm that uses what we have proven.

$$\begin{array}{lll} \overline{\gamma}(m,C) = |[& \textbf{var}\ S,T:\mathcal{P}(\mathbb{B}(\boldsymbol{A'})\times T_{\mathbb{B}});\ \textbf{var}\ b:T_{\mathbb{B}}; & \\ | & & \\ & \textbf{if}\ \ m=[] & \rightarrow\ S:=\{\langle [], \mathit{true}\rangle\} \\ & [\!]\ \ m=[a(\overrightarrow{d})]\oplus n & \rightarrow\ S,T:=\phi([a(\overrightarrow{d})],\overrightarrow{d},[],n,C),\overline{\gamma}(n,C) \\ & & ;\ b:=\forall_{\langle r,e\rangle\in S}(\neg e) \\ & & ;\ S:=S\cup\{\langle [a(\overrightarrow{d})]\oplus r, e\wedge b\rangle \mid \langle r,e\rangle\in T\} \\ & \textbf{fi} & \\ ; & \textbf{return}\ S & \\ ]| & & \end{array}$$

$$\begin{array}{llll} \phi(m,\overrightarrow{d},w,n,C) = |[& \textbf{var}\ S,T:\mathcal{P}(\mathbb{B}(\boldsymbol{A'})\times T_{\mathbb{B}}); & & \\ | & & & \\ & \textbf{if}\ \ n=[] & \rightarrow\ \textbf{if}\ \ \exists_{\langle b,c\rangle\in C}(b=\mu(m)) & \rightarrow\ T:=\overline{\gamma}(w,C) \\ & & & ;\ S:=\{\langle [c(\overrightarrow{d})]\oplus r,e\rangle \mid \langle r,e\rangle\in T\} \\ & & \quad [\!]\ \ \neg\exists_{\langle b,c\rangle\in C}(b=\mu(m)) & \rightarrow\ S:=\emptyset \\ & & \quad \textbf{fi} & \\ & [\!]\ \ n=[a(\overrightarrow{f})]\oplus o & \rightarrow\ T:=\phi(m\oplus[a(\overrightarrow{f})],\overrightarrow{d},w,o,C) & \\ & & ;\ T:=\{\langle r,e\wedge(\overrightarrow{f}=\overrightarrow{d})\rangle \mid \langle r,e\rangle\in T\} & \\ & & ;\ S:=T\cup\phi(m,\overrightarrow{d},w\oplus[a(\overrightarrow{f})],o,C) & \\ & \textbf{fi} & & \\ ; & \textbf{return}\ S & & \\ ]| & & & \end{array}$$

If we analyse this algorithm focussing on the lenght of input m, we see that it is $O(2^{|m|})$. It basically takes the first action in m and computes the result given that this action participates in a communication and the result given that it does not.

However, looking at ϕ, we can see that the algorithm needlessly tries to find a part in n, such that m with this part can communicate, if m is not even a subbag of a left hand side of a communication in C. So, we propose to add an extra check to ϕ to prevent this behaviour and making the algorithm more (or precisely) in the order of $O(2^{|m_1|} + |m_2|)$, with $m = m_1 \oplus m_2$ and m_1 contains actions that occur in a left hand side of a communication in C and m_2 actions that do not.

$$
\begin{array}{l}
\phi(m, \overrightarrow{d}, w, n, C) = |[\ \ \textbf{var}\ S, T : \mathcal{P}(\mathbb{B}(\boldsymbol{A'}) \times T_{\mathbb{B}});\\
\quad\quad \textbf{var}\ b : \textbf{bool};\\
\quad |\\
\quad\quad b := \exists_{o,c}(\langle \mu(m) \oplus o, c\rangle \in C)\\
\quad ;\ \textbf{if}\ \neg b \rightarrow S := \emptyset\\
\quad\quad [\!]\ b \wedge n = [] \rightarrow \textbf{if}\ \exists_{\langle b,c\rangle \in C}(b = \mu(m)) \rightarrow T := \overline{\gamma}(w, C)\\
\quad\quad\quad\quad ;\ S := \{\langle [c(\overrightarrow{d})] \oplus r, e\rangle \mid \langle r, e\rangle \in T\}\\
\quad\quad\quad [\!]\ \neg\exists_{\langle b,c\rangle \in C}(b = \mu(m)) \rightarrow S := \emptyset\\
\quad\quad\quad \textbf{fi}\\
\quad\quad [\!]\ b \wedge n = [a(\overrightarrow{f})] \oplus o \rightarrow T := \phi(m \oplus [a(\overrightarrow{f})], \overrightarrow{d}, w, o, C)\\
\quad\quad\quad ;\ T := \{\langle r, e \wedge (\overrightarrow{f} = \overrightarrow{d})\rangle \mid \langle r, e\rangle \in T\}\\
\quad\quad\quad ;\ S := T \cup \phi(m, \overrightarrow{d}, w \oplus [a(\overrightarrow{f})], o, C)\\
\quad\quad \textbf{fi}\\
\quad ;\ \textbf{return}\ S\\
\quad]|
\end{array}
$$

Another problem with the above code is that it can generate a lot of negative conditions to indicate that certain communication do not happen. This appears to be at least exponential.

We solve this by removing the problematic $\forall$ in $\overline{\gamma}$. Instead we add an extra parameter to $\overline{\gamma}$ and ϕ indicating which actions will not communicate. Then, in the final case of $\overline{\gamma}$, where $m = []$, we use a new function ψ to calculate a more reasonable condition indicating that the remaining actions do not communicate.

Note that the following algorithm deviates in a significant way of the previous version, which means that it validity is not guaranteed and additional proofs will be needed.

$$
\begin{array}{l}
\overline{\gamma}(m, C, r) = |[\ \ \textbf{var}\ S, T : \mathcal{P}(\mathbb{B}(\boldsymbol{A'}) \times T_{\mathbb{B}});\\
\quad |\\
\quad\quad \textbf{if}\ m = [] \rightarrow S := \{\langle r, \psi(r, C)\rangle\}\\
\quad\quad [\!]\ m = [a(\overrightarrow{d})] \oplus n \rightarrow S, T := \phi([a(\overrightarrow{d})], \overrightarrow{d}, [], n, C, r), \overline{\gamma}(n, C, [a(\overrightarrow{d})] \oplus r)\\
\quad\quad\quad ;\ S := S \cup T\\
\quad\quad \textbf{fi}\\
\quad ;\ \textbf{return}\ S\\
\quad]|
\end{array}
$$

$$
\begin{array}{l}
\phi(m, \overrightarrow{d}, w, n, C, r) = |[\ \ \textbf{var}\ S, T : \mathcal{P}(\mathbb{B}(\boldsymbol{A'}) \times T_{\mathbb{B}});\\
\quad\quad \textbf{var}\ b : \textbf{bool};\\
\quad |\\
\quad\quad b := \exists_{o,c}(\langle \mu(m) \oplus o, c\rangle \in C)\\
\quad ;\ \textbf{if}\ \neg b \rightarrow S := \emptyset\\
\quad\quad [\!]\ b \wedge n = [] \rightarrow \textbf{if}\ \exists_{\langle b,c\rangle \in C}(b = \mu(m)) \rightarrow T := \overline{\gamma}(w, C, r)\\
\quad\quad\quad\quad ;\ S := \{\langle [c(\overrightarrow{d})] \oplus r, e\rangle \mid \langle r, e\rangle \in T\}\\
\quad\quad\quad [\!]\ \neg\exists_{\langle b,c\rangle \in C}(b = \mu(m)) \rightarrow S := \emptyset\\
\quad\quad\quad \textbf{fi}\\
\quad\quad [\!]\ b \wedge n = [a(\overrightarrow{f})] \oplus o \rightarrow T := \phi(m \oplus [a(\overrightarrow{f})], \overrightarrow{d}, w, o, C, r)\\
\quad\quad\quad ;\ T := \{\langle r, e \wedge (\overrightarrow{f} = \overrightarrow{d})\rangle \mid \langle r, e\rangle \in T\}\\
\quad\quad\quad ;\ S := T \cup \phi(m, \overrightarrow{d}, w \oplus [a(\overrightarrow{f})], o, C, r)\\
\quad\quad \textbf{fi}\\
\quad ;\ \textbf{return}\ S\\
\quad]|
\end{array}
$$

$$
\begin{array}{lll}
\psi(m,C) = |[& \textbf{var}\ b : T_{\mathbb{B}}; & \\
| & & \\
 & \textbf{if}\ \ m = [] & \rightarrow\ b := \mathit{true} \\
 & []\ \ m = [a(\overrightarrow{d})] \oplus n & \rightarrow\ b := \psi'(a(\overrightarrow{d}), n, C) \wedge \psi(n, C) \\
 & \textbf{fi} & \\
; & \textbf{return}\ b & \\
]| & &
\end{array}
$$

$$
\begin{array}{llll}
\psi'(a(\overrightarrow{d}), m, C) = |[& \textbf{var}\ b : T_{\mathbb{B}}; & & \\
 & \textbf{var}\ c : \textbf{bool}; & & \\
| & & & \\
 & \textbf{if}\ \ m = [] & \rightarrow\ b := \mathit{true} & \\
 & []\ \ m = [b(\overrightarrow{e})] \oplus n & \rightarrow\ c := \exists_{o,d}(\langle [a,b] \oplus o, d\rangle \in C) & \\
 & & ;\ \textbf{if}\ \ c \wedge \xi([a(\overrightarrow{d}), b(\overrightarrow{e})], n, C) & \rightarrow\ b := \psi'(a(\overrightarrow{d}), n, C) \wedge (\overrightarrow{d} \neq \overrightarrow{e}) \\
 & & \ \ []\ \ \neg c \vee \neg\xi([a(\overrightarrow{d}), b(\overrightarrow{e})], n, C) & \rightarrow\ b := \psi'(a(\overrightarrow{d}), n, C) \\
 & & \ \ \textbf{fi} & \\
 & \textbf{fi} & & \\
; & \textbf{return}\ b & & \\
]| & & &
\end{array}
$$

$$
\begin{array}{llll}
\xi(m,n,C) = |[& \textbf{var}\ b : \textbf{bool}; & & \\
| & & & \\
 & \textbf{if}\ \ n = [] & \rightarrow\ b := \exists_d(\langle m, d\rangle \in C) & \\
 & []\ \ n = [a(\overrightarrow{d})] \oplus o & \rightarrow\ \textbf{if}\ \ \exists_d(\langle [a] \oplus m, d\rangle \in C) & \rightarrow\ b := \mathit{true} \\
 & & \ \ \ \ []\ \ \exists_{b,o',d}(\langle [a,b] \oplus m \oplus o', d\rangle \in C) & \rightarrow\ b := \xi([a] \oplus m, o, C) \vee \xi(m, o, C) \\
 & & \ \ \ \ []\ \ \neg\exists_{o',d}(\langle [a] \oplus m \oplus o', d\rangle \in C) & \rightarrow\ b := \xi(m, o, C) \\
 & & \ \ \ \ \textbf{fi} & \\
 & \textbf{fi} & & \\
; & \textbf{return}\ b & & \\
]| & & &
\end{array}
$$

Naturally, functions ψ and ψ' can easily be transformed to the following non-recursive implementation.

$$
\begin{array}{lll}
\psi(m,C) = |[& \textbf{var}\ b : T_{\mathbb{B}}; & \\
| & & \\
 & b := \mathit{true} & \\
; & \textbf{do}\ \ m = [a(\overrightarrow{d})] \oplus n & \rightarrow\ b, m := b \wedge \psi'(a(\overrightarrow{d}), n, C), n \\
 & \textbf{od} & \\
; & \textbf{return}\ b & \\
]| & &
\end{array}
$$

$$
\begin{array}{lll}
\psi'(a(\overrightarrow{d}), m, C) = |[& \mathbf{var}\ b : T_{\mathbb{B}}; & \\
 & \mathbf{var}\ c : \mathbf{bool}; & \\
| & & \\
 & b := \mathit{true} & \\
; & \mathbf{do}\ \ m = [b(\overrightarrow{e})] \oplus n \ \rightarrow & c := \exists_{o,d}(\langle [a,b] \oplus o, d\rangle \in C) \\
 & & ;\ \mathbf{if}\ \ c \wedge \xi([a(\overrightarrow{d}), b(\overrightarrow{e})], n, C) \quad \rightarrow\ b := b \wedge (\overrightarrow{d} \neq \overrightarrow{e}) \\
 & & \quad [\!]\ \ \neg c \vee \neg \xi([a(\overrightarrow{d}), b(\overrightarrow{e})], n, C) \rightarrow\ \mathbf{skip} \\
 & & \quad \mathbf{fi} \\
 & & ;\ m := n \\
 & \mathbf{od} & \\
; & \mathbf{return}\ b & \\
]| & &
\end{array}
$$

Theorem 0.11.

$$\overline{\gamma}(m, C, r) = \{\langle r \oplus r', e \wedge \psi(r, C)\rangle \ :\ \langle r', e\rangle \in \overline{\gamma}(m, C)\}$$

Proof 0.11.

$\{\langle r \oplus r', e \wedge \psi(r, C)\rangle \ :\ \langle r', e\rangle \in \overline{\gamma}([], C)\}$

$=\quad \{\ \}$

$\{\langle r \oplus r', e \wedge \psi(r, C)\rangle \ :\ \langle r', e\rangle \in \{\langle []\rangle \mathit{true}\}\}$

$=\quad \{\ \}$

$\{\langle r \oplus [], \mathit{true} \wedge \psi(r, C)\rangle\}$

$=\quad \{\ \}$

$\{\langle r, \psi(r, C)\rangle\}$

$=\quad \{\ \}$

$\overline{\gamma}([], C, r)$

$\{\langle r \oplus r', e \wedge \psi(r, C)\rangle \ :\ \langle r', e\rangle \in \overline{\gamma}([a(\overrightarrow{d})] \oplus m, C)\}$

$=\quad \{\ \}$

$\{\langle r \oplus r', e \wedge \psi(r, C)\rangle \ :\ \langle r', e\rangle \in \overline{\gamma}'([a(\overrightarrow{d})] \oplus m, C) \cup \{\langle [a(\overrightarrow{d})] \oplus r, e \wedge \forall_{\langle r', e'\rangle \in \overline{\gamma}'([a(\overrightarrow{d})] \oplus m, C)}(\neg e')\rangle \mid \langle r, e\rangle \in \overline{\gamma}(m$

$=\quad \{\ \}$

$\{\langle r \oplus r', e \wedge \psi(r, C)\rangle \ :\ \langle r', e\rangle \in \phi([a(\overrightarrow{d})], \overrightarrow{d}, [], m, C) \cup \{\langle [a(\overrightarrow{d})] \oplus r, e \wedge \forall_{\langle r', e'\rangle \in \phi([a(\overrightarrow{d})], \overrightarrow{d}, [], m, C)}(\neg e')\rangle \mid \langle r, e\rangle \in$

$=\quad \{\ \}$

$\{\langle r \oplus r', e \wedge \psi(r, C)\rangle \ :\ \langle r', e\rangle \in \phi([a(\overrightarrow{d})], \overrightarrow{d}, [], m, C)\} \cup \{\langle r \oplus r', e \wedge \psi(r, C)\rangle \ :\ \langle r', e\rangle \in \{\langle [a(\overrightarrow{d})] \oplus r, e \wedge \forall_{\langle r', e}$

$=\quad \{\ \}$

$\phi([a(\overrightarrow{d})], \overrightarrow{d}, [], m, C, r) \cup \{\langle r \oplus r', e \wedge \psi(r, C)\rangle \ :\ \langle r', e\rangle \in \{\langle [a(\overrightarrow{d})] \oplus r, e \wedge \forall_{\langle r', e'\rangle \in \phi([a(\overrightarrow{d})], \overrightarrow{d}, [], m, C)}(\neg e')\rangle \mid \langle r, e\rangle$

$=\quad \{\ \}$

$\phi([a(\overrightarrow{d})], \overrightarrow{d}, [], m, C, r) \cup \{\langle r \oplus [a(\overrightarrow{d})] \oplus r', e \wedge \forall_{\langle r', e'\rangle \in \phi([a(\overrightarrow{d})], \overrightarrow{d}, [], m, C)}(\neg e') \wedge \psi(r, C)\rangle \ :\ \langle r', e\rangle \in \overline{\gamma}(m, C)\}$

$=\quad \{\ \mathrm{X}\ \}$

$\phi([a(\overrightarrow{d})], \overrightarrow{d}, [], m, C, r) \cup \{\langle [a(\overrightarrow{d})] \oplus r \oplus r', e \wedge \psi([a(\overrightarrow{d})] \oplus r, C)\rangle \ :\ \langle r', e\rangle \in \overline{\gamma}(m, C)\}$

$=\quad \{\ \}$

$\phi([a(\overrightarrow{d})], \overrightarrow{d}, [], m, C, r) \cup \overline{\gamma}(m, C, [a(\overrightarrow{d})] \oplus r)$

$=\quad \{\ \}$

$\overline{\gamma}([a(\overrightarrow{d})] \oplus m, C, r)$

□

Corollary 0.12.

$$\overline{\gamma}(m, C) = \overline{\gamma}(m, C, [])$$

Proof 0.12.

$$\begin{array}{ll} & \overline{\gamma}(m, C, []) \\ = & \quad \{ \quad \} \\ & \{\langle [] \oplus r', e \wedge \psi([], C)\rangle \ : \ \langle r', e\rangle \in \overline{\gamma}(m, C)\} \\ = & \quad \{ \quad \} \\ & \{\langle r', e \wedge \mathit{true}\rangle \ : \ \langle r', e\rangle \in \overline{\gamma}(m, C)\} \\ = & \quad \{ \quad \} \\ & \{\langle r', e\rangle \ : \ \langle r', e\rangle \in \overline{\gamma}(m, C)\} \\ = & \quad \{ \quad \} \\ & \overline{\gamma}(m, C) \end{array}$$

□

References

[1] M.J. van Weerdenburg, *GenSpect Process Algebra*, Master's thesis, Eindhoven University of Technology, 2004