Appendix C

Refinement Laws

Simulation

Law C.1 (*Skip*)

$$Skip \preceq Skip$$

Law C.2 (*Stop*)

$$Stop \preceq Stop$$

Law C.3 (*Chaos*)

$$Chaos \preceq Chaos$$

Law C.4 (Schema expressions)

$$ASExp \preceq CSExp$$

provided

- ➯ $\forall P_1.State;\ P_2.State;\ L \bullet R \land \text{pre } ASExp \Rightarrow \text{pre } CSExp$
- ➯ $\forall P_1.State;\ P_2.State;\ P_2.State';\ L \bullet R \land \text{pre } ASExp \land CSExp \Rightarrow$ $(\exists P_1.State';\ L' \bullet R' \land ASExp)$ ❑

Law C.5 (Prefix distribution*)

$$c \rightarrow A_1 \preceq c \rightarrow A_2$$

provided $A_1 \preceq A_2$ ❑

Law C.6 (Simple prefix distribution*)

$$c.ae \rightarrow Skip \preceq c.ce \rightarrow Skip$$

provided

➪ $\forall P_1.State;\ P_2.State;\ L \bullet R \Rightarrow ae = ce$ ❑

Law C.7 (Output prefix distribution)

$$c!ae \rightarrow A_1 \preceq c!ce \rightarrow A_2$$

provided

➪ $\forall P_1.State;\ P_2.State;\ L \bullet R \Rightarrow ae = ce$

➪ $A_1 \preceq A_2$ ❑

Law C.8 (Input prefix distribution)

$$c?x \rightarrow A_1 \preceq c?x \rightarrow A_2$$

provided $A_1 \preceq A_2$ ❑

Law C.9 (Input constrained prefix distribution*)

$$c?x : T_1 \rightarrow A_1 \preceq c?x : T_1 \rightarrow A_2$$

provided

➪ $A_1 \preceq A_2$

➪ $\forall A_1.State;\ A_2.State;\ L \bullet R \Rightarrow (T_1 \Leftrightarrow T_2)$ ❑

Law C.10 (Multiple prefix distribution*)
For every channel c and communication parameters as and cs,

$$c\ as \rightarrow A_1 \preceq c\ cs \rightarrow A_2$$

provided

➪ $A_1 \preceq A_2$

➪ *For every abstract expression* e_{a_i} *in as and its corresponding concrete expression* e_{c_i} *in cs:* $\forall P_1.State;\ P_2.State;\ L \bullet R \Rightarrow (e_{a_i} \Leftrightarrow e_{a_i})$

➪ *The names of all input variables are not changed from as to cs.*

➪ *Type of c is finite.* ❑

Law C.11 (Guard distribution)

$$ag \ \& \ A_1 \preceq cg \ \& \ A_2$$

provided

- ➩ $\forall P_1.State; \ P_2.State; \ L \bullet R \Rightarrow (ag \Leftrightarrow cg)$
- ➩ $A_1 \preceq A_2$ ❑

Law C.12 (Sequence distribution)

$$A_1; \ A_2 \preceq B_1; \ B_2$$

provided

- ➩ $A_1 \preceq B_1$
- ➩ $A_2 \preceq B_2$ ❑

Law C.13 (External choice distribution*)

$$A_1 \Box A_2 \preceq B_1 \Box B_2$$

provided

- ➩ $A_1 \preceq B_1$
- ➩ $A_2 \preceq B_2$
- ➩ *R is a function from the concrete to the abstract state* ❑

Law C.14 (External choice/Prefix distribution*)

$$\Box \ i \bullet c_i \rightarrow A_i \preceq \Box \ i \bullet c_i \rightarrow B_i$$

provided $\forall i \bullet A_i \preceq B_i$ ❑

Law C.15 (External choice/Simple prefix distribution*)

$$\Box \ i \bullet c_i.ae_i \rightarrow A_i \preceq \Box \ i \bullet c_i.ce_i \rightarrow B_i$$

provided

- ➩ $\forall i \bullet A_i \preceq B_i$
- ➩ $\forall i \bullet \forall P_1.State; \ P_2.State; \ L \bullet R \Rightarrow ae_i = ce_i$ ❑

Law C.16 (External choice/Output Prefix distribution*)

$$\Box\, i \bullet c_i!ae_i \rightarrow A_i \preceq \Box\, i \bullet c_i!ce_i \rightarrow B_i$$

provided

- $\forall\, i \bullet A_i \preceq B_i$
- $\forall\, i \bullet \forall\, P_1.State;\ P_2.State;\ L \bullet R \Rightarrow ae_i = ce_i$ ❑

Law C.17 (External choice/Input Prefix distribution*)

$$\Box\, i \bullet c_i?x_i \rightarrow A_i \preceq \Box\, i \bullet c_i?x_i \rightarrow B_i$$

provided $\forall\, i \bullet A_i \preceq B_i$ ❑

Law C.18 (External choice/Constrained Input Prefix distribution*)

$$\Box\, i \bullet c_i?x_i : T_{A_i} \rightarrow A_i \preceq \Box\, i \bullet (c_i?x_i : T_{B_i} \rightarrow B_i$$

provided

- $\forall\, i \bullet A_i \preceq B_i$
- $\forall\, i \bullet \forall\, A.State;\ B.State;\ L \bullet R \Rightarrow (T_{A_i} \Leftrightarrow T_{B_i})$ ❑

Law C.19 (External choice/Multiple Prefix distribution*)
For every channel c_i and communication parameters as_i, and cs_i,

$$\Box\, i \bullet c_i\ as_i \rightarrow A_i \preceq \Box\, i \bullet c_i\ cs_i \rightarrow B_i$$

provided

- *Type of c is finite*
- $\forall\, i \bullet A_i \preceq B_i$
- *For every i, and every abstract expression e_a in as_i and its corresponding concrete expression*
  e_c in cs_i: $\forall\, P_1.State;\ P_2.State;\ L \bullet R \Rightarrow e_a \Leftrightarrow e_c$
- *For every i, the names of all input variables are not changed neither from as_i to cs_i* ❑

Law C.20 (Internal choice distribution*)

$$A_1 \sqcap A_2 \preceq B_1 \sqcap B_2$$

provided

- $A_1 \preceq A_2$
- $B_1 \preceq B_2$ ❑

Law C.21 (Parallelism composition distribution*)

$$A_1 \llbracket ns_{1_A} \mid cs \mid ns_{2_A} \rrbracket A_2 \preceq B_1 \llbracket ns_{1_B} \mid cs \mid ns_{2_B} \rrbracket B_2$$

provided

- $A_1 \preceq B_1$
- $A_2 \preceq B_2$
- $\forall v_A, v_B \bullet R(v_A, v_B) \Rightarrow (v_A \in ns_{1_A} \Rightarrow v_B \in ns_{1_B})$
- $\forall v_A, v_B \bullet R(v_A, v_B) \Rightarrow (v_A \in ns_{2_A} \Rightarrow v_B \in ns_{2_B})$ ❑

Law C.22 (Interleave distribution*)

$$A_1 \interleave\![ns_1 \mid ns_2]\!\interleave A_2 \preceq B_1 \interleave\![ns_1 \mid ns_2]\!\interleave B_2$$

provided

- $A_1 \preceq A_2$
- $B_1 \preceq B_2$
- $\forall v_A, v_B \bullet R(v_A, v_B) \Rightarrow (v_A \in ns_{1_A} \Rightarrow v_B \in ns_{1_B})$
- $\forall v_A, v_B \bullet R(v_A, v_B) \Rightarrow (v_A \in ns_{2_A} \Rightarrow v_B \in ns_{2_B})$ ❑

Law C.23 (Recursion distribution*)

$$\mu X \bullet F_A(X) \preceq \mu X \bullet F_C(X)$$

provided $F_A \preceq F_C$ ❑

Law C.24 (Specification Statement Distribution*)

$$w_A : [pre_A, post_A] \preceq w_B : [pre_B, post_B]$$

provided

- $\neg\, pre_A \preceq \neg\, pre_B$
- $post_A \land u'_A = u_A \preceq post_B \land u'_B = u_B$, *where u are the state variables that are not in the frame w.* ❑

Law C.25 (Variable Block Distribution*)

$$\mathbf{var}\ x \bullet A_1 \preceq \mathbf{var}\ x \bullet A_2$$

provided

$A_1 \preceq A_2$ ❑

Action Refinement

Assumptions

Law C.26 (Assumption Conjunction*)

$$\{g_1\};\ \{g_2\} = \{g_1 \land g_2\}$$

Law C.27 (Assumption introduction*)

$$\{g\} = \{g\};\ \{g_1\}$$

provided $g \Rightarrow g_1$ ❑

In the following two laws we refer to a predicate $assump'$. In general, for any predicate p, the predicate p' is formed by dashing all its free undecorated variables.

Law C.28 (Schema Expression/Assumption—introduction)

$$[\Delta State;\ i? : T_i;\ o! : T_o \mid p \land assump']$$
$$=$$
$$[\Delta State;\ i? : T_i;\ o! : T_o \mid p \land assump'];\ \{assump\}$$

The schema in this law is an arbitrary schema that specifies an action in *Circus*: it acts on a state schema *State* and, optionally, has input variables $i?$ of type T_i, and output variables $o!$ of type T_o.

Law C.29 (Initialisation schema/Assumption—introduction*)

$$[State' \mid p \land assump']$$
$$=$$
$$[State' \mid p \land assump'];\ \{assump\}$$

Law C.30 (Assumption/Guard—introduction)

$$\{\, g \,\};\ A = \{\, g \,\};\ g \,\&\, A$$

Law C.31 (Guard/Assumption—introduction 1*)

$$g \,\&\, A = g \,\&\, \{g\};\ A$$

Law C.32 (Assumption/Guard—elimination 1)

$$\{\, g_1 \,\};\ (g_2 \,\&\, A) = \{\, g_1 \,\};\ A$$

provided $g_1 \Rightarrow g_2$ ❑

Law C.33 (Assumption/Guard—elimination 2)

$$\{ g_1 \};\ (g_2 \,\&\, A) = \{ g_1 \};\ Stop$$

provided $g_1 \Rightarrow \neg\, g_2$ ❑

Law C.34 (Assumption/Guard—replacement)

$$\{ g_1 \};\ (g_2 \,\&\, A) = \{ g_1 \};\ (g_3 \,\&\, A)$$

provided $g_1 \Rightarrow (g_2 \Leftrightarrow g_3)$ ❑

Law C.35 (Assumption elimination)

$$\{p\} \sqsubseteq_{\mathcal{A}} Skip$$

Law C.36 (Assumption substitution 1*)

$$\{g_1\} \sqsubseteq_{\mathcal{A}} \{g_2\}$$

provided $g_1 \Rightarrow g_2$ ❑

Law C.37 (Assumption/External choice—distribution)

$$\{p\};\ (A_1 \Box A_2) = (\{p\};\ A_1) \Box (\{p\};\ A_2)$$

Law C.38 (Assumption/Parallelism composition—distribution)

$$\{p\};\ (A_1 \llbracket ns_1 \mid cs \mid ns_2 \rrbracket A_2) = (\{p\};\ A_1) \llbracket ns_1 \mid cs \mid ns_2 \rrbracket (\{p\};\ A_2)$$

Law C.39 (Assumption/Interleaving—distribution)

$$\{p\};\ (A_1 \, |\!|\!|[ns_1 \mid ns_2]|\!|\!| \, A_2) = (\{p\};\ A_1) \, |\!|\!|[ns_1 \mid ns_2]|\!|\!| \, (\{p\};\ A_2)$$

Law C.40 (Assumption/Mutual recursion—distribution*)

$$\{g\};\ \mu\, X_1, \ldots, X_i, \ldots, X_n \bullet \left\langle \begin{array}{l} F_1(X_1, \ldots, X_i, \ldots, X_n), \ldots, \\ F_i(X_1, \ldots, X_i, \ldots, X_n), \ldots, \\ F_n(X_1, \ldots, X_i, \ldots, X_n) \end{array} \right\rangle$$

$$\sqsubseteq_{\mathcal{A}}$$

$$\mu\, X_1, \ldots, X_i, \ldots, X_n \bullet \left\langle \begin{array}{l} F_1(X_1, \ldots, X_i, \ldots, X_n), \ldots, \\ \{g\};\ F_i(X_1, \ldots, X_i, \ldots, X_n), \ldots, \\ F_n(X_1, \ldots, X_i, \ldots, X_n) \end{array} \right\rangle$$

provided *for all* j, *such that* $1 \leq j \leq n$,

$$\{g\};\ F_j(X_1, \ldots, X_i, \ldots, X_n) \sqsubseteq_{\mathcal{A}} F_j(\{g\};\ X_1, \ldots, \{g\};\ X_i, \ldots, \{g\};\ X_n),$$

❑

Law C.41 (Assumption/Prefix—distribution*)

$$\{g\};\ c \to A \sqsubseteq_{\mathcal{A}} c \to \{g\};\ A$$

Law C.42 (Assumption/Prefix—distribution 2*)

$$\{g\};\ c \to A = \{g\};\ c \to \{g\};\ A$$

Law C.43 (Assumption/Simple Prefix—distribution*)

$$\{g\};\ c.e \to A \sqsubseteq_{\mathcal{A}}\ ;\ c.e \to \{g\};\ A$$

Law C.44 (Assumption/Simple Prefix—distribution 2*)

$$\{g\};\ c.e \to A = \{g\};\ c.e \to \{g\};\ A$$

Law C.45 (Assumption/Output prefix—distribution*)

$$\{g\};\ c!x \to A \sqsubseteq_{\mathcal{A}} c!x \to \{g\};\ A$$

Law C.46 (Assumption/Output prefix—distribution 2*)

$$\{g\};\ c!x \to A = \{g\};\ c!x \to \{g\};\ A$$

Law C.47 (Assumption/Input prefix—distribution*)

$$\{g\};\ c?x \to A \sqsubseteq_{\mathcal{A}} c?x \to \{g\};\ A$$

provided $x \notin FV(g)$ ❑

Law C.48 (Assumption/Input Prefix—distribution 2*)

$$\{g\};\ c?x \rightarrow A = \{g\};\ c?x \rightarrow \{g\};\ A$$

provided $x \notin FV(g)$ ❑

Law C.49 (Assumption/Constrained Input prefix—distribution*)

$$\{g\};\ c?x : T \rightarrow A \sqsubseteq_{\mathcal{A}} c?x : T \rightarrow \{g\};\ A$$

provided $x \notin FV(g)$ ❑

Law C.50 (Assumption/Constrained Input Prefix—distribution 2*)

$$\{g\};\ c?x : T \rightarrow A = \{g\};\ c?x : T \rightarrow \{g\};\ A$$

provided $x \notin FV(g)$ ❑

Law C.51 (Assumption/Multiple prefix—distribution*)
For every channel c and communication parameters as,

$$\{g\};\ c\ as \rightarrow A \sqsubseteq_{\mathcal{A}} c\ as \rightarrow \{g\};\ A$$

provided

➩ *The names of all input variables are not free in g.* ❑

Law C.52 (Assumption/Multiple Prefix—distribution 2*)
For every channel c and communication parameters as,

$$\{g\};\ c\ as \rightarrow A = c\ as \rightarrow \{g\};\ A$$

provided

➩ *The names of all input variables are not free in g.* ❑

Law C.53 (Assumption/Schema—distribution*)

$$\{g\};\ [decl \mid p] \sqsubseteq_{\mathcal{A}} [decl \mid p];\ \{g\}$$

provided $g \wedge p \Rightarrow g'$ ❑

Law C.54 (Assumption/Assignment—distribution*)

$$\{g\};\ x := e = \{g\};\ x := e;\ \{g\}$$

provided $x \notin FV(g)$ ❑

Law C.55 (Assumption Unit*)

$$\{true\} = Skip$$

Law C.56 (Assumption Zero*)

$$\{false\} = Chaos$$

Guards

Law C.57 (Guard combination)

$$g_1 \mathbin{\&} (g_2 \mathbin{\&} A) = (g_1 \wedge g_2) \mathbin{\&} A$$

Law C.58 (Guards expansion*)

$$(g_1 \vee g_2) \mathbin{\&} A = g_1 \mathbin{\&} A \,\Box\, g_2 \mathbin{\&} A$$

Law C.59 (Guard/Sequence—associativity)

$$(g \mathbin{\&} A_1);\ A_2 = g \mathbin{\&} (A_1;\ A_2)$$

Law C.60 (Guard/External choice—distribution)

$$g \mathbin{\&} (A_1 \,\Box\, A_2) = (g \mathbin{\&} A_1) \,\Box\, (g \mathbin{\&} A_2)$$

Law C.61 (Guard/Internal choice—distribution)

$$g \mathbin{\&} (A_1 \sqcap A_2) = (g \mathbin{\&} A_1) \sqcap (g \mathbin{\&} A_2)$$

Law C.62 (Guard/Parallelism composition—distribution 1)

$$g \mathbin{\&} (A_1 \llbracket ns_1 \mid cs \mid ns_2 \rrbracket A_2) = (g \mathbin{\&} A_1) \llbracket ns_1 \mid cs \mid ns_2 \rrbracket (g \mathbin{\&} A_2)$$

Law C.63 (Guard/Parallelism composition—distribution 2)

$$(g_1 \mathbin{\&} A_1) \mathbin{[\![} ns_1 \mid cs \mid ns_2 \mathbin{]\!]} (g_2 \mathbin{\&} A_2)$$
$$=$$
$$(g_1 \lor g_2) \mathbin{\&} ((g_1 \mathbin{\&} A_1) \mathbin{[\![} ns_1 \mid cs \mid ns_2 \mathbin{]\!]} (g_2 \mathbin{\&} A_2))$$

provided

➩ $g_1 \Leftrightarrow g_2$ ❑

Law C.64 (Guards/Parallelism composition—distribution 3*)

$$(g_1 \land g_2) \mathbin{\&} (A_1 \mathbin{[\![} ns_1 \mid cs \mid ns_2 \mathbin{]\!]} A_2)$$
$$=$$
$$(g_1 \mathbin{\&} A_1) \mathbin{[\![} ns_1 \mid cs \mid ns_2 \mathbin{]\!]} (g_2 \mathbin{\&} A_2)$$

provided

➩ $g_1 \Leftrightarrow g_2$ ❑

Law C.65 (Guard/Interleaving—distribution 1)

$$g \mathbin{\&} (A_1 \mathbin{|\!|\![} ns_1 \mid ns_2 \mathbin{]\!|\!|} A_2) = (g \mathbin{\&} A_1) \mathbin{|\!|\![} ns_1 \mid ns_2 \mathbin{]\!|\!|} (g \mathbin{\&} A_2)$$

Law C.66 (Guard/Interleaving—distribution 2)

$$(g_1 \mathbin{\&} A_1) \mathbin{|\!|\![} ns_1 \mid ns_2 \mathbin{]\!|\!|} (g_2 \mathbin{\&} A_2)$$
$$=$$
$$(g_1 \lor g_2) \mathbin{\&} ((g_1 \mathbin{\&} A_1) \mathbin{|\!|\![} ns_1 \mid ns_2 \mathbin{]\!|\!|} (g_2 \mathbin{\&} A_2))$$

Law C.67 (True guard)

$$true \mathbin{\&} A = A$$

Law C.68 (False guard)

$$false \mathbin{\&} A = Stop$$

Law C.69 (Guarded *Stop*)

$$g \mathbin{\&} Stop = Stop$$

Schema Expressions

Law C.70 (Schema disjunction elimination)

$$\text{pre } SExp_1 \,\&\, (SExp_1 \lor SExp_2) \;\sqsubseteq_{\mathcal{A}}\; \text{pre } SExp_1 \,\&\, SExp_1$$

Law C.71 (Schema expression/Sequence—introduction)

$$[\Delta S_1;\ \Delta S_2;\ i? : T \mid preS_1 \land preS_2 \land CS_1 \land CS_2]$$
$$\sqsubseteq_{\mathcal{A}}$$
$$[\Delta S_1;\ \Xi S_2;\ i? : T \mid preS_1 \land CS_1];\ [\Xi S_1;\ \Delta S_2;\ i? : T \mid preS_2 \land CS_2]$$

provided

- ➩ $\alpha(S_1) \cap \alpha(S_2) = \varnothing$
- ➩ $FV(preS_1) \subseteq \alpha(S_1) \cup \{i?\}$
- ➩ $FV(preS_2) \subseteq \alpha(S_2) \cup \{i?\}$
- ➩ $DFV(CS_1) \subseteq \alpha(S_1')$
- ➩ $DFV(CS_2) \subseteq \alpha(S_2')$
- ➩ $UDFV(CS_2) \cap DFV(CS_1) = \varnothing$ ❑

Law C.72 (Initialisation schema/Sequence—introduction*)

$$[\, S_1';\ S_2' \mid CS_1 \land CS_2 \,]$$
$$=$$
$$[\, S_1' \mid CS_1 \,];\ [\, S_2' \mid CS_2 \,]$$

provided

- ➩ $\alpha(S_1) \cap \alpha(S_2) = \varnothing$
- ➩ $DFV(CS_1) \subseteq \alpha(S_1')$
- ➩ $DFV(CS_2) \subseteq \alpha(S_2')$ ❑

Law C.73 (Schemas/Parallelism composition—distribution*)

$$SExp;\ (A_1 \llbracket\, ns_1 \mid cs \mid ns_2 \,\rrbracket\, A_2)$$
$$=$$
$$(SExp;\ A_1) \llbracket\, ns_1 \mid cs \mid ns_2 \,\rrbracket\, A_2$$

provided

- ➩ $wrtV(SExp) \subseteq ns_1$
- ➩ $wrtV(SExp) \cap usedV(A_2) = \varnothing$ ❑

Law C.74 (Schemas/Interleaving—distribution*)

$$(\Box\, i \bullet g_i \,\&\, SExp_i);\ (A_1 \parallel\!\![ns_1 \mid ns_2]\!\!\parallel A_2)$$
$$=$$
$$((\Box\, i \bullet g_i \,\&\, SExp_i);\ A_1) \parallel\!\![ns_1 \mid ns_2]\!\!\parallel A_2$$

provided

- ➪ $\bigcup_i wrtV(SExp_i) \subseteq ns_1$
- ➪ $\bigcup_i wrtV(SExp_i) \cap usedV(A_2) = \varnothing$ ❑

Law C.75 (Schemas refinement*)

$$SExp_1 \sqsubseteq_{\mathcal{A}} SExp_2$$

where

- $SExp_1 \mathrel{\widehat=} [\Delta S;\ di?;\ do! \mid P_1]$
- $SExp_2 \mathrel{\widehat=} [\Delta S;\ di?;\ do! \mid P_2]$

provided

- ➪ $\text{pre}\ SExp_1 \Rightarrow \text{pre}\ SExp_2$
- ➪ $(\text{pre}\ SExp_1 \wedge P_2) \Rightarrow P_1$ ❑

Parallelism composition

Law C.76 (Parallelism composition commutativity*)

$$A_1 \parallel\![\, ns_1 \mid cs \mid ns_2 \,]\!\parallel A_2 = A_2 \parallel\![\, ns_2 \mid cs \mid ns_1 \,]\!\parallel A_1$$

Law C.77 (Partition expansion*)

$$\mathbf{var}\ x : T \bullet A_1;\ (A_2 \parallel\![\, ns_1 \mid cs \mid ns_2 \,]\!\parallel A_3)$$
$$=$$
$$\mathbf{var}\ x : T \bullet A_1;\ (A_2 \parallel\![\, ns_1 \cup \{x\} \mid cs \mid ns_2 \,]\!\parallel A_3)$$

provided $x \notin ns_2$ ❑

Law C.78 (Parallelism composition introduction 1*)

$$c \rightarrow A = (c \rightarrow A \llbracket ns_1 \mid \{\!| c |\!\} \mid ns_2 \rrbracket c \rightarrow Skip)$$
$$c.e \rightarrow A = (c.e \rightarrow A \llbracket ns_1 \mid \{\!| c |\!\} \mid ns_2 \rrbracket c.e \rightarrow Skip)$$

provided

- $c \notin usedC(A)$
- $wrtV(A) \subseteq ns_1$ ❑

Law C.79 (Sequence/Parallelism composition—introduction 1)

$$A_1;\ A_2(e)$$
$$=$$
$$((A_1;\ c!e \rightarrow Skip) \llbracket \overline{wrtV(A_2)} \mid \{\!|c|\!\} \mid wrtV(A_2) \rrbracket c?y \rightarrow A_2(y)) \setminus \{\!|c|\!\}$$

provided

- $c \notin usedC(A_1) \cup usedC(A_2)$
- $y \notin FV(A_2)$
- $wrtV(A_1) \cap usedV(A_2) = \varnothing$
- $FV(e) \cap wrtV(A_2\ before\ e) = \varnothing$ ❑

Law C.80 (Channel extension 1)

$$A_1 \llbracket ns_1 \mid cs \mid ns_2 \rrbracket A_2 = A_1 \llbracket ns_1 \mid cs \cup \{\!|c|\!\} \mid ns_2 \rrbracket A_2$$

provided $c \notin usedC(A_1) \cup usedC(A_2)$ ❑

Law C.81 (Channel extension 2)

$$A_1 \llbracket ns_1 \mid cs \mid ns_2 \rrbracket A_2(e)$$
$$=$$
$$(c!e \rightarrow A_1 \llbracket ns_1 \mid cs \cup \{\!|c|\!\} \mid ns_2 \rrbracket c?x \rightarrow A_2(x)) \setminus \{\!|c|\!\}$$

provided

- $c \notin usedC(A_1) \cup usedC(A_2)$
- $x \notin FV(A_2)$
- $FV(e) \cap wrtV(A_2\ before\ e) = \varnothing$ ❑

Law C.82 (Channel extension 3*)

$$(A_1 \llbracket ns_1 \mid cs_1 \mid ns_2 \rrbracket A_2(e)) \setminus cs_2$$
$$=$$
$$((c!e \to A_1) \llbracket ns_1 \mid cs_1 \mid ns_2 \rrbracket (c?x \to A_2(x))) \setminus cs_2$$

provided

- ➪ $c \in cs_1$
- ➪ $c \in cs_2$
- ➪ $x \notin FV(A_2)$ ❑

Law C.83 (Channel extension 4*)

$$(A_1 \llbracket ns_1 \mid cs_1 \mid ns_2 \rrbracket A_2) \setminus cs_2 = ((c \to A_1) \llbracket ns_1 \mid cs_1 \mid ns_2 \rrbracket (c \to A_2)) \setminus cs_2$$
$$(A_1 \llbracket ns_1 \mid cs_1 \mid ns_2 \rrbracket A_2) \setminus cs_2 = ((c.e \to A_1) \llbracket ns_1 \mid cs_1 \mid ns_2 \rrbracket (c.e \to A_2)) \setminus cs_2$$

provided

- ➪ $c \in cs_1$
- ➪ $c \in cs_2$ ❑

Law C.84 (Parallelism composition/Sequence—step*)

$$(A_1;\ A_2) \llbracket ns_1 \mid cs \mid ns_2 \rrbracket A_3 = A_1;\ (A_2 \llbracket ns_1 \mid cs \mid ns_2 \rrbracket A_3)$$

provided

- ➪ $initials(A_3) \subseteq cs$
- ➪ $cs \cap usedC(A_1) = \varnothing$
- ➪ $wrtV(A_1) \cap usedV(A_3) = \varnothing$
- ➪ A_3 *is divergence-free*
- ➪ $wrtV(A_1) \subseteq ns_1$ ❑

Law C.85 (Parallelism composition/External choice—exchange)

$$(A_1 \llbracket ns_1 \mid cs \mid ns_2 \rrbracket A_2) \,\square\, (B_1 \llbracket ns_1 \mid cs \mid ns_2 \rrbracket B_2)$$
$$=$$
$$(A_1 \,\square\, B_1) \llbracket ns_1 \mid cs \mid ns_2 \rrbracket (A_2 \,\square\, B_2)$$

provided $A_1 \llbracket ns_1 \mid cs \mid ns_2 \rrbracket B_2 = A_2 \llbracket ns_1 \mid cs \mid ns_2 \rrbracket B_1 = Stop$ ❑

Law C.86 (Parallelism composition/External choice—expansion*)

$$(\Box\, i \bullet a_i \to A_i) \llbracket ns_1 \mid cs \mid ns_2 \rrbracket (\Box\, j \bullet b_j \to B_j)$$
$$=$$
$$(\Box\, i \bullet a_i \to A_i) \llbracket ns_1 \mid cs \mid ns_2 \rrbracket ((\Box\, j \bullet b_j \to B_j) \Box (c \to C))$$

provided

- $\bigcup_i \{a_i\} \subseteq cs$
- $c \in cs$
- $c \notin \bigcup_i \{a_i\}$
- $c \notin \bigcup_j \{b_j\}$

Law C.87 (Parallelism composition/External choice—distribution*)

$$\Box\, i \bullet (A_i \llbracket ns_1 \mid cs \mid ns_2 \rrbracket A) = (\Box\, i \bullet A_i) \llbracket ns_1 \mid cs \mid ns_2 \rrbracket A$$

provided

- ➪ $initials(A) \subseteq cs$
- ➪ A *is deterministic* ❑

Law C.88 (Parallelism composition/Sequence—distribution*)

$$(A_1 \llbracket ns_1 \mid cs \mid ns_2 \rrbracket A_2);\ (B_1 \llbracket ns_1 \mid cs \mid ns_2 \rrbracket B_2)$$
$$=$$
$$(A_1;\ B_1) \llbracket ns_1 \mid cs \mid ns_2 \rrbracket (A_2;\ B_2)$$

provided

- ➪ $initials(B_1) \cup initials(B_2) \subseteq cs$
- ➪ $usedC(A_1) \cap initials(B_2) = \varnothing$
- ➪ $usedC(A_2) \cap initials(B_1) = \varnothing$
- ➪ $usedV(B_1) \cap ns_2 = usedV(B_2) \cap ns_1 = \varnothing$ ❑

Law C.89 (Parallelism composition Assignment/*Skip**)

$$vl := el \llbracket ns_1 \mid cs \mid ns_2 \rrbracket Skip = vl := el$$

provided

- ➪ ns_1 *and* ns_2 *partition the variables in scope*
- ➪ $vl \in ns_1$ ❑

Law C.90 (Parallelism composition unit*)

$$Skip \llbracket ns_1 \mid cs \mid ns_2 \rrbracket Skip = Skip$$

Law C.91 (Parallelism composition unit 2*)

$$Stop \llbracket ns_1 \mid cs \mid ns_2 \rrbracket Stop = Stop$$

Law C.92 (Parallelism composition Deadlocked 1*)

$$(c_1 \to A_1) \llbracket ns_1 \mid cs \mid ns_2 \rrbracket (c_2 \to A_2) = Stop = Stop \llbracket ns_1 \mid cs \mid ns_2 \rrbracket (c_2 \to A_2)$$

provided

- ➪ $c_1 \neq c_2$
- ➪ $\{c_1, c_2\} \subseteq cs$ ❑

Law C.93 (Parallelism composition Deadlocked 2)

$$g1 \mathbin{\&} c_1 \to A_1 \llbracket ns_1 \mid cs \cup \{\!| c_1, c_2 |\!\} \mid ns_2 \rrbracket g_2 \mathbin{\&} c_2 \to A_2 = Stop$$

provided

- ➪ $c_1 \neq c_2$
- ➪ $\{c_1, c_2\} \subseteq cs$ ❑

Law C.94 (Parallelism composition Zero*)

$$Chaos \llbracket ns_1 \mid cs \mid ns_2 \rrbracket A = Chaos$$

Interleaving

Law C.95 (Interleaving/Sequence—distribution*)

$$(A_1 \parallel\!\![ns_1 \mid ns_2]\!\!\parallel A_2); (B_1 \llbracket ns_1 \mid cs \mid ns_2 \rrbracket B_2)$$
$$=$$
$$(A_1; B_1) \llbracket ns_1 \mid cs \mid ns_2 \rrbracket (A_2; B_2)$$

provided

- ➪ $(usedC(A_1) \cup usedC(A_2)) \cap cs = \varnothing$
- ➪ $initials(B_1) \cup initials(B_2) \subseteq cs$ ❑

Law C.96 (Interleaving Zero*)

$$Chaos \; |\!|\![ns_1 \mid ns_2]\!|\!| \; A = Chaos$$

Law C.97 (Interleaving *Stop)**

$$Stop \; |\!|\![ns_1 \mid ns_2]\!|\!| \; Stop = Stop$$

Law C.98 (Parallelism composition/Interleaving Equivalence*)

$$A_1 \; |\!|\![ns_2 \mid ns_2]\!|\!| \; A_2 = A_1 \; [\![\, ns_2 \mid \varnothing \mid ns_2 \,]\!] \; A_2$$

Law C.99 (Interleaving Choices*)

$$\begin{array}{l} (c_1 \to A_1) \; |\!|\![ns_1 \mid ns_2]\!|\!| \; (c_2 \to A_2) \\ = \\ c_1 \to (A_1 \; |\!|\![ns_1 \mid ns_2]\!|\!| \; (c_2 \to A_2)) \; \Box \; c_2 \to ((c_1 \to A_1) \; |\!|\![ns_1 \mid ns_2]\!|\!| \; A_2) \end{array}$$

Prefix

Law C.100 (Prefix/*Skip)**

$$c \to A = (c \to Skip); \; A$$

$$c.e \to A = (c.e \to Skip); \; A$$

Law C.101 (Prefix/Sequential composition—associativity)

$$c \to (A_1; \; A_2) = (c \to A_1); \; A_2$$

$$c.e \to (A_1; \; A_2) = (c.e \to A_1); \; A_2$$

provided $FV(A_2) \cap \alpha(c) = \varnothing$ ❑

Law C.102 (Prefix/Hiding*)

$$(c \to Skip) \setminus \{c\} = Skip$$

$$(c.e \to Skip) \setminus \{c\} = Skip$$

Law C.103 (Prefix introduction*)

$$A = (c \to A) \setminus \{\!|\; c \;|\!\}$$

provided $c \notin usedC(A)$ ❑

Law C.104 (Prefix/External choice—distribution*)

$$c \to \Box\, i \bullet g_i \mathbin{\&} A_i = \Box\, i \bullet g_i \mathbin{\&} c \to A_i$$

provided

- $\vee\, i \bullet g_i$
- $\forall\, i, j \mid i \neq j \bullet \neg\, (g_i \wedge g_j)$ *(guards are mutually exclusive).* ❑

Law C.105 (Prefix/Internal choice—distribution)

$$c \to (A_1 \sqcap A_2) = (c \to A_1) \sqcap (c \to A_2)$$

$$c.e \to (A_1 \sqcap A_2) = (c.e \to A_1) \sqcap (c.e \to A_2)$$

Law C.106 (Prefix/Parallelism composition—distribution)

$$c \to (A_1 \llbracket ns_1 \mid cs \mid ns_2 \rrbracket A_2) = (c \to A_1) \llbracket ns_1 \mid cs \cup \{\!|c|\!\} \mid ns_2 \rrbracket (c \to A_2)$$

$$c.e \to (A_1 \llbracket ns_1 \mid cs \mid ns_2 \rrbracket A_2) = (c.e \to A_1) \llbracket ns_1 \mid cs \cup \{\!|c|\!\} \mid ns_2 \rrbracket (c.e \to A_2)$$

provided $c \notin usedC(A_1) \cup usedC(A_2)$ *or* $c \in cs$ ❑

Law C.107 (Communication/Parallelism composition—distribution)

$$(c!e \to A_1) \llbracket ns_1 \mid cs \mid ns_2 \rrbracket (c?x \to A_2(x)) = c.e \to (A_1 \llbracket ns_1 \mid cs \mid ns_2 \rrbracket A_2(e))$$

provided

- $c \in cs$
- $x \notin FV(A_2)$. ❑

Law C.108 (Input prefix/Parallelism composition—distribution*)

$$c?x \to (A_1 \llbracket ns_1 \mid cs \mid ns_2 \rrbracket A_2) = (c?x \to A_1) \llbracket ns_1 \mid cs \mid ns_2 \rrbracket (c?x \to A_2)$$

provided

$c \in cs$ ❑

Law C.109 (Input prefix/Parallelism composition—distribution 2*)

$$c?x \rightarrow (A_1 \llbracket ns_1 \mid cs \mid ns_2 \rrbracket A_2) = (c?x \rightarrow A_1) \llbracket ns_1 \mid cs \mid ns_2 \rrbracket A_2$$

provided

- $c \notin cs$
- $x \notin usedV(A_2)$
- $initials(A_2) \subseteq cs$
- A_2 *is deterministic* ❑

External choice

Law C.110 (External choice commutativity*)

$$A_1 \Box A_2 = A_2 \Box A_1$$

Law C.111 (External choice elimination*)

$$A \Box A = A$$

Law C.112 (External choice/Sequence—distribution)

$$(\Box\, i \bullet g_i \,\&\, c_i \rightarrow A_i);\ B = \Box\, i \bullet g_i \,\&\, c_i \rightarrow A_i;\ B$$

Law C.113 (External choice/Sequence—distribution 2*)

$$((g_1 \,\&\, A_1) \Box (g_2 \,\&\, A_2));\ B = ((g_1 \,\&\, A_1);\ B) \Box ((g_2 \,\&\, A_2);\ B)$$

provided $g_1 \Rightarrow \neg\, g_2$ ❑

Law C.114 (External choice unit)

$$Stop \Box A = A$$

Internal Choice

Law C.115 (Sequence/Internal choice—distribution*)

$$A_1;\ (A_2 \sqcap A_3) = (A_1;\ A_2) \sqcap (A_1;\ A_3)$$

Law C.116 (Internal choice elimination*)

$$A \sqcap A = A$$

Law C.117 (Internal choice elimination 2*)

$$A_1 \sqcap A_2 \sqsubseteq_{\mathcal{A}} A_1$$

Law C.118 (Internal choice zero*)

$$A \sqcap Chaos = Chaos$$

Law C.119 (Internal choice/Parallelism composition Distribution*)

$$(A_1 \sqcap A_2) \llbracket ns_1 \mid cs \mid ns_2 \rrbracket A_3$$
$$=$$
$$(A_1 \llbracket ns_1 \mid cs \mid ns_2 \rrbracket A_3) \sqcap (A_2 \llbracket ns_1 \mid cs \mid ns_2 \rrbracket A_3)$$

Hiding

Law C.120 (Hiding Identity*)

$$A \setminus cs = A$$

provided $cs \cap usedC(A) = \varnothing$ ❑

Law C.121 (Hiding combination)

$$(A \setminus cs_1) \setminus cs_2 = A \setminus (cs_1 \cup cs_2)$$

Law C.122 (Hiding/External choice—distribution*)

$$(A_1 \square A_2) \setminus cs = (A_1 \setminus cs) \square (A_2 \setminus cs)$$

provided $(initials(A_1) \cup initials(A_2)) \cap cs = \varnothing$ ❑

Law C.123 (Hiding/External choice—distribution 2*)

$$((g_1 \,\&\, A_1) \square (g_2 \,\&\, A_2)) \setminus cs = ((g_1 \,\&\, A_1) \setminus cs) \square ((g_2 \,\&\, A_2) \setminus cs)$$

provided $\neg (g_1 \wedge g_2)$ *or* $(initials(A_1) \cup initials(A_2)) \cap cs = \varnothing$ ❑

Law C.124 (Hiding expansion 2*)

$$A \setminus cs = A \setminus cs \cup \{c\}$$

provided $c \notin usedC(A)$ ❑

Law C.125 (Hiding/Sequence—distribution*)

$$(A_1;\ A_2) \setminus cs = (A_1 \setminus cs);\ (A_2 \setminus cs)$$

Law C.126 (Hiding/*Chaos*—distribution*)

$$Chaos \setminus cs = Chaos$$

Law C.127 (Hiding/Parallelism composition—distribution*)

$$(A_1 \llbracket ns_1 \mid cs_1 \mid ns_2 \rrbracket A_2) \setminus cs_2 = (A_1 \setminus cs_2) \llbracket ns_1 \mid cs_1 \mid ns_2 \rrbracket (A_2 \setminus cs_2)$$

provided $cs_1 \cap cs_2 = \varnothing$ ❑

Recursion

Law C.128 (Recursion unfold)

$$\mu X \bullet F(X) = F(\mu X \bullet F(X))$$

Law C.129 (Recursion—least fixed-point)

$$F(Y) \sqsubseteq_{\mathcal{A}} Y \Rightarrow \mu X \bullet F(X) \sqsubseteq_{\mathcal{A}} Y$$

Law C.130 (Recursion Refinement*)

$$\mu X \bullet F_1(X) \sqsubseteq_{\mathcal{A}} \mu X \bullet F_2(X)$$

provided $F_1 \sqsubseteq_{\mathcal{A}} F_2$ ❑

Law C.131 (Recursion—divergence introduction*)

$$(\mu X \bullet (c \to X)) \setminus \{c\} = (\mu X \bullet (c.e \to X)) \setminus \{c\} = Chaos$$

Sequence

Law C.132 (Sequence unit)

$(A) Skip;\ A$
$(B) A = A;\ Skip$

Law C.133 (Sequence zero)

$$Stop;\ A = Stop$$

Law C.134 (Sequence zero 2*)

$$Chaos;\ A = Chaos$$

Chaos

Law C.135 ($Chaos$ Refinement*)

$$Chaos \sqsubseteq_{\mathcal{A}} A$$

Variable Blocks

Law C.136 (Variable block introduction*)

$$A = \mathbf{var}\ x : T \bullet A$$

provided $x \notin FV(A)$ ❑

Law C.137 (Variable block/Sequence—extension*)

$$A_1;\ (\mathbf{var}\ x : T \bullet A_2);\ A_3 = (\mathbf{var}\ x : T \bullet A_1;\ A_2;\ A_3)$$

provided $x \notin FV(A_1) \cup FV(A_3)$ ❑

Law C.138 (Variable block/Parallelism composition—extension*)

$$\begin{array}{l}(\mathbf{var}\ x : T \bullet A_1) \llbracket ns_1 \mid cs \mid ns_2 \rrbracket A_2 \\ = \\ (\mathbf{var}\ x : T \bullet A_1 \llbracket ns_1 \cup \{x\} \mid cs \mid ns_2 \rrbracket A_2)\end{array}$$

provided $x \notin FV(A_2) \cup ns_1 \cup ns_2$ ❑

Law C.139 (Variable Substitution*)

$$A(x) = \mathbf{var}\ y \bullet y : [y' = x];\ A(y)$$

provided y *is not free in* A ❑

Alternation

Law C.140 (Alternation Introduction*)

$$w : [pre, post] \sqsubseteq_{\mathcal{A}} \textbf{if} \; \|_i g_i \rightarrow \; w : [g_i \wedge pre, post] \; \textbf{fi}$$

provided $pre \Rightarrow \bigvee_i g_i$ ❑

Law C.141 (Alternation/Guarded Actions—interchange*)

$$\textbf{if} \; g_1 \rightarrow \; A_1 \| \; g_2 \rightarrow A_2 \; \textbf{fi} = g_1 \; \& \; A_1 \; \Box \; g_2 \; \& \; A_2$$

provided

- ➪ $g_1 \vee g_2$
- ➪ $g_1 \Rightarrow \neg \; g_2$ ❑

Substitution

Law C.142 (Substitution introduction*)

$$A = A[old_1, ..., old_n := new_1, ..., new_n]$$

provided $\{old_1, ..., old_n\} \cap FV(A) = \varnothing$ ❑

Law C.143 (Substitution expansion*)

$$F(A[old_1, \ldots, old_n := new_1, \ldots, new_n]) = F(A)[old_1, \ldots, old_n := new_1, \ldots, new_n]$$

provided $\{old_1, \ldots, old_n\} \cap FV(F(\_)) = \varnothing$ ❑

Law C.144 (Substitution combination*)

$$A[old_1, \ldots, old_n := mid_1, \ldots, mid_n][mid_1, \ldots, mid_n := new_1, \ldots, new_n]$$
$$=$$
$$A[old_1, \ldots, old_n := new_1, \ldots, new_n]$$

provided $\{mid_1, \ldots, mid_n\} \cap FV(A) = \varnothing$ ❑

Law C.145 (Substitution combination 2*)

$$A[old_1, ..., old_n := new_1, \ldots, new_n][old_{n+1}, ..., old_m := new_{n+1}, ..., new_m]$$
$$=$$
$$A[old_1, ..., old_m := new_1, ..., new_m]$$

provided $\{new_1, ..., new_n\} \cap \{old_{n+1}, ..., old_m\} = \varnothing$ ❑

Process Refinement

Law C.146 (Process splitting)

Let qd and rd stand for the declarations of the processes Q and R, determined by Q.State, Q.PPar, and Q.Act, and R.State, R.PPar, and R.Act, respectively, and pd stand for the process declaration.

$$\begin{array}{l} \textbf{process } P \mathrel{\widehat=} \textbf{begin state } State \mathrel{\widehat=} Q.State \land R.State \\ \qquad\qquad Q.PPar \land_\Xi R.State \\ \qquad\qquad R.PPar \land_\Xi Q.State \\ \qquad\qquad \bullet F(Q.Act, R.Act) \\ \qquad \textbf{end} \end{array}$$

Then

$$pd = (qd\ rd\ \textbf{process } P \mathrel{\widehat=} F(Q, R))$$

provided *Q.PPar and R.PPar are disjoint with respect to R.State and Q.State.* ❑

Law C.147 (Process Splitting 2*)

$$
\begin{array}{l}
\textbf{process}\ G \mathrel{\widehat=} \textbf{begin} \\
\quad LState \mathrel{\widehat=} [\, id : Range;\ comps \mid pred_l \,] \\
\quad \textbf{state}\ GState \mathrel{\widehat=} \\
\qquad\qquad [\, f : Range \rightarrow LState \mid \forall j : Range \bullet (f\ j).id = j \land pred_g(j) \,] \\
\quad L.schema_j \land_{\Xi}\ GState \\
\quad L.action_k \land_{\Xi}\ GState
\end{array}
$$

Promotion

$\Delta LState;\ \Delta GState;\ id? : Range$

$\theta LState = f\ id? \land f' = f \oplus \{id? \mapsto \theta LState'\}$

$$
\begin{array}{l}
\quad G.schema_j \mathrel{\widehat=} \forall\, id? : Range \bullet L.schema_j \land Promotion \\
\quad G.action_k \mathrel{\widehat=} \\
\qquad \llbracket\, cs \,\rrbracket\ i : Range \bullet \llbracket\, \alpha\,(f\ i) \,\rrbracket \bullet (\textbf{promote}_\mathbf{2}\ L.action_k)\,[id, id? := i, i] \\
\quad \bullet\ G.action\ \textbf{end} \\
= \\
\textbf{process}\ L \mathrel{\widehat=}\ (id : Range \bullet \textbf{begin state}\ LState \mathrel{\widehat=} [\, comps \mid pred_l \,] \\
\qquad\qquad\qquad\qquad\qquad\qquad L.schema_j\ L.action_k \\
\qquad\qquad\qquad\qquad\qquad\qquad \bullet\ L.action\ \textbf{end}) \\
\textbf{process}\ G \mathrel{\widehat=}\ \llbracket cs \rrbracket\ id : Range \bullet L(id)
\end{array}
$$