Meld 2.0 Semantics

Flavio Cruz

February 11, 2014

1 Static Semantics

1.1 Types

$$\frac{}{\textsf{addr typ}}\ \textsf{addr} \qquad \frac{}{\textsf{int typ}}\ \textsf{int} \qquad \frac{}{\textsf{float typ}}\ \textsf{float} \qquad \frac{}{\textsf{bool typ}}\ \textsf{bool} \qquad \frac{}{\textsf{string typ}}\ \textsf{string}$$

$$\frac{\tau\ \textsf{typ}}{\textsf{list}\ \tau\ \textsf{typ}}\ \textsf{list} \qquad \frac{\textsf{A typ} \quad \textsf{B typ}}{\textsf{struct A} \times \textsf{B typ}}\ \textsf{struct} \qquad \frac{}{\textsf{struct} \cdot \textsf{typ}}\ \textsf{struct}$$

1.2 Expressions

$$\frac{}{\Psi; \Gamma \vdash \textsf{addr}(N) : \textsf{addr}}\ \textsf{addr literal}$$

$$\frac{\textsf{N is a literal}}{\Psi; \Gamma \vdash \textsf{int}(N) : \textsf{int}}\ \textsf{int literal} \qquad \frac{\textsf{F is a float literal}}{\Psi; \Gamma \vdash \textsf{float}(F) : \textsf{float}}\ \textsf{float literal}$$

$$\frac{\textsf{S is a string literal}}{\Psi; \Gamma \vdash \textsf{string}(S) : \textsf{string}}\ \textsf{string literal} \qquad \frac{}{\Psi; \Gamma, X : \tau \vdash X : \tau}\ \textsf{var}$$

$$\frac{\tau}{\Psi; \Gamma \vdash [] : \textsf{list}\ \tau}\ \textsf{nil} \qquad \frac{\Psi; \Gamma \vdash e_1 : \tau \quad \Psi; \Gamma \vdash e_2 : \textsf{list}\ \tau}{\Psi; \Gamma \vdash [e_1 \mid e_2] : \textsf{list}\ \tau}\ \textsf{cons}$$

$$\frac{\Psi; \Gamma \vdash e_1 : \tau_1 \quad \Psi; \Gamma \vdash e_2 : \tau_2}{\Psi; \Gamma \vdash \textsf{struct}\ e_1 \times e_2 : \textsf{struct}\ \tau_1 \times \tau_2}\ \textsf{make struct}$$

$$\frac{\Psi; \Gamma \vdash e : \textsf{struct}\ \tau_1 \times \tau_2}{\Psi; \Gamma \vdash fst(e) : \tau_1}\ \textsf{struct fst} \qquad \frac{\Psi; \Gamma \vdash e : \textsf{struct}\ \tau_1 \times \tau_2}{\Psi; \Gamma \vdash snd(e) : \tau_2}\ \textsf{struct snd}$$

$$\frac{\Psi; \Gamma \vdash e_1 : \textsf{int} \quad \Psi; \Gamma \vdash e_2 : \textsf{int}}{\Psi; \Gamma \vdash e_1\ \textsf{op}\ e_2 : \textsf{int}}\ \textsf{math int} \qquad \frac{\Psi; \Gamma \vdash e_1 : \textsf{float} \quad \Psi; \Gamma \vdash e_2 : \textsf{float}}{\Psi; \Gamma \vdash e_1\ \textsf{op}\ e_2 : \textsf{float}}\ \textsf{math float}$$

$$\frac{\Psi; \Gamma \vdash e_1 : \textsf{int} \quad \Psi; \Gamma \vdash e_2 : \textsf{float}}{\Psi; \Gamma \vdash e_1\ \textsf{op}\ e_2 : \textsf{float}}\ \textsf{math cast1} \qquad \frac{\Psi; \Gamma \vdash e_1 : \textsf{float} \quad \Psi; \Gamma \vdash e_2 : \textsf{int}}{\Psi; \Gamma \vdash e_1\ \textsf{op}\ e_2 : \textsf{float}}\ \textsf{math cast2}$$

$$\frac{\Psi;\Gamma \vdash c : \mathsf{bool} \quad \Psi;\Gamma \vdash e_1 : \tau \quad \Psi;\Gamma \vdash e_2 : \tau}{\Psi;\Gamma \vdash \mathsf{if}\ c\ \mathsf{then}\ e_1\ \mathsf{else}\ e_2\ \mathsf{end} : \tau}\ \mathsf{if}$$

$$\frac{\Psi;\Gamma \vdash e_1 : \mathsf{int} \quad \Psi;\Gamma \vdash e_2 : \mathsf{int}}{\Psi;\Gamma \vdash e_1\ \mathsf{cmp}\ e_2 : \mathsf{bool}}\ \mathsf{cmp\ int} \qquad \frac{\Psi;\Gamma \vdash e_1 : \mathsf{float} \quad \Psi;\Gamma \vdash e_2 : \mathsf{float}}{\Psi;\Gamma \vdash e_1\ \mathsf{cmp}\ e_2 : \mathsf{bool}}\ \mathsf{cmp\ float}$$

$$\frac{\Psi;\Gamma \vdash e_1 : \mathsf{bool} \quad \Psi;\Gamma \vdash e_2 : \mathsf{bool}}{\Psi;\Gamma \vdash e_1\ \mathsf{cmp}\ e_2 : \mathsf{bool}}\ \mathsf{cmp\ bool} \qquad \frac{\Psi;\Gamma \vdash e_1 : \mathsf{string} \quad \Psi;\Gamma \vdash e_2 : \mathsf{string}}{\Psi;\Gamma \vdash e_1\ \mathsf{cmp}\ e_2 : \mathsf{bool}}\ \mathsf{cmp\ string}$$

$$\frac{\Psi;\Gamma \vdash e_1 : \mathsf{addr} \quad \Psi;\Gamma \vdash e_2 : \mathsf{addr}}{\Psi;\Gamma \vdash e_1\ \mathsf{cmp}\ e_2 : \mathsf{bool}}\ \mathsf{cmp\ addr}$$

$$\frac{\Psi;\Gamma \vdash e_1 : \mathsf{bool} \quad \Psi;\Gamma \vdash e_2 : \mathsf{bool}}{\Psi;\Gamma \vdash e_1\ \mathsf{or}\ e_2 : \mathsf{bool}}\ \mathsf{or}$$

$$\frac{\Psi;\Gamma \vdash e_1 : \tau_1 \quad \Psi;\Gamma, X : \tau_1 \vdash e_2 : \tau}{\Psi;\Gamma \vdash \mathsf{let}\ X\ =\ e_1\ \mathsf{in}\ e_2\ \mathsf{end} : \tau}\ \mathsf{let}$$

$$\frac{}{\Psi, \mathsf{const}(name, v) : \tau;\Gamma \vdash \mathsf{getconst}(name) : \tau}\ \mathsf{const}$$

$$\frac{\Psi;\Gamma \vdash e : \mathsf{struct}\ \tau \quad \mathsf{external}(name, arg : \mathsf{struct}\ \tau) : \tau_{\mathsf{r}} \in \Psi}{\Psi;\Gamma \vdash \mathsf{callexternal}(name, e) : \tau_{\mathsf{r}}}\ \mathsf{external}$$

$$\frac{\Psi;\Gamma \vdash e : \mathsf{struct}\ \tau \quad \mathsf{fun}(name, arg : \mathsf{struct}\ \tau, b) : \tau_{\mathsf{r}} \in \Psi}{\Psi;\Gamma \vdash \mathsf{callfun}(name, e) : \tau_{\mathsf{r}}}\ \mathsf{fun}$$

$$\frac{}{\Psi;\Gamma \vdash \mathsf{world} : \mathsf{int}}\ \mathsf{world} \qquad \frac{}{\Psi;\Gamma \vdash \mathsf{arg}(N) : \mathsf{string}}\ \mathsf{arg}$$

1.3 Declarations

$$\frac{\mathsf{addr\ typ} \quad \tau_1\ \mathsf{typ} \quad \ldots \quad \tau_{\mathsf{n}}\ \mathsf{typ}}{\mathsf{decl}\ name\ [\mathsf{addr}, \tau_1, ..., \tau_n]}\ \mathsf{decl} \qquad \frac{\mathsf{addr\ typ} \quad \tau_1\ \mathsf{typ} \quad \ldots \quad \tau_{\mathsf{n}}\ \mathsf{typ}}{!\mathsf{decl}\ name\ [\mathsf{addr}, \tau_1, ..., \tau_n]}\ !\mathsf{decl}$$

$$\frac{\Psi;\Gamma \vdash e : \tau \quad \Psi\ ;\ e \rightarrow v \quad \mathsf{val}\ v : \tau}{\mathsf{const}(name, v)\ \mathsf{of}\ \tau}\ \mathsf{const}$$

$$\frac{\Psi; arg : \mathsf{struct}\ \tau \vdash b : \tau_{\mathsf{r}}}{\mathsf{fun}(name, arg : \mathsf{struct}\ \tau, \mathsf{b} : \tau_r)\ \mathsf{of}\ \tau_r}\ \mathsf{fun}$$

1.4 Rules

$$\frac{\Psi; \cdot; H \vdash_1 A \text{ rule}}{\text{rule } \Psi \ / \ \forall H : \text{addr}.A} \text{ rule start}$$

$$\frac{\Psi; \Gamma, X : \tau; H \vdash_N A \text{ rule}}{\Psi; \Gamma; H \vdash_N \forall X : \tau.A \text{ rule}} \text{ rule add var}$$

$$\frac{\Psi; \Gamma, H : \text{addr}; H; \Gamma \vdash A \text{ body} \quad \Psi; \Gamma, H : \text{addr}; H \vdash_N B \text{ head}}{\Psi; \Gamma; H \vdash_N A \multimap B \text{ rule}} \text{ rule body head}$$

$$\frac{\Psi; \Gamma; H; \Gamma'' \vdash B \text{ body} \quad \Psi; \Gamma; H; \Gamma' \vdash A \text{ body}}{\Psi; \Gamma; H; \Gamma', \Gamma'' \vdash A \otimes B \text{ body}} \text{ rule body tensor}$$

$$\frac{}{\Psi; \Gamma; H; \cdot \vdash 1 \text{ body}} \text{ rule body 1}$$

$$\frac{\Psi; \Gamma, X : \tau; H; \Gamma', X : \tau \vdash A \text{ body}}{\Psi; \Gamma; H; \Gamma' \vdash \exists X : \tau.A \text{ body}} \text{ rule body exists}$$

$$\frac{\text{decl } name \ [\text{addr}, \tau_1, ..., \tau_n] \in \Psi}{\Psi; \Gamma, H_{fact} : \text{addr}; H; X_1 : \tau_1, ..., X_n : \tau_n \vdash name[@H_{fact}](X_1, ..., X_n) \text{ body}} \text{ rule body fact}$$

$$\frac{\text{!decl } name \ [\text{addr}, \tau_1, ..., \tau_n] \in \Psi}{\Psi; \Gamma, H_{fact} : \text{addr}; H; X_1 : \tau_1, ..., X_n : \tau_n \vdash !name[@H_{fact}](X_1, ..., X_n) \text{ body}} \text{ rule body !fact}$$

$$\frac{\Psi; \Gamma \models e : \text{bool}}{\Psi; \Gamma; H; \cdot \vdash !(\text{constraint } e) \text{ body}} \text{ rule}$$

$$\frac{\Psi; \Gamma; H \vdash_N A \text{ head} \quad \Psi; \Gamma; H \vdash_N B \text{ head}}{\Psi; \Gamma; H \vdash_N A \otimes B \text{ head}} \text{ rule head tensor}$$

$$\frac{}{\Psi; \Gamma; H \vdash_N 1 \text{ head}} \text{ rule head 1}$$

$$\frac{\Psi; \Gamma \models e : \text{addr} \quad \Psi; \Gamma \models e_1 : \tau_1 \quad ... \quad \Psi; \Gamma \models e_n : \tau_\text{n} \quad \text{decl } name \ [\text{addr}, \tau_1, ..., \tau_n] \in \Psi}{\Psi; \Gamma; H \vdash_N name[@e](e_1, ..., e_n) \text{ head}} \text{ rule head fact}$$

$$\frac{\Psi; \Gamma \models e : \text{addr} \quad \Psi; \Gamma \models e_1 : \tau_1 \quad ... \quad \Psi; \Gamma \models e_n : \tau_\text{n} \quad \text{!decl } name \ [\text{addr}, \tau_1, ..., \tau_n] \in \Psi}{\Psi; \Gamma; H \vdash_N !name[@e](e_1, ..., e_n) \text{ head}} \text{ rule head !fact}$$

$$\frac{\Psi; \Gamma, X : \text{addr}; H \vdash_N A \text{ head}}{\Psi; \Gamma; H \vdash_N \exists X : \text{addr}.A \text{ head}} \text{ rule head exists}$$

$$\frac{\Psi; \Gamma; H \vdash_2 A \text{ rule}}{\Psi; \Gamma; H \vdash_1 \text{comp } A \text{ head}} \text{ rule head comprehension}$$

$$\frac{[\text{Op}] \ / \ \tau_1 \rightsquigarrow \tau_2 \quad \Psi; \Gamma, X : \tau_1; H \vdash_2 A \multimap 1 \text{ rule} \quad \Psi; \Gamma, X : \tau_2; H \vdash_2 B \text{ head}}{\Psi; \Gamma; H \vdash_1 [\text{Op}; X; A \Rightarrow B] \text{ head}} \text{ rule head aggregate}$$

1.5 Aggregate Types

$$\frac{}{[\mathsf{count}]\ /\ \mathsf{int} \rightsquigarrow \mathsf{int}}\ \mathsf{agg\ count} \qquad \frac{}{[\mathsf{collect\ int}]\ /\ \mathsf{int} \rightsquigarrow \mathsf{list\ int}}\ \mathsf{agg\ collect\ int}$$

$$\frac{}{[\mathsf{sum\ int}]\ /\ \mathsf{int} \rightsquigarrow \mathsf{int}}\ \mathsf{agg\ int\ sum}$$

$$\frac{}{[\mathsf{max\ int}]\ /\ \mathsf{int} \rightsquigarrow \mathsf{int}}\ \mathsf{agg\ int\ max} \qquad \frac{}{[\mathsf{min\ int}]\ /\ \mathsf{int} \rightsquigarrow \mathsf{int}}\ \mathsf{agg\ int\ min}$$

2 Dynamic Semantics

2.1 Expression Values

$$\frac{}{\mathsf{val\ int}(N) : \mathsf{int}}\ \mathsf{int} \qquad \frac{}{\mathsf{val\ bool}(B) : \mathsf{bool}}\ \mathsf{bool} \qquad \frac{}{\mathsf{val\ float}(F) : \mathsf{float}}\ \mathsf{float}$$

$$\frac{}{\mathsf{val\ string}(S) : \mathsf{string}}\ \mathsf{string} \qquad \frac{}{\mathsf{val\ addr}(A) : \mathsf{addr}}\ \mathsf{addr}$$

$$\frac{}{\mathsf{val}\ [] : \mathsf{list}\ \tau}\ \mathsf{nil} \qquad \frac{\mathsf{val}\ x : \tau \quad \mathsf{val}\ ls : \mathsf{list}\ \tau}{\mathsf{val}\ x :: ls : \mathsf{list}\ \tau}\ \mathsf{cons}$$

$$\frac{}{\mathsf{val\ struct} \cdot : \mathsf{struct} \cdot}\ \mathsf{struct} \qquad \frac{\mathsf{val}\ v_1 : \tau_1 \quad \mathsf{val}\ v_2 : \tau_2}{\mathsf{val}\ v_1 \times v_2 : \mathsf{struct}\ \tau_1 \times \tau_2}\ \mathsf{struct}$$

2.2 Expression Evaluation

$$\frac{}{\Psi\ ;\ \mathsf{int}(N) \to \mathsf{int}(N)}\ \mathsf{int} \qquad \frac{}{\Psi\ ;\ \mathsf{float}(F) \to \mathsf{float}(F)}\ \mathsf{float} \qquad \frac{}{\Psi\ ;\ \mathsf{addr}(A) \to \mathsf{addr}(A)}\ \mathsf{addr}$$

$$\frac{}{\Psi\ ;\ \mathsf{bool}(B) \to \mathsf{bool}(B)}\ \mathsf{bool}$$

$$\frac{}{\Psi\ ;\ \mathsf{string}(S) \to \mathsf{string}(S)}\ \mathsf{string} \qquad \frac{}{\Psi\ ;\ [] \to []}\ \mathsf{list\ nil} \qquad \frac{}{\Psi\ ;\ L \to L}\ \mathsf{list}$$

$$\frac{\Psi\ ;\ e_1 \to v_1 \quad \Psi\ ;\ e_2 \to v_2}{\Psi\ ;\ [e_1|e_2] \to v_1 :: v_2}\ \mathsf{cons}$$

$$\frac{}{\Psi\ ;\ \mathsf{struct} \cdot \to \mathsf{struct} \cdot}\ \mathsf{struct} \qquad \frac{}{\Psi\ ;\ v_1 \times v_2 \to v_1 \times v_2}\ \mathsf{struct}$$

$$\frac{\Psi\ ;\ e_1 \to v_1 \quad \Psi\ ;\ e_2 \to v_2}{\Psi\ ;\ \mathsf{struct\ e_1 \times e_2} \to v_1 \times v_2}\ \mathsf{make\ struct}$$

$$\frac{\Psi\ ;\ e \to v_1 \times v_2}{\Psi\ ;\ \mathsf{fst(e)} \to v_1}\ \mathsf{struct\ fst} \qquad \frac{\Psi\ ;\ e \to v_1 \times v_2}{\Psi\ ;\ \mathsf{snd(e)} \to v_2}\ \mathsf{struct\ snd}$$

$$\frac{\Psi; \cdot \vdash e_1 : \mathsf{int} \quad \Psi; \cdot \vdash e_2 : \mathsf{int} \quad \Psi\ ;\ e_1 \to \mathsf{int}(A) \quad \Psi\ ;\ e_2 \to \mathsf{int}(B) \quad V = A\ \mathsf{op}\ B}{\Psi\ ;\ e_1\ \mathsf{op}\ e_2 \to \mathsf{int}(V)}\ \mathsf{math\ int}$$

$$\frac{\Psi;\cdot\vdash e_1 : \mathsf{float} \quad \Psi;\cdot\vdash e_2 : \mathsf{float} \quad \Psi\ ;\ e_1 \rightarrow \mathsf{float}(A) \quad \Psi\ ;\ e_2 \rightarrow \mathsf{float}(B) \quad V = A\ \mathsf{op}\ B}{\Psi\ ;\ e_1\ \mathsf{op}\ e_2 \rightarrow \mathsf{float}(V)}\ \textsf{math float}$$

$$\frac{\Psi;\cdot\vdash e_1 : \mathsf{int} \quad \Psi;\cdot\vdash e_2 : \mathsf{float} \quad \Psi\ ;\ e_1 \rightarrow \mathsf{int}(A) \quad \Psi\ ;\ e_2 \rightarrow \mathsf{float}(B) \quad V = A\ \mathsf{op}\ B}{\Psi\ ;\ e_1\ \mathsf{op}\ e_2 \rightarrow \mathsf{float}(V)}\ \textsf{math cast1}$$

$$\frac{\Psi;\cdot\vdash e_1 : \mathsf{float} \quad \Psi;\cdot\vdash e_2 : \mathsf{int} \quad \Psi\ ;\ e_1 \rightarrow \mathsf{float}(A) \quad \Psi\ ;\ e_2 \rightarrow \mathsf{int}(B) \quad V = A\ \mathsf{op}\ B}{\Psi\ ;\ e_1\ \mathsf{op}\ e_2 \rightarrow \mathsf{float}(V)}\ \textsf{math cast2}$$

$$\frac{\Psi\ ;\ c \rightarrow \mathsf{bool}(true) \quad \Psi\ ;\ e_1 \rightarrow v_1}{\Psi\ ;\ \mathsf{if}\ c\ \mathsf{then}\ e_1\ \mathsf{else}\ e_2\ \mathsf{end} \rightarrow v_1}\ \textsf{if true} \qquad \frac{\Psi\ ;\ c \rightarrow \mathsf{bool}(false) \quad \Psi\ ;\ e_2 \rightarrow v_2}{\Psi\ ;\ \mathsf{if}\ c\ \mathsf{then}\ e_1\ \mathsf{else}\ e_2\ \mathsf{end} \rightarrow v_2}\ \textsf{if false}$$

$$\frac{\Psi\ ;\ e_1 \rightarrow \mathsf{int}(A) \quad \Psi\ ;\ e_2 \rightarrow \mathsf{int}(B) \quad V = A\ \mathsf{cmp}\ B}{\Psi\ ;\ e_1\ \mathsf{cmp}\ e_2 \rightarrow \mathsf{bool}(V)}\ \textsf{cmp int}$$

$$\frac{\Psi\ ;\ e_1 \rightarrow \mathsf{float}(A) \quad \Psi\ ;\ e_2 \rightarrow \mathsf{float}(B) \quad V = A\ \mathsf{cmp}\ B}{\Psi\ ;\ e_1\ \mathsf{cmp}\ e_2 \rightarrow \mathsf{bool}(V)}\ \textsf{cmp float}$$

$$\frac{\Psi\ ;\ e_1 \rightarrow \mathsf{bool}(A) \quad \Psi\ ;\ e_2 \rightarrow \mathsf{bool}(B) \quad V = A\ \mathsf{cmp}\ B}{\Psi\ ;\ e_1\ \mathsf{cmp}\ e_2 \rightarrow \mathsf{bool}(V)}\ \textsf{cmp bool}$$

$$\frac{\Psi\ ;\ e_1 \rightarrow \mathsf{string}(A) \quad \Psi\ ;\ e_2 \rightarrow \mathsf{string}(B) \quad V = A\ \mathsf{cmp}\ B}{\Psi\ ;\ e_1\ \mathsf{cmp}\ e_2 \rightarrow \mathsf{bool}(V)}\ \textsf{cmp string}$$

$$\frac{\Psi\ ;\ e_1 \rightarrow \mathsf{addr}(A) \quad \Psi\ ;\ e_2 \rightarrow \mathsf{addr}(B) \quad V = A\ \mathsf{cmp}\ B}{\Psi\ ;\ e_1\ \mathsf{cmp}\ e_2 \rightarrow \mathsf{bool}(V)}\ \textsf{cmp addr}$$

$$\frac{\Psi\ ;\ e_1 \rightarrow \mathsf{bool}(A) \quad \Psi\ ;\ e_2 \rightarrow \mathsf{bool}(B) \quad V = A\ \mathsf{or}\ B}{\Psi\ ;\ e_1\ \mathsf{or}\ e_2 \rightarrow \mathsf{bool}(V)}\ \textsf{or}$$

$$\frac{\Psi\ ;\ e_1 \rightarrow v_1 \quad \Psi\ ;\ [v_1/x]e_2 \rightarrow v}{\Psi\ ;\ \mathsf{let}\ X\ =\ e_1\ \mathsf{in}\ e_2\ \mathsf{end} \rightarrow v}\ \textsf{let}$$

$$\frac{\mathsf{const}(name, v) : \tau \in \Psi}{\Psi\ ;\ \mathsf{getconst}(name) \rightarrow v}\ \textsf{const}$$

$$\frac{\begin{array}{l}\Psi\ ;\ e_1 \rightarrow v \\ \mathsf{external}(name, arg : \mathsf{struct}\ \tau) : \tau_{\mathsf{r}} \in \Psi \\ \quad v = \mathsf{callexternal}(name, v)\end{array}}{\Psi\ ;\ \mathsf{callexternal}(name, e) \rightarrow v}\ \textsf{external}$$

$$\frac{\begin{array}{c}\Psi\ ;\ e \to v' \\ \mathsf{fun}(name, arg : \mathsf{struct}\ \tau, e) : \tau_{\mathsf{r}} \in \Psi \\ \Psi\ ;\ [v'/\mathsf{arg}]e \to v\end{array}}{\Psi\ ;\ \mathsf{callfun}(name, e) \to v}\ \mathsf{fun}$$

$$\frac{}{\Psi\ ;\ \mathsf{world} \to \mathsf{int}(N)}\ \mathsf{world} \qquad \frac{}{\Psi\ ;\ \mathsf{arg}(N) \to \mathsf{string}(S)}\ \mathsf{arg}$$

2.3 Aggregates

$$\frac{}{[\mathsf{count}] \hookrightarrow \mathsf{int}(0)}\ \mathsf{init\ count}$$

$$\frac{}{[\mathsf{collect\ int}] \hookrightarrow []}\ \mathsf{init\ collect\ int}$$

$$\frac{}{[\mathsf{sum}] \hookrightarrow \mathsf{int}(0)}\ \mathsf{init\ sum} \qquad \frac{}{[\mathsf{max}] \hookrightarrow \mathsf{int}(-\infty)}\ \mathsf{init\ max} \qquad \frac{}{[\mathsf{min}] \hookrightarrow \mathsf{int}(+\infty)}\ \mathsf{init\ min}$$

$$\frac{}{[\mathsf{sum}]\ A/B \Rightarrow A + B}\ \mathsf{op\ sum}$$

$$\frac{}{[\mathsf{count}]\ A/B \Rightarrow A + 1}\ \mathsf{op\ count}$$

$$\frac{}{[\mathsf{collect\ int}]\ A/B \Rightarrow [B|A]}\ \mathsf{op\ collect\ int}$$

$$\frac{}{[\mathsf{min}]\ A/B \Rightarrow \mathsf{if}\ A \leq B\ \mathsf{then}\ A\ \mathsf{else}\ B\ \mathsf{end}}\ \mathsf{op\ min}$$

$$\frac{}{[\mathsf{max}]\ A/B \Rightarrow \mathsf{if}\ A \leq B\ \mathsf{then}\ B\ \mathsf{else}\ A\ \mathsf{end}}\ \mathsf{op\ max}$$

2.4 Global Semantics

Meaning of variables:

Ψ : Program state: constants, functions, external functions and declarations.

Θ : Rules with priority.

Φ : Rules without priority.

Γ : Persistent fact context.

Δ : Linear fact context.

$$\frac{\mathsf{apply}\ \Psi; \Gamma; \Delta, [R] \to \Gamma'; \Delta'; \Xi}{\Psi; \Theta, \Phi, R; \Gamma; \Delta \Rightarrow \Gamma'; \Delta'; \Xi}\ \mathsf{rule\ app}$$

$$\frac{\mathsf{match}\ \Psi; \Gamma; \Delta_1 \to [A]; \Delta_1 \quad \mathsf{derive}\ \Psi; \Gamma; \Delta_2; [B]; \cdot; \cdot \to \Gamma'; \Delta'; \Xi}{\mathsf{apply}\ \Psi; \Gamma; \Delta_1, \Delta_2, [A \multimap B] \to \Gamma'; \Delta'; \Delta_1, \Xi}\ \multimap \mathsf{L}$$

$$\frac{\mathsf{val}\ M : \tau \quad \mathsf{apply}\ \Psi; \Gamma; \Delta, [A\{M/X\}] \to \Gamma'; \Delta'; \Xi}{\mathsf{apply}\ \Psi; \Gamma; \Delta, [\forall X : \tau . A] \to \Gamma'; \Delta'; \Xi}\ \forall \mathsf{L}$$

2.4.1 Match

$$\frac{\text{match } \Psi; \Gamma; \Delta \rightarrow [[M/X]A]; \Xi}{\text{match } \Psi; \Gamma; \Delta \rightarrow [\exists X.A]; \Xi} \text{ match exists}$$

$$\frac{}{\text{match } \Psi; \Gamma; \cdot \rightarrow [1]; \cdot} \text{ match one}$$

$$\frac{\text{match } \Psi; \Gamma; \Delta \rightarrow [A]; \Xi_1 \quad \text{match } \Psi; \Gamma; \Delta' \rightarrow [B]; \Xi_2}{\text{match } \Psi; \Gamma; \Delta, \Delta' \rightarrow [A \otimes B]; \Xi_1, \Xi_2} \text{ match split}$$

$$\frac{\Psi \; ; \; e \rightarrow \text{bool(true)}}{\text{match } \Psi; \Gamma; \cdot \rightarrow [!\text{constraint } e]; \cdot} \text{ match end constraint}$$

$$\frac{v_1 = v_1' \quad ... \quad v_n = v_n'}{\text{match } \Psi; \Gamma; name[@v_1](v_2, ..., v_n) \rightarrow [name[@v_1](v_2, ..., v_n)]; name[@v_1](v_2, ..., v_n)} \text{ match end linear}$$

$$\frac{v_1 = v_1' \quad ... \quad v_n = v_n'}{\text{match } \Psi; \Gamma, !name[@v_1](v_2, ..., v_n); \cdot \rightarrow [!name[@v_1'](v_2', ..., v_n')]; \cdot} \text{ match end persistent}$$

$$\frac{}{\text{int}(N) = \text{int}(N)} \text{ equal int} \qquad \frac{}{\text{float}(F) = \text{float}(F)} \text{ equal float} \qquad \frac{}{\text{addr}(A) = \text{addr}(A)} \text{ equal addr}$$

$$\frac{}{\text{string}(S) = \text{string}(S)} \text{ equal string} \qquad \frac{}{\text{bool}(B) = \text{bool}(B)} \text{ equal bool}$$

$$\frac{}{[] = []} \text{ equal nil} \qquad \frac{x = x' \quad ls = ls'}{x :: ls = x' :: ls'} \text{ equal cons}$$

2.5 Derive

$$\frac{\text{derive } \Psi; \Gamma; \Delta; A; \Delta_1; \Gamma_1 \rightarrow \Gamma'; \Delta'; \Xi}{\text{derive } \Psi; \Gamma; \Delta; [A]; \Delta_1; \Gamma_1 \rightarrow \Gamma'; \Delta'; \Xi} \text{ derive blur}$$

$$\frac{\text{derive } \Psi; \Gamma; \Delta; A, B, \Omega; \Delta_1; \Gamma_1 \rightarrow \Gamma'; \Delta'; \Xi}{\text{derive } \Psi; \Gamma; \Delta; A \otimes B, \Omega; \Delta_1; \Gamma_1 \rightarrow \Gamma'; \Delta'; \Xi} \text{ derive } \otimes$$

$$\frac{\text{derive } \Psi; \Gamma; \Delta; [x/X]A, \Omega; \Delta_1; \Gamma_1 \rightarrow \Gamma'; \Delta'; \Xi \quad x = \text{new addr}(A)}{\text{derive } \Psi; \Gamma; \Delta; \exists X : \text{addr}.A, \Omega; \Delta_1; \Gamma_1 \rightarrow \Gamma'; \Delta'; \Xi} \text{ derive exists}$$

$$\frac{\begin{array}{c}\Psi \; ; \; e_1 \rightarrow v_1 \quad ... \quad \Psi \; ; \; e_n \rightarrow v_n \\ \text{derive } \Psi; \Gamma; \Delta; \Omega; \Delta_1, name[@v_1](v_2, ..., v_n); \Gamma_1 \rightarrow \Gamma'; \Delta'; \Xi\end{array}}{\text{derive } \Psi; \Gamma; \Delta; name[@e_1](e_2, ..., e_n), \Omega; \Delta_1; \Gamma_1 \rightarrow \Gamma'; \Delta'; \Xi} \text{ derive fact}$$

$$\frac{\begin{array}{c}\Psi \; ; \; e_1 \rightarrow v_1 \quad ... \quad \Psi \; ; \; e_n \rightarrow v_n \\ \text{derive } \Psi; \Gamma; \Delta; \Omega; \Delta_1; \Gamma_1, !name[@v_1](v_2, ..., v_n) \rightarrow \Gamma'; \Delta'; \Xi\end{array}}{\text{derive } \Psi; \Gamma; \Delta; !name[@e_1](e_2, ..., e_n), \Omega; \Delta_1; \Gamma_1 \rightarrow \Gamma'; \Delta'; \Xi} \text{ derive !fact}$$

$$\frac{\text{derive } \Psi;\Gamma;\Delta;\Omega;\Delta_1;\Gamma_1 \to \Gamma';\Delta';\Xi}{\text{derive } \Psi;\Gamma;\Delta;1,\Omega;\Delta_1;\Gamma_1 \to \Gamma';\Delta';\Xi}\ \text{derive 1}$$

$$\frac{\text{derive } \Psi;\Gamma;\Delta;1 \mathbin{\&} (A \otimes \text{comp } A),\Omega;\Delta_1;\Gamma_1 \to \Gamma';\Delta';\Xi}{\text{derive } \Psi;\Gamma;\Delta;\text{comp } A,\Omega;\Delta_1;\Gamma_1 \to \Gamma';\Delta';\Xi}\ \text{derive comprehension}$$

$$\frac{\text{derive } \Psi;\Gamma;\Delta;A,\Omega;\Delta_1;\Gamma_1 \to \Gamma';\Delta';\Xi}{\text{derive } \Psi;\Gamma;\Delta;A \mathbin{\&} B,\Omega;\Delta_1;\Gamma_1 \to \Gamma';\Delta';\Xi}\ \text{derive } \mathbin{\&} \text{ left}$$

$$\frac{\text{derive } \Psi;\Gamma;\Delta;B,\Omega;\Delta_1;\Gamma_1 \to \Gamma';\Delta';\Xi}{\text{derive } \Psi;\Gamma;\Delta;A \mathbin{\&} B,\Omega;\Delta_1;\Gamma_1 \to \Gamma';\Delta';\Xi}\ \text{derive } \mathbin{\&} \text{ right}$$

$$\frac{[\text{Op}] \hookrightarrow V \quad \text{derive } \Psi;\Gamma;\Delta;\text{agg Op } V\ (x.A(x))\ (y.B(y)),\Omega;\Delta_1;\Gamma_1 \to \Gamma';\Delta';\Xi}{\text{derive } \Psi;\Gamma;\Delta;[\text{Op};X;A \Rightarrow B],\Omega;\Delta_1;\Gamma_1 \to \Gamma';\Delta';\Xi}\ \text{derive aggregate}$$

$$\frac{\begin{array}{r}\text{derive } \Psi;\Gamma;\Delta;B(V) \mathbin{\&} (\forall X'.A(X') \multimap \text{agg Op } E\ (x.A(x))\ (y.B(y))),\Omega;\Delta_1;\Gamma_1 \to \Gamma';\Delta';\Xi \\ [\text{Op}]\ V'/X' \Rightarrow E\end{array}}{\text{derive } \Psi;\Gamma;\Delta;\text{agg Op } V'\ (x.A(x))\ (y.B(y)),\Omega;\Delta_1;\Gamma_1 \to \Gamma';\Delta';\Xi}\ \text{derive aggregate unfold}$$

$$\frac{\text{derive } \Psi;\Gamma;\Delta;[M/X]A,\Omega;\Delta_1;\Gamma_1 \to \Gamma';\Delta';\Xi \quad \text{val } M : \tau}{\text{derive } \Psi;\Gamma;\Delta;\forall X : \tau.A,\Omega;\Delta_1;\Gamma_1 \to \Gamma';\Delta';\Xi}\ \text{derive forall}$$

$$\frac{\text{match } \Psi;\Gamma;\Delta \to A;\Xi' \quad \text{derive } \Psi;\Gamma;\Delta - \Xi';B,\Omega;\Delta_1;\Gamma_1 \to \Gamma';\Delta';\Xi}{\text{derive } \Psi;\Gamma;\Delta;A \multimap B,\Omega;\Delta_1;\Gamma_1 \to \Gamma';\Delta';\Xi,\Xi'}\ \text{derive lolli}$$

$$\frac{}{\text{derive } \Psi;\Gamma;\Delta;\cdot;\Delta_1;\Gamma_1 \to \Gamma_1;\Delta_1;\cdot}\ \text{derive end}$$

2.6 Local Semantics

$$\frac{\text{apply } \Psi;\Gamma;\Delta,[R] \to \Gamma';\Delta';\Xi \mathbin{@} \pi \quad N = \Delta - \Xi}{\Psi;\Theta,R;\Gamma;\Delta \Rightarrow \Gamma,\Gamma';\Delta',N;\ \mathbin{@} \pi}\ \text{rule app}$$

$$\frac{\text{match } \Psi;\Gamma;\Delta_1 \to [A];\Xi \mathbin{@} \pi \quad \text{derive } \Psi;\Gamma;\Delta_2;[B];\cdot;\cdot \to \Gamma';\Delta';\Xi}{\text{apply } \Psi;\Gamma;\Delta_1,\Delta_2,[A \multimap B] \to \Gamma';\Delta';\Xi \mathbin{@} \pi}\ \multimap\text{L}$$

$$\frac{\text{val } M : \tau \quad \text{apply } \Psi;\Gamma;\Delta,[A\{M/X\}] \to \Gamma';\Delta';\Xi \mathbin{@} \pi}{\text{apply } \Psi;\Gamma;\Delta,[\forall X : \tau.A] \to \Gamma';\Delta';\Xi \mathbin{@} \pi}\ \forall\text{L}$$

2.6.1 Match

$$\frac{\textsf{match } \Psi;\Gamma;\Delta \rightarrow [[M/X]A];\Xi \mathbin{@} \pi}{\textsf{match } \Psi;\Gamma;\Delta \rightarrow [\exists X.A];\Xi \mathbin{@} \pi}\ \textsf{match exists}$$

$$\frac{}{\textsf{match } \Psi;\Gamma;\cdot \rightarrow [1];\cdot \mathbin{@} \pi}\ \textsf{match one}$$

$$\frac{\textsf{match } \Psi;\Gamma;\Delta \rightarrow [A];\Xi_1 \mathbin{@} \pi \quad \textsf{match } \Psi;\Gamma;\Delta' \rightarrow [B];\Xi_2 \mathbin{@} \pi}{\textsf{match } \Psi;\Gamma;\Delta,\Delta' \rightarrow [A \otimes B];\Xi_1,\Xi_2 \mathbin{@} \pi}\ \textsf{match split}$$

$$\frac{\Psi\ ;\ e \rightarrow \textsf{bool(true)}}{\textsf{match } \Psi;\Gamma;\cdot \rightarrow [!\textsf{constraint } e];\cdot \mathbin{@} \pi}\ \textsf{match end constraint}$$

$$\frac{v_1 = v_1' \quad \ldots \quad v_n = v_n' \quad v_1 = \textsf{addr}(\pi)}{\textsf{match } \Psi;\Gamma; name[@v_1](v_2,\ldots,v_n) \rightarrow [name[@v_1](v_2,\ldots,v_n)]; name[@v_1](v_2,\ldots,v_n) \mathbin{@} \pi}\ \textsf{match end linear}$$

$$\frac{v_1 = v_1' \quad \ldots \quad v_n = v_n' \quad v_1 = \textsf{addr}(\pi)}{\textsf{match } \Psi;\Gamma, !name[@v_1](v_2,\ldots,v_n);\cdot \rightarrow [!name[@v_1'](v_2',\ldots,v_n')];\cdot \mathbin{@} \pi}\ \textsf{match end persistent}$$

2.7 Extending Linear Logic with Comprehensions

$$\textsf{comp } A\ B \triangleq 1 \mathbin{\&} ((\forall X.A \multimap B) \otimes \textsf{comp } A\ B)$$

$$\textsf{agg } V\ A\ C \triangleq C \mathbin{\&} (\forall X.A \multimap \textsf{agg } (X+V)\ A\ C)$$

An example from Meld:

```
a(H) -o [sum => S | B | !edge(H, B), !weight(H, B, S) | total(H, S)].

a(H) -o agg1(H, 0).

agg1(H, V) := total(H, V) &
            (forall B, S. !edge(H, B), !weight(H, B, S) -o agg1(H, S + V)).
```

These would be the left and right rules for definitions:

$$\frac{\Delta, B\theta \vdash C \quad A \triangleq B \quad A' \doteq A\theta}{\Delta, \textsf{def } A' \vdash C}\ \textsf{def } L$$

$$\frac{\Delta \vdash B\theta \quad A \triangleq B \quad A' \doteq A\theta}{\Delta \vdash \textsf{def } A'}\ \textsf{def } R$$

Identity expansion:

$$\frac{\dfrac{\dfrac{}{B\theta \vdash B\theta}\ \textsf{id} \quad A \triangleq B \quad A' \doteq A\theta}{\textsf{def } A' \vdash B\theta}\ \textsf{def } L \quad A \triangleq B \quad A' \doteq A\theta}{\textsf{def } A' \vdash \textsf{def } A'}\ \textsf{def } R$$

Cut reduction:

$$\dfrac{\dfrac{\Delta \vdash B\theta \quad A \overset{\triangle}{=} B \quad A' \doteq A'\theta}{\Delta \vdash \mathsf{def}\ A'}\ \mathsf{def}\ R \qquad \dfrac{\Delta, B\theta \vdash C \quad A \overset{\triangle}{=} B \quad A' \doteq A\theta}{\Delta, \mathsf{def}\ A' \vdash C}\ \mathsf{def}\ L}{\Delta \vdash C}\ \mathsf{cut}$$

Reduces to:

$$\dfrac{\Delta, B\theta \vdash C \quad \Delta \vdash B\theta}{\Delta \vdash C}\ \mathsf{cut}$$

3 Linear Logic

$$\dfrac{}{\Psi; \Gamma; \cdot \vdash \mathbf{1}}\ \mathbf{1}R \qquad \dfrac{\Psi; \Gamma; \Delta \vdash C}{\Psi; \Gamma; \Delta, \mathbf{1} \vdash C}\ \mathbf{1}L$$

$$\dfrac{\Psi; \Gamma; \Delta \vdash A \quad \Gamma; \Delta \vdash B}{\Psi; \Gamma; \Delta \vdash A \mathbin{\&} B}\ \&R \qquad \dfrac{\Psi; \Gamma; \Delta, A \vdash C}{\Psi; \Gamma; \Delta, A \mathbin{\&} B \vdash C}\ \&L_1 \qquad \dfrac{\Psi; \Gamma; \Delta, B \vdash C}{\Psi; \Gamma; \Delta, B \mathbin{\&} B \vdash C}\ \&L_2$$

$$\dfrac{\Psi; \Gamma; \Delta \vdash A \quad \Gamma; \Delta \vdash B}{\Psi; \Gamma; \Delta, \Delta' \vdash A \otimes B}\ \otimes R \qquad \dfrac{\Psi; \Gamma; \Delta, A, B \vdash C}{\Psi; \Gamma; \Delta, A \otimes B \vdash C}\ \otimes L$$

$$\dfrac{\Psi; \Gamma; \Delta, A \vdash B}{\Psi; \Gamma; \Delta \vdash A \multimap B}\ \multimap R \qquad \dfrac{\Psi; \Gamma; \Delta \vdash A \quad \Psi; \Gamma; \Delta', B \vdash C}{\Psi; \Gamma; \Delta, \Delta', A \multimap B \vdash C}\ \multimap L$$

$$\dfrac{\Psi, m : \tau; \Gamma; \Delta \vdash A\{m/n\}}{\Psi; \Gamma; \Delta \vdash \forall n : \tau.A}\ \forall R \qquad \dfrac{\Psi \vdash M : \tau \quad \Psi; \Gamma; \Delta, A\{M/n\} \vdash C}{\Psi; \Gamma; \Delta, \forall n : \tau.A \vdash C}\ \forall L$$

$$\dfrac{\Psi \vdash M : \tau \quad \Psi\ \Gamma; \Delta \vdash A\{M/n\}}{\Psi\ \Gamma; \Delta \vdash \exists n : \tau.A}\ \exists R \qquad \dfrac{\Psi, m : \tau; \Gamma; \Delta, A\{m/n\} \vdash C}{\Psi; \Gamma; \Delta, \exists n : \tau.A \vdash C}\ \exists L$$

$$\dfrac{\Psi; \Gamma; \cdot \vdash A}{\Psi; \Gamma; \cdot \vdash !A}\ !R \qquad \dfrac{\Psi; \Gamma, A; \Delta \vdash C}{\Psi; \Gamma; \Delta, !A \vdash C}\ !L \qquad \dfrac{\Psi; \Gamma, A; \Delta, A \vdash C}{\Psi; \Gamma, A; \Delta \vdash C}\ \mathsf{copy}$$

$$\dfrac{\Psi; \Gamma; \Delta \vdash B\theta \quad A \overset{\triangle}{=} B \quad A' \doteq A\theta}{\Psi; \Gamma; \Delta \vdash \mathsf{def}\ A'}\ \mathsf{def}\ R \qquad \dfrac{\Psi; \Gamma; \Delta, B\theta \vdash C \quad A \overset{\triangle}{=} B \quad A' \doteq A\theta}{\Psi; \Gamma; \Delta, \mathsf{def}\ A' \vdash C}\ \mathsf{def}\ L$$

$$\dfrac{\Psi; \Gamma; \Delta \vdash A \quad \Psi; \Gamma; \Delta', A \vdash C}{\Psi; \Gamma; \Delta, \Delta' \vdash C}\ \mathsf{cut} \qquad \dfrac{\Psi; \Gamma; \cdot \vdash A \quad \Psi; \Gamma, A; \Delta \vdash C}{\Psi; \Gamma; \Delta \vdash C}\ \mathsf{cut!}$$

3.1 Cut Free System

$$\frac{}{\Psi;\Gamma;\cdot \Rightarrow \mathbf{1}}\ \mathbf{1}R \qquad \frac{\Psi;\Gamma;\Delta \Rightarrow C}{\Psi;\Gamma;\Delta,\mathbf{1} \Rightarrow C}\ \mathbf{1}L$$

$$\frac{\Psi;\Gamma;\Delta \Rightarrow A \quad \Gamma;\Delta \Rightarrow B}{\Psi;\Gamma;\Delta \Rightarrow A \,\&\, B}\ \&R \qquad \frac{\Psi;\Gamma;\Delta, A \Rightarrow C}{\Psi;\Gamma;\Delta, A \,\&\, B \Rightarrow C}\ \&L_1 \qquad \frac{\Psi;\Gamma;\Delta, B \Rightarrow C}{\Psi;\Gamma;\Delta, B \,\&\, B \Rightarrow C}\ \&L_2$$

$$\frac{\Psi;\Gamma;\Delta \Rightarrow A \quad \Gamma;\Delta \Rightarrow B}{\Psi;\Gamma;\Delta,\Delta' \Rightarrow A \otimes B}\ \otimes R \qquad \frac{\Psi;\Gamma;\Delta, A, B \Rightarrow C}{\Psi;\Gamma;\Delta, A \otimes B \Rightarrow C}\ \otimes L$$

$$\frac{\Psi;\Gamma;\Delta, A \Rightarrow B}{\Psi;\Gamma;\Delta \Rightarrow A \multimap B}\ \multimap R \qquad \frac{\Psi;\Gamma;\Delta \Rightarrow A \quad \Psi;\Gamma;\Delta', B \Rightarrow C}{\Gamma;\Delta,\Delta', A \multimap B \Rightarrow C}\ \multimap L$$

$$\frac{\Psi, m:\tau;\Gamma;\Delta \Rightarrow A\{m/n\}}{\Psi;\Gamma;\Delta \Rightarrow \forall n:\tau.A}\ \forall R \qquad \frac{\Psi \vdash M:\tau \quad \Psi;\Gamma;\Delta, A\{M/n\} \Rightarrow C}{\Psi;\Gamma;\Delta, \forall n:\tau.A \Rightarrow C}\ \forall L$$

$$\frac{\Psi \vdash M:\tau \quad \Psi;\Gamma;\Delta \Rightarrow A\{M/n\}}{\Psi;\Gamma;\Delta \Rightarrow \exists n:\tau.A}\ \exists R \qquad \frac{\Psi, m:\tau;\Gamma;\Delta, A\{m/n\} \Rightarrow C}{\Psi;\Gamma;\Delta, \exists n:\tau.A \Rightarrow C}\ \exists L$$

$$\frac{\Psi;\Gamma;\cdot \Rightarrow A}{\Psi;\Gamma;\cdot \Rightarrow !A}\ !R \qquad \frac{\Psi;\Gamma, A;\Delta \Rightarrow C}{\Psi;\Gamma;\Delta, !A \Rightarrow C}\ !L \qquad \frac{\Psi;\Gamma, A;\Delta, A \Rightarrow C}{\Psi;\Gamma, A;\Delta \Rightarrow C}\ \mathsf{copy}$$

$$\frac{\Psi;\Gamma;\Delta \Rightarrow B\theta \quad A \triangleq B \quad A' \doteq A\theta}{\Psi;\Gamma;\Delta \Rightarrow \mathsf{def}\ A'}\ \mathsf{def}\ R \qquad \frac{\Psi;\Gamma;\Delta, B\theta \Rightarrow C \quad A \triangleq B \quad A' \doteq A\theta}{\Psi;\Gamma;\Delta, \mathsf{def}\ A' \Rightarrow C}\ \mathsf{def}\ L$$

3.2 Cut Elimination Theorem

If $\Gamma;\Delta \Rightarrow A$ and $\Gamma;\Delta', A \Rightarrow C$ then $\Gamma;\Delta,\Delta' \Rightarrow C$

4 Basic Low Level System With Comprehensions

4.1 High Level System

4.1.1 Match

$$\frac{}{\mathsf{match}\ \cdot \to 1}\ \mathsf{match}\ 1 \qquad \frac{}{\mathsf{match}\ p \to p}\ \mathsf{match}\ p$$

$$\frac{\mathsf{match}\ \Delta_1 \to A \quad \mathsf{match}\ \Delta_2 \to B}{\mathsf{match}\ \Delta_1,\Delta_2 \to A \otimes B}\ \mathsf{match}\ \otimes$$

4.1.2 Derive

$$\frac{\mathsf{derive}\ \Delta; \Xi; p, \Delta_1; \Omega \to \Xi'; \Delta'}{\mathsf{derive}\ \Delta; \Xi; \Delta_1; p, \Omega \to \Xi'; \Delta'}\ \mathsf{derive}\ p$$

$$\frac{\mathsf{derive}\ \Delta; \Xi; \Delta_1; A, B, \Omega \to \Xi'; \Delta'}{\mathsf{derive}\ \Delta; \Xi; \Delta_1; A \otimes B, \Omega \to \Xi'; \Delta'}\ \mathsf{derive}\ \otimes$$

$$\frac{\mathsf{derive}\ \Delta; \Xi; \Delta_1; \Omega \to \Xi'; \Delta'}{\mathsf{derive}\ \Delta; \Xi; \Delta_1; 1, \Omega \to \Xi'; \Delta'}\ \mathsf{derive}\ 1$$

$$\frac{}{\mathsf{derive}\ \Delta; \Xi'; \Delta'; \cdot \to \Xi'; \Delta'}\ \mathsf{derive}\ end$$

$$\frac{\mathsf{derive}\ \Delta; \Xi; \Delta_1; 1 \,\&\, (A \multimap B \otimes \mathsf{comp}\ A \multimap B), \Omega \to \Xi'; \Delta'}{\mathsf{derive}\ \Delta; \Xi; \Delta_1; \mathsf{comp}\ A \multimap B, \Omega \to \Xi'; \Delta'}\ \mathsf{derive}\ comp$$

$$\frac{\mathsf{match}\ \Delta_a \to A \quad \mathsf{derive}\ \Delta_b; \Xi, \Delta_a; \Delta_1; B, \Omega \to \Xi'; \Delta'}{\mathsf{derive}\ \Delta_a, \Delta_b; \Xi; \Delta_1; A \multimap B, \Omega \to \Xi'; \Delta'}\ \mathsf{derive}\ \multimap$$

$$\frac{\mathsf{derive}\ \Delta; \Xi; \Delta_1; A, \Omega \to \Xi'; \Delta'}{\mathsf{derive}\ \Delta; \Xi; \Delta_1; A \,\&\, B, \Omega \to \Xi'; \Delta'}\ \mathsf{derive}\ \&L \qquad \frac{\mathsf{derive}\ \Delta; \Xi; \Delta_1; B, \Omega \to \Xi'; \Delta'}{\mathsf{derive}\ \Delta; \Xi; \Delta_1; A \,\&\, B, \Omega \to \Xi'; \Delta'}\ \mathsf{derive}\ \&R$$

4.1.3 Apply

$$\frac{\mathsf{match}\ \Delta \to A \quad \mathsf{derive}\ \Delta''; \Delta; \cdot; B \to \Xi'; \Delta'}{\mathsf{a}_0\ \Delta, \Delta''; A \multimap B \to \Xi'; \Delta'}\ \mathsf{a}_0\ rule$$

$$\frac{\mathsf{do}_0\ \Delta; R \to \Xi'; \Delta'}{\mathsf{do}_0\ \Delta; R, \Phi \to \Xi'; \Delta'}\ \mathsf{do}_0\ rule$$

4.2 Low Level System

4.2.1 Match

$$\frac{\mathsf{m}_1\ \Delta; \Xi, p; \Omega; H; C \to \Xi'; \Delta'}{\mathsf{m}_1\ \Delta, p; \Xi; p, \Omega; H; C \to \Xi'; \Delta'}\ ok \qquad \frac{p \notin \Delta \quad \mathsf{cont}\ C; H; \Xi'; \Delta'}{\mathsf{m}_1\ \Delta; \Xi; p, \Omega; H; C \to \Xi'; \Delta'}\ fail$$

$$\frac{\mathsf{m}_1\ \Delta; \Xi; A, B, \Omega; H; C \to \Xi'; \Delta'}{\mathsf{m}_1\ \Delta; \Xi; A \otimes B, \Omega; H; C \to \Xi'; \Delta'}\ \otimes$$

$$\frac{\mathsf{d}_1\ \Delta; \Xi; \cdot; H; \cdot \to \Xi'; \Delta'}{\mathsf{m}_1\ \Delta; \Xi; \cdot; H; (\Phi; \Delta'') \to \Xi'; \Delta'}\ \mathsf{rule\ cont}$$

4.2.2 Derive

$$\frac{\mathsf{d}_1\ \Delta; \Xi; p, \Delta_1; \Omega; C \to \Xi'; \Delta'}{\mathsf{d}_1\ \Delta; \Xi; \Delta_1; p, \Omega; C \to \Xi'; \Delta'}\ p \qquad \frac{\mathsf{d}_1\ \Delta; \Xi; \Delta_1; \Omega; C \to \Xi'; \Delta'}{\mathsf{d}_1\ \Delta; \Xi; \Delta_1; 1, \Omega; C \to \Xi'; \Delta'}\ 1$$

$$\frac{\mathsf{d}_1\ \Delta; \Xi; \Delta_1; A, B, \Omega; C \to \Xi'; \Delta'}{\mathsf{d}_1\ \Delta; \Xi; \Delta_1; A \otimes B, \Omega; C \to \Xi'; \Delta'}\ \otimes$$

$$\frac{\mathsf{a}_1\ \Delta; A \multimap B; (\mathsf{d}_1\ \Delta; \Xi; \Delta_1; \mathsf{comp}\ A \multimap B, \Omega; \cdot) \to \Xi'; \Delta'}{\mathsf{d}_1\ \Delta; \Xi; \Delta_1; \mathsf{comp}\ A \multimap B, \Omega; \cdot \to \Xi'; \Delta'}\ \mathsf{comp}$$

$$\frac{\mathsf{d}_1\ \Delta; \Xi; \cdot; H; (\mathsf{d}_1\ \Delta''; \Xi''; \Delta_1; \mathsf{comp}\ A \multimap B, \Omega; \cdot) \to \Xi'; \Delta'}{\mathsf{m}_1\ \Delta; \Xi; \cdot; H; (\mathsf{d}_1\ \Delta''; \Xi''; \Delta_1; \mathsf{comp}\ A \multimap B, \Omega; \cdot) \to \Xi'; \Delta'}\ \mathsf{comp\ derivation}$$

$$\frac{}{\mathsf{d}_1\ \Delta; \Xi; \Delta_1; \cdot; \cdot \to \Xi; \Delta_1}\ \mathsf{end}$$

$$\frac{\mathsf{d}_1\ \Delta; \Xi, \Xi''; \Delta_1, \Delta_1''; \mathsf{comp}\ A \multimap B, \Omega; \cdot \to \Xi'; \Delta'}{\mathsf{d}_1\ \Delta; \Xi; \Delta_1; \cdot; (\mathsf{d}_1\ \Delta''; \Xi''; \Delta_1''; \mathsf{comp}\ A \multimap B, \Omega) \to \Xi'; \Delta'}\ \mathsf{next\ comp}$$

4.2.3 Continuations

$$\frac{\mathsf{do}_1\ \Delta; \Phi \to \Xi'; \Delta'}{\mathsf{cont}\ (\Phi; \Delta); \Xi'; \Delta'}\ \mathsf{rule\ fail}$$

$$\frac{\mathsf{d}_1\ \Delta; \Xi''; \Delta_1; \Omega; \cdot \to \Xi'; \Delta'}{\mathsf{cont}\ (\mathsf{d}_1\ \Delta''; \Xi''; \Delta_1; \mathsf{comp}\ A \multimap B, \Omega); \Xi'; \Delta'}\ \mathsf{comp\ done}$$

4.2.4 Apply

$$\frac{\mathsf{m}_1\ \Delta; \cdot; A; B; C \to \Xi'; \Delta'}{\mathsf{a}_1\ \Delta; A \multimap B; C \to \Xi'; \Delta'}\ \mathsf{apply\ rule}$$

$$\frac{\mathsf{a}_1\ \Delta; R; (\Phi; \Delta) \to \Xi'; \Delta'}{\mathsf{do}_1\ \Delta; R, \Phi \to \Xi'; \Delta'}\ \mathsf{pick\ rule}$$

4.3 Low level comprehension match succeeds or fails

If $\mathsf{m}_1\ \Delta'', \Delta_1, ..., \Delta_n; \Xi; \Omega; H; (\mathsf{d}_1\ \Delta'''; \Xi''; \Delta_1; \mathsf{comp}\ A \multimap B, \Omega') \to \Xi'; \Delta'$ then either:

- $\mathsf{cont}\ (\mathsf{d}_1\ \Delta'''; \Xi''; \Delta_1; \mathsf{comp}\ A \multimap B, \Omega'); \Xi'; \Delta'$ or
- $\mathsf{m}_1\ \Delta''; \Xi, \Delta_1, ..., \Delta_n'\cdot; H; (\mathsf{d}_1\ ...) \to \Xi'; \Delta'$ and $\Omega = \Omega_1, ..., \Omega_n$ where $\mathsf{match}\ \Delta_1 \to \Omega_1$, ..., $\mathsf{match}\ \Delta_n \to \Omega_n$ and $\Delta_1, ..., \Delta_n$ is not empty.

It's trivial by induction on the assumption, except the case p, Ω and $A \otimes B, \Omega$.

4.4 Low level comprehension gives one match

If $\mathsf{m}_1\ \Delta'', \Xi''; \cdot; A; H; (\mathsf{d}_1\ \Delta'''; \Xi; \Delta_1; \mathsf{comp}\ A \multimap B, \Omega') \rightarrow \Xi'; \Delta'$ then either

- $\mathsf{cont}\ (\mathsf{d}_1\ \Delta'''; \Xi; \Delta_1; \mathsf{comp}\ A \multimap B, \Omega'); \Xi'; \Delta'$ or
- $\mathsf{m}_1\ \Delta''; \Xi''; \cdot; H; (\mathsf{d}_1\ \Delta'''; \Xi; \Delta_1; \mathsf{comp}\ A \multimap B, \Omega') \rightarrow \Xi'; \Delta'$ and $\mathsf{match}\ \Xi'' \rightarrow A$, where Ξ'' is not empty

This follows trivially from the previous theorem.

4.5 Comprehension head is another derivation theorem

If $\mathsf{d}_1\ \Delta; \Xi; \Delta_1; \Omega'; (\mathsf{d}_1\ \Delta''; \Xi''; \Delta_1'; \mathsf{comp}\ A \multimap B, \Omega) \rightarrow \Xi'; \Delta'$
then
$\mathsf{d}_1\ \Delta; \Xi, \Xi''; \Delta_1, \Delta_1'; \Omega', \mathsf{comp}\ A \multimap B, \Omega; \cdot \rightarrow \Xi'; \Delta'$.

- p

  $\mathsf{d}_1\ \Delta; \Xi; \Delta_1; p, \Omega''; (\mathsf{d}_1\ ...) \rightarrow \Xi'; \Delta'$ (1) assumption
  $\Omega' = p, \Omega''$ (2) from (1)
  $\mathsf{d}_1\ \Delta; \Xi; p, \Delta_1; \Omega''; (\mathsf{d}_1\ ...) \rightarrow \Xi'; \Delta'$ (3) inversion of (1)
  $\mathsf{d}_1\ \Delta; \Xi, \Xi''; p, \Delta_1, \Delta_1'; \Omega'', \mathsf{comp}\ A \multimap B, \Omega; \cdot \rightarrow \Xi'; \Delta'$ (4) i.h. on (3)
  $\mathsf{d}_1\ \Delta; \Xi, \Xi''; \Delta_1, \Delta_1'; p, \Omega'', \mathsf{comp}\ A \multimap B, \Omega; \cdot \rightarrow \Xi'; \Delta'$ (5) apply rule on (4)

- $A \otimes B$

  $\mathsf{d}_1\ \Delta; \Xi, \Xi''; \Delta_1, \Delta_1'; A, B, \Omega''; \mathsf{comp}\ A \multimap B, \Omega; \cdot \rightarrow \Xi'; \Delta'$ (1) by i.h.
  $\mathsf{d}_1\ \Delta; \Xi, \Xi''; \Delta_1, \Delta_1'; A \otimes B, \Omega''; \mathsf{comp}\ A \multimap B, \Omega; \cdot \rightarrow \Xi'; \Delta'$ (2) rule application on (1)

- $\cdot$

  $\mathsf{d}_1\ \Delta; \Xi; \Delta_1; \cdot; (\mathsf{d}_1\ \Delta''; \Xi''; \Delta_1'; \mathsf{comp}\ A \multimap B, \Omega) \rightarrow \Xi'; \Delta'$ (1) assumption
  $\mathsf{d}_1\ \Delta; \Xi, \Xi''; \Delta_1, \Delta_1'; \mathsf{comp}\ A \multimap B, \Omega; \cdot \rightarrow \Xi'; \Delta'$ (2) inversion of (1)

4.6 Low level matching gives high level matching theorem

If
$\mathsf{m}_1\ \Delta, \Delta_1, ..., \Delta_n; \Xi; A_1, ..., A_n; H; \cdot \rightarrow \Xi'; \Delta'$
then
$\mathsf{match}\ \Delta_1 \rightarrow A_1$ through $\mathsf{match}\ \Delta_n \rightarrow A_n$ and
$\mathsf{m}_1\ \Delta; \Xi, \Delta_1, ..., \Delta_n; \cdot; H; \cdot \rightarrow \Xi'; \Delta'$ and
$\Delta_1, ..., \Delta_n$ is not empty if $A_1, ..., A_n$ is not $\cdot$.
Induction on $\Omega = A_1, ..., A_n$.

- p, Ω and $p \notin \Delta$

  Not applicable.

- p, Ω

  $\mathsf{m}_1\ \Delta, \Delta_1, ..., \Delta_n, p; \Xi; p, A_1, ..., A_n; H; \cdot \rightarrow \Xi'; \Delta'$ (1) assumption
  $\mathsf{m}_1\ \Delta, \Delta_1, ..., \Delta_n; \Xi, p; A_1, ..., A_n; H; \cdot \rightarrow \Xi'; \Delta'$ (2) inversion of (1)
  $\mathsf{match}\ \Delta_1 \rightarrow A_1\ ...\ \mathsf{match}\ \Delta_n \rightarrow A_n$ (3) induction on (2)
  $\mathsf{m}_1\ \Delta; \Xi, p, \Delta_1, ..., \Delta_n; \cdot; H; \cdot \rightarrow \Xi'; \Delta'$ (4) induction on (2)
  $\mathsf{match}\ p \rightarrow p$ (5) axiom
  $p, \Delta_1, ..., \Delta_n$ is not empty

- $A \otimes B, \Omega$

  $\mathsf{m}_1\ \Delta, \Delta_1, ..., \Delta_n; \Xi; A \otimes B, A_1, ..., A_{n-2}; H; \cdot \rightarrow \Xi'; \Delta'$ (1) assumption
  $\mathsf{m}_1\ \Delta, \Delta_1, ..., \Delta_n; \Xi; A, B, A_1, ..., A_{n-2}; H; \cdot \rightarrow \Xi'; \Delta'$ (2) inversion of (1)
  $\mathsf{m}_1\ \Delta; \Xi, \Delta_1, ..., \Delta_n; \cdot; H; \cdot \rightarrow \Xi'; \Delta'$ (3) induction on (2)
  $\mathsf{match}\ \Delta_1 \rightarrow A, \quad \mathsf{match}\ \Delta_2 \rightarrow B, \quad ... \quad \mathsf{match}\ \Delta_n \rightarrow A_{n-2}$ (4) induction on (2)
  $\mathsf{match}\ \Delta_1, \Delta_2 \rightarrow A \otimes B$ (5) rule on (4)
  $\mathsf{match}\ \Delta_1, \Delta_2 \rightarrow A \otimes B \quad ... \quad \mathsf{match}\ \Delta_n \rightarrow A_{n-2}$ (6) from (5)
  $\Delta_1, ..., \Delta_n$ is not empty (7) from induction on (2)

- $\cdot$

  $\mathsf{m}_1\ \Delta; \Xi; \cdot; H; (\cdot; \Delta'') \rightarrow \Xi'; \Delta'$ (1) assumption
  $n = 0$ since $\Omega = \cdot$

4.7 Derive soundness

If $\mathsf{d}_1\ \Delta; \Xi; \Delta_1; \Omega; \cdot \rightarrow \Xi'; \Delta'$ then
$\mathsf{derive}\ \Delta; \Xi; \Delta_1; \Omega \rightarrow \Xi'; \Delta'$

By induction first on Ω and then on the size of Δ.

- p, Ω

  $\mathsf{d}_1\ \Delta; \Xi; \Delta_1; p, \Omega; \cdot \rightarrow \Xi'; \Delta'$ (1) assumption
  $\mathsf{d}_1\ \Delta; \Xi; \Delta_1, p; \Omega; \cdot \rightarrow \Xi'; \Delta'$ (2) inversion of (1)
  $\mathsf{derive}\ \Delta; \Xi; \Delta_1, p; \Omega \rightarrow \Xi'; \Delta'$ (3) by induction on (2)
  $\mathsf{derive}\ \Delta; \Xi; \Delta_1; p, \Omega \rightarrow \Xi'; \Delta'$ (4) rule application on (3)

- $1, \Omega$

  Same as before.

- $A \otimes B, \Omega$

  Same as before.

- $\mathsf{comp}\ A \multimap B, \Omega$

  $\mathsf{d}_1\ \Delta; \Xi; \Delta_1; \mathsf{comp}\ A \multimap B, \Omega; \cdot \rightarrow \Xi'; \Delta'$ (1) assumption
  $\mathsf{a}_1\ \Delta; A \multimap B; (\mathsf{d}_1\ \Delta; \Xi; \Delta_1; \mathsf{comp}\ A \multimap B, \Omega) \rightarrow \Xi'; \Delta'$ (2) inversion of (1)
  $\mathsf{m}_1\ \Delta; \cdot; A; B; (\mathsf{d}_1\ \Delta; \Xi; \Delta_1; \mathsf{comp}\ A \multimap B, \Omega) \rightarrow \Xi'; \Delta'$ (3) inversion of (2)
  Using (3) on theorem "Low level comprehension gives one match" we get two subcases:

  - Comprehension fails:
    $\mathsf{cont}\ (\mathsf{d}_1\ \Delta; \Xi; \Delta_1; \mathsf{comp}\ A \multimap B, \Omega); \Xi'\Delta'$ (4) from theorem
    $\mathsf{d}_1\ \Delta; \Xi; \Delta_1; \Omega; \cdot \rightarrow \Xi'; \Delta'$ (5) inversion of (4)
    $\mathsf{derive}\ \Delta; \Xi; \Delta_1; \Omega \rightarrow \Xi'; \Delta'$ (6) induction on (5)
    $\mathsf{derive}\ \Delta; \Xi; \Delta_1; 1, \Omega \rightarrow \Xi'; \Delta'$ (7) rule on (6)
    $\mathsf{derive}\ \Delta; \Xi; \Delta_1; 1 \mathbin{\&} (A \multimap B \otimes \mathsf{comp}\ A \multimap B), \Omega \rightarrow \Xi'; \Delta'$ (8) rule application on (7)
    $\mathsf{derive}\ \Delta; \Xi; \Delta_1; \mathsf{comp}\ A \multimap B, \Omega \rightarrow \Xi'; \Delta'$ (9) rule application on (8)

  - Comprehension succeeds: $\Delta = \Delta'', \Xi''$ (4) from theorem
    $\mathsf{m}_1\ \Delta''; \Xi''; \cdot; B; (\mathsf{d}_1\ \Delta; \Xi; \Delta_1; \mathsf{comp}\ A \multimap B, \Omega) \rightarrow \Xi'; \Delta'$ (5) from theorem "Low level comprehension gives one match"

$\mathsf{match}\ \Xi'' \to A$ (6) from the same theorem
Δ'' is smaller than Δ (7) from the same theorem
$\mathsf{d}_1\ \Delta''; \Xi''; \cdot; B; (\mathsf{d}_1\ \Delta; \Xi; \Delta_1; \mathsf{comp}\ A \multimap B, \Omega) \to \Xi'; \Delta'$ (8) inversion of (5)
$\mathsf{d}_1\ \Delta''; \Xi'', \Xi; \Delta_1; B, \mathsf{comp}\ A \multimap B, \Omega; \cdot \to \Xi'; \Delta'$ (9) using theorem "Comprehension head is another derivation theorem" on (8)
$\mathsf{derive}\ \Delta''; \Xi''; \Xi; \Delta_1; B, \mathsf{comp}\ A \multimap B, \Omega \to \Xi'; \Delta'$ (10) by i.h. on (9) because of (7)
$\mathsf{derive}\ \Xi'', \Delta''; \Xi'; \Delta_1; A \multimap B, \mathsf{comp}\ A \multimap B, \Omega \to \Xi'; \Delta'$ (11) using rule on (10) and (6)
$\mathsf{derive}\ \Xi'', \Delta''; \Xi'; \Delta_1; (A \multimap B) \otimes (\mathsf{comp}\ A \multimap B), \Omega \to \Xi'; \Delta'$ (12) using rule on (11)
$\mathsf{derive}\ \Xi'', \Delta''; \Xi'; \Delta_1; 1 \mathbin{\&} ((A \multimap B) \otimes (\mathsf{comp}\ A \multimap B)), \Omega \to \Xi'; \Delta'$ (13) rule on (12)
$\mathsf{derive}\ \Xi'', \Delta''; \Xi'; \Delta_1; \mathsf{comp}\ A \multimap B, \Omega \to \Xi'; \Delta'$ (14) rule on (13)

4.8 Apply Soundness Theorem

If $\mathsf{a}_1\ \Delta; R; (\cdot; \Delta) \to \Xi'; \Delta'$ then
$\mathsf{a}_0\ \Delta; R \to \Xi'; \Delta'$

Case by case analysis:

- $R = A \multimap B$

  $\mathsf{a}_1\ \Delta; A \multimap B; (\cdot; \Delta) \to \Xi'; \Delta'$ (1) assumption
  $\mathsf{m}_1\ \Delta_1, \Delta_2; \cdot; A; B; (\cdot; \Delta) \to \Xi'; \Delta'$ (2) inversion of (1)
  $\mathsf{m}_1\ \Delta_2; \Delta_1; \cdot; B; (\cdot; \Delta) \to \Xi'; \Delta'$ (3) using theorem "Low level matching gives high level matching theorem" on (2)
  $\mathsf{match}\ \Delta_1 \to A$ (4) from same theorem on (2)
  $\Delta_2; \Delta_1; \cdot; B; (\cdot; \Delta) \to \Xi'; \Delta'$ (5) inversion of (3)
  $\mathsf{derive}\ \Delta_2; \Delta_1; \cdot; B \to \Xi'; \Delta'$ (6) derive soundness on (5)
  $\mathsf{a}_0\ \Delta_1, \Delta_2; A \multimap B \to \Xi'; \Delta'$ (7) using rule on (6) and (4)

4.9 Success or continuation

If $\mathsf{m}_1\ \Delta; \Xi; \Delta_1; \Omega; H; (\Phi; \Delta'') \to \Xi'; \Delta'$ then either:

- $\mathsf{cont}\ (\Phi; \Delta''); \Xi'; \Delta'$
- $\mathsf{m}_1\ \Delta; \Xi; \Delta_1; \Omega; H; (\cdot; \Delta'') \to \Xi'; \Delta'$

Induction on Ω:

- p, Ω and $p \in \Omega$

  $\mathsf{m}_1\ \Delta, p; \Xi; p, \Omega; H; (\Phi; \Delta'') \to \Xi'; \Delta'$ (1) assumption
  $\mathsf{m}_1\ \Delta; \Xi, p; \Omega; H; (\Phi; \Delta'') \to \Xi'; \Delta'$ (2) inversion of (1)
  $\mathsf{cont}\ (\Phi; \Delta''); \Xi'; \Delta'$ or $\mathsf{m}_1\ \Delta; \Xi, p; \Omega; H; (\cdot; \Delta'') \to \Xi'; \Delta'$ (3) induction of (2)
  $\mathsf{cont}\ (\Phi; \Delta''); \Xi'; \Delta'$ or $\mathsf{m}_1\ \Delta, p; \Xi; p, \Omega; H; (\cdot; \Delta'') \to \Xi'; \Delta'$ (4) rule application on (3)

- p, Ω and $p \notin \Delta$

  $\mathsf{m}_1\ \Delta; \Xi; p, \Omega; H; (\Phi; \Delta'') \to \Xi'; \Delta'$ (1) assumption
  $\mathsf{cont}\ (\Phi; \Delta''); \Xi'; \Delta'$ (2) inversion of (1)

- $A \otimes B, \Omega$

$\mathsf{m}_1\ \Delta; \Xi; A \otimes B, \Omega; H; (\Phi; \Delta'') \to \Xi'; \Delta'$	(1) assumption
$\mathsf{m}_1\ \Delta; \Xi; A, B, \Omega; H; (\Phi; \Delta'') \to \Xi'; \Delta'$	(2) inversion of (1)
$\mathsf{cont}\ \ (\Phi; \Delta''); \Xi'; \Delta'$ or $\mathsf{m}_1\ \Delta; \Xi; A, B, \Omega; H; (\cdot; \Delta'') \to \Xi'; \Delta'$	(3) i.h. on (2)
$\mathsf{cont}\ \ (\Phi; \Delta''); \Xi'; \Delta'$ or $\mathsf{m}_1\ \Delta; \Xi; A \otimes B, \Omega; H; (\cdot; \Delta'') \to \Xi'; \Delta'$	(4) rule on (3)

- $\cdot$

$\mathsf{m}_1\ \Delta; \Xi; \cdot; H; (\Phi; \Delta'') \to \Xi'; \Delta'$	(1) assumption
$\mathsf{d}_1\ \Delta; \Xi; \cdot; H; \cdot \to \Xi'; \Delta'$	(2) inversion of (1)
$\mathsf{m}_1\ \Delta; \Xi; \cdot; H; (\cdot; \Delta'') \to \Xi'; \Delta'$	(3) rule application on (2)

4.10 One Rule Theorem

If $\mathsf{do}_1\ \Delta; \Phi \to \Xi'; \Delta'$ then $\exists R \in \Phi. \mathsf{do}_1\ \Delta; R \to \Xi'; \Delta'$.
Induction on the size of Φ.

- $\Phi = \cdot$

  Not applicable.

- $\Phi = R', \Phi'$

$\mathsf{do}_1\ \Delta; R', \Phi' \to \Xi' \Delta'$	(1) assumption
$\mathsf{a}_1\ \Delta; R'; (\Phi'; \Delta) \to \Xi'; \Delta'$	(2) inversion of (1)
$\mathsf{m}_1\ \Delta; \cdot; A; B; (\Phi'; \Delta) \to \Xi'; \Delta'$	(3) inversion of (2) where $R' = A \multimap B$

From theorem "Success or continuation" we have two cases:

- Match Success

$\mathsf{m}_1\ \Delta; \cdot; A; B; (\cdot; \Delta) \to \Xi'; \Delta'$	(4) from theorem
$\mathsf{a}_1\ \Delta; A \multimap B; (\cdot; \Delta) \to \Xi'; \Delta'$	(5) rule application on (4)
$\mathsf{do}_1\ \Delta; A \multimap B \to \Xi'; \Delta'$	(6) rule application on (5)
$\mathsf{do}_1\ \Delta; R \to \Xi'; \Delta'$	(7) rewrite of (6)

- Continuation

$\mathsf{cont}\ (\Phi'; \Delta); \Xi'; \Delta'$	(4) from theorem
$\mathsf{do}_1\ \Delta; \Phi' \to \Xi'; \Delta'$	(5) inversion of (4)
$\exists R \in \Phi'. \mathsf{do}_1\ \Delta; R \to \Xi'; \Delta'$	(6) i.h. on (5)
$\exists R \in \Phi.$	(7) from (6)

5 Term Equivalence

The judgment $A \equiv B$ tests if two multi-sets of terms A, B are equal. Two multisets are equal if, when deconstructed, they have the same atoms.

$$\frac{A \equiv B}{p, A \equiv p, B} \equiv p$$

$$\frac{A \equiv B}{!p, A \equiv !, B} \equiv !p$$

$$\frac{A \equiv B}{1, A \equiv B} \equiv 1\ left \quad \frac{A \equiv B}{A \equiv 1, B} \equiv 1\ right$$

$$\frac{}{\cdot \equiv \cdot} \equiv \cdot$$

$$\frac{A, B, C \equiv D}{A \otimes B, C \equiv D} \equiv \otimes\ left \quad \frac{A \equiv B, C, D}{A \equiv B \otimes C, D} \equiv \otimes\ right$$

5.1 Equivalence commutativity

Theorem 1. *If $A \equiv B$ then $B \equiv A$.*

Proof. By induction on the structure of the assumption.

- p

  $p, A \equiv p, B$ (1) assumption
  $A \equiv B$ (2) inversion of (1)
  $B \equiv A$ (3) i.h. on (2)
  $p, B \equiv p, A$ (4) rule on (3)

- $!p$

  $!p, A \equiv !p, B$ (1) assumption
  $A \equiv B$ (2) inversion of (1)
  $B \equiv A$ (3) i.h. on (2)
  $!p, B \equiv !p, A$ (4) rule on (3)

- $1\ left$

  $1, A \equiv B$ (1) assumption
  $A \equiv B$ (2) assumption
  $B \equiv A$ (3) i.h. on (2)
  $B \equiv 1, A$ (4) apply right rule on (3)

- $1\ right$

  Same thing as before.

- $\cdot$

  Immediate.

- $\otimes\ left$

  $A \otimes B, C \equiv D$ (1) assumption
  $A, B, C \equiv D$ (2) inversion of (1)
  $D \equiv A, B, C$ (3) i.h. on (2)
  $D \equiv A \otimes B, C$ (4) right rule on (3)

- $\otimes\ right$

  Same thing as the last case.

□

5.2 Match Equivalence Theorem

Theorem 2. *If* $A_1, ..., A_n \equiv B_1, ..., B_m$ *and* $\mathsf{match}\ \Gamma; \Delta \to A_1 \otimes ... \otimes A_n$ *then* $\mathsf{match}\ \Gamma; \Delta \to B_1 \otimes ... \otimes B_m$.

Proof. By induction on the structure of the assumption.

- p

$p, A \equiv p, B$	(1) assumption
$A \equiv B$	(2) inversion of (1)
$\mathsf{match}\ p \to p$	(3) axiom
$\mathsf{match}\ p, \Delta \to p \otimes A$	(4) assumption
$\mathsf{match}\ \Delta \to A$	(5) inversion of (4)
$\mathsf{match}\ \Delta \to B$	(6) i.h. on (5) and (2)
$\mathsf{match}\ p, \Delta \to p \otimes B$	(7) rule application on (6) and (3)

- $!p$

$!p, A \equiv\ !p, B$	(1) assumption
$A \equiv B$	(2) inversion of (1)
$\mathsf{match}\ \Gamma', p; \Delta \to\ !p \otimes A$	(3) assumption
$\mathsf{match}\ \Gamma', p; \cdot \to\ !p$	(4) inversion of (4)
$\mathsf{match}\ \Gamma', p; \Delta \to A$	(5) inversion of (3)
$\mathsf{match}\ \Gamma', p; \Delta \to B$	(6) i.h. on (5) and (2)
$\mathsf{match}\ \Gamma', p; \Delta \to p \otimes B$	(7) merge (6) with (4)

- $1\ left$

$1, A \equiv B$	(1) assumption
$A \equiv B$	(2) inversion of (1)
$\mathsf{match}\ \Delta \to 1 \otimes A$	(3) assumption
$\mathsf{match}\ \Delta \to A$	(4) inversion of (4)
$\mathsf{match}\ \cdot \to 1$	(5) axiom
$\mathsf{match}\ \Delta \to B$	(6) i.h. on (2) and (4)
$\mathsf{match}\ \Delta \to 1 \otimes B$	(7) rule application on (6) and (5)

- $1\ right$

  Same thing as before.

- $\cdot$

  Immediate since match fails.

- $\otimes\ left$

$A \otimes B, C \equiv D$	(1) assumption
$A, B, C \equiv D$	(2) inversion of (1)
$\mathsf{match}\ \Delta_1, \Delta_2, \Delta_3 \to (A \otimes B) \otimes C$	(3) assumption
$\mathsf{match}\ \Delta_1, \Delta_2 \to A \otimes B$	(4) inversion of (3)
$\mathsf{match}\ \Delta_3 \to C$	(5) inversion of (3)
$\mathsf{match}\ \Delta_1 \to A$ and $\mathsf{match}\ \Delta_2 \to B$	(6) inversion of (4)
$\mathsf{match}\ \Delta_1, \Delta_2, \Delta_3 \to A \otimes B \otimes C$	(7) apply (5) with (6)
$\mathsf{match}\ \Delta_1, \Delta_2, \Delta_3 \to D$	(8) i.h. on (7) and (2)

- $\otimes$ *right*

  Apply equivalence commutativity theorem and follow the previous case.

$\square$

6 Low Level System With Matching Continuations

6.1 Low Level System

For this system, we include only linear facts but we use a continuation stack to match facts.

6.1.1 Match

$$\frac{\mathsf{m}_1\ \Delta, \Delta''; \Xi, p_1; \Omega; H; (\Delta, p_1; \Delta''; p; \Omega; \Xi; \cdot); R \to \Xi'; \Delta'}{\mathsf{m}_1\ \Delta, p_1, \Delta''; \Xi; p, \Omega; H; \cdot; R \to \Xi'; \Delta'}\ \mathsf{m}_1\ ok\ first$$

$$\frac{\mathsf{m}_1\ \Delta, \Delta''; \Xi, p_1; \Omega; H; (\Delta, p_1; \Delta''; p; \Omega; \Xi; q, \Lambda), C_1, C; R \to \Xi'; \Delta' \quad C_1 = (\Delta_{old}; \Delta'_{old}; \Xi_{old}; q; \Omega_{old}; \Lambda)}{\mathsf{m}_1\ \Delta, p_1, \Delta''; \Xi; p, \Omega; H; C_1, C; R \to \Xi'; \Delta'}\ \mathsf{m}_1\ ok\ other$$

$$\frac{p \notin \Delta \quad \mathsf{cont}\ C; H; R; \Xi'; \Delta'}{\mathsf{m}_1\ \Delta; \Xi; p, \Omega; H; C; R \to \Xi'; \Delta'}\ \mathsf{m}_1\ fail$$

$$\frac{\mathsf{m}_1\ \Delta; \Xi; A, B, \Omega; H; C; R \to \Xi'; \Delta'}{\mathsf{m}_1\ \Delta; \Xi; A \otimes B, \Omega; H; C; R \to \Xi'; \Delta'}\ \mathsf{m}_1\ \otimes$$

$$\frac{\mathsf{d}_1\ \Delta; \Xi; \cdot; H; \cdot \to \Xi'; \Delta'}{\mathsf{m}_1\ \Delta; \Xi; \cdot; H; C; R \to \Xi'; \Delta'}\ \mathsf{m}_1\ end$$

6.1.2 Derive

$$\frac{\mathsf{d}_1\ \Delta; \Xi; p, \Delta_1; \Omega; C \to \Xi'; \Delta'}{\mathsf{d}_1\ \Delta; \Xi; \Delta_1; p, \Omega; C \to \Xi'; \Delta'}\ \mathsf{d}_1\ p \qquad \frac{\mathsf{d}_1\ \Delta; \Xi; \Delta_1; \Omega; C \to \Xi'; \Delta'}{\mathsf{d}_1\ \Delta; \Xi; \Delta_1; 1, \Omega; C \to \Xi'; \Delta'}\ \mathsf{d}_1\ 1$$

$$\frac{\mathsf{d}_1\ \Delta; \Xi; \Delta_1; A, B, \Omega; C \to \Xi'; \Delta'}{\mathsf{d}_1\ \Delta; \Xi; \Delta_1; A \otimes B, \Omega; C \to \Xi'; \Delta'}\ \mathsf{d}_1\ \otimes$$

$$\frac{}{\mathsf{d}_1\ \Delta; \Xi; \Delta_1; \cdot \to \Xi; \Delta_1}\ \mathsf{d}_1\ end$$

6.1.3 Continuation

$$\frac{\mathsf{do}_1\ \Delta; \Phi \to \Xi'; \Delta'}{\mathsf{cont}\ \cdot; H; (\Phi; \Delta); \Xi'; \Delta'}\ \mathsf{cont}\ next\ rule$$

$$\frac{\mathsf{m}_1\ \Delta, \Delta''; \Xi, p_1; \Omega; H; (\Delta, p_1; \Delta''; p; \Omega; \Xi; \Lambda), C; R \to \Xi'; \Delta'}{\mathsf{cont}\ (\Delta; p_1, \Delta''; p; \Omega; \Xi; \Lambda), C; H; R; \Xi'; \Delta'}\ \mathsf{cont}\ next$$

$$\frac{\mathsf{cont}\ C; H; R; \Xi'; \Delta'}{\mathsf{cont}\ (\Delta; \cdot; p; \Omega; \Xi; \Lambda), C; H; R; \Xi'; \Delta'}\ \mathsf{cont}\ no\ more$$

6.1.4 Apply

$$\frac{\mathsf{m}_1\ \Delta; \cdot; A; B; \cdot; R \to \Xi'; \Delta'}{\mathsf{a}_1\ \Delta; A \multimap B; R \to \Xi'; \Delta'}\ \mathsf{a}_1\ start$$

$$\frac{\mathsf{a}_1\ \Delta; R; (\Phi; \Delta) \to \Xi'; \Delta'}{\mathsf{do}_1\ \Delta; R, \Phi \to \Xi'; \Delta'}\ \mathsf{do}_1\ rule$$

6.2 High Level System

6.2.1 Match

$$\frac{}{\mathsf{match}\ \cdot \to 1}\ \mathsf{match}\ 1 \qquad \frac{}{\mathsf{match}\ p \to p}\ \mathsf{match}\ p$$

$$\frac{\mathsf{match}\ \Delta_1 \to A \quad \mathsf{match}\ \Delta_2 \to B}{\mathsf{match}\ \Delta_1, \Delta_2 \to A \otimes B}\ \mathsf{match}\ \otimes$$

6.2.2 Derive

$$\frac{\mathsf{derive}\ \Delta; \Xi; p, \Delta_1; \Omega \to \Xi'; \Delta'}{\mathsf{derive}\ \Delta; \Xi; \Delta_1; p, \Omega \to \Xi'; \Delta'}\ \mathsf{derive}\ p$$

$$\frac{\mathsf{derive}\ \Delta; \Xi; \Delta_1; A, B, \Omega \to \Xi'; \Delta'}{\mathsf{derive}\ \Delta; \Xi; \Delta_1; A \otimes B, \Omega \to \Xi'; \Delta'}\ \mathsf{derive}\ \otimes$$

$$\frac{}{\mathsf{derive}\ \Delta; \Xi'; \Delta'; \cdot \to \Xi'; \Delta'}\ \mathsf{derive}\ end$$

6.2.3 Apply

$$\frac{\mathsf{match}\ \Delta \to A \quad \mathsf{derive}\ \Delta''; \Delta; \cdot; B \to \Xi'; \Delta'}{\mathsf{a}_0\ \Delta, \Delta''; A \multimap B \to \Xi'; \Delta'}\ \mathsf{a}_0$$

$$\frac{\mathsf{do}_0\ \Delta; R \to \Xi'; \Delta'}{\mathsf{do}_0\ \Delta; R, \Phi \to \Xi'; \Delta'}\ \mathsf{do}_0\ rule$$

6.3 Definitions

6.3.1 Well Formed Frame

Definition 3. *Given a frame $f = (\Delta, p_1; \Delta'; \Xi; p; \Omega_1, ..., \Omega_n; \Lambda_1, ..., \Lambda_m)$ and a body term A and a context Δ_{inv}, we say that f is well formed iff:*

1. $\Lambda_1, ..., \Lambda_m$ *are atomic terms* p_i.
2. $\Xi = \Xi_1, ..., \Xi_m$
3. $p, \Omega_1, ..., \Omega_n, \Lambda_1, ..., \Lambda_m \equiv A$
4. $\mathsf{match}\ \Xi \to \Lambda_1 \otimes ... \otimes \Lambda_m$.
5. $\Delta, \Delta', \Xi, p_1 = \Delta_{inv}$.

6.3.2 Well Formed Stack

Definition 4. *A continuation stack C is well formed iff every frame is well formed.*

6.3.3 Related Match

Definition 5. $\mathsf{m}_1\ \Delta;\Xi;\Omega;H;C;R \to \Xi';\Delta'$ *is related to a term A and a context Δ_{inv} iff:*

1. *C is well formed in relation to A and Δ_{inv};*
2. $\Delta,\Xi = \Delta_{inv}$
3. *Stack relatedness:*
   (a) $C = \cdot$
      i. $\Omega \equiv A$
   (b) $C = (\Delta_a;\Delta_b;\Xi'';p;\Omega';\Lambda_1,...,\Lambda_m),C'$
      i. $\Omega' \equiv \Omega$
      ii. $p_1 \in \Xi$ *and* $\mathsf{match}\ p_1 \to p$
      iii. $\Xi = \Xi'',p_1$

6.4 Theorems

6.4.1 Lexicographical Ordering

1. $\mathsf{cont}\ C;H;R;\Xi';\Delta' \prec \mathsf{cont}\ C',C;H;R;\Xi';\Delta'$
2. $\mathsf{cont}\ C',(\Delta,\Delta_1;\Delta_2),C;H;R;\Xi';\Delta' \prec \mathsf{cont}\ C'',(\Delta;\Delta_1,\Delta_2),C;H;R;\Xi';\Delta'$
3. $\mathsf{cont}\ C'',C;H;R;\Xi';\Delta' \prec \mathsf{m}_1\ \Delta;\Xi;\Omega;H;C',C;R \to \Xi';\Delta'$ as long as $C'' \prec C$
4. $\mathsf{m}_1\ \Delta'';\Xi'';\Omega';H;C',C;R \to \Xi';\Delta' \prec \mathsf{m}_1\ \Delta;\Xi;\Omega;H;C;R \to \Xi';\Delta'$ as long as $\Omega' \prec \Omega$
5. $\mathsf{m}_1\ \Delta;\Xi;\Omega;H;C;R \to \Xi';\Delta' \prec \mathsf{cont}\ C',C;H;R;\Xi';\Delta'$
6. $\mathsf{m}_1\ \Delta;\Xi;\Omega;H;C'',(\Delta,\Delta_1;\Delta_2),C;R \to \Xi';\Delta' \prec \mathsf{m}_1\ \Delta'';\Xi'';\Omega';C',(\Delta;\Delta_1,\Delta_2),C;R \to \Xi';\Delta'$

6.4.2 Match Soundness Theorem

Theorem 6. *If a match* $\mathsf{m}_1\ \Delta_1,\Delta_2;\Xi;\Omega;H;C;R \to \Xi';\Delta'$ *is related to term A and context Δ_1,Δ_2,Ξ then either:*

1. $\mathsf{cont}\ \cdot;H;R;\Xi';\Delta'$ *or*
2. $\mathsf{match}\ \Delta_x \to A$ *(where Δ_x is a subset of Δ_1,Δ_2,Ξ) and one of the two subcases are true:*
   (a) $\mathsf{m}_1\ \Delta_1;\Xi,\Delta_2;\cdot;H;C'',C;R \to \Xi';\Delta'$ *(related) and $\Delta_x = \Xi,\Delta_2$*
   (b) $\exists f = (\Delta_a;\Delta_{b_1},p_2,\Delta_{b_2};\Xi_1,...,\Xi_m;p;\Omega_1,...,\Omega_k;\Lambda_1,...,\Lambda_m) \in C$ *where $C = C',f,C''$ and f turns into some* $f' = (\Delta_a,\Delta_{b_1},p_2;\Delta_{b_2};\Xi_1,...,\Xi_m;p;\Omega_1,...,\Omega_k;\Lambda_1,...,\Lambda_m)$ *such that:*
      - $\mathsf{m}_1\ \ \Delta_c;\Xi_1,...,\Xi_m,p_2,\Xi_c;\cdot;H;C''',f',C'';R \to \Xi';\Delta'$ *(related) where* $\Delta_c = (\Delta_1,\Delta_2,\Xi) - (\Xi_1,...,\Xi_m,p_2,\Xi_c)$

If $\mathsf{cont}\ C;H;R;\Xi';\Delta'$ *and C is well formed in relation to A and Δ_1,Δ_2,Ξ then either:*

1. $\mathsf{cont}\ \cdot; H; R; \Xi'; \Delta'$

2. $\mathsf{match}\ \Delta_x \rightarrow A$ *(where* Δ_x *is a subset of* Δ_1, Δ_2, Ξ*) where:*

   (a) $\exists f = (\Delta_a; \Delta_{b_1}, p_2, \Delta_{b_2}; \Xi_1, ..., \Xi_m; p; \Omega_1, ..., \Omega_k; \Lambda_1, ..., \Lambda_m) \in C$ *where* $C = C', f, C''$ *and* f *turns into some* $f' = (\Delta_a, \Delta_{b_1}, p_2; \Delta_{b_2}; \Xi_1, ..., \Xi_m; p; \Omega_1, ..., \Omega_k; \Lambda_1, ..., \Lambda_m)$ *such that:*

   - $\mathsf{m}_1\ \Delta_c; \Xi_1, ..., \Xi_m, p_2, \Xi_c; \cdot; H; C''', f', C''; R \rightarrow \Xi'; \Delta'$ *(related)*

Proof. Proof by mutual induction. In m_1 on the size of Ω and on cont, first on the size of Δ'' and then on the size of C. Use the stack constraints and the Match Equivalence Theorem to prove $\mathsf{match}\ \Delta_x \rightarrow A$.

- m_1 *ok first*

  First prove that the inverted match is related and then use induction.

- m_1 *ok other*

  First we prove that the inverted match is related. We know that $q \in \Xi$ due to our assumption, so that proves $\mathsf{match}\ q \rightarrow q$.
  Second, we prove that the new stack frame is related and then apply induction.

- m_1 *fail*

  $\mathsf{m}_1\ \Delta; \Xi; p, \Omega; H; C; R \rightarrow \Xi'; \Delta'$ (1) assumption
  $\mathsf{cont}\ C; H; R; \Xi'; \Delta'$ (2) inversion of (1)
  Apply i.h. on (2) to get $\mathsf{cont}\ \cdot; H; R; \Xi'; \Delta'$ or the $\mathsf{match}\ \Delta_x \rightarrow A$.

- $\mathsf{m}_1\ \otimes$

  By induction.

- m_1 *end*

  $\mathsf{m}_1\ \Delta; \Xi; \cdot; H; C; R \rightarrow \Xi'; \Delta'$ (1) assumption.
  Our stack must have some frames, thus $p, \Lambda_c \equiv A$ (p and Λ_c both arguments of the last frame). We also know that $p \in \Xi$ due to our assumption and that $\mathsf{match}\ \Xi_c, p \rightarrow \Lambda \otimes p$ is true. Therefore $\Xi = \Xi_c, p$ and thus $\mathsf{match}\ \Xi \rightarrow A$.

- cont *next rule*

  $\mathsf{cont}\ \cdot; H; (\Phi; \Delta); \Xi'; \Delta'$ (1) assumption

- cont *next*

  $\mathsf{cont}\ (\Delta; p_1, \Delta''; p, \Omega; \Xi; \Lambda), C; H; R; \Xi'; \Delta'$ (1) assumption
  $\mathsf{m}_1\ \Delta, \Delta''; \Xi, p_1; \Omega; (\Delta, p_1; \Delta''; p, \Omega; H; \Xi; \Lambda), C; R \rightarrow \Xi'; \Delta'$ (2) inversion of (1)
  Since the frame we push into the match makes the match related, we can use induction (2):

  1. $\mathsf{cont}\ \cdot; H; R; \Xi'; \Delta'$
  2. $\mathsf{match}\ \Delta_x \rightarrow A$
     - $\mathsf{m}_1\ \Delta_a; \Xi, p_1, \Delta_1, ..., \Delta_k; \cdot; H; C''; R \rightarrow \Xi'; \Delta'$
       where $\Delta_a, \Delta_1, ..., \Delta_k = \Delta, \Delta'$

       $\exists f = (\Delta; p_1; \Delta''; p, \Omega; H; \Xi; \Lambda)$
     - $\exists f \in (\Delta, p_1; \Delta''; p, \Omega; H; \Xi; \Lambda), C$
       f can be $(\Delta, p_1; \Delta''; p, \Omega; H; \Xi; \Lambda)$ (which is contained in the original cont)
       or $f \in C$

- cont *no more*

  $\mathsf{cont}\ (\Delta; \cdot; p, \Omega; \Xi), C; H; R; \Xi'; \Delta'$ (1) assumption
  $\mathsf{cont}\ C; H; R; \Xi'; \Delta'$ (2) inversion of (1)

  Apply induction.

□

6.5 Match Soundness Lemma

Theorem 7. *If* $\mathsf{m}_1\ \Delta, \Delta''; \cdot; \Omega; H; \cdot; R \to \Xi'; \Delta'$ *then either:*

1. $\mathsf{cont}\ \cdot; R; \Xi'; \Delta'$ *or;*
2. $\mathsf{m}_1\ \Delta; \Delta''; \cdot; H; C'; R \to \Xi'; \Delta'$ *and* $\mathsf{match}\ \Delta'' \to \Omega$

Proof. By direct use of the match soundness theorem. □

6.6 Derivation Soundness Theorem

Theorem 8. *If* $\mathsf{d}_1\ \Delta; \Xi; \Delta_1; \Omega; C \to \Xi'; \Delta'$ *then* $\mathsf{derive}\ \Delta; \Xi; \Delta_1; \Omega \to \Xi'; \Delta'$.

Proof. By simple induction on Ω.

- $\mathsf{d}_1\ p$

  By induction.

- $\mathsf{d}_1\ 1$

  By induction.

- $\mathsf{d}_1\ \otimes$

  By induction.

- $\mathsf{d}_1\ end$

  Use axioms.

□

7 Low Level System With Comprehensions

7.1 High Level System

7.1.1 Match

$$\frac{}{\mathsf{match}\ \cdot \to 1}\ \mathsf{match}\ 1 \qquad \frac{}{\mathsf{match}\ p \to p}\ \mathsf{match}\ p$$

$$\frac{\mathsf{match}\ \Delta_1 \to A \quad \mathsf{match}\ \Delta_2 \to B}{\mathsf{match}\ \Delta_1, \Delta_2 \to A \otimes B}\ \mathsf{match}\ \otimes$$

7.1.2 Derive

$$\frac{\mathsf{derive}\ \Delta; \Xi; p, \Delta_1; \Omega \rightarrow \Xi'; \Delta'}{\mathsf{derive}\ \Delta; \Xi; \Delta_1; p, \Omega \rightarrow \Xi'; \Delta'}\ \mathsf{derive}\ p$$

$$\frac{\mathsf{derive}\ \Delta; \Xi; \Delta_1; A, B, \Omega \rightarrow \Xi'; \Delta'}{\mathsf{derive}\ \Delta; \Xi; \Delta_1; A \otimes B, \Omega \rightarrow \Xi'; \Delta'}\ \mathsf{derive}\ \otimes$$

$$\frac{\mathsf{derive}\ \Delta; \Xi; \Delta_1; \Omega \rightarrow \Xi'; \Delta'}{\mathsf{derive}\ \Delta; \Xi; \Delta_1; 1, \Omega \rightarrow \Xi'; \Delta'}\ \mathsf{derive}\ 1$$

$$\frac{}{\mathsf{derive}\ \Delta; \Xi'; \Delta'; \cdot \rightarrow \Xi'; \Delta'}\ \mathsf{derive}\ \mathit{end}$$

$$\frac{\mathsf{derive}\ \Delta; \Xi; \Delta_1; 1 \mathbin{\&} (A \multimap B \otimes \mathsf{comp}\ A \multimap B), \Omega \rightarrow \Xi'; \Delta'}{\mathsf{derive}\ \Delta; \Xi; \Delta_1; \mathsf{comp}\ A \multimap B, \Omega \rightarrow \Xi'; \Delta'}\ \mathsf{derive}\ \mathit{comp}$$

$$\frac{\mathsf{match}\ \Delta_a \rightarrow A \quad \mathsf{derive}\ \Delta_b; \Xi, \Delta_a; \Delta_1; B, \Omega \rightarrow \Xi'; \Delta'}{\mathsf{derive}\ \Delta_a, \Delta_b; \Xi; \Delta_1; A \multimap B, \Omega \rightarrow \Xi'; \Delta'}\ \mathsf{derive}\ \multimap$$

$$\frac{\mathsf{derive}\ \Delta; \Xi; \Delta_1; A, \Omega \rightarrow \Xi'; \Delta'}{\mathsf{derive}\ \Delta; \Xi; \Delta_1; A \mathbin{\&} B, \Omega \rightarrow \Xi'; \Delta'}\ \mathsf{derive}\ \mathbin{\&} L \qquad \frac{\mathsf{derive}\ \Delta; \Xi; \Delta_1; B, \Omega \rightarrow \Xi'; \Delta'}{\mathsf{derive}\ \Delta; \Xi; \Delta_1; A \mathbin{\&} B, \Omega \rightarrow \Xi'; \Delta'}\ \mathsf{derive}\ \mathbin{\&} R$$

7.1.3 Apply

$$\frac{\mathsf{match}\ \Delta \rightarrow A \quad \mathsf{derive}\ \Delta''; \Delta; \cdot; B \rightarrow \Xi'; \Delta'}{\mathsf{a}_0\ \Delta, \Delta''; A \multimap B \rightarrow \Xi'; \Delta'}\ \mathsf{a}_0\ \mathit{rule}$$

$$\frac{\mathsf{do}_0\ \Delta; R \rightarrow \Xi'; \Delta'}{\mathsf{do}_0\ \Delta; R, \Phi \rightarrow \Xi'; \Delta'}\ \mathsf{do}_0\ \mathit{rule}$$

7.2 Low Level System

We extend the previous system with comprehensions. Like the previous system, we use a stack of continuation frames to match the body of rule. For comprehensions, we use the same mechanism but reuse the stack to apply the comprehension several times. Since we use only linear resources, we remove all frames except the first since their state no longer makes sense because the first linear resource used is no longer available due to the application of the comprehension. The last frame is modified in order to remove the facts consumed.

7.2.1 Continuation Frames

The continuation stack is an ordered list of continuation frames. Each frame has the structure $(\Delta_a; \Delta_b; p; \Omega; \Xi; \Lambda)$:

Δ_a : A multi-set of linear resources that are not of type p plus all the other p's we have already tried, including the current p.

Δ_b : All the other p's we haven't tried yet. It is a multi-set of linear resources.

p : The current term that originated this choice point.

Ω : The remaining terms we need to match past this choice point. This is an ordered list.

Ξ : A multi-set of linear resources we have consumed to reach this point.

Λ : A multi-set of atomic terms that we have matched to reach this choice point. All the linear resources that match these terms are located in Ξ.

7.2.2 Match

The matching judgment uses the form $\mathsf{m}_1\ \Delta; \Xi; \Omega; H; C; R \to \Xi'; \Delta'$:

Δ : The multi-set of linear resources still available to complete the matching process.

Ξ : The multi-set of linear resources consumed up to this point.

Ω : Ordered list of terms we want to match.

H : The head of the rule.

C : The continuation stack is an ordered list of frames.

R : The rule continuation. If the matching process fails, we try another rule.

Ξ' : The linear resources consumed to apply some rule of the system.

Δ' : The linear resources created during the application of some rule of the system.

$$\frac{\mathsf{m}_1\ \Delta, \Delta''; \Xi, p_1; \Omega; H; (\Delta, p_1; \Delta''; p; \Omega; \Xi; \cdot); R \to \Xi'; \Delta'}{\mathsf{m}_1\ \Delta, p_1, \Delta''; \Xi; p, \Omega; H; \cdot; R \to \Xi'; \Delta'}\ \mathsf{m}_1\ ok\ first$$

$$\frac{\begin{array}{c}\mathsf{m}_1\ \Delta, \Delta''; \Xi, p_1; \Omega; H; (\Delta, p_1; \Delta''; p; \Omega; \Xi; q, \Lambda), C_1, C; R \to \Xi'; \Delta' \\ C_1 = (\Delta_{old}; \Delta'_{old}; \Xi_{old}; q; \Omega_{old}; \Lambda)\end{array}}{\mathsf{m}_1\ \Delta, p_1, \Delta''; \Xi; p, \Omega; H; C_1, C; R \to \Xi'; \Delta'}\ \mathsf{m}_1\ ok\ other$$

$$\frac{p \notin \Delta \quad \mathsf{cont}\ C; H; R; \Xi'; \Delta'}{\mathsf{m}_1\ \Delta; \Xi; p, \Omega; H; C; R \to \Xi'; \Delta'}\ \mathsf{m}_1\ fail$$

$$\frac{\mathsf{m}_1\ \Delta; \Xi; A, B, \Omega; H; C; R \to \Xi'; \Delta'}{\mathsf{m}_1\ \Delta; \Xi; A \otimes B, \Omega; H; C; R \to \Xi'; \Delta'}\ \mathsf{m}_1 \otimes$$

$$\frac{\mathsf{d}_1\ \Delta; \Xi; \cdot; H; \cdot \to \Xi'; \Delta'}{\mathsf{m}_1\ \Delta; \Xi; \cdot; H; C; R \to \Xi'; \Delta'}\ \mathsf{m}_1\ end$$

7.2.3 Continuation

If the previous matching process fails, we pick the top frame of the stack C and restore the matching process using another fact and/or context. The continuation judgment $\mathsf{cont}\ C; H; R; \Xi'; \Delta'$ deals with the backtracking process where the meaning of each argument is as follows:

C : The continuation stack as an ordered list of frames.

H : The head of the current rule we are trying to match.

R : The next available rules if the current one does not match.

Ξ' : The linear resources consumed to apply some rule of the system.

Δ' : The linear resources created during the application of some rule of the system.

$$\frac{\mathsf{do}_1\ \Delta; \Phi \to \Xi'; \Delta'}{\mathsf{cont}\ \cdot; H; (\Phi; \Delta); \Xi'; \Delta'}\ \mathsf{cont}\ \textit{next rule}$$

$$\frac{\mathsf{m}_1\ \Delta, \Delta''; \Xi, p_1; \Omega; H; (\Delta, p_1; \Delta''; p; \Omega; \Xi; \Lambda), C; R \to \Xi'; \Delta'}{\mathsf{cont}\ (\Delta; p_1, \Delta''; p; \Omega; \Xi; \Lambda), C; H; R; \Xi'; \Delta'}\ \mathsf{cont}\ \textit{next}$$

$$\frac{\mathsf{cont}\ C; H; R; \Xi'; \Delta'}{\mathsf{cont}\ (\Delta; \cdot; p; \Omega; H; \Xi; \Lambda), C; H; R; \Xi'; \Delta'}\ \mathsf{cont}\ \textit{no more}$$

7.2.4 Derive

Once the list of terms Ω is exhausted, we derive the terms of the head of rule represented as R. The derivation judgment uses the form $\mathsf{d}_1\ \Delta; \Xi; \Delta_1; \Omega \to \Xi'; \Delta'$:

- Δ : The multi-set of linear resources we started with minus the linear resources consumed during the matching of the body of the rule.
- Ξ : A multi-set of linear resources consumed during the matching process of the body of the rule.
- Δ_1 : A multi-set of linear resources derived up to this point in the derivation process.
- Ω : The remaining terms to derive as an ordered list. We started with R.
- Ξ' : A multi-set of facts consumed during the whole process.
- Δ' : A multi-set of linear facts derived.

$$\frac{\mathsf{d}_1\ \Delta; \Xi; p, \Delta_1; \Omega \to \Xi'; \Delta'}{\mathsf{d}_1\ \Delta; \Xi; \Delta_1; p, \Omega \to \Xi'; \Delta'}\ \mathsf{d}_1\ p \qquad \frac{\mathsf{d}_1\ \Delta; \Xi; \Delta_1; \Omega \to \Xi'; \Delta'}{\mathsf{d}_1\ \Delta; \Xi; \Delta_1; 1, \Omega \to \Xi'; \Delta'}\ \mathsf{d}_1\ 1$$

$$\frac{\mathsf{d}_1\ \Delta; \Xi; \Delta_1; A, B, \Omega \to \Xi'; \Delta'}{\mathsf{d}_1\ \Delta; \Xi; \Delta_1; A \otimes B, \Omega \to \Xi'; \Delta'}\ \mathsf{d}_1\ \otimes$$

$$\frac{}{\mathsf{d}_1\ \Delta; \Xi; \Delta_1; \cdot \to \Xi; \Delta_1}\ \mathsf{d}_1\ \textit{end}$$

$$\frac{\mathsf{mc}\ \Delta; \Xi; \Delta_1; \cdot; A; B; \cdot; \mathsf{comp}\ A \multimap B; \Omega; \Delta \to \Xi'; \Delta'}{\mathsf{d}_1\ \Delta; \Xi; \Delta_1; \mathsf{comp}\ A \multimap B, \Omega \to \Xi'; \Delta'}\ \mathsf{d}_1\ \textit{comp}$$

7.2.5 Match Comprehension

The matching process for comprehensions is similar to the process used for matching the body of the rule. Note that the structure of the continuation frames is the same. The full judgment as the form $\mathsf{mc}\ \Delta; \Xi_N; \Delta_{N1}; \Xi; \Omega; C; AB; \Omega_N; \Delta_N \to \Xi'; \Delta'$:

- Δ : The multi-set of linear resources remaining up to this point in the matching process.
- Ξ_N : The multi-set of linear resources used during the matching process of the body of the rule.
- Δ_{N1} : The multi-set of linear resources derived by the head of the rule.

Ξ : The multi-set of linear resources consumed up to this point.

Ω : The ordered list of terms that need to be matched for the comprehension to be applied.

C : The continuation stack for the comprehension.

AB : The comprehension $\mathsf{comp}\ A \multimap B$ that is being matched.

Ω_N : The ordered list of remaining terms of the head of the rule to be derived.

Δ_N : The multi-set of linear resources that were still available after matching the body of the rule. Note that $\Delta, \Xi = \Delta_N$.

Ξ' : A multi-set of facts consumed during the whole process.

Δ' : A multi-set of linear facts derived.

$$\frac{\mathsf{mc}\ \Delta, \Delta''; \Xi_N; \Delta_{N1}; \Xi, p_1; \Omega; (\Delta, p_1; \Delta''; \Xi; p; \Omega; \cdot); AB; \Omega_N; \Delta_N \to \Xi'; \Delta'}{\mathsf{mc}\ \Delta, p_1, \Delta''; \Xi_N; \Delta_{N1}; \Xi; p, \Omega; \cdot; AB; \Omega_N; \Delta_N \to \Xi'; \Delta'}\ \mathsf{mc}\ p\ ok\ first$$

$$\frac{\begin{array}{l}\mathsf{mc}\ \Delta, \Delta''; \Xi_N; \Delta_{N1}; \Xi, p_1; \Omega; (\Delta, p_1; \Delta''; \Xi; p; \Omega; q, \Lambda), C_1, C; AB; \Omega_N; \Delta_N \to \Xi'; \Delta' \\ C_1 = (\Delta_{old}; \Delta'_{old}; \Xi_{old}; q; \Omega_{old}; \Lambda)\end{array}}{\mathsf{mc}\ \Delta, p_1, \Delta''; \Xi_N; \Delta_{N1}; \Xi; p, \Omega; C_1, C; AB; \Omega_N; \Delta_N \to \Xi'; \Delta'}\ \mathsf{mc}\ p\ ok\ other$$

$$\frac{\mathsf{contc}\ \Delta_N; \Xi_N; \Delta_{N1}; C; \mathsf{comp}\ A \multimap B; \Omega_N \to \Xi'; \Delta'}{\mathsf{mc}\ \Delta; \Xi_N; \Delta_{N1}; \Xi; p, \Omega; C; \mathsf{comp}\ A \multimap B; \Omega_N; \Delta_N \to \Xi'; \Delta'}\ \mathsf{mc}\ p\ fail$$

$$\frac{\mathsf{mc}\ \Delta; \Xi_N; \Delta_{N1}; \Xi; X, Y, \Omega; C; \mathsf{comp}\ A \multimap B; \Omega_N; \Delta_N \to \Xi'; \Delta'}{\mathsf{mc}\ \Delta; \Xi_N; \Delta_{N1}; \Xi; X \otimes Y, \Omega; C; \mathsf{comp}\ A \multimap B; \Omega_N; \Delta_N \to \Xi'; \Delta'}\ \mathsf{mc}\ \otimes$$

$$\frac{\mathsf{dall}\ \Xi_N; \Delta_{N1}; \Xi; C; \mathsf{comp}\ A \multimap B; \Omega_N; \Delta_N \to \Xi'; \Delta'}{\mathsf{mc}\ \Delta; \Xi_N; \Delta_{N1}; \Xi; \cdot; C; \mathsf{comp}\ A \multimap B; \Omega_N; \Delta_N \to \Xi'; \Delta'}\ \mathsf{mc}\ end$$

7.2.6 Match Comprehension Continuation

If the matching process fails, we need to backtrack to the previous frame and restore the matching process at that point. The judgment that backtracks has the form $\mathsf{contc}\ \Delta_N; \Xi_N; \Delta_{N1}; C; AB; \Omega_N \to \Xi'; \Delta'$:

Δ_N : The multi-set of linear resources that were still available after matching the body of the rule.

Ξ_N : The multi-set of linear resources used during the matching process of the body of the rule.

Δ_{N1} : The multi-set of linear resources derived by the head of the rule.

C : The continuation stack.

AB : The comprehension $\mathsf{comp}\ A \multimap B$ that is being matched.

Ω_N : The ordered list of remaining terms of the head of the rule to be derived.

Ξ' : A multi-set of facts consumed during the whole process.

Δ' : A multi-set of linear facts derived.

$$\frac{\mathsf{d}_1\ \Delta_N;\Xi_N;\Delta_{N1};\Omega \rightarrow \Xi';\Delta'}{\mathsf{contc}\ \Delta_N;\Xi_N;\Delta_{N1};\cdot;\mathsf{comp}\ A \multimap B;\Omega \rightarrow \Xi';\Delta'}\ \mathsf{contc}\ \mathit{end}$$

$$\frac{\mathsf{mc}\ \Delta;\Xi_N;\Delta_{N1};\Xi;\Omega;(\Delta,p_1;\Delta'';\Xi;p;\Omega;\Lambda),C;AB;\Omega_N;\Delta_N \rightarrow \Xi';\Delta'}{\mathsf{contc}\ \Delta_N;\Xi_N;\Delta_{N1};(\Delta;p_1,\Delta'';\Xi;p;\Omega;\Lambda),C;AB;\Omega_N \rightarrow \Xi';\Delta'}\ \mathsf{contc}\ \mathit{next}$$

$$\frac{\mathsf{contc}\ \Delta_N;\Xi_N;\Delta_{N1};C;AB;\Omega_N \rightarrow \Xi';\Delta'}{\mathsf{contc}\ \Delta_N;\Xi_N;\Delta_{N1};(\Delta;\cdot;\Xi;p;\Omega;\Lambda),C;AB;\Omega_N \rightarrow \Xi';\Delta'}\ \mathsf{contc}\ \mathit{next\ empty}$$

7.2.7 Comprehension Stack Fixing

After a comprehension is matched and before we derive the head of such comprehension, we need to "fix" the comprehension stack. In a nutshell, we remove all the frames except the first frame and remove the consumed linear resources from this first frame so that the frame is still valid for the next application of the comprehension. The judgment that fixes the stack has the form $\mathsf{dall}\ \Xi_N;\Delta_{N1};\Xi;C;AB;\Omega_N;\Delta_N \rightarrow \Xi';\Delta'$:

Ξ_N : The multi-set of linear resources used during the matching process of the body of the rule.

Δ_{N1} : The multi-set of linear resources derived by the head of the rule.

Ξ : The multi-set of linear resources consumed by the previous application of the comprehension.

C : The continuation stack for the comprehension.

AB : The comprehension $\mathsf{comp}\ A \multimap B$ that is being matched.

Ω_N : The ordered list of remaining terms of the head of the rule to be derived.

Δ_N : The multi-set of linear resources that were still available after matching the body of the rule.

Ξ' : A multi-set of facts consumed during the whole process.

Δ' : A multi-set of linear facts derived.

$$\frac{\mathsf{dc}\ \Xi_N,\Xi;\Delta_{N1};(\Delta_x-\Xi;\Delta''-\Xi;\cdot;p;\Omega;\cdot);\mathsf{comp}\ A \multimap B;\Omega_N;(\Delta_N-\Xi) \rightarrow \Xi';\Delta'}{\mathsf{dall}\ \Xi_N;\Delta_{N1};\Xi;(\Delta_x;\Delta'';\cdot;p,\Omega;\cdot);\mathsf{comp}\ A \multimap B;\Omega_N;\Delta_N \rightarrow \Xi';\Delta'}\ \mathsf{dall}\ \mathit{end}$$

$$\frac{\mathsf{dall}\ \Xi_N;\Delta_{N1};\Xi;X,C;\mathsf{comp}\ A \multimap B;\Omega_N;\Delta_N \rightarrow \Xi';\Delta'}{\mathsf{dall}\ \Xi_N;\Delta_{N1};\Xi;\_,X,C;\mathsf{comp}\ A \multimap B;\Omega_N;\Delta_N \rightarrow \Xi';\Delta'}\ \mathsf{dall}\ \mathit{more}$$

7.2.8 Comprehension Derivation

After the fixing process, we commit to the facts consumed during the matching process by adding them to Ξ_N (see rule dall *more*) and start deriving the head of the comprehension. All the facts derived go directly to Δ_{N1}. The judgment has the form $\mathsf{dc}\ \Xi;\Delta_1;\Omega;C;AB;\Omega_N;\Delta_N \rightarrow \Xi';\Delta'$:

Ξ : The multi-set of linear resources consumed both by the body of the rule and all the comprehension applications.

Δ_1 : The multi-set of linear resources derived by the head of the rule and the all the comprehension applications.

Ω : Ordered list of terms to derive.

C	: The fixed continuation stack.
AB	: The comprehension $\mathsf{comp}\ A \multimap B$ that is being derived.
Ω_N	: The ordered list of remaining terms of the head of the rule to be derived.
Δ_N	: The multi-set of remaining linear resources that can be used for the next comprehension applications.
Ξ'	: A multi-set of facts consumed during the whole process.
Δ'	: A multi-set of linear facts derived.

$$\frac{\mathsf{dc}\ \Xi; \Delta_1, p; \Omega; C; \mathsf{comp}\ A \multimap B; \Omega_N; \Delta_N \rightarrow \Xi'; \Delta'}{\mathsf{dc}\ \Xi; \Delta_1; p, \Omega; C; \mathsf{comp}\ A \multimap B; \Omega_N; \Delta_N \rightarrow \Xi'; \Delta'}\ \mathsf{dc}\ p$$

$$\frac{\mathsf{dc}\ \Xi; \Delta_1; A, B, \Omega; C; \mathsf{comp}\ A \multimap B; \Omega_N; \Delta_N \rightarrow \Xi'; \Delta'}{\mathsf{dc}\ \Xi; \Delta_1; A \otimes B, \Omega; C; \mathsf{comp}\ A \multimap B; \Omega_N; \Delta_N \rightarrow \Xi'; \Delta'}\ \mathsf{dc}\ \otimes$$

$$\frac{\mathsf{contc}\ \Delta_N; \Xi; \Delta_1; C; \mathsf{comp}\ A \multimap B; \Omega_N \rightarrow \Xi'; \Delta'}{\mathsf{dc}\ \Xi; \Delta_1; \cdot; C; \mathsf{comp}\ A \multimap B; \Omega_N; \Delta_N \rightarrow \Xi'; \Delta'}\ \mathsf{dc}\ \mathit{end}$$

7.2.9 Apply

This whole process is started by the do_1 and a_1 judgments. The $\mathsf{do}_1\ \Delta; \Phi \rightarrow \Xi'; \Delta'$ judgment starts with a multi-set of linear resources Δ and an ordered list of rules that can be applied Φ. The $\mathsf{a}_1\ \Delta; A \multimap B; R \rightarrow \Xi'; \Delta'$ tries to apply the rule $A \multimap B$ and stores the rule continuation R so that if the current rule fails, we can try another one (in order).

$$\frac{\mathsf{m}_1\ \Delta; \cdot; A; B; \cdot; R \rightarrow \Xi'; \Delta'}{\mathsf{a}_1\ \Delta; A \multimap B; R \rightarrow \Xi'; \Delta'}\ \mathsf{a}_1\ \mathit{start\ matching}$$

$$\frac{\mathsf{a}_1\ \Delta; R; (\Phi; \Delta) \rightarrow \Xi'; \Delta'}{\mathsf{do}_1\ \Delta; R, \Phi \rightarrow \Xi'; \Delta'}\ \mathsf{do}_1\ \mathit{rule}$$

7.3 Definitions

7.3.1 Related Comprehension Match

Definition 9. *A match* $\mathsf{mc}\ \Delta_1, \Delta_2; \Xi_N; \Delta_{N1}; \Xi; \Omega; C; \mathsf{comp}\ A \multimap B; \Omega_N; \Delta_N \rightarrow \Xi'; \Delta'$ *is related to a term* A *and a context* Δ_{inv} *if it follows the same conditions described in 6.3.3.*

7.4 Theorems

7.4.1 Body Match Soundness Theorem

Theorem 10. *If* $\mathsf{m}_1\ \Delta, \Delta''; \cdot; \Omega; H; \cdot; R \rightarrow \Xi'; \Delta'$ *then either:*

1. $\mathsf{cont}\ \cdot; R; \Xi'; \Delta'$ *or;*
2. $\mathsf{m}_1\ \Delta; \Delta''; \cdot; H; C'; R \rightarrow \Xi'; \Delta'$ *and* $\mathsf{match}\ \Delta'' \rightarrow \Omega$

Proof. Proved in Section 6.4.2. □

7.4.2 Comprehension Match Soundness Theorem

With this theorem, we want to prove that when trying to match a giving comprehension $A \multimap B$ we either fail to apply it or we are able to apply it and get the corresponding match derivation at the high level.

Theorem 11.

- *Match sub-theorem*

  If a match $\mathsf{mc}\ \Delta_1, \Delta_2; \Xi_N; \Delta_{N1}; \Xi; \Omega; C; \mathsf{comp}\ A \multimap B; \Omega_N; \Delta_N \rightarrow \Xi'; \Delta'$ *is related to term* A *and context* $\Delta_N = \Delta_1, \Delta_2, \Xi$ *then either:*

  1. $\mathsf{d}_1\ \Delta_N; \Xi_N; \Delta_{N1}; \Omega_N \rightarrow \Xi'; \Delta'$*;*
  2. $\mathsf{match}\ \Delta_x \rightarrow A$ *(where* $\Delta_x \subseteq \Delta_N$*) and one of the following sub-cases is true:*
     (a) $\mathsf{mc}\ \Delta_1; \Xi_N; \Delta_{N1}, \Xi, \Delta_2; \cdot; C', C; \mathsf{comp}\ A \multimap B; \Omega_N; \Delta_N \rightarrow \Xi'; \Delta'$ *(related) and* $\Delta_x = \Delta_2$*.*
     (b) $\exists f = (\Delta_a; \Delta_{b_1}, p_2, \Delta_{b_2}; \Xi_1, ..., \Xi_m; p; \Omega_1, .., \Omega_k; \Lambda_1, ..., \Lambda_m) \in C$ *where* $C = C', f, C''$ *that turns into* $f' = (\Delta_a, \Delta_{b_1}, p_2; \Delta_{b_2}; \Xi_1, ..., \Xi_m; p; \Omega_1, ..., \Omega_k; \Lambda_1, ..., \Lambda_m)$ *such that:*
       – $\mathsf{mc}\ \Delta_c; \Xi_N; \Delta_{N1}; \Xi_1, ..., \Xi_m, p_2, \Xi_c; \cdot; C''', f', C''; \mathsf{comp}\ A \multimap B; \Omega_N; \Delta_N \rightarrow \Xi'; \Delta'$ *(related)*

- *Continuation sub-theorem*

  If $\mathsf{contc}\ \Delta_N; \Xi_N; \Delta_{N1}; C; \mathsf{comp}\ A \multimap B; \Omega_N \rightarrow \Xi'; \Delta'$ *and* C *is well formed in relation to* A *and* Δ_1, Δ_2, Ξ *then either:*

  1. $\mathsf{d}_1\ \Delta_N; \Xi_N; \Delta_{N1}; \Omega_N \rightarrow \Xi'; \Delta'$*;*
  2. $\exists f = (\Delta_a; \Delta_{b_1}, p_2, \Delta_{b_2}; \Xi_1, ..., \Xi_m; p; \Omega_1, .., \Omega_k; \Lambda_1, ..., \Lambda_m) \in C$ *where* $C = C', f, C''$ *that turns into* $f' = (\Delta_a, \Delta_{b_1}, p_2; \Delta_{b_2}; \Xi_1, ..., \Xi_m; p; \Omega_1, ..., \Omega_k; \Lambda_1, ..., \Lambda_m)$ *such that:*
     – $\mathsf{mc}\ \Delta_c; \Xi_N; \Delta_{N1}; \Xi_1, ..., \Xi_m, p_2, \Xi_c; \cdot; C''', f', C''; \mathsf{comp}\ A \multimap B; \Omega_N; \Delta_N \rightarrow \Xi'; \Delta'$ *(related)*

Proof. By nested induction. In mc on the size of Ω. In contc , first on the size of Δ'' of the continuation frame and then on the continuation stack C. Use the stack constraints and the Match Equivalence Theorem to prove $\mathsf{match}\ \Delta_x \rightarrow A$.

- mc p *ok first*

  By induction on Ω.
  Stack frame $(\Delta, p_1; \Delta''; \Xi; p; \Omega; \cdot)$ is related.

- mc p *ok other*

  By induction on Ω.
  First we prove that the inverted match is related. We know that $q \in \Xi$ due to our assumption, so that proves $\mathsf{match}\ q \rightarrow q$. Second, we prove that the new stack frame $(\Delta, p_1; \Delta''; \Xi; p; \Omega; q, \Lambda)$ is related and then apply induction.

- mc p *fail*

  By mutual induction on contc .

- mc $\otimes$

  By induction on Ω.

- mc *end*

  The stack must have some frames, thus $p, \Lambda_c \equiv A$ (p and Λ_c both arguments of the last frame). We also know that $p \in \Xi$ due to our assumption and that $\mathsf{match}\ \Xi_c, p \rightarrow \Delta \otimes p$ is true. Therefore $\Xi = \Xi_c, p$ and thus $\mathsf{match}\ \Xi \rightarrow A$.

- contc *end*

  Select case 1 by inverting the assumption.

- contc *next*

  By induction on Δ''.

- contc *next empty*

  By induction on the size of C.

□

7.4.3 Comprehension Soundness Lemma

Lemma 1. *If* $\mathsf{mc}\ \Delta, \Xi; \Xi_N; \Delta_{N1}; \cdot; A; \cdot; \mathsf{comp}\ A \multimap B; \Omega_N; \Delta_N \to \Xi'; \Delta'$ *where* $\Delta, \Xi = \Delta_N$ *and the match is related to both* Δ_N *and* A *then either:*

1. $\mathsf{d}_1\ \Delta_N; \Xi_N; \Delta_{N1}; \Omega_N \to \Xi'; \Delta'$ *or*
2. $\mathsf{mc}\ \Delta; \Xi_N; \Delta_{N1}; \Xi; \cdot; C'; \mathsf{comp}\ A \multimap B; \Omega_N; \Delta_N \to \Xi'; \Delta'$ *and*
   (a) $\mathsf{match}\ \Xi \to A$
   (b) C' *is well formed in relation to* A *and* Δ, Ξ.

Proof. Direct application of the previous theorem. □

7.4.4 Dall Transformation Theorem

Now, we want to prove that by applying the dall judgment, our continuation stack will only have the first frame and that such frame will be fixed.

Theorem 12. *If* $\mathsf{dall}\ \Xi_N; \Delta_{N1}; \Xi; C, (\Delta_a; \Delta_b; \cdot; \Omega; \cdot); \mathsf{comp}\ A \multimap B; \Omega_N; \Delta_N \to \Xi'; \Delta'$ *then* $\mathsf{dc}\ \Xi_N, \Xi; \Delta_{N1}; B; (\Delta_a - \Xi; \Delta_b - \Xi; \cdot; p; \Omega; \cdot); \mathsf{comp}\ A \multimap B; \Omega_N; (\Delta_N - \Xi) \to \Xi'; \Delta'$.

Proof. By induction on the size of the continuation stack C.

- $C = \_, X, C'$

  $\mathsf{dall}\ \Xi_N; \Delta_{N1}; \Xi; \_, X, C'; \mathsf{comp}\ A \multimap B; \Omega_N; \Delta_N \to \Xi'; \Delta'$ (1) assumption
  $\mathsf{dall}\ \Xi_N; \Delta_{N1}; \Xi; X, C'; \mathsf{comp}\ A \multimap B; \Omega_N; \Delta_N \to \Xi'; \Delta'$ (2) inversion of (1)
  $\mathsf{dc}\ \Xi_N, \Xi; \Delta_{N1}; B; (\Delta_a - \Xi; \Delta_b - \Xi; \cdot; \Omega; \cdot); \mathsf{comp}\ A \multimap B; \Omega_N; (\Delta_N - \Xi) \to \Xi'; \Delta'$ (3) i.h. on (2)

- $C = (\Delta_a - \Xi; \Delta_b - \Xi; \cdot; \Omega; B)$

  $\mathsf{dall}\ \Xi_N; \Delta_{N1}; \Xi; (\Delta_a; \Delta_b; \cdot; \Omega; \cdot); \mathsf{comp}\ A \multimap B; \Omega_N; \Delta_N \to \Xi'; \Delta'$ (1) assumption
  $\mathsf{dc}\ \Xi_N, \Xi; \Delta_{N1}; B; (\Delta_a - \Xi; \Delta_b - \Xi; \cdot; \Omega; B); \mathsf{comp}\ A \multimap B; \Omega_N; (\Delta_N - \Xi) \to \Xi'; \Delta'$ (2) inversion of (1)

□

7.4.5 Successful Comprehension Match Gives Derivation

We can apply the previous theorem to know that after a successful matching we will start the derivation process.

Lemma 2. *If* $\mathsf{mc}\ \Delta; \Xi_N; \Delta_{N1}; \Xi; \cdot; B; C, (\Delta_a; \Delta_b; \cdot; p; \Omega; \cdot); \mathsf{comp}\ A \multimap B; \Omega_N; \Delta_N \to \Xi'; \Delta'$ *then* $\mathsf{dc}\ \Xi_N, \Xi; \Delta_{N1}; B; (\Delta_a - \Xi; \Delta_b - \Xi; \cdot; p; \Omega; \cdot); \mathsf{comp}\ A \multimap B; \Omega_N; (\Delta_N - \Xi) \to \Xi'; \Delta'$.

Proof. Invert the assumption and then apply dall transformation theorem. □

7.4.6 Comprehension Derivation Theorem

We now prove that if we need to derive the head of the comprehension, we will finish this process and that giving a derivation at the high level with the same results, we can restore the terms we have derived. This second result will be used to prove the soundness of the comprehension mechanism.

Theorem 13. *If* $\mathsf{dc}\ \Xi_N; \Delta_{N1}; \Omega_1, ..., \Omega_n; C; \mathsf{comp}\ A \multimap B; \Omega_N; \Delta_N \rightarrow \Xi'; \Delta'$ *then:*

1. $\mathsf{dc}\ \Xi_N; \Delta_{N1}, \Delta_1, ..., \Delta_n; \cdot; C; \mathsf{comp}\ A \multimap B; \Omega_N; \Delta_N \rightarrow \Xi'; \Delta'$ *and*
2. *If* $\mathsf{derive}\ \Delta; \Xi_N; \Delta_{N1}, \Delta_1, ..., \Delta_n; \Omega \rightarrow \Xi'; \Delta'$ *then* $\mathsf{derive}\ \Delta; \Xi_N; \Delta_{N1}; \Omega_1, ..., \Omega_n, \Omega \rightarrow \Xi'; \Delta'$

Proof. By induction on the size of $\Omega_1, ..., \Omega_n$.

- p, Ω

  $\mathsf{dc}\ \Xi_N; \Delta_{N1}; p, \Omega_2, ..., \Omega_n; C; \mathsf{comp}\ A \multimap B; \Omega_N; \Delta_N \rightarrow \Xi'; \Delta'$ (1) assumption
  $\mathsf{dc}\ \Xi_N; \Delta_{N1}, p; \Omega_2, ..., \Omega_n; C; \mathsf{comp}\ A \multimap B; \Omega_N; \Delta_N \rightarrow \Xi'; \Delta'$ (2) inversion of (1)
  $\mathsf{dc}\ \Xi_N; \Delta_{N1}, p, \Delta_2, ..., \Delta_n; \cdot; C; \mathsf{comp}\ A \multimap B; \Omega_N; \Delta_N \rightarrow \Xi'; \Delta'$ (3) i.h. on (2)
  if $\mathsf{derive}\ \Delta; \Xi_N; \Delta_{N1}, p, \Delta_2, ..., \Delta_n; \Omega \rightarrow \Xi'; \Delta'$ then $\mathsf{derive}\ \Delta; \Xi_N; \Delta_{N1}; p, \Omega_2, ..., \Omega_n, \Omega \rightarrow \Xi'; \Delta'$ (4) i.h. on (2)
  $\mathsf{derive}\ \Delta; \Xi_N; \Delta_{N1}, p, \Delta_2, ..., \Delta_n; \Omega \rightarrow \Xi'; \Delta'$ (5) from (4)
  $\mathsf{derive}\ \Delta; \Xi_N; \Delta_{N1}; p, \Omega_2, ..., \Omega_n, \Omega \rightarrow \Xi'; \Delta'$ (6) using (5) on (4)

- $1, \Omega$

  By induction.

- $A \otimes B, \Omega$

  $\mathsf{dc}\ \Xi_N; \Delta_{N1}; A \otimes B, \Omega_2, ..., \Omega_n; C; \mathsf{comp}\ A \multimap B; \Omega_N; \Delta_N \rightarrow \Xi'; \Delta'$ (1) assumption
  $\mathsf{dc}\ \Xi_N; \Delta_{N1}; A, B, \Omega_2, ..., \Omega_n; C; \mathsf{comp}\ A \multimap B; \Omega_N; \Delta_N \rightarrow \Xi'; \Delta'$ (2) inversion of (1)
  Solve by induction on (2)

□

7.4.7 Comprehension Derivation Lemma

Lemma 3. *If* $\mathsf{dc}\ \Xi_N; \Delta_{N1}; \Omega_x; C; \mathsf{comp}\ A \multimap B; \Omega_N; \Delta_N \rightarrow \Xi'; \Delta'$ *then:*

1. $\mathsf{dc}\ \Xi_N; \Delta_{N1}, \Delta_x; \cdot; C; \mathsf{comp}\ A \multimap B; \Omega_N; \Delta_N \rightarrow \Xi'; \Delta'$ *and*
2. *If* $\mathsf{derive}\ \Delta; \Xi_N; \Delta_{N1}, \Delta_x; \Omega \rightarrow \Xi'; \Delta'$ *then* $\mathsf{derive}\ \Delta; \Xi_N; \Delta_{N1}; \Omega_x, \Omega \rightarrow \Xi'; \Delta'$

Proof. By direct application of the comprehension derivation theorem. □

7.4.8 Comprehension Theorem

This is the main theorem of the system. If we have a matching process with a single continuation frame we can derive $n \geq 0$ comprehensions and have n valid matching processes and n derivations at the high level. Since the second argument of the continuation frame will be reduced after each application, we have a valid induction hypotheses.

Theorem 14. *If* $\mathsf{mc}\ \Delta_a, \Delta'_b; \Xi_N; \Delta_{N1}; p_1; \Omega; (\Delta_a, p_1; \Delta'_b; \cdot; p; \Omega; \cdot); \mathsf{comp}\ A \multimap B; \Omega_N; \Delta, \Xi_1, ..., \Xi_n \rightarrow \Xi'; \Delta'$ *(related to A and* $\Delta_N = \Delta_a, \Delta_b$*) and* $\Delta_a, \Delta_b = \Delta, \Xi_1, ..., \Xi_n$ *then* $\exists n \geq 0$ *such that:*

1. $\mathsf{d}_1\ \Delta; \Xi_N, \Xi_1, ..., \Xi_n; \Delta_{N1}, \Delta_1, ..., \Delta_n; \Omega_N \rightarrow \Xi'; \Delta'$

2. $\mathsf{match}\ \Xi_1 \to A$ ... $\mathsf{match}\ \Xi_n \to A$

3. n implications from $1...i...n$ *such that:* $\forall \Omega_x, \Delta_x$. *if* $\mathsf{derive}\ \Delta_x; \Xi_N, \Xi_1, ..., \Xi_i; \Delta_{N1}, \Delta_1, ..., \Delta_i; \Omega_x \to \Xi'; \Delta'$ *then* $\mathsf{derive}\ \Delta_x; \Xi_N, \Xi_1, ..., \Xi_i; \Delta_{N1}, \Delta_1, ..., \Delta_{i-1}; B, \Omega_x \to \Xi'; \Delta'$.

Proof. By induction on the size of Δ'_b.

$\Delta_b = p_1, \Delta'_b$ (1) from assumption
$\Delta_a, p_1, \Delta'_b = \Delta, p, \Xi'_1, ..., \Xi_n$ (2) from assumption
By applying the comprehension soundness theorem on the assumption:

- Failure:
  $\mathsf{d_1}\ \Delta, \Xi_1, ..., \Xi_n; \Xi_N; \Delta_{N1}; \Omega_N \to \Xi'; \Delta'$ (3) assumption (so $n = 0$)

- Success $\mathsf{match}\ \Delta_x \to A$:

  1. Without backtracking:
     $\mathsf{mc}\ \Delta, \Xi_2, ..., \Xi_n; \Xi_N; \Delta_{N1}; p, \Xi'_1; \cdot; C', (\Delta_a, p_1; \Delta'_b; \cdot; p; \Omega; \cdot); \mathsf{comp}\ A \multimap B; \Omega_N; \Delta, \Xi_1, ..., \Xi_n \to \Xi'; \Delta'$ (3) assumption
     $\mathsf{match}\ p_1, \Xi'_1 \to A$ (4) from theorem
     $\mathsf{dc}\ \Xi_N, p, \Xi'_1; \Delta_{N1}; B; (\Delta_a, p_1 - (p_1, \Xi'_1); \Delta'_b - (p_1, \Xi'_1); \cdot; p; \Omega; \cdot); \mathsf{comp}\ A \multimap B; \Omega_N; (\Delta, \Xi_1, ..., \Xi_n) - \Xi_1 \to \Xi'; \Delta'$ (5) apply successful comprehension matches gives derivation lemma to (3)
     $\mathsf{dc}\ \Xi_N, p, \Xi'_1; \Delta_{N1}; B; (\Delta_a - \Xi'_1; \Delta'_b - \Xi'_1; \cdot; p; \Omega; \cdot); \mathsf{comp}\ A \multimap B; \Omega_N; \Delta, \Xi_2, ..., \Xi_n \to \Xi'; \Delta'$ (6) simplification of (5)
     $\mathsf{dc}\ \Xi_N, p, \Xi'_1; \Delta_{N1}, \Delta_1; \cdot; (\Delta_a - \Xi'_1; \Delta'_b - \Xi'_1; \cdot; p; \Omega; \cdot); \mathsf{comp}\ A \multimap B; \Omega_N; \Delta, \Xi_2, ..., \Xi_n \to \Xi'; \Delta'$ (7) from comprehension derivation lemma on (6)
     if $\forall \Omega_x, \Delta_x.\mathsf{derive}\ \Delta_x; \Xi_N, p, \Xi'_1; \Delta_{N1}, \Delta_1; \Omega_x \to \Xi'; \Delta'$ then $\mathsf{derive}\ \Delta_x; \Xi_N, p, \Xi'_1; \Delta_{N1}; B, \Omega_x \to \Xi'; \Delta'$ (8) using the same lemma
     $\mathsf{contc}\ \Delta, \Xi_2, ..., \Xi_n; \Xi_N, p, \Xi'_1; \Delta_{N1}, \Delta_1; (\Delta_a - \Xi'_1; \Delta'_b - \Xi'_1; \cdot; p; \Omega; \cdot); \mathsf{comp}\ A \multimap B; \Omega_N \to \Xi'; \Delta'$ (9) inversion of (7)

     When inverting (9), we have two sub-cases:
     $\Delta_a - \Xi'_1 = \Delta''_a$ (10)
     $\Delta'_b - \Xi'_1 = \Delta''_b$ (11)
     $\Delta''_a, \Delta''_b = \Delta, \Xi_2, ..., \Xi_n$ (12)

     - End ($n = 1$):
       $\mathsf{contc}\ \Delta, \Xi_2, ..., \Xi_n; \Xi_N, \Xi_1; \Delta_{N1}, \Delta_1; \cdot; \mathsf{comp}\ A \multimap B; \Omega_N \to \Xi'; \Delta'$ (13) inversion of (9)
       $\mathsf{d_1}\ \Delta, \Xi_2, ..., \Xi_n; \Xi_N, \Xi_1; \Delta_{N1}, \Delta_1; \Omega_N \to \Xi'; \Delta'$ (14) inverting (13), which is what we want

     - Next ($n = n' + 1$):
       $\Delta'''_b = \Delta''_b, p_2$ (13) from inversion
       $\mathsf{mc}\ \Delta''_a, \Delta'''_b; \Xi_N, \Xi_1; \Delta_{N1}, \Delta_1; \cdot; \Omega; (\Delta''_a, p_2; \Delta'''_b; \cdot; p; \Omega; \cdot); \mathsf{comp}\ A \multimap B; \Omega_N; \Delta, \Xi_2, ..., \Xi_n \to \Xi'; \Delta'$ (14) inversion of (9)
       Apply induction hypotheses to (14) to get results from n'.

  2. With backtracking:
     $f = (\Delta_a, p_1; \Delta'_b; \cdot; p; \Omega; \cdot)$ (3) from theorem
     turns into $f' = (\Delta_a, p_1, \Delta'''_b, p_2; \Delta''_b; \cdot; p; \Omega; \cdot)$ (4) from theorem (3)

$\mathsf{mc}\ \Delta, \Xi_2, ..., \Xi_n; \Xi_N; \Delta_{N1}; p_2, \Xi_1'; \cdot; C', f'; \mathsf{comp}\ A \multimap B; \Omega_N; \Delta, \Xi_1, ..., \Xi_n \rightarrow \Xi'; \Delta'$ (5) from theorem (3)
$\mathsf{match}\ p_2, \Xi_1' \rightarrow p \otimes \Omega$ (6) from theorem (3) where $p_2, \Xi_1' = \Delta_x$
$p, \Omega \equiv A$ (7) from assumptions
$\mathsf{match}\ p_2, \Xi_1' \rightarrow A$ (8) using match equivalence theorem

Use the same approach as the last sub-case, but using p_2 instead of p_1 and using the fact that Δ_b was already reduced because we had to backtrack.

□

7.4.9 Comprehension Lemma

For this lemma, we start the matching process with an empty continuation stack. If the first application of the comprehension succeeds we get a continuation with a single frame that can be used to prove the other n applications of the comprehension by using the main theorem.

Lemma 4. *If* $\mathsf{mc}\ \Delta, \Xi_1, ..., \Xi_n; \Xi_N; \Delta_{N1}; \cdot; A; \cdot; \mathsf{comp}\ A \multimap B; \Omega_N; \Delta, \Xi_1, ..., \Xi_n \rightarrow \Xi'; \Delta'$ *then we can apply the comprehension* $n >= 0$ *times:*

1. $\mathsf{d}_1\ \Delta; \Xi_N, \Xi_1, ..., \Xi_n; \Delta_{N1}, \Delta_1, .., \Delta_n; \Omega_N \rightarrow \Xi'; \Delta'$ *for* $\exists n \geq 0$
2. $\mathsf{match}\ \Xi_1 \rightarrow A\ ...\ \mathsf{match}\ \Xi_n \rightarrow A.$
3. *n implications from $1...i...n$ such that:* $\forall \Omega_x, \Delta_x$. *if* $\mathsf{derive}\ \Delta_x; \Xi_N, \Xi_1, ..., \Xi_i; \Delta_{N1}, \Delta_1, ..., \Delta_i; \Omega_x \rightarrow \Xi'; \Delta'$ *then* $\mathsf{derive}\ \Delta_x; \Xi_N, \Xi_1, ..., \Xi_i; \Delta_{N1}, \Delta_1, ..., \Delta_{i-1}; B, \Omega_x \rightarrow \Xi'; \Delta'$.

Proof. Applying the comprehension soundness lemma, we get two cases:

- Failure:

  $\mathsf{d}_1\ \Delta, \Xi_1, ..., \Xi_n; \Xi_N; \Delta_{N1}; \Omega_N \rightarrow \Xi'; \Delta'$ no comprehension application was possible.

- Success:

  $\mathsf{mc}\ \Delta, \Xi_2, ..., \Xi_n; \Xi_N; \Delta_{N1}; \Xi_1; \cdot; C'; \mathsf{comp}\ A \multimap B; \Omega_N; \Delta, \Xi_1, ..., \Xi_n \rightarrow \Xi'; \Delta'$ (1) result from theorem
  $\mathsf{match}\ \Xi_1 \rightarrow A$ (2) from theorem
  mc in (1) is related (3) from theorem
  C is well formed in relation to A and $\Delta, \Xi_1, ..., \Xi_n$ (4) from theorem

  $\mathsf{dc}\ \Xi_N, \Xi_1; \Delta_{N1}; B; (\Delta_a - \Xi_1; \Delta_b - \Xi_1; \cdot; p; \Omega; \cdot); \mathsf{comp}\ A \multimap B; \Omega_N; (\Delta, \Xi_1, ..., \Xi_n - \Xi_1) \rightarrow \Xi'; \Delta'$ (5) applying successful comprehension match gives derivation
  $\mathsf{dc}\ \Xi_N, \Xi_1; \Delta_{N1}; B; (\Delta_a - \Xi_1; \Delta_b - \Xi_1; \cdot; p; \Omega; \cdot); \mathsf{comp}\ A \multimap B; \Omega_N; \Delta, \Xi_2, ..., \Xi_n \rightarrow \Xi'; \Delta'$ (6) simplifying (5)
  $\Delta_a, \Delta_b = \Delta, \Xi_1, ..., \Xi_n$ (7) from (4)
  $(\Delta_a - \Xi_1), (\Delta_b - \Xi_1) = (\Delta_a, \Delta_b) - \Xi_1 = \Delta, \Xi_2, .., \Xi_n$ and $\Delta_a' = \Delta_a - \Xi_1$ and $\Delta_b' = \Delta_b - \Xi_1$ (8) from (7) and (6)
  $p, \Omega \equiv A$ (9) from (4) and (6)
  $\mathsf{dc}\ \Xi_N, \Xi_1; \Delta_{N1}, \Delta_1; \cdot; (\Delta_a'; \Delta_b'; \cdot; p; \Omega; \cdot); \mathsf{comp}\ A \multimap B; \Omega_N; \Delta, \Xi_2, ..., \Xi_n \rightarrow \Xi'; \Delta'$ (10) using comprehension derivation lemma
  if $\forall \Omega_x, \Delta_x. \mathsf{derive}\ \Delta_x; \Xi_N, \Xi_1; \Delta_{N1}, \Delta_1; \Omega_x \rightarrow \Xi'; \Delta'$ then $\mathsf{derive}\ \Delta_x; \Xi_N, \Xi_1; \Delta_{N1}; B, \Omega_x \rightarrow \Xi'; \Delta'$ (11) using the same lemma
  $\mathsf{contc}\ \Delta, \Xi_2, ..., \Xi_n; \Xi_N, \Xi_1; \Delta_{N1}, \Delta_1; (\Delta_a'; \Delta_b'; \cdot; p; \Omega; \cdot); \mathsf{comp}\ A \multimap B; \Omega_N \rightarrow \Xi'; \Delta'$ (12) inversion of (10)

  Invert (12) to get either 0 applications of the comprehension or apply the comprehension theorem to the inversion to get the remaining $n - 1$.

□

7.4.10 Derivation Soundness Theorem

Finally, we can prove that the derivation of the head of the rule is sound. The only complicated case is the comprehension. Since we know that applying the comprehension will result in n applications, we can use the results of the main lemma to reconstruct the derivation tree at the high level by using the n applications and the induction hypothesis.

Theorem 15. *If* $\mathsf{d}_1\ \Delta; \Xi; \Delta_1; \Omega \rightarrow \Xi'; \Delta$ *then* $\mathsf{derive}\ \Delta; \Xi; \Delta_1; \Omega \rightarrow \Xi'; \Delta'$.

Proof. By induction on Ω.

- p, Ω

  By induction.

- $1, \Omega$

  By induction.

- $A \otimes B, \Omega$

  By induction.

- $\cdot$

  Use axiom.

- $\mathsf{comp}\ A \multimap B, \Omega$

  By using the comprehension lemma, we get n applications of the comprehension.
  $\Delta = \Delta, \Xi_1, ..., \Xi_n$ (1)
  $\mathsf{d}_1\ \Delta; \Xi, \Xi_1, ..., \Xi_n; \Delta_1, \Delta_{c_1}, ..., \Delta_{c_n}; \Omega \rightarrow \Xi'; \Delta'$ (2) from lemma
  $\mathsf{derive}\ \Delta; \Xi, \Xi_1, ..., \Xi_n; \Delta_1, \Delta_{c_1}, ..., \Delta_{c_n}; \Omega \rightarrow \Xi'; \Delta'$ (3) i.h. on (2)
  $\mathsf{derive}\ \Delta; \Xi, \Xi_1, ..., \Xi_n; \Delta_1, \Delta_{c_1}, ..., \Delta_{c_n}; 1, \Omega \rightarrow \Xi'; \Delta'$ (4) apply $\mathsf{derive}\ 1$ on (3)
  $\mathsf{derive}\ \Delta; \Xi, \Xi_1, ..., \Xi_n; \Delta_1, \Delta_{c_1}, ..., \Delta_{c_n}; 1 \mathbin{\&} (A \multimap B \otimes \mathsf{comp}\ A \multimap B), \Omega \rightarrow \Xi'; \Delta'$ (5) apply $\mathsf{derive}\ \mathbin{\&} L$ on (4)
  $\mathsf{derive}\ \Delta; \Xi, \Xi_1, ..., \Xi_n; \Delta_1, \Delta_{c_1}, ..., \Delta_{c_n}; \mathsf{comp}\ A \multimap B, \Omega \rightarrow \Xi'; \Delta'$ (6) apply $\mathsf{derive}\ \mathit{comp}$ on (5)
  Using the n^{th} implication of the comprehension theorem:
  $\mathsf{derive}\ \Delta; \Xi, \Xi_1, ..., \Xi_n; \Delta_1, \Delta_{c_1}, ..., \Delta_{c_{n-1}}; B, \mathsf{comp}\ A \multimap B, \Omega \rightarrow \Xi'; \Delta'$ (7)
  Using the $\mathsf{match}\ \Xi_n \rightarrow A$ result:
  $\mathsf{derive}\ \Delta, \Xi_n; \Xi_1, ..., \Xi_{n-1}; \Delta_1, \Delta_{c_1}, ..., \Delta_{c_{n-1}}; A \multimap B, \mathsf{comp}\ A \multimap B, \Omega \rightarrow \Xi'; \Delta'$ (8) using match on (7)
  $\mathsf{derive}\ \Delta, \Xi_n; \Xi_1, ..., \Xi_{n-1}; \Delta_1, \Delta_{c_1}, ..., \Delta_{c_{n-1}}; A \multimap B \otimes \mathsf{comp}\ A \multimap B, \Omega \rightarrow \Xi'; \Delta'$ (9) applying rule $\mathsf{derive} \otimes$ on (8)
  $\mathsf{derive}\ \Delta, \Xi_n; \Xi_1, ..., \Xi_{n-1}; \Delta_1, \Delta_{c_1}, ..., \Delta_{c_{n-1}}; 1 \mathbin{\&} (A \multimap B \otimes \mathsf{comp}\ A \multimap B), \Omega \rightarrow \Xi'; \Delta'$ (10) applying rule $\mathsf{derive}\ \mathbin{\&} R$ on (9)
  $\mathsf{derive}\ \Delta, \Xi_n; \Xi_1, ..., \Xi_{n-1}; \Delta_1, \Delta_{c_1}, ..., \Delta_{c_{n-1}}; \mathsf{comp}\ A \multimap B, \Omega \rightarrow \Xi'; \Delta'$ (11) applying rule $\mathsf{derive}\ \mathit{comp}$ on (10)
  By applying the other $n-1$ comprehensions we will get:
  $\mathsf{derive}\ \Delta, \Xi_1, ..., \Xi_n; \Delta_1; \mathsf{comp}\ A \multimap B, \Omega \rightarrow \Xi'; \Delta'$ (12)

□

8 Low Level System With Persistent Facts

8.1 High Level System

8.1.1 Match

$$\frac{}{\mathsf{match}\ \Gamma; \cdot \rightarrow 1}\ \mathsf{match}\ 1 \qquad \frac{}{\mathsf{match}\ \Gamma; p \rightarrow p}\ \mathsf{match}\ p \qquad \frac{}{\mathsf{match}\ \Gamma, p; \cdot \rightarrow !p}\ \mathsf{match}\ !p$$

$$\frac{\mathsf{match}\ \Gamma; \Delta_1 \rightarrow A \quad \mathsf{match}\ \Delta_2 \rightarrow B}{\mathsf{match}\ \Gamma; \Delta_1, \Delta_2 \rightarrow A \otimes B}\ \mathsf{match} \otimes$$

8.1.2 Derive

$$\frac{\mathsf{derive}\ \Gamma; \Delta; \Xi; \Gamma_1; p, \Delta_1; \Omega \rightarrow \Xi'; \Delta'; \Gamma'}{\mathsf{derive}\ \Gamma; \Delta; \Xi; \Gamma_1; \Delta_1; p, \Omega \rightarrow \Xi'; \Delta'; \Gamma'}\ \mathsf{derive}\ p$$

$$\frac{\mathsf{derive}\ \Gamma; \Delta; \Xi; \Gamma_1, p; \Delta_1; \Omega \rightarrow \Xi'; \Delta'; \Gamma'}{\mathsf{derive}\ \Gamma; \Delta; \Xi; \Gamma_1; \Delta_1; !p, \Omega \rightarrow \Xi'; \Delta'; \Gamma'}\ \mathsf{derive}\ !p$$

$$\frac{\mathsf{derive}\ \Gamma; \Delta; \Xi; \Gamma_1; \Delta_1; A, B, \Omega \rightarrow \Xi'; \Delta'; \Gamma'}{\mathsf{derive}\ \Gamma; \Delta; \Xi; \Gamma_1; \Delta_1; A \otimes B, \Omega \rightarrow \Xi'; \Delta'; \Gamma'}\ \mathsf{derive} \otimes$$

$$\frac{\mathsf{derive}\ \Gamma; \Delta; \Xi; \Gamma_1; \Delta_1; \Omega \rightarrow \Xi'; \Delta'; \Gamma'}{\mathsf{derive}\ \Gamma; \Delta; \Xi; \Gamma_1; \Delta_1; 1, \Omega \rightarrow \Xi'; \Delta'; \Gamma'}\ \mathsf{derive}\ 1$$

$$\frac{}{\mathsf{derive}\ \Gamma; \Delta; \Xi'; \Gamma'; \Delta'; \cdot \rightarrow \Xi'; \Delta'; \Gamma'}\ \mathsf{derive}\ end$$

$$\frac{\mathsf{derive}\ \Gamma; \Delta; \Xi; \Gamma_1; \Delta_1; 1 \,\&\, (A \multimap B \otimes \mathsf{comp}\ A \multimap B), \Omega \rightarrow \Xi'; \Delta'; \Gamma'}{\mathsf{derive}\ \Gamma; \Delta; \Xi; \Gamma_1; \Delta_1; \mathsf{comp}\ A \multimap B, \Omega \rightarrow \Xi'; \Delta'; \Gamma'}\ \mathsf{derive}\ comp$$

$$\frac{\mathsf{match}\ \Gamma; \Delta_a \rightarrow A \quad \mathsf{derive}\ \Gamma; \Delta_b; \Xi, \Delta_a; \Gamma_1; \Delta_1; B, \Omega \rightarrow \Xi'; \Delta'; \Gamma'}{\mathsf{derive}\ \Gamma; \Delta_a, \Delta_b; \Xi; \Gamma_1; \Delta_1; A \multimap B, \Omega \rightarrow \Xi'; \Delta'; \Gamma'}\ \mathsf{derive} \multimap$$

$$\frac{\mathsf{derive}\ \Gamma; \Delta; \Xi; \Gamma_1; \Delta_1; A, \Omega \rightarrow \Xi'; \Delta'; \Gamma'}{\mathsf{derive}\ \Gamma; \Delta; \Xi; \Gamma_1; \Delta_1; A \,\&\, B, \Omega \rightarrow \Xi'; \Delta'; \Gamma'}\ \mathsf{derive}\ \& L \qquad \frac{\mathsf{derive}\ \Gamma; \Delta; \Xi; \Gamma_1; \Delta_1; B, \Omega \rightarrow \Xi'; \Delta'; \Gamma'}{\mathsf{derive}\ \Gamma; \Delta; \Xi; \Gamma_1; \Delta_1; A \,\&\, B, \Omega \rightarrow \Xi'; \Delta'; \Gamma'}\ \mathsf{derive}\ \& R$$

8.1.3 Apply

$$\frac{\mathsf{match}\ \Gamma; \Delta; A \rightarrow \Delta \quad \mathsf{derive}\ \Gamma; \Delta''; \Delta; \cdot; \cdot; B \rightarrow \Xi'; \Delta'; \Gamma'}{\mathsf{a}_0\ \Gamma; \Delta, \Delta''; A \multimap B \rightarrow \Xi'; \Delta'; \Gamma'}\ \mathsf{a}_0\ rule$$

$$\frac{\mathsf{do}_0\ \Gamma; \Delta; R \rightarrow \Xi'; \Delta'; \Gamma'}{\mathsf{do}_0\ \Gamma; \Delta; R, \Phi \rightarrow \Xi'; \Delta'; \Gamma'}\ \mathsf{do}_0\ rule$$

8.2 Low Level System

We extend the previous system with persistent facts. Most judgments are extended with the Γ context for persistent facts. While the matching mechanism has new frames for persistent facts, everything else remains the same. Because persistent facts are never consumed, we only need to make sure that the frames read the facts from the Γ context.

All the judgments in this system have been extended with Γ', the context that contains the persistent resources created during the application of some rule.

8.2.1 Continuation Frames

The system adds a new continuation frame for persistent facts with the form $[\Gamma'; \Delta; !p; \Omega; \Xi; \Lambda; \Upsilon]$:

Γ' : A multi-set of persistent resources that can be used if the current one fails.

Δ : The multi-set of linear resources at this point of the matching process.

$!p$: The persistent atomic term that created this frame.

Ω : The remaining terms we need to match past this choice point. This is an ordered list.

Ξ : A multi-set of linear resources we have consumed to reach this point.

Λ : A multi-set of linear atomic terms that we have matched to reach this choice point. All the linear resources that match these terms are located in Ξ.

Υ : A multi-set of persistent atomic terms that we have matched to reach this point. All the persistent resources used for matching must be located in the Γ of the matching judgment.

Please note that the old continuation frame is also extended with Υ.

8.2.2 Match

The m_1 judgments is simply extended with arguments Γ and Γ'. However, the number of inference rules has duplicated due to the presence of persistent frames and persistent terms.

$$\frac{\mathsf{m}_1\ \Gamma; \Delta, \Delta''; \Xi, p_1; \Omega; H; (\Delta, p_1; \Delta''; p; \Omega; \Xi; \cdot; \cdot); R \to \Xi'; \Delta'; \Gamma'}{\mathsf{m}_1\ \Gamma; \Delta, p_1, \Delta''; \Xi; p, \Omega; H; C; R \to \Xi'; \Delta'; \Gamma'}\ \mathsf{m}_1\ p\ ok\ first$$

$$\frac{\begin{array}{l}\mathsf{m}_1\ \Gamma; \Delta, \Delta''; \Xi, p_1; \Omega; H; (\Delta, p_1; \Delta''; p; \Omega; \Xi; q, \Lambda; \Upsilon), C_1, C; R \to \Xi'; \Delta'; \Gamma' \\ C_1 = (\Delta_{old}; \Delta'_{old}; q; \Omega_{old}; \Xi_{old}; \Lambda; \Upsilon)\end{array}}{\mathsf{m}_1\ \Gamma; \Delta, p_1, \Delta''; \Xi; p, \Omega; H; C_1, C; R \to \Xi'; \Delta'; \Gamma'}\ \mathsf{m}_1\ p\ ok\ other\ q$$

$$\frac{\begin{array}{l}\mathsf{m}_1\ \Gamma; \Delta, \Delta''; \Xi, p_1; \Omega; H; (\Delta, p_1; \Delta''; p; \Omega; \Xi; \Lambda; q, \Upsilon), C_1, C; R \to \Xi'; \Delta'; \Gamma' \\ C_1 = [\Gamma_{old}; \Delta_{old}; !q; \Omega_{old}; \Xi_{old}; \Lambda; \Upsilon]\end{array}}{\mathsf{m}_1\ \Gamma; \Delta, p_1, \Delta''; \Xi; p, \Omega; H; C_1, C; R \to \Xi'; \Delta'; \Gamma'}\ \mathsf{m}_1\ p\ ok\ other\ !q$$

$$\frac{p \notin \Delta \quad \mathsf{cont}\ C; H; R; \Gamma; \Xi'; \Delta'; \Gamma'}{\mathsf{m}_1\ \Gamma; \Delta; \Xi; p, \Omega; H; C; R \to \Xi'; \Delta'; \Gamma'}\ \mathsf{m}_1\ p\ fail$$

$$\frac{\mathsf{m}_1\ \Gamma, p, \Gamma'; \Delta; \Xi; \Omega; H; [\Gamma'; \Delta; !p; \Omega; \Xi; \Lambda; \cdot]; R \to \Xi'; \Delta'; \Gamma'}{\mathsf{m}_1\ \Gamma, p, \Gamma'; \Delta; \Xi; !p, \Omega; H; \cdot; R \to \Xi'; \Delta'; \Gamma'}\ \mathsf{m}_1\ !p\ ok\ first$$

$$\frac{\begin{array}{c}\mathsf{m}_1\ \Gamma, p, \Gamma'; \Delta; \Xi; \Omega; H; [\Gamma'; \Delta; !p; \Omega; \Xi; q, \Lambda; \Upsilon], C_1, C; R \to \Xi'; \Delta'; \Gamma' \\ C_1 = (\Delta_{old}; \Delta'_{old}; q; \Omega_{old}; \Xi_{old}; \Lambda; \Upsilon)\end{array}}{\mathsf{m}_1\ \Gamma, p, \Gamma'; \Delta; \Xi; !p, \Omega; H; C_1, C; R \to \Xi'; \Delta'; \Gamma'}\ \mathsf{m}_1\ !p\ \textit{ok other}\ q$$

$$\frac{\begin{array}{c}\mathsf{m}_1\ \Gamma, p, \Gamma'; \Delta; \Xi; \Omega; H; [\Gamma'; \Delta; !p; \Omega; \Xi; \Lambda; q, \Upsilon], C_1, C; R \to \Xi'; \Delta'; \Gamma' \\ C_1 = [\Gamma_{old}; \Delta_{old}; !q; \Omega_{old}; \Xi_{old}; \Lambda; \Upsilon]\end{array}}{\mathsf{m}_1\ \Gamma, p, \Gamma'; \Delta; \Xi; !p, \Omega; H; C_1, C; R \to \Xi'; \Delta'; \Gamma'}\ \mathsf{m}_1\ !p\ \textit{ok other}\ !q$$

$$\frac{p \notin \Gamma \quad \mathsf{cont}\ C; H; R; \Gamma; \Xi'; \Delta'; \Gamma'}{\mathsf{m}_1\ \Gamma; \Delta; \Xi; !p, \Omega; H; C; R \to \Xi'; \Delta'; \Gamma'}\ \mathsf{m}_1\ !p\ \textit{fail}$$

$$\frac{\mathsf{m}_1\ \Gamma; \Delta; \Xi; A, B, \Omega; H; C; R \to \Xi'; \Delta'; \Gamma'}{\mathsf{m}_1\ \Gamma; \Delta; \Xi; A \otimes B, \Omega; H; C; R \to \Xi'; \Delta'; \Gamma'}\ \mathsf{m}_1\ \otimes$$

$$\frac{\mathsf{d}_1\ \Gamma; \Delta; \Xi; \cdot; H; \cdot \to \Xi'; \Delta'; \Gamma'}{\mathsf{m}_1\ \Gamma; \Delta; \Xi; \cdot; H; C; R \to \Xi'; \Delta'; \Gamma'}\ \mathsf{m}_1\ \textit{end}$$

8.2.3 Continuation

The cont judgment has been extended with the Γ context. We also need to handle the case where the top of the stack contains persistent frames.

$$\frac{\mathsf{do}_1\ \Gamma; \Delta; \Phi \to \Xi'; \Delta'; \Gamma'}{\mathsf{cont}\ \cdot; H; (\Phi, \Delta); \Gamma; \Xi'; \Delta'; \Gamma'}\ \mathsf{cont}\ \textit{next rule}$$

$$\frac{\mathsf{m}_1\ \Gamma; \Delta, \Delta''; \Xi, p_1; \Omega; H; (\Delta, p_1; \Delta''; p, \Omega; H; \Xi; \Lambda; \Upsilon), C; R \to \Xi'; \Delta'; \Gamma'}{\mathsf{cont}\ (\Delta; p_1, \Delta''; p, \Omega; \Xi; \Lambda; \Upsilon), C; H; R; \Gamma; \Xi'; \Delta'; \Gamma'}\ \mathsf{cont}\ p\ \textit{next}$$

$$\frac{\mathsf{cont}\ C; H; R; \Gamma; \Xi'; \Delta'; \Gamma'}{\mathsf{cont}\ (\Delta; \cdot; p, \Omega; \Xi; \Lambda; \Upsilon), C; H; R; \Gamma; \Xi'; \Delta'; \Gamma'}\ \mathsf{cont}\ p\ \textit{no more}$$

$$\frac{\mathsf{m}_1\ \Gamma; \Delta; \Xi; \Omega; H; [\Gamma'; \Delta; !p, \Omega; \Xi; \Lambda; \Upsilon], C; R \to \Xi'; \Delta'; \Gamma'}{\mathsf{cont}\ [p_1, \Gamma'; \Delta; !p, \Omega; \Xi; \Lambda; \Upsilon], C; H; R; \Gamma; \Xi'; \Delta'; \Gamma'}\ \mathsf{cont}\ !p\ \textit{next}$$

$$\frac{\mathsf{cont}\ C; H; R; \Gamma; \Xi'; \Delta'; \Gamma'}{\mathsf{cont}\ [\cdot; \Delta; !p, \Omega; \Xi; \Lambda; \Upsilon], C; H; R; \Gamma; \Xi'; \Delta'; \Gamma'}\ \mathsf{cont}\ !p\ \textit{no more}$$

8.2.4 Derivation

We extended with d_1 judgment with Γ and Γ_1. Γ_1 contains the derived persistent resources.

$$\frac{\mathsf{d}_1\ \Gamma; \Delta; \Xi; \Gamma_1; p, \Delta_1; \Omega \to \Xi'; \Delta'; \Gamma'}{\mathsf{d}_1\ \Gamma; \Delta; \Xi; \Gamma_1; \Delta_1; p, \Omega \to \Xi'; \Delta'; \Gamma'}\ \mathsf{d}_1\ p \qquad \frac{\mathsf{d}_1\ \Gamma; \Delta; \Xi; \Gamma_1; \Delta_1; \Omega \to \Xi'; \Delta'; \Gamma'}{\mathsf{d}_1\ \Gamma; \Delta; \Xi; \Gamma_1; \Delta_1; 1, \Omega \to \Xi'; \Delta'; \Gamma'}\ \mathsf{d}_1\ 1$$

$$\frac{\mathsf{d}_1\ \Gamma; \Delta; \Xi; \Gamma_1, p; \Delta_1; \Omega \to \Xi'; \Delta'; \Gamma'}{\mathsf{d}_1\ \Gamma; \Delta; \Xi; \Gamma_1; \Delta_1; !p, \Omega \to \Xi'; \Delta'; \Gamma'}\ \mathsf{d}_1\ !p$$

$$\frac{\mathsf{d}_1\ \Gamma; \Delta; \Xi; \Gamma_1; \Delta_1; A, B, \Omega \to \Xi'; \Delta'; \Gamma'}{\mathsf{d}_1\ \Gamma; \Delta; \Xi; \Gamma_1; \Delta_1; A \otimes B, \Omega \to \Xi'; \Delta'; \Gamma'}\ \mathsf{d}_1 \otimes$$

$$\frac{}{\mathsf{d}_1\ \Gamma; \Delta; \Xi; \Gamma_1; \Delta_1; \cdot \to \Xi; \Delta_1; \Gamma_1}\ \mathsf{d}_1\ end$$

$$\frac{\mathsf{mc}\ \Gamma; \Delta; \Xi; \Gamma_1; \Delta_1; \cdot; A; B; \cdot; \cdot; \mathsf{comp}\ A \multimap B; \Omega; \Delta \to \Xi'; \Delta'; \Gamma'}{\mathsf{d}_1\ \Gamma; \Delta; \Xi; \Gamma_1; \Delta_1; \mathsf{comp}\ A \multimap B, \Omega \to \Xi'; \Delta'; \Gamma'}\ \mathsf{d}_1\ comp$$

8.2.5 Match Comprehension

For the matching process of the comprehensions, we use two stacks, C and P. In P, we put all the initial persistent frames and in C we put the first linear frame and then everything else. With this we can easily find out the first linear frame and remove everything that was pushed on top of such frame. The match comprehension judgment mc has been extended with persistent frames and a few other arguments:

Γ	: The multi-set of persistent resources.
Γ_{N1}	: Multi-set of persistent resources derived up to this point in the head of the rule.
C	: The continuation stack that contains both linear and persistent frames. The first frame must be linear.
P	: The second part of the continuation stack with only persistent frames.
Γ'	: Multi-set of derived persistent resources.

Like the m_1 judgment, we can see a duplication of inference rules due to the presence of persistent frames.

$$\frac{\mathsf{mc}\ \Gamma; \Delta, \Delta''; \Xi_N; \Gamma_{N1}; \Delta_{N1}; \Xi, p_1; \Omega; (\Delta, p_1; \Delta''; \cdot; p; \Omega; \cdot; \cdot); \cdot; AB; \Omega_N; \Delta_N \to \Xi'; \Delta'; \Gamma'}{\mathsf{mc}\ \Gamma; \Delta, p_1, \Delta''; \Xi_N; \Gamma_{N1}; \Delta_{N1}; \cdot; p, \Omega; \cdot; \cdot; AB; \Omega_N; \Delta_N \to \Xi'; \Delta'; \Gamma'}\ \mathsf{mc}\ p\ ok\ first$$

$$\frac{\begin{array}{l}\mathsf{mc}\ \Gamma; \Delta, \Delta''; \Xi_N; \Gamma_{N1}; \Delta_{N1}; \Xi, p_1; \Omega; (\Delta, p_1; \Delta''; \Xi; p; \Omega; q, \Lambda; \Upsilon), C_1, C; P; AB; \Omega_N; \Delta_N \to \Xi'; \Delta'; \Gamma' \\ C_1 = (\Delta_{old}; \Delta'_{old}; \Xi_{old}; q; \Omega_{old}; \Lambda; \Upsilon)\end{array}}{\mathsf{mc}\ \Gamma; \Delta, p_1, \Delta''; \Xi_N; \Gamma_{N1}; \Delta_{N1}; \Xi; p, \Omega; C_1, C; P; AB; \Omega_N; \Delta_N \to \Xi'; \Delta'; \Gamma'}\ \mathsf{mc}\ p\ ok\ other\ q$$

$$\frac{\begin{array}{l}\mathsf{mc}\ \Gamma; \Delta, \Delta''; \Xi_N; \Gamma_{N1}; \Delta_{N1}; \Xi, p_1; \Omega; (\Delta, p_1; \Delta''; \Xi; p; \Omega; \Lambda; q, \Upsilon), C_1, C; P; AB; \Omega_N; \Delta_N \to \Xi'; \Delta'; \Gamma' \\ C_1 = [\Gamma_{old}; \Delta_{old}; \Xi_{old}; q; \Omega_{old}; \Lambda; \Upsilon]\end{array}}{\mathsf{mc}\ \Gamma; \Delta, p_1, \Delta''; \Xi_N; \Gamma_{N1}; \Delta_{N1}; \Xi; p, \Omega; C_1, C; P; AB; \Omega_N; \Delta_N \to \Xi'; \Delta'; \Gamma'}\ \mathsf{mc}\ p\ ok\ other\ !qC$$

$$\frac{\begin{array}{l}\mathsf{mc}\ \Gamma; \Delta, \Delta''; \Xi_N; \Gamma_{N1}; \Delta_{N1}; p_1; \Omega; (\Delta, p_1; \Delta''; \cdot; p; \Omega; \cdot; q, \Upsilon); P_1, P; AB; \Omega_N; \Delta_N \to \Xi'; \Delta'; \Gamma' \\ P_1 = [\Gamma_{old}; \Delta_N; \cdot; q; \Omega_{old}; \cdot; \Upsilon] \\ \Delta_N = \Delta, p_1, \Delta''\end{array}}{\mathsf{mc}\ \Gamma; \Delta, p_1, \Delta''; \Xi_N; \Gamma_{N1}; \Delta_{N1}; \cdot; p, \Omega; \cdot; P_1, P; AB; \Omega_N; \Delta_N \to \Xi'; \Delta'; \Gamma'}\ \mathsf{mc}\ p\ ok\ other\ !qP$$

$$\frac{\mathsf{contc}\ \Gamma; \Delta_N; \Xi_N; \Gamma_{N1}; \Delta_{N1}; C; P; \mathsf{comp}\ A \multimap B; \Omega_N \rightarrow \Xi'; \Delta'; \Gamma'}{\mathsf{mc}\ \Gamma; \Delta; \Xi_N; \Gamma_{N1}; \Delta_{N1}; \Xi; p, \Omega; C; P; \mathsf{comp}\ A \multimap B; \Omega_N; \Delta_N \rightarrow \Xi'; \Delta'; \Gamma'}\ \mathsf{mc}\ p\ fail$$

$$\frac{\mathsf{mc}\ \Gamma, \Gamma', p; \Delta_N; \Xi_N; \Gamma_{N1}; \Delta_{N1}; \cdot; \Omega; \cdot; [\Gamma'; \Delta_N; \cdot; !p; \cdot; \Omega; \cdot; \cdot]; AB; \Omega_N; \Delta_N \rightarrow \Xi'; \Delta'; \Gamma'}{\mathsf{mc}\ \Gamma, \Gamma', p; \Delta_N; \Xi_N; \Gamma_{N1}; \Delta_{N1}; \cdot; !p, \Omega; \cdot; \cdot; AB; \Omega_N; \Delta_N \rightarrow \Xi'; \Delta'; \Gamma'}\ \mathsf{mc}\ !p\ first$$

$$\frac{\begin{array}{l}\mathsf{mc}\ \Gamma, \Gamma', p; \Delta_N; \Xi_N; \Gamma_{N1}; \Delta_{N1}; \cdot; \Omega; [\Gamma'; \Delta_N; \cdot; !p; \cdot; \Omega; \cdot; q, \Upsilon], P_1, P; AB; \Omega_N; \Delta_N \rightarrow \Xi'; \Delta'; \Gamma' \\ P_1 = [\Gamma_{old}; \Delta_N; \cdot; !q; \Omega_{old}; \cdot; \Upsilon]\end{array}}{\mathsf{mc}\ \Gamma, \Gamma', p; \Delta_N; \Xi_N; \Gamma_{N1}; \Delta_{N1}; \cdot; !p, \Omega; \cdot; P_1, P; AB; \Omega_N; \Delta_N \rightarrow \Xi'; \Delta'; \Gamma'}\ \mathsf{mc}\ !p\ other\ !qP$$

$$\frac{\begin{array}{l}\mathsf{mc}\ \Gamma, \Gamma', p; \Delta; \Xi_N; \Gamma_{N1}; \Delta_{N1}; \Xi; \Omega; [\Gamma'; \Delta; \Xi; !p; \cdot; \Omega; \Lambda; q, \Upsilon], C_1, C; P; AB; \Omega_N; \Delta_N \rightarrow \Xi'; \Delta'; \Gamma' \\ C_1 = [\Gamma_{old}; \Delta_{old}; \Xi_{old}; !q; \Omega_{old}; \Lambda; \Upsilon]\end{array}}{\mathsf{mc}\ \Gamma, \Gamma', p; \Delta; \Xi_N; \Gamma_{N1}; \Delta_{N1}; \Xi; !p, \Omega; C_1, C; P; AB; \Omega_N; \Delta_N \rightarrow \Xi'; \Delta'; \Gamma'}\ \mathsf{mc}\ !p\ other\ !qC$$

$$\frac{\begin{array}{l}\mathsf{mc}\ \Gamma, \Gamma', p; \Delta; \Xi_N; \Gamma_{N1}; \Delta_{N1}; \Xi; \Omega; [\Gamma'; \Delta; \Xi; !p; \cdot; \Omega; \Lambda, q; \Upsilon], C_1, C; P; AB; \Omega_N; \Delta_N \rightarrow \Xi'; \Delta'; \Gamma' \\ C_1 = (\Delta_{old}; \Delta'_{old}; \Xi_{old}; q; \Omega_{old}; \Lambda; \Upsilon)\end{array}}{\mathsf{mc}\ \Gamma, \Gamma', p; \Delta; \Xi_N; \Gamma_{N1}; \Delta_{N1}; \Xi; !p, \Omega; C_1, C; P; AB; \Omega_N; \Delta_N \rightarrow \Xi'; \Delta'; \Gamma'}\ \mathsf{mc}\ !p\ other\ qC$$

$$\frac{\mathsf{mc}\ \Delta; \Xi_N; \Delta_{N1}; \Xi; X, Y, \Omega; C; P; \mathsf{comp}\ A \multimap B; \Omega_N; \Delta_N \rightarrow \Xi'; \Delta'}{\mathsf{mc}\ \Delta; \Xi_N; \Delta_{N1}; \Xi; X \otimes Y, \Omega; C; P; \mathsf{comp}\ A \multimap B; \Omega_N; \Delta_N \rightarrow \Xi'; \Delta'}\ \mathsf{mc}\ \otimes$$

$$\frac{\mathsf{dall}\ \Gamma; \Xi_N; \Gamma_{N1}; \Delta_{N1}; \Xi; C; P; \mathsf{comp}\ A \multimap B; \Omega_N; \Delta_N \rightarrow \Xi'; \Delta'; \Gamma'}{\mathsf{mc}\ \Gamma; \Delta; \Xi_N; \Gamma_{N1}; \Delta_{N1}; \Xi; \cdot; C; P; \mathsf{comp}\ A \multimap B; \Omega_N; \Delta_N \rightarrow \Xi'; \Delta'; \Gamma'}\ \mathsf{mc}\ end$$

8.2.6 Match Comprehension Continuation

When backtracking to a previous frame, we need to be carefully when using the stacks C and P.

$$\frac{\mathsf{d}_1\ \Gamma; \Delta_N; \Xi_N; \Gamma_{N1}; \Delta_{N1}; \Omega \rightarrow \Xi'; \Delta'; \Gamma'}{\mathsf{contc}\ \Gamma; \Delta_N; \Xi_N; \Gamma_{N1}; \Delta_{N1}; \cdot; \cdot; \mathsf{comp}\ A \multimap B; \Omega \rightarrow \Xi'; \Delta'; \Gamma'}\ \mathsf{contc}\ end$$

$$\frac{\mathsf{mc}\ \Gamma; \Delta; \Xi_N; \Gamma_{N1}; \Delta_{N1}; \Xi; \Omega; (\Delta, p_1; \Delta''; \Xi; p; \Omega; \Lambda; \Upsilon), C; P; AB; \Omega_N; \Delta_N \rightarrow \Xi'; \Delta'; \Gamma'}{\mathsf{contc}\ \Gamma; \Delta_N; \Xi_N; \Gamma_{N1}; \Delta_{N1}; (\Delta; p_1, \Delta''; \Xi; p; \Omega; \Lambda; \Upsilon), C; P; AB; \Omega_N \rightarrow \Xi'; \Delta'; \Gamma'}\ \mathsf{contc}\ nextC\ p$$

$$\frac{\mathsf{mc}\ \Gamma; \Delta; \Xi_N; \Gamma_{N1}; \Delta_{N1}; \Xi; \Omega; [\Gamma'; \Delta; \Xi; !p; \Omega; \Lambda; \Upsilon], C; P; AB; \Omega_N; \Delta_N \rightarrow \Xi'; \Delta'; \Gamma'}{\mathsf{contc}\ \Gamma; \Delta_N; \Xi_N; \Gamma_{N1}; \Delta_{N1}; [p_1, \Gamma'; \Delta; \Xi; !p; \Omega; \Lambda; \Upsilon], C; P; AB; \Omega_N \rightarrow \Xi'; \Delta'; \Gamma'}\ \mathsf{contc}\ nextC\ !p$$

$$\frac{\mathsf{contc}\ \Gamma; \Delta_N; \Xi_N; \Gamma_{N1}; \Delta_{N1}; C; P; AB; \Omega_N \rightarrow \Xi'; \Delta'; \Gamma'}{\mathsf{contc}\ \Gamma; \Delta_N; \Xi_N; \Gamma_{N1}; \Delta_{N1}; (\Delta; \cdot; \Xi; p; \Omega; \Lambda; \Upsilon), C; P; AB; \Omega_N \rightarrow \Xi'; \Delta'; \Gamma'}\ \mathsf{contc}\ nextC\ empty\ p$$

$$\frac{\textsf{contc}\ \Gamma; \Delta_N; \Xi_N; \Gamma_{N1}; \Delta_{N1}; C; P; AB; \Omega_N \rightarrow \Xi'; \Delta'; \Gamma'}{\textsf{contc}\ \Gamma; \Delta_N; \Xi_N; \Gamma_{N1}; \Delta_{N1}; [\cdot; \Delta; \Xi; !p; \Omega; \Lambda; \Upsilon], C; P; AB; \Omega_N \rightarrow \Xi'; \Delta'; \Gamma'}\ \textsf{contc}\ \mathit{nextC\ empty}\ !p$$

$$\frac{\textsf{mc}\ \Gamma; \Delta_N; \Xi_N; \Gamma_{N1}; \Delta_{N1}; \cdot; \Omega; \cdot; [\Gamma'; \Delta_N; \cdot; !p; \Omega; \cdot; \Upsilon], P; AB; \Omega_N; \Delta_N \rightarrow \Xi'; \Delta'; \Gamma'}{\textsf{contc}\ \Gamma; \Delta_N; \Xi_N; \Gamma_{N1}; \Delta_{N1}; \cdot; [p_1, \Gamma'; \Delta_N; \cdot; !p; \Omega; \cdot; \Upsilon], P; AB; \Omega_N \rightarrow \Xi'; \Delta'; \Gamma'}\ \textsf{contc}\ \mathit{nextP}\ !p$$

$$\frac{\textsf{contc}\ \Gamma; \Delta_N; \Xi_N; \Gamma_{N1}; \Delta_{N1}; \cdot; P; AB; \Omega_N \rightarrow \Xi'; \Delta'; \Gamma'}{\textsf{contc}\ \Gamma; \Delta_N; \Xi_N; \Gamma_{N1}; \Delta_{N1}; \cdot; [\cdot; \Delta_N; \cdot; !p; \Omega; \cdot; \Upsilon], P; AB; \Omega_N \rightarrow \Xi'; \Delta'; \Gamma'}\ \textsf{contc}\ \mathit{nextP\ empty}\ !p$$

8.2.7 Stack Transformation

Like the previous system, we need to transform continuation stack to be able to used again. First, we remove everything except the first frame in the C stack. Next, we transform the Δ argument in the first frame of C and in every frame of P to remove recently consumed facts.

The strans $\Xi; P; P'$ judgments transforms the P stack and has the following meaning:

- Ξ : Consumed linear resources during the last application of the comprehension.
- P : The P stack.
- P' : The transformed P stack.

$$\frac{\textsf{strans}\ \Xi; P; P'}{\textsf{strans}\ \Xi; [\Gamma'; \Delta_N; \cdot; !p; \Omega; \cdot; \Upsilon], P; [\Gamma'; \Delta_N - \Xi; \cdot; !p, \Omega; \cdot; \Upsilon], P'}\ \textsf{strans}$$

$$\frac{\textsf{strans}\ \Xi; \cdot; \cdot}{\textsf{strans}\ \Xi; \cdot; \cdot}\ \textsf{strans}\ \mathit{end}$$

$$\frac{\textsf{strans}\ \Xi; P; P' \quad \textsf{dc}\ \Gamma; \Xi_N, \Xi; \Gamma_{N1}; \Delta_{N1}; (\Delta_x - \Xi; \Delta'' - \Xi; \cdot; p; \Omega; \cdot; \Upsilon); P'; \textsf{comp}\ A \multimap B; \Omega_N; (\Delta_N - \Xi) \rightarrow \Xi'; \Delta'; \Gamma'}{\textsf{dall}\ \Gamma; \Xi_N; \Gamma_{N1}; \Delta_{N1}; \Xi; (\Delta_x; \Delta''; \cdot; p; \Omega; \cdot; \Upsilon); P; \textsf{comp}\ A \multimap B; \Omega_N; \Delta_N \rightarrow \Xi'; \Delta'; \Gamma'}\ \textsf{dall}\ \mathit{end\ linear}$$

$$\frac{\textsf{strans}\ \Xi; P; P' \quad \textsf{dc}\ \Gamma; \Xi_N, \Xi; \Gamma_{N1}; \Delta_{N1}; \cdot; P'; \textsf{comp}\ A \multimap B; \Omega_N; (\Delta_N - \Xi) \rightarrow \Xi'; \Delta'; \Gamma'}{\textsf{dall}\ \Gamma; \Xi_N; \Gamma_{N1}; \Delta_{N1}; \Xi; \cdot; P; \textsf{comp}\ A \multimap B; \Omega_N; \Delta_N \rightarrow \Xi'; \Delta'; \Gamma'}\ \textsf{dall}\ \mathit{end\ empty}$$

$$\frac{\textsf{dall}\ \Gamma; \Xi_N; \Gamma_{N1}; \Delta_{N1}; \Xi; X, C; P; \textsf{comp}\ A \multimap B; \Omega_N; \Delta_N \rightarrow \Xi'; \Delta'; \Gamma'}{\textsf{dall}\ \Gamma; \Xi_N; \Gamma_{N1}; \Delta_{N1}; \Xi; \_, X, C; P; \textsf{comp}\ A \multimap B; \Omega_N; \Delta_N \rightarrow \Xi'; \Delta'; \Gamma'}\ \textsf{dall}\ \mathit{more}$$

8.2.8 Comprehension Derivation

The dc is identical to the previous system, however it has been extended with Γ, Γ_1, C, P and Γ'.

$$\frac{\mathsf{dc}\ \Gamma; \Xi; \Gamma_1; \Delta_1, p; \Omega; C; P; \mathsf{comp}\ A \multimap B; \Omega_N; \Delta_N \rightarrow \Xi'; \Delta'; \Gamma'}{\mathsf{dc}\ \Gamma; \Xi; \Gamma_1; \Delta_1; p, \Omega; C; P; \mathsf{comp}\ A \multimap B; \Omega_N; \Delta_N \rightarrow \Xi'; \Delta'; \Gamma'}\ \mathsf{dc}\ p$$

$$\frac{\mathsf{dc}\ \Gamma; \Xi; \Gamma_1, p; \Delta_1; \Omega; C; P; \mathsf{comp}\ A \multimap B; \Omega_N; \Delta_N \rightarrow \Xi'; \Delta'; \Gamma'}{\mathsf{dc}\ \Gamma; \Xi; \Gamma_1; \Delta_1; !p, \Omega; C; P; \mathsf{comp}\ A \multimap B; \Omega_N; \Delta_N \rightarrow \Xi'; \Delta'; \Gamma'}\ \mathsf{dc}\ !p$$

$$\frac{\mathsf{dc}\ \Gamma; \Xi; \Gamma_1; \Delta_1; A, B, \Omega; C; P; \mathsf{comp}\ A \multimap B; \Omega_N; \Delta_N \rightarrow \Xi'; \Delta'; \Gamma'}{\mathsf{dc}\ \Gamma; \Xi; \Gamma_1; \Delta_1; A \otimes B, \Omega; C; P; \mathsf{comp}\ A \multimap B; \Omega_N; \Delta_N \rightarrow \Xi'; \Delta'; \Gamma'}\ \mathsf{dc} \otimes$$

$$\frac{\mathsf{contc}\ \Gamma; \Delta_N; \Xi; \Gamma_1; \Delta_1; C; P; \mathsf{comp}\ A \multimap B; \Omega_N \rightarrow \Xi'; \Delta'; \Gamma'}{\mathsf{dc}\ \Gamma; \Xi; \Gamma_1; \Delta_1; \cdot; C; P; \mathsf{comp}\ A \multimap B; \Omega_N; \Delta_N \rightarrow \Xi'; \Delta'; \Gamma'}\ \mathsf{dc}\ \mathit{end}$$

8.2.9 Application

Finally, we add Γ and Γ' to the main inference rules to complete the system.

$$\frac{\mathsf{m}_1\ \Gamma; \Delta; \cdot; \cdot; A; B; \cdot; R \rightarrow \Xi'; \Delta'; \Gamma'}{\mathsf{a}_1\ \Gamma; \Delta; A \multimap B; R \rightarrow \Xi'; \Delta'; \Gamma'}\ \mathsf{a}_1\ \mathit{start\ matching}$$

$$\frac{\mathsf{a}_1\ \Gamma; \Delta; R; (\Phi, \Delta) \rightarrow \Xi'; \Delta'; \Gamma'}{\mathsf{do}_1\ \Gamma; \Delta; R, \Phi \rightarrow \Xi'; \Delta'; \Gamma'}\ \mathsf{do}_1\ \mathit{rule}$$

8.3 Definitions

8.3.1 Persistent Frame

Definition 16. *A persistent frame f has the form* $[\Gamma'; \Delta; \Xi_1, ..., \Xi_m; !p; \Omega_1, ..., \Omega_n; \Lambda_1, ..., \Lambda_m]$.

8.3.2 Linear Frame

Definition 17. *A linear frame f has the form* $(\Delta; \Delta'; \Xi_1, ..., \Xi_m; p; \Omega_1, ..., \Omega_n; \Lambda_1, ..., \Lambda_m, \Upsilon_1, ..., \Upsilon_k)$.

8.3.3 Well Formed Frame

Definition 18. *Given a triplet $A; \Gamma; \Delta_{inv}$ where A is a term, Γ is a multi-set of persistent resources and Δ_{inv} a multi-set of linear resources we say that a frame f is well-formed iff:*

1. *Linear frame* $f = (\Delta, p_1; \Delta'; \Xi_1, ..., \Xi_m; p; \Omega_1, ..., \Omega_n; \Lambda_1, ..., \Lambda_m, \Upsilon_1, ..., \Upsilon_k)$
   - *(a)* $\Lambda_1, ..., \Lambda_m$ *are linear atomic terms* p_i*;*
   - *(b)* $\Upsilon_1, ..., \Upsilon_k$ *are persistent atomic terms* $!p_i$.
   - *(c)* $p, \Omega_1, ..., \Omega_n, \Lambda_1, ..., \Lambda_m, \Upsilon_1, ..., \Upsilon_k \equiv A$*;*
   - *(d)* $\mathsf{match}\ \Xi \rightarrow \Lambda_1 \otimes ... \otimes \Lambda_m$*;*
   - *(e)* $\Delta, \Delta', \Xi, p_1 = \Delta_{inv}$*;*
   - *(f)* $\mathsf{match}\ \Gamma; \cdot \rightarrow \Upsilon_1 \otimes ... \otimes \Upsilon_k$.
2. *Persistent frame* $f = [\Gamma'; \Delta; \Xi_1, ..., \Xi_m; !p; \Omega_1, ..., \Omega_n; \Lambda_1, ..., \Lambda_m]$

   (a) $\Lambda_1, ..., \Lambda_m$ *are linear atomic terms* p_i*;*
   (b) $\Upsilon_1, ..., \Upsilon_k$ *are persistent atomic terms* $!p_i$*.*
   (c) $!p, \Omega_1, ..., \Omega_n, \Lambda_1, ..., \Lambda_m, \Upsilon_1, ..., \Upsilon_k \equiv A$*;*
   (d) $\mathsf{match}\ \Xi \rightarrow \Lambda_1 \otimes ... \otimes \Lambda_m$*;*
   (e) $\Delta, \Xi = \Delta_{inv}$*;*
   (f) $\mathsf{match}\ \Gamma; \cdot \rightarrow !p \otimes \Upsilon_1 \otimes ... \otimes \Upsilon_k$*;*
   (g) $\Gamma' \subset \Gamma$*.*

8.3.4 Well Formed Stack

Definition 19. *A continuation stack C is well formed iff every frame is well formed.*

8.3.5 Well Related Match

Definition 20. $\mathsf{m}_1\ \Gamma; \Delta; \Xi; \Omega; H; C; R \rightarrow \Xi'; \Delta'$ *is related to a term A, a context Δ_{inv} and a context Γ iff:*

1. *C is well formed in relation to $A; \Gamma; \Delta_{inv}$;*
2. $\Delta, \Xi = \Delta_{inv}$
3. *Stack relatedness:*
   (a) $C = \cdot$
   $\Omega \equiv A$
   (b) $C = (\Delta_a; \Delta_b; \Xi''; p; \Omega'; \Lambda_1, ..., \Lambda_m; \Upsilon_1, ..., \Upsilon_k), C'$
      i. $\Omega' \equiv \Omega$
      ii. $p_1 \in \Xi$ *and* $\mathsf{match}\ p_1 \rightarrow p$
      iii. $\Xi = \Xi'', p_1$
   (c) $C = [\Gamma'; \Delta_a; \Xi''; !p; \Omega'; \Lambda_1, ..., \Lambda_m; \Upsilon_1, ..., \Upsilon_k], C'$
      i. $\Omega \equiv \Omega'$
      ii. $\Xi = \Xi''$

Definition 21. $\mathsf{mc}\ \Gamma; \Delta; \Xi_N; \Gamma_{N1}; \Delta_{N1}; \Xi; \Omega; C; P; \mathsf{comp}\ A \multimap B; \Omega_N; \Delta_N \rightarrow \Xi'; \Delta'; \Gamma'$ *is related to a term A, a context Δ_{inv} and a context Γ iff:*

1. *P is only composed of persistent frames.*
2. *C is composed of either linear or persistent frames, but the first frame is linear.*
3. *C and P are well formed in relation to $A; \Gamma; \Delta_{inv}$;*
4. $\Delta, \Xi = \Delta_{inv}$
5. *Stack relatedness:*
   (a) $C = \cdot$ *and* $P = \cdot$
   $\Omega \equiv A$
   (b) $C = \cdot$ *and* $P = [\Gamma'; \Delta_a; \Xi''; !p; \Omega'; \Lambda_1, ..., \Lambda_m; \Upsilon_1, ..., \Upsilon_k], P'$
      i. $\Omega \equiv \Omega'$
      ii. $\Xi = \Xi''$

(c) $C = (\Delta_a; \Delta_b; \Xi''; p; \Omega'; \Lambda_1, ..., \Lambda_m; \Upsilon_1, ..., \Upsilon_k), C'$

i. $\Omega' \equiv \Omega$

ii. $p_1 \in \Xi$ *and* $\mathsf{match}\ p_1 \to p$

iii. $\Xi = \Xi'', p_1$

(d) $C = [\Gamma'; \Delta_a; \Xi''; !p; \Omega'; \Lambda_1, ..., \Lambda_m; \Upsilon_1, ..., \Upsilon_k], C'$

i. $\Omega \equiv \Omega'$

ii. $\Xi = \Xi''$

8.4 Theorems

8.4.1 Body Match Soundness Theorem

Theorem 22. *If a* $\mathsf{m}_1\ \Gamma; \Delta_1, \Delta_2; \Xi; \Omega; H; C; R \to \Xi'; \Delta'; \Gamma'$ *is related to term* A *and a context* Δ_1, Δ_2, Ξ *and a context* Γ *then either:*

1. $\mathsf{cont}\ \cdot; H; R; \Gamma; \Xi'; \Delta'; \Gamma'$

2. $\mathsf{match}\ \Gamma; \Delta_x \to A$ *(where* Δ_x *is a subset of* Δ_1, Δ_2, Ξ*) and one of the two sub-cases are true:*

   (a) $\mathsf{m}_1\ \Gamma; \Delta_1; \Xi, \Delta_2; \cdot; H; C'', C; R \to \Xi'; \Delta'; \Gamma'$ *(related) and* $\Delta_x = \Xi, \Delta_2$

   (b) $\exists f = (\Delta_a; \Delta_{b_1}, p_2, \Delta_{b_2}; p; \Omega_1, ..., \Omega_k; \Xi_1, ..., \Xi_m; \Lambda_1, ..., \Lambda_m; \Upsilon_1, ..., \Upsilon_n) \in C$ *where* $C = C', f, C''$ *and* f *turns into* $f' = (\Delta_a, \Delta_{b_1}, p_2; \Delta_{b_2}; p; \Omega_1, ..., \Omega_k; \Xi_1, ..., \Xi_m; \Lambda_1, ..., \Lambda_m; \Upsilon_1, ..., \Upsilon_n)$ *such that:*

      i. $\mathsf{m}_1\ \Gamma; \Delta_c; \Xi_1, ..., \Xi_m, p_2, \Xi_c; \cdot; H; C''', f', C''; R \to \Xi'; \Delta'; \Gamma'$ *(related) where* $\Delta_c = (\Delta_1, \Delta_2, \Xi) - (\Xi_1, ..., \Xi_m, p_2, \Xi_c)$

   (c) $\exists f = [\Gamma_1, p_2, \Gamma_2; \Delta_{c_1}, \Delta_{c_2}; \Xi_c; !p; \Omega_1, ..., \Omega_k; \Lambda_1, ..., \Lambda_m; \Upsilon_1, ..., \Upsilon_n] \in C$ *where* $C = C', f, C''$ *and* f *turns into* $f' = [\Gamma_2; \Delta_{c_1}, \Delta_{c_2}; \Xi_1, ..., \Xi_m; !p; \Omega_1, ..., \Omega_k; \Lambda_1, ..., \Lambda_m; \Upsilon_1, ..., \Upsilon_n]$ *such that:*

      i. $\mathsf{m}_1\ \Gamma; \Delta_{c_1}; \Xi_1, ..., \Xi_m, \Delta_{c_2}; \cdot; H; C'', f', C''; R \to \Xi'; \Delta'; \Gamma'$ *(related) where* $\Delta_{c_1}, \Delta_{c_2} = \Delta_1, \Delta_2, \Xi - (\Xi_1, ..., \Xi_m)$

If $\mathsf{cont}\ C; H; R; \Gamma; \Xi'; \Delta'; \Gamma'$ *and* C *is well formed in relation to* A *and* Δ_1, Δ_2, Ξ *then either:*

1. $\mathsf{cont}\ \cdot; H; R; \Gamma; \Xi'; \Delta'; \Gamma'$

2. $\mathsf{match}\ \Delta_x \to A$ *(where* $\Delta_x \subseteq \Delta_1, \Delta_2, \Xi$*) where one sub-case is true:*

   (a) $\exists f = (\Delta_a; \Delta_{b_1}, p_2, \Delta_{b_2}; \Xi_1, ..., \Xi_m; p; \Omega_1, ..., \Omega_k; \Lambda_1, ..., \Lambda_m; \Upsilon_1, ..., \Upsilon_n) \in C$ *where* $C = C', f, C''$ *and* f *turns into* $f' = (\Delta_a, \Delta_{b_1}, p_2; \Delta_{b_2}; p; \Omega_1, ..., \Omega_k; \Xi_1, ..., \Xi_m; \Lambda_1, ..., \Lambda_m; \Upsilon_1, ..., \Upsilon_n)$ *such that:*

      i. $\mathsf{m}_1\ \Gamma; \Delta_c; \Xi_1, ..., \Xi_m, p_2, \Xi_c; \cdot; H; C''', f', C''; R \to \Xi'; \Delta'; \Gamma'$ *(related) where* $\Delta_c = (\Delta_1, \Delta_2, \Xi) - (\Xi_1, ..., \Xi_m, p_2, \Xi_c)$

   (b) $\exists f = [\Gamma_1, p_2, \Gamma_2; \Delta_{c_1}, \Delta_{c_2}; \Xi_c; !p; \Omega_1, ..., \Omega_k; \Lambda_1, ..., \Lambda_m; \Upsilon_1, ..., \Upsilon_n] \in C$ *where* $C = C', f, C''$ *and* f *turns into* $f' = [\Gamma_2; \Delta_{c_1}, \Delta_{c_2}; \Xi_1, ..., \Xi_m; !p; \Omega_1, ..., \Omega_k; \Lambda_1, ..., \Lambda_m; \Upsilon_1, ..., \Upsilon_n]$ *such that:*

      i. $\mathsf{m}_1\ \Gamma; \Delta_{c_1}; \Xi_1, ..., \Xi_m, \Delta_{c_2}; \cdot; H; C'', f', C''; R \to \Xi'; \Delta'; \Gamma'$ *(related) where* $\Delta_{c_1}, \Delta_{c_2} = \Delta_1, \Delta_2, \Xi - (\Xi_1, ..., \Xi_m)$

Proof. Proof by mutual induction. In m_1 on the size of Ω and on cont , first on the size of Δ'' and Γ'' and then on the size of C. Use the stack constraints and the Match Equivalence Theorem to prove $\mathsf{match}\ \Delta_x \to A$.

1. $\mathsf{m}_1\ p\ ok\ first$

   By induction on Ω.

2. m_1 *p ok other q*

   By induction on Ω.

3. m_1 *p ok other* $!q$

   By induction on Ω.

4. m_1 $p\ fail$

   Induction on the inverted cont judgment.

5. m_1 $!p\ ok\ first$

   Induction on Ω.

6. m_1 $!p$ *ok other* $!q$

   Induction on Ω.

7. m_1 $!p\ fail$

   Induction on the inverted cont judgment.

8. $\mathsf{m}_1 \otimes$

   Induction on Ω.

9. m_1 *end*

   Use assumption.

10. cont *next rule*

    Use assumption.

11. cont *p next*

    Induction on Δ''.

12. cont *p no more*

    Induction on C.

13. cont $!p$ *next*

    Induction on Γ'.

14. cont $!p$ *no more*

    Induction on C.

□

8.4.2 Body Match Soundness Lemma

Lemma 5. *Given a match* $\mathsf{m}_1\ \Gamma; \Delta_1, \Delta_2; \cdot; A; B; \cdot; R \to \Xi'; \Delta'; \Gamma'$ *that is related to* A*,* Δ_1, Δ_2 *and* Γ*, we get either:*

1. $\mathsf{cont}\ \cdot; B; R; \Gamma; \Xi'; \Delta'; \Gamma'$*;*
2. $\mathsf{match}\ \Delta_2 \to A$*:*

   (a) $\mathsf{m}_1\ \Gamma; \Delta_1; \Delta_2; \cdot; B; C'; R \to \Xi'; \Delta'; \Gamma'$ *(related)*

Proof. Use the Body Match Soundness Theorem. □

8.4.3 Comprehension Match Soundness Theorem

Theorem 23. • *If a match* $\mathsf{mc}\ \Gamma; \Delta_1, \Delta_2; \Xi_N; \Gamma_{N1}; \Delta_{N1}; \Xi; \Omega; C; P; \mathsf{comp}\ A \multimap B; \Omega_N; \Delta_N \to \Xi'; \Delta'; \Gamma'$ *is related to term A, context* $\Delta_N = \Delta_1, \Delta_2, \Xi$ *and context* Γ *then either:*

1. $\Gamma; \Delta_N; \Xi_N; \Gamma_{N1}; \Delta_{N1}; \Omega_N \to \Xi'; \Delta'; \Gamma'$*;*
2. $\mathsf{match}\ \Delta_x \to A$ *(where* $\Delta_x \subseteq \Delta_N$*) and one of the following sub-cases is true:*
   (a) $\mathsf{mc}\ \Gamma; \Delta_1; \Xi_N; \Gamma_{N1}; \Delta_{N1}; \Xi, \Delta_2; \cdot; C', C; P; \mathsf{comp}\ A \multimap B; \Omega_N; \Delta_N \to \Xi'; \Delta'; \Gamma'$ *(related) and* $\Delta_x = \Delta_2$*, if* $C \neq \cdot$
   (b) $\mathsf{mc}\ \Gamma; \Delta_1; \Xi_N; \Gamma_{N1}; \Delta_{N1}; \Xi, \Delta_2; \cdot; C'; P', P; \mathsf{comp}\ A \multimap B; \Omega_N; \Delta_N \to \Xi'; \Delta'; \Gamma'$ *(related) and* $\Delta_x = \Delta_2$*, if* $C = \cdot$*.*
   (c) $\exists f = (\Delta_a; \Delta_{b_1}, p_2, \Delta_{b_2}; p; \Xi_1, ..., \Xi_n; \Omega_1, ..., \Omega_k; \Lambda_1, ..., \Lambda_n; \Upsilon_1, ..., \Upsilon_m) \in C$ *where* $C = C', f, C''$ *that turns into* $f' = (\Delta_a, \Delta_{b_1}, p_2; \Delta_{b_2}; p; \Xi_1, ..., \Xi_n; \Omega_1, ..., \Omega_k; \Lambda_1, ..., \Lambda_n; \Upsilon_1, ..., \Upsilon_m)$ *such that:*
      i. $\mathsf{mc}\ \Gamma; \Delta_c; \Xi_N; \Gamma_{N1}; \Delta_{N1}; \Xi_1, ..., \Xi_n, \Xi_c; \cdot; C''', f', C''; P; \mathsf{comp}\ A \multimap B; \Omega_N; \Delta_N \to \Xi'; \Delta'; \Gamma'$ *(related)*
   (d) $\exists f = [\Gamma_1, p_2, \Gamma_2; \Delta_{c_1}, \Delta_{c_2}; \Xi_c; !p; \Omega_1, ..., \Omega_k; \Lambda_1, ..., \Lambda_n; \Upsilon_1, ..., \Upsilon_m] \in C$ *where* $C = C', f, C''$ *that turns into* $f' = [\Gamma_2; \Delta_{c_1}, \Delta_{c_2}; \Xi_c; !p; \Omega_1, ..., \Omega_k; \Lambda_1, ..., \Lambda_n; \Upsilon_1, ..., \Upsilon_m]$ *such that:*
      i. $\mathsf{mc}\ \Gamma; \Delta_{c_1}; \Xi_N; \Gamma_{N1}; \Delta_{N1}; \Delta_{c_2}, \Xi_c; \cdot; C''', f', C''; P; \mathsf{comp}\ A \multimap B; \Omega_N; \Delta_N \to \Xi'; \Delta'; \Gamma'$ *(related)*
   (e) $\exists f = [\Gamma_1, p_2, \Gamma_2; \Delta_{c_1}, \Delta_{c_2}; \Xi_c; !p; \Omega_1, ..., \Omega_k; \Lambda_1, ..., \Lambda_n; \Upsilon_1, ..., \Upsilon_m] \in P$ *where* $P = P', f, P''$ *that turns into* $f' = [\Gamma_2; \Delta_{c_1}, \Delta_{c_2}; \Xi_c; !p; \Omega_1, ..., \Omega_k; \Lambda_1, ..., \Lambda_n; \Upsilon_1, ..., \Upsilon_m]$ *such that:*
      i. $\mathsf{mc}\ \Gamma; \Delta_{c_1}; \Xi_N; \Gamma_{N1}; \Delta_{N1}; \Delta_{c_2}, \Xi_c; \cdot; C'; P''', f', P''; \mathsf{comp}\ A \multimap B; \Omega_N; \Delta_N \to \Xi'; \Delta'; \Gamma'$ *(related)*

• *If* $\mathsf{contc}\ \Gamma; \Delta_N; \Xi_N; \Gamma_{N1}; \Delta_{N1}; C; P; \mathsf{comp}\ A \multimap B \to \Xi'; \Delta'; \Gamma'$ *and* C *and* P *is well formed in relation to* $A; \Gamma; \Delta_N$ *then either:*

1. $\exists f = (\Delta_a; \Delta_{b_1}, p_2, \Delta_{b_2}; p; \Xi_1, ..., \Xi_n; \Omega_1, ..., \Omega_k; \Lambda_1, ..., \Lambda_n; \Upsilon_1, ..., \Upsilon_m) \in C$ *where* $C = C', f, C''$ *that turns into* $f' = (\Delta_a, \Delta_{b_1}, p_2; \Delta_{b_2}; p; \Xi_1, ..., \Xi_n; \Omega_1, ..., \Omega_k; \Lambda_1, ..., \Lambda_n; \Upsilon_1, ..., \Upsilon_m)$ *such that:*
   (a) $\mathsf{mc}\ \Gamma; \Delta_c; \Xi_N; \Gamma_{N1}; \Delta_{N1}; \Xi_1, ..., \Xi_n, \Xi_c; \cdot; C''', f', C''; P; \mathsf{comp}\ A \multimap B; \Omega_N; \Delta_N \to \Xi'; \Delta'; \Gamma'$ *(related)*
2. $\exists f = [\Gamma_1, p_2, \Gamma_2; \Delta_{c_1}, \Delta_{c_2}; \Xi_c; !p; \Omega_1, ..., \Omega_k; \Lambda_1, ..., \Lambda_n; \Upsilon_1, ..., \Upsilon_m] \in C$ *where* $C = C', f, C''$ *that turns into* $f' = [\Gamma_2; \Delta_{c_1}, \Delta_{c_2}; \Xi_c; !p; \Omega_1, ..., \Omega_k; \Lambda_1, ..., \Lambda_n; \Upsilon_1, ..., \Upsilon_m]$ *such that:*
   (a) $\mathsf{mc}\ \Gamma; \Delta_{c_1}; \Xi_N; \Gamma_{N1}; \Delta_{N1}; \Delta_{c_2}, \Xi_c; \cdot; C''', f', C''; P; \mathsf{comp}\ A \multimap B; \Omega_N; \Delta_N \to \Xi'; \Delta'; \Gamma'$ *(related)*
3. $\exists f = [\Gamma_1, p_2, \Gamma_2; \Delta_{c_1}, \Delta_{c_2}; \Xi_c; !p; \Omega_1, ..., \Omega_k; \Lambda_1, ..., \Lambda_n; \Upsilon_1, ..., \Upsilon_m] \in P$ *where* $P = P', f, P''$ *that turns into* $f' = [\Gamma_2; \Delta_{c_1}, \Delta_{c_2}; \Xi_c; !p; \Omega_1, ..., \Omega_k; \Lambda_1, ..., \Lambda_n; \Upsilon_1, ..., \Upsilon_m]$ *such that:*
   (a) $\mathsf{mc}\ \Gamma; \Delta_{c_1}; \Xi_N; \Gamma_{N1}; \Delta_{N1}; \Delta_{c_2}, \Xi_c; \cdot; C'; P''', f', P''; \mathsf{comp}\ A \multimap B; \Omega_N; \Delta_N \to \Xi'; \Delta'; \Gamma'$ *(related)*

Proof. Mutual induction on m_1 and contc . m_1 on the size of Ω and contc first on the size of Δ' or Γ' and then on the size of C and P merged. We use the same approach as in Section 7.4.2. □

8.4.4 Comprehension Match Soundness Lemma

Lemma 6. *If* $\mathsf{mc}\ \Gamma; \Delta, \Xi; \Xi_N; \Gamma_{N1}; \Delta_{N1}; \cdot; A; \cdot; \cdot; \mathsf{comp}\ A \multimap B; \Omega_N; \Delta_N \to \Xi'; \Delta'; \Gamma'$ *and the match is related to both A,* Δ_N *and* Γ *then either:*

1. $\Gamma; \Delta_N; \Xi_N; \Gamma_{N1}; \Delta_{N1}; \Omega_N \to \Xi'; \Delta'; \Gamma'$*;*
2. $\mathsf{mc}\ \Gamma; \Delta; \Xi_N; \Gamma_{N1}; \Delta_{N1}; \Xi; \cdot; C'; P'; \mathsf{comp}\ A \multimap B; \Omega_N; \Delta_N \to \Xi'; \Delta'; \Gamma'$ *and*

(a) $\mathsf{match}\ \Gamma; \Xi \to A$

(b) C' *and* P' *are well formed in relation to* $A; \Gamma; \Delta, \Xi$.

Proof. Direct application of the previous theorem. □

8.4.5 Strans Theorem

Theorem 24. *If* $\mathsf{strans}\ \Xi; P; P'$ *then* P' *will be the transformation of stack* P *where* $\forall f = [\Gamma'; \Delta_N; \cdot; !p; \Omega; \cdot; \Upsilon] \in P$ *will turn into* $f' = [\Gamma'; \Delta_N - \Xi; \cdot; !p; \Omega; \cdot; \Upsilon]$.

Proof. Induction on the size of P. □

8.4.6 Dall Transformation Theorem

Theorem 25. *If* $\mathsf{dall}\ \Gamma; \Xi_N; \Gamma_{N1}; \Delta_{N1}; \Xi; C; P; \mathsf{comp}\ A \multimap B; \Omega_N; \Delta_N \to \Xi'; \Delta'; \Gamma'$ *then* $\mathsf{dc}\ \Gamma; \Xi_N, \Xi; \Gamma_{N1}; \Delta_{N1}; B; C'; P'; \mathsf{comp}$
$B; \Omega_N; (\Delta_N - \Xi) \to \Xi'; \Delta'; \Gamma'$, *where:*

1. *If* $C = \cdot$ *then* $C' = \cdot$;
2. *If* $C = C_1, (\Delta_a; \Delta_b; \cdot; p; \Omega; \cdot; \Upsilon)$ *then* $C' = (\Delta_a - \Xi; \Delta_b - \Xi; \cdot; p; \Omega; \cdot; \Upsilon)$;
3. P' *is the transformation of stack* P, *where* $\forall f = [\Gamma'; \Delta_N; \cdot; !p; \Omega; \cdot; \Upsilon] \in P$ *will turn into* $f' = [\Gamma'; \Delta_N - \Xi; \cdot; !p; \Omega; \cdot; \Upsilon]$.

Proof. Use induction on the size of the stack C to get the result of C' then apply the strans theorem to get P'. □

8.4.7 Successful Comprehension Match Gives Derivation

We can apply the previous theorem to know that after a successful matching we will start the derivation process:

Lemma 7. *If* $\mathsf{mc}\ \Gamma; \Delta; \Xi_N; \Gamma_{N1}; \Delta_{N1}; \Xi; \cdot; B; C; P; \mathsf{comp}\ A \multimap B; \Omega_N; \Delta_N \to \Xi'; \Delta'; \Gamma'$ *then* $\mathsf{dc}\ \Gamma; \Xi_N, \Xi; \Gamma_{N1}; \Delta_{N1}; B; C'; P'; \mathsf{c}$
$B; \Omega_N; (\Delta_N - \Xi) \to \Xi'; \Delta'; \Gamma'$ *where:*

1. *If* $C = \cdot$ *then* $C' = \cdot$;
2. *If* $C = C_1, (\Delta_a; \Delta_b; \cdot; p; \Omega; \cdot; \Upsilon)$ *then* $C' = (\Delta_a - \Xi; \Delta_b - \Xi; \cdot; p; \Omega; \cdot; \Upsilon)$ *then* $C' = (\Delta_a - \Xi; \Delta_b - \Xi; \cdot; p; \Omega; \cdot; \Upsilon)$;
3. P' *is the transformation of stack* P, *where* $\forall f = [\Gamma'; \Delta_N; \cdot; !p; \Omega; \cdot; \Upsilon] \in P$ *will turn into* $f' = [\Gamma'; \Delta_N - \Xi; \cdot; !p; \Omega; \cdot; \Upsilon]$.

Proof. Invert the assumption and then apply dall transformation theorem. □

8.4.8 Comprehension Derivation Theorem

We prove that if we try to derive the head of the comprehension, we will finish this process and that giving a derivation at the high level with the same results we can restore the terms we have derived. We will use this result to prove the soundness of the comprehension mechanism.

Theorem 26. *If* $\mathsf{dc}\ \Gamma; \Xi_N; \Gamma_{N1}; \Delta_{N1}; \Omega_1, ..., \Omega_n; C; P; \mathsf{comp}\ A \multimap B; \Omega_N; \Delta_N \to \Xi'; \Delta'; \Gamma'$ *then:*

1. $\mathsf{dc}\ \Gamma; \Xi_N; \Delta_{N1}, \Delta_1, ..., \Delta_n; \Gamma_{N1}, \Gamma_1, ..., \Gamma_n; \Omega \to \Xi'; \Delta'; \Gamma'$;
2. *If* $\mathsf{derive}\ \Gamma; \Delta; \Xi_N; \Gamma_{N1}, \Gamma_1, ..., \Gamma_n; \Delta_{N1}, \Delta_1, ..., \Delta_n; \Omega \to \Xi'; \Delta'; \Gamma'$ *then* $\mathsf{derive}\ \Gamma; \Delta; \Xi_N; \Gamma_{N1}; \Delta_{N1}; \Omega_1, ..., \Omega_n, \Omega \to \Xi'; \Delta'; \Gamma'$

Proof. Induction on $\Omega_1, ..., \Omega_n$. □

8.4.9 Comprehension Derivation Lemma

Now using the previous theorem we prove that we can do the same for the head of the comprehension.

Theorem 27. *If* $\mathsf{dc}\ \Gamma; \Xi_N; \Gamma_{N1}; \Delta_{N1}; B; C; P; \mathsf{comp}\ A \multimap B; \Omega_N; \Delta_N \to \Xi'; \Delta'; \Gamma'$ *then:*

1. $\mathsf{dc}\ \Gamma; \Xi_N; \Delta_{N1}, \Delta_B; \cdot; C; P; \mathsf{comp}\ A \multimap B; \Omega_N; \Delta_N \to \Xi'; \Delta'; \Gamma'$;
2. *If* $\mathsf{derive}\ \Gamma; \Delta; \Xi_N; \Gamma_{N1}, \Gamma_B; \Delta_{N1}, \Delta_B; \Omega \to \Xi'; \Delta'; \Gamma'$ *then* $\mathsf{derive}\ \Gamma; \Delta; \Xi_N; \Gamma_{N1}; \Delta_{N1}; B, \Omega \to \Xi'; \Delta'; \Gamma'$.

Proof. Use the theorem above. □

8.4.10 Comprehension Theorem

If we have a matching process with at most a continuation frame in C and a continuation stack P, where C, P are not empty, we can derive $n \neq 0$ comprehensions and have n valid matching processes and n derivations at the high level. Since C, P will be reduced in size, either by frame use or the arguments in the frames get reduced each time a comprehension is applied, we have a valid induction hypothesis.

Theorem 28. *If* $\mathsf{mc}\ \Gamma; \Delta_a, \Delta'_b; \Xi_N; \Gamma_{N1}; \Delta_{N1}; p_1; \Omega; (\Delta_a, p_1; \Delta'_b; \cdot; p; \Omega; \cdot; \Upsilon); P; \mathsf{comp}\ A \multimap B; \Omega_N; \Delta, \Xi_1, ..., \Xi_n \to \Xi'; \Delta'; \Gamma'$ *(related to A,* $\Delta_a, \Delta_b, p_1 = \Delta_N$ *and* Γ*) and* $\Delta_a, \Delta_b, p_1 = \Delta, \Xi_1, ..., \Xi_n$ *then* $\exists n \geq 0$ *such that:*

1. $\Gamma; \Delta_N; \Xi_N, \Xi_1, ..., \Xi_n; \Gamma_{N1}, \Gamma_1, ..., \Gamma_n; \Delta_{N1}, \Delta_1, ..., \Delta_n; \Omega_N \to \Xi'; \Delta'; \Gamma'$
2. $\mathsf{match}\ \Gamma; \Xi_1 \to A$ ... $\mathsf{match}\ \Gamma; \Xi_n \to A$
3. *n implications from* $1...i...n$ *such that:* $\forall \Omega_x, \Delta_x$. *if* $\Gamma; \Delta_x; \Xi_N, \Xi_1, ..., \Xi_i; \Gamma_{N1}, \Gamma_1, ..., \Gamma_i; \Delta_{N1}, \Delta_1, ..., \Delta_i; \Omega_x \to \Xi'; \Delta'; \Gamma'$ *then* $\mathsf{derive}\ \Gamma; \Delta_x; \Xi_N, \Xi_1, ..., \Xi_i; \Delta_{N1}, \Delta_1, ..., \Delta_{i-1}; B, \Omega_X \to \Xi'; \Delta'$

If $\mathsf{mc}\ \Gamma; \Delta_N; \Xi_N; \Gamma_{N1}; \Delta_{N1}; \cdot; \Omega; \cdot; P; \mathsf{comp}\ A \multimap B; \Omega_N; \Delta, \Xi_1, ..., \Xi_n \to \Xi'; \Delta'; \Gamma'$ *(related to A,* Δ_N *and* Γ*) and* $\Delta_N = \Delta, \Xi_1, ..., \Xi_n$ *then* $\exists n \geq 0$ *such that:*

1. $\Gamma; \Delta_N; \Xi_N, \Xi_1, ..., \Xi_n; \Gamma_{N1}, \Gamma_1, ..., \Gamma_n; \Delta_{N1}, \Delta_1, ..., \Delta_n; \Omega_N \to \Xi'; \Delta'; \Gamma'$
2. $\mathsf{match}\ \Gamma; \Xi_1 \to A$ ... $\mathsf{match}\ \Gamma; \Xi_n \to A$
3. *n implications from* $1...i...n$ *such that:* $\forall \Omega_x, \Delta_x$. *if* $\Gamma; \Delta_x; \Xi_N, \Xi_1, ..., \Xi_i; \Gamma_{N1}, \Gamma_1, ..., \Gamma_i; \Delta_{N1}, \Delta_1, ..., \Delta_i; \Omega_x \to \Xi'; \Delta'; \Gamma'$ *then* $\mathsf{derive}\ \Gamma; \Delta_x; \Xi_N, \Xi_1, ..., \Xi_i; \Delta_{N1}, \Delta_1, ..., \Delta_{i-1}; B, \Omega_X \to \Xi'; \Delta'$

Proof. By mutual induction, first on either the size of Δ'_b (first argument) or Γ' (second argument) and then on the size of C, P.

1. first implication:
   $\Delta_b = p_1, \Delta'_b$ (1) from assumption
   $\Delta_a, p_1, \Delta'_b = \Delta, p_1, \Xi'_1, ..., \Xi_n = \Delta_N$ $(\Xi_1 = p_1, \Xi_1)$ (2) from assumption
   By applying the comprehension soundness theorem to the assumption, we get:

   - Failure:
     $\Gamma; \Delta_N; \Xi_N; \Gamma_{N1}; \Delta_{N1}; \Omega_N \to \Xi'; \Delta'; \Gamma'$ (3) from theorem (so $n = 0$)

   - Success:
     $\mathsf{match}\ \Gamma; \Xi_1 \to A$ (3) from theorem

(a) Without backtracking:
$\mathsf{mc}\ \Gamma; \Delta, \Xi_2, ..., \Xi_n; \Xi_N; \Gamma_{N1}; \Delta_{N1}; p_1, \Xi'_1; \cdot; C', (\Delta_a, p_1; \Delta'_b; \cdot; p; \Omega; \cdot; \Upsilon); P; \mathsf{comp}\ A \multimap B; \Omega_N; \Delta_N \rightarrow \Xi'; \Delta'; \Gamma'$ (4) from theorem
$\mathsf{dc}\ \Gamma; \Xi_N, \Xi_1; \Gamma_{N1}; \Delta_{N1}; B; (\Delta_a; \Delta'_b - (\Xi_1); \cdot; p; \Omega; \cdot; \Upsilon); P; \mathsf{comp}\ A \multimap B; \Omega_N; \Delta, \Xi_2, ..., \Xi_n \rightarrow \Xi'; \Delta'; \Gamma'$ (5) using successful comprehension matches gives derivation lemma to (4)
$\mathsf{dc}\ \Gamma; \Xi_N, \Xi_1; \Gamma_{N1}, \Gamma_1; \Delta_{N1}, \Delta_1; \cdot; (\Delta_a; \Delta'_b - (\Xi_1); \cdot; p; \Omega; \cdot; \Upsilon); P; \mathsf{comp}\ A \multimap B; \Omega_N; \Delta, \Xi_2, ..., \Xi_n \rightarrow \Xi'; \Delta'$ (6) applying comprehension derivation lemma on (5)
if $\forall \Omega_x, \Delta_x.\mathsf{derive}\ \Gamma; \Delta_x; \Xi_N, \Xi_1; \Gamma_{N1}, \Gamma_1; \Delta_{N1}, \Delta_1; \Omega_x \rightarrow \Xi'; \Delta'; \Gamma'$ then $\mathsf{derive}\ \Gamma; \Delta_x; \Xi_N, \Xi_1; \Gamma_{N1}; \Delta_{N1}; B, \Omega_x -$
$\Xi'; \Delta'; \Gamma'$ (7) from the same lemma
$\mathsf{contc}\ \Gamma; \Delta, \Xi_2, ..., \Xi_n; \Xi_N, \Xi_1; \Gamma_{N1}, \Gamma_1; \Delta_{N1}, \Delta_1; (\Delta_a; \Delta'_b - (\Xi_1); \cdot; p; \Omega; \cdot; \Upsilon); P; \mathsf{comp}\ A \multimap B; \Omega_N \rightarrow \Xi'; \Delta'; \Gamma'$ (8) inversion of (7)

By inverting (8) we either fail ($n = 1$) or we get a new match. For the latter case, we apply mutual induction to get the remaining $n - 1$ comprehensions.

(b) With backtracking:

i. Linear frame
$f = (\Delta_a, p_1; \Delta'_b; \cdot; p; \Omega; \cdot; \Upsilon)$ (4) from theorem
turns into $f' = (\Delta_a, p_1, \Delta'''_b, p_2; \Delta''_b; \cdot; p; \Omega; \cdot; \Upsilon)$ (5) from theorem
$\mathsf{mc}\ \Gamma; \Delta, \Xi_2, ..., \Xi_n; \Xi_N; \Gamma_{N1}; \Delta_{N1}; p_2, \Xi'_1; \cdot; C', f'; P; \mathsf{comp}\ A \multimap B; \Omega_N; \Delta_N \rightarrow \Xi'; \Delta'; \Gamma'$ (6) from theorem

Use the same approach as the case without backtracking.

ii. Persistent frame
$f = [\Gamma''_1, p_2, \Gamma''_2; \Delta_N; \cdot; !p; \Omega; \cdot; \Upsilon]$ (4) from theorem
turns into $f' = [\Gamma''_2; \Delta_N; \cdot; !p; \Omega; \cdot; \Upsilon]$ (5) from theorem
$\mathsf{mc}\ \Gamma; \Delta, \Xi_2, ..., \Xi_n; \Xi_N; \Gamma_{N1}; \Delta_{N1}; \Xi_1; \cdot; C'; P', f', P''; \mathsf{comp}\ A \multimap B; \Omega_N; \Delta_N \rightarrow \Xi'; \Delta'; \Gamma'$ (6) from theorem

Use the same approach as the case without backtracking.

2. second implication:
Use the same approach as the one used in the first implication.

□

8.4.11 Comprehension Lemma

We prove that by starting with empty continuation stacks, we get n comprehensions.

Lemma 8. *If* $\mathsf{mc}\ \Gamma; \Delta, \Xi_1, ..., \Xi_n; \Xi_N; \Gamma_{N1}; \Delta_{N1}; \cdot; A; \cdot; \cdot; \mathsf{comp}\ A \multimap B; \Omega_N; \Delta, \Xi_1, ..., \Xi_n \rightarrow \Xi'; \Delta'; \Gamma'$ *(related to A, $\Delta, \Xi_1, ..., \Xi_n = \Delta_N$ and Γ) then $\exists n \geq 0$ such that:*

1. $\Gamma; \Delta_N; \Xi_N, \Xi_1, ..., \Xi_n; \Gamma_{N1}, \Gamma_1, ..., \Gamma_n; \Delta_{N1}, \Delta_1, ..., \Delta_n; \Omega_N \rightarrow \Xi'; \Delta'; \Gamma'$
2. $\mathsf{match}\ \Gamma; \Xi_1 \rightarrow A\ ...\ \mathsf{match}\ \Gamma; \Xi_n \rightarrow A$
3. *n implications from $1...i...n$ such that:* $\forall \Omega_x, \Delta_x$. *if* $\Gamma; \Delta_x; \Xi_N, \Xi_1, ..., \Xi_i; \Gamma_{N1}, \Gamma_1, ..., \Gamma_i; \Delta_{N1}, \Delta_1, ..., \Delta_i; \Omega_x \rightarrow \Xi'; \Delta'; \Gamma'$ *then* $\mathsf{derive}\ \Gamma; \Delta_x; \Xi_N, \Xi_1, ..., \Xi_i; \Delta_{N1}, \Delta_1, ..., \Delta_{i-1}; B, \Omega_X \rightarrow \Xi'; \Delta'$

Proof. If we apply the comprehension soundness lemma, we get two cases:

1. Failure:
$\Gamma; \Delta_N; \Xi_N; \Gamma_{N1}; \Delta_{N1}; \Omega_N \rightarrow \Xi'; \Delta'; \Gamma'$ (1) no comprehension application was possible ($n = 0$)

2. Success:

   $\mathsf{mc}\ \Gamma; \Xi_2, ..., \Xi_n; \Xi_N; \Gamma_{N1}; \Delta_{N1}; \Xi_1; \cdot; C; \mathsf{comp}\ A \multimap B; \Omega_N; \Delta_N \rightarrow \Xi'; \Delta'; \Gamma'$ (related) (1) result from theorem
   $\mathsf{match}\ \Gamma; \Xi_1 \rightarrow A$ (2) from theorem
   $\mathsf{dc}\ \Gamma; \Xi_N, \Xi_1; \Gamma_{N1}; \Delta_{N1}; B; C'; P'; \mathsf{comp}\ A \multimap B; \Omega_N; \Delta, \Xi_2, ..., \Xi_n \rightarrow \Xi'; \Delta'; \Gamma'$ (3) applying successful comprehension match gives derivation
   $\mathsf{dc}\ \Gamma; \Xi_N, \Xi_1; \Gamma_{N1}, \Gamma_1; \Delta_{N1}, \Delta_1; \cdot; C'; P'; \mathsf{comp}\ A \multimap B; \Omega_N; \Delta, \Xi_2, ..., \Xi_n \rightarrow \Xi'; \Delta'; \Gamma'$ (4) using comprehension derivation lemma
   $\mathsf{if}\ \forall \Omega_x, \Delta_x.\mathsf{derive}\ \Gamma; \Delta_x; \Xi_N, \Xi_1; \Gamma_{N1}, \Gamma_1; \Delta_{N1}, \Delta_1; \Omega_x \rightarrow \Xi'; \Delta'; \Gamma'$ then $\mathsf{derive}\ \Gamma; \Delta_x; \Xi_N, \Xi_1; \Gamma_{N1}; \Delta_{N1}; B, \Omega_x \rightarrow \Xi'; \Delta'; \Gamma'$ (5) from the same lemma
   $\mathsf{contc}\ \Gamma; \Delta, \Xi_2, ..., \Xi_n; \Xi_N, \Xi_1; \Gamma_{N1}, \Gamma_1; \Delta_{N1}, \Delta_1; C'; P'; \mathsf{comp}\ A \multimap B; \Omega_N \rightarrow \Xi'; \Delta'; \Gamma'$ (6) inversion of (5)

   Invert (6) to get either 0 applications of the comprehension or apply the comprehension theorem to the inversion to get the remaining $n - 1$.

□

8.4.12 Derivation Soundness Theorem

Theorem 29. *If* $\Gamma; \Delta; \Xi; \Gamma_1; \Delta_1; \Omega \rightarrow \Xi'; \Delta'; \Gamma'$ *then* $\mathsf{derive}\ \Gamma; \Delta; \Gamma_1; \Delta_1; \Omega \rightarrow \Xi'; \Delta'; \Gamma'$.

Proof. Use the same approach used in 7.4.10, but using the theorems presented in this section. □

8.4.13 Comprehension Soundness Theorem

9 Low Level System With Aggregates

9.1 High Level System

9.1.1 Match

$$\frac{}{\mathsf{match}\ \Gamma; \cdot \rightarrow 1}\ \mathsf{match}\ 1 \qquad \frac{}{\mathsf{match}\ \Gamma; p \rightarrow p}\ \mathsf{match}\ p \qquad \frac{}{\mathsf{match}\ \Gamma, p; \cdot \rightarrow !p}\ \mathsf{match}\ !p$$

$$\frac{\mathsf{match}\ \Gamma; \Delta_1 \rightarrow A \quad \mathsf{match}\ \Delta_2 \rightarrow B}{\mathsf{match}\ \Gamma; \Delta_1, \Delta_2 \rightarrow A \otimes B}\ \mathsf{match}\ \otimes$$

9.1.2 Derive

$$\frac{\mathsf{derive}\ \Gamma; \Delta; \Xi; \Gamma_1; p, \Delta_1; \Omega \rightarrow \Xi'; \Delta'; \Gamma'}{\mathsf{derive}\ \Gamma; \Delta; \Xi; \Gamma_1; \Delta_1; p, \Omega \rightarrow \Xi'; \Delta'; \Gamma'}\ \mathsf{derive}\ p$$

$$\frac{\mathsf{derive}\ \Gamma; \Delta; \Xi; \Gamma_1, p; \Delta_1; \Omega \rightarrow \Xi'; \Delta'; \Gamma'}{\mathsf{derive}\ \Gamma; \Delta; \Xi; \Gamma_1; \Delta_1; !p, \Omega \rightarrow \Xi'; \Delta'; \Gamma'}\ \mathsf{derive}\ !p$$

$$\frac{\mathsf{derive}\ \Gamma; \Delta; \Xi; \Gamma_1; \Delta_1; A, B, \Omega \rightarrow \Xi'; \Delta'; \Gamma'}{\mathsf{derive}\ \Gamma; \Delta; \Xi; \Gamma_1; \Delta_1; A \otimes B, \Omega \rightarrow \Xi'; \Delta'; \Gamma'}\ \mathsf{derive}\ \otimes$$

$$\frac{\mathsf{derive}\ \Gamma; \Delta; \Xi; \Gamma_1; \Delta_1; \Omega \rightarrow \Xi'; \Delta'; \Gamma'}{\mathsf{derive}\ \Gamma; \Delta; \Xi; \Gamma_1; \Delta_1; 1, \Omega \rightarrow \Xi'; \Delta'; \Gamma'}\ \mathsf{derive}\ 1$$

$$\frac{}{\mathsf{derive}\ \Gamma; \Delta; \Xi'; \Gamma'; \Delta'; \cdot \to \Xi'; \Delta'; \Gamma'}\ \mathsf{derive}\ end$$

$$\frac{\mathsf{derive}\ \Gamma; \Delta; \Xi; \Gamma_1; \Delta_1; \mathsf{agg\ Op}\ V\ (x.A(x))\ (y.B(y)), \Omega \to \Xi'; \Delta'; \Gamma' \quad [\mathsf{Op}] \hookrightarrow V}{\mathsf{derive}\ \Gamma; \Delta; \Xi; \Gamma_1; \Delta_1; [\mathsf{Op}; X; A \Rightarrow B], \Omega \to \Xi'; \Delta'; \Gamma'}\ \mathsf{derive}\ agg$$

$$\frac{\begin{array}{r}\mathsf{derive}\ \Gamma; \Delta; \Gamma_1; \Delta_1; B(V) \mathbin{\&} (\forall X'.A(X') \multimap \mathsf{agg\ Op}\ E\ (x.A(x))\ (y.B(y))), \Omega \to \Xi'; \Delta'; \Gamma' \\ [\mathsf{Op}]\ V'/X' \Rightarrow E\end{array}}{\mathsf{derive}\ \Gamma; \Delta; \Xi; \Gamma_1; \Delta_1; \mathsf{agg\ Op}\ V'\ (x.A(x))\ (y.B(y)), \Omega \to \Xi'; \Delta'; \Gamma'}\ \mathsf{derive}\ unfold\ agg$$

$$\frac{\mathsf{derive}\ \Gamma; \Delta; \Xi; \Gamma_1; \Delta_1; [M/X]A, \Omega; \to \Xi'; \Delta'; \Gamma'}{\mathsf{derive}\ \Gamma; \Delta; \Xi; \Gamma_1; \Delta_1; \forall X : \tau.A, \Omega \to \Xi'; \Delta'; \Gamma'}\ \mathsf{derive}\ \forall$$

$$\frac{\mathsf{derive}\ \Gamma; \Delta; \Xi; \Gamma_1; \Delta_1; 1 \mathbin{\&} (A \multimap B \otimes \mathsf{comp}\ A \multimap B), \Omega \to \Xi'; \Delta'; \Gamma'}{\mathsf{derive}\ \Gamma; \Delta; \Xi; \Gamma_1; \Delta_1; \mathsf{comp}\ A \multimap B, \Omega \to \Xi'; \Delta'; \Gamma'}\ \mathsf{derive}\ comp$$

$$\frac{\mathsf{match}\ \Gamma; \Delta_a \to A \quad \mathsf{derive}\ \Gamma; \Delta_b; \Xi, \Delta_a; \Gamma_1; \Delta_1; B, \Omega \to \Xi'; \Delta'; \Gamma'}{\mathsf{derive}\ \Gamma; \Delta_a, \Delta_b; \Xi; \Gamma_1; \Delta_1; A \multimap B, \Omega \to \Xi'; \Delta'; \Gamma'}\ \mathsf{derive}\ \multimap$$

$$\frac{\mathsf{derive}\ \Gamma; \Delta; \Xi; \Gamma_1; \Delta_1; A, \Omega \to \Xi'; \Delta'; \Gamma'}{\mathsf{derive}\ \Gamma; \Delta; \Xi; \Gamma_1; \Delta_1; A \mathbin{\&} B, \Omega \to \Xi'; \Delta'; \Gamma'}\ \mathsf{derive}\ \& L \qquad \frac{\mathsf{derive}\ \Gamma; \Delta; \Xi; \Gamma_1; \Delta_1; B, \Omega \to \Xi'; \Delta'; \Gamma'}{\mathsf{derive}\ \Gamma; \Delta; \Xi; \Gamma_1; \Delta_1; A \mathbin{\&} B, \Omega \to \Xi'; \Delta'; \Gamma'}\ \mathsf{derive}\ \& R$$

9.1.3 Apply

$$\frac{\mathsf{match}\ \Gamma; \Delta; A \to \Delta \quad \mathsf{derive}\ \Gamma; \Delta''; \Delta; \cdot; \cdot; B \to \Xi'; \Delta'; \Gamma'}{\mathsf{a}_0\ \Gamma; \Delta, \Delta''; A \multimap B \to \Xi'; \Delta'; \Gamma'}\ \mathsf{a}_0\ rule$$

$$\frac{\mathsf{do}_0\ \Gamma; \Delta; R \to \Xi'; \Delta'; \Gamma'}{\mathsf{do}_0\ \Gamma; \Delta; R, \Phi \to \Xi'; \Delta'; \Gamma'}\ \mathsf{do}_0\ rule$$

9.2 Low Level System

We complement the previous system with aggregate derivation. For this, we add the judgements ma , da and conta that will match, derive and apply continuations when applying an aggregate.

9.2.1 Match

$$\frac{\mathsf{m}_1\ \Gamma; \Delta, \Delta''; \Xi, p_1; \Omega; H; (\Delta, p_1; \Delta''; p; \Omega; \Xi; \cdot; \cdot); R \to \Xi'; \Delta'; \Gamma'}{\mathsf{m}_1\ \Gamma; \Delta, p_1, \Delta''; \Xi; p, \Omega; H; C; R \to \Xi'; \Delta'; \Gamma'}\ \mathsf{m}_1\ p\ ok\ first$$

$$\frac{\begin{array}{l}\mathsf{m}_1\ \Gamma; \Delta, \Delta''; \Xi, p_1; \Omega; H; (\Delta, p_1; \Delta''; p; \Omega; \Xi; q, \Lambda; \Upsilon), C_1, C; R \to \Xi'; \Delta'; \Gamma' \\ C_1 = (\Delta_{old}; \Delta'_{old}; q; \Omega_{old}; \Xi_{old}; \Lambda; \Upsilon)\end{array}}{\mathsf{m}_1\ \Gamma; \Delta, p_1, \Delta''; \Xi; p, \Omega; H; C_1, C; R \to \Xi'; \Delta'; \Gamma'}\ \mathsf{m}_1\ p\ ok\ other\ q$$

$$\frac{\begin{array}{l}\mathsf{m}_1\ \Gamma; \Delta, \Delta''; \Xi, p_1; \Omega; H; (\Delta, p_1; \Delta''; p; \Omega; \Xi; \Lambda; q, \Upsilon), C_1, C; R \to \Xi'; \Delta'; \Gamma' \\ C_1 = [\Gamma_{old}; \Delta_{old}; !q; \Omega_{old}; \Xi_{old}; \Lambda; \Upsilon]\end{array}}{\mathsf{m}_1\ \Gamma; \Delta, p_1, \Delta''; \Xi; p, \Omega; H; C_1, C; R \to \Xi'; \Delta'; \Gamma'}\ \mathsf{m}_1\ p\ ok\ other\ !q$$

$$\frac{p \notin \Delta \quad \mathsf{cont}\ C; H; R; \Gamma; \Xi'; \Delta'; \Gamma'}{\mathsf{m}_1\ \Gamma; \Delta; \Xi; p, \Omega; H; C; R \to \Xi'; \Delta'; \Gamma'}\ \mathsf{m}_1\ p\ \mathit{fail}$$

$$\frac{\mathsf{m}_1\ \Gamma, p, \Gamma'; \Delta; \Xi; \Omega; H; [\Gamma'; \Delta; !p; \Omega; \Xi; \Lambda; \cdot]; R \to \Xi'; \Delta'; \Gamma'}{\mathsf{m}_1\ \Gamma, p, \Gamma'; \Delta; \Xi; !p, \Omega; H; \cdot; R \to \Xi'; \Delta'; \Gamma'}\ \mathsf{m}_1\ !p\ \mathit{ok\ first}$$

$$\frac{\begin{array}{l}\mathsf{m}_1\ \Gamma, p, \Gamma'; \Delta; \Xi; \Omega; H; [\Gamma'; \Delta; !p; \Omega; \Xi; q, \Lambda; \Upsilon], C_1, C; R \to \Xi'; \Delta'; \Gamma' \\ C_1 = (\Delta_{old}; \Delta'_{old}; q; \Omega_{old}; \Xi_{old}; \Lambda; \Upsilon)\end{array}}{\mathsf{m}_1\ \Gamma, p, \Gamma'; \Delta; \Xi; !p, \Omega; H; C_1, C; R \to \Xi'; \Delta'; \Gamma'}\ \mathsf{m}_1\ !p\ \mathit{ok\ other}\ q$$

$$\frac{\begin{array}{l}\mathsf{m}_1\ \Gamma, p, \Gamma'; \Delta; \Xi; \Omega; H; [\Gamma'; \Delta; !p; \Omega; \Xi; \Lambda; q, \Upsilon], C_1, C; R \to \Xi'; \Delta'; \Gamma' \\ C_1 = [\Gamma_{old}; \Delta_{old}; !q; \Omega_{old}; \Xi_{old}; \Lambda; \Upsilon]\end{array}}{\mathsf{m}_1\ \Gamma, p, \Gamma'; \Delta; \Xi; !p, \Omega; H; C_1, C; R \to \Xi'; \Delta'; \Gamma'}\ \mathsf{m}_1\ !p\ \mathit{ok\ other}\ !q$$

$$\frac{p \notin \Gamma \quad \mathsf{cont}\ C; H; R; \Gamma; \Xi'; \Delta'; \Gamma'}{\mathsf{m}_1\ \Gamma; \Delta; \Xi; !p, \Omega; H; C; R \to \Xi'; \Delta'; \Gamma'}\ \mathsf{m}_1\ !p\ \mathit{fail}$$

$$\frac{\mathsf{m}_1\ \Gamma; \Delta; \Xi; A, B, \Omega; H; C; R \to \Xi'; \Delta'; \Gamma'}{\mathsf{m}_1\ \Gamma; \Delta; \Xi; A \otimes B, \Omega; H; C; R \to \Xi'; \Delta'; \Gamma'}\ \mathsf{m}_1\ \otimes$$

$$\frac{\mathsf{d}_1\ \Gamma; \Delta; \Xi; \cdot; H; \cdot \to \Xi'; \Delta'; \Gamma'}{\mathsf{m}_1\ \Gamma; \Delta; \Xi; \cdot; H; C; R \to \Xi'; \Delta'; \Gamma'}\ \mathsf{m}_1\ \mathit{end}$$

9.2.2 Continuation

$$\frac{\mathsf{do}_1\ \Gamma; \Delta; \Phi \to \Xi'; \Delta'; \Gamma'}{\mathsf{cont}\ \cdot; H; (\Phi, \Delta); \Gamma; \Xi'; \Delta'; \Gamma'}\ \mathsf{cont}\ \mathit{next\ rule}$$

$$\frac{\mathsf{m}_1\ \Gamma; \Delta, \Delta''; \Xi, p_1; \Omega; H; (\Delta, p_1; \Delta''; p, \Omega; H; \Xi; \Lambda; \Upsilon), C; R \to \Xi'; \Delta'; \Gamma'}{\mathsf{cont}\ (\Delta; p_1, \Delta''; p, \Omega; \Xi; \Lambda; \Upsilon), C; H; R; \Gamma; \Xi'; \Delta'; \Gamma'}\ \mathsf{cont}\ p\ \mathit{next}$$

$$\frac{\mathsf{cont}\ C; H; R; \Gamma; \Xi'; \Delta'; \Gamma'}{\mathsf{cont}\ (\Delta; \cdot; p, \Omega; \Xi; \Lambda; \Upsilon), C; H; R; \Gamma; \Xi'; \Delta'; \Gamma'}\ \mathsf{cont}\ p\ \mathit{no\ more}$$

$$\frac{\mathsf{m}_1\ \Gamma; \Delta; \Xi; \Omega; H; [\Gamma'; \Delta; !p, \Omega; \Xi; \Lambda; \Upsilon], C; R \to \Xi'; \Delta'; \Gamma'}{\mathsf{cont}\ [p_1, \Gamma'; \Delta; !p, \Omega; \Xi; \Lambda; \Upsilon], C; H; R; \Gamma; \Xi'; \Delta'; \Gamma'}\ \mathsf{cont}\ !p\ \mathit{next}$$

$$\frac{\mathsf{cont}\ C; H; R; \Gamma; \Xi'; \Delta'; \Gamma'}{\mathsf{cont}\ [\cdot; \Delta; !p, \Omega; \Xi; \Lambda; \Upsilon], C; H; R; \Gamma; \Xi'; \Delta'; \Gamma'}\ \mathsf{cont}\ !p\ \mathit{no\ more}$$

9.2.3 Derivation

We add a new rule for aggregates.

$$\frac{\mathsf{d}_1\ \Gamma; \Delta; \Xi; \Gamma_1; p, \Delta_1; \Omega \to \Xi'; \Delta'; \Gamma'}{\mathsf{d}_1\ \Gamma; \Delta; \Xi; \Gamma_1; \Delta_1; p, \Omega \to \Xi'; \Delta'; \Gamma'}\ \mathsf{d}_1\ p \qquad \frac{\mathsf{d}_1\ \Gamma; \Delta; \Xi; \Gamma_1; \Delta_1; \Omega \to \Xi'; \Delta'; \Gamma'}{\mathsf{d}_1\ \Gamma; \Delta; \Xi; \Gamma_1; \Delta_1; 1, \Omega \to \Xi'; \Delta'; \Gamma'}\ \mathsf{d}_1\ 1$$

$$\frac{\mathsf{d}_1\ \Gamma; \Delta; \Xi; \Gamma_1, p; \Delta_1; \Omega \to \Xi'; \Delta'; \Gamma'}{\mathsf{d}_1\ \Gamma; \Delta; \Xi; \Gamma_1; \Delta_1; !p, \Omega \to \Xi'; \Delta'; \Gamma'}\ \mathsf{d}_1\ !p$$

$$\frac{\mathsf{d}_1\ \Gamma; \Delta; \Xi; \Gamma_1; \Delta_1; A, B, \Omega \to \Xi'; \Delta'; \Gamma'}{\mathsf{d}_1\ \Gamma; \Delta; \Xi; \Gamma_1; \Delta_1; A \otimes B, \Omega \to \Xi'; \Delta'; \Gamma'}\ \mathsf{d}_1 \otimes$$

$$\frac{}{\mathsf{d}_1\ \Gamma; \Delta; \Xi; \Gamma_1; \Delta_1; \cdot \to \Xi; \Delta_1; \Gamma_1}\ \mathsf{d}_1\ end$$

$$\frac{\mathsf{mc}\ \Gamma; \Delta; \Xi; \Gamma_1; \Delta_1; \cdot; A; B; \cdot; \cdot; \mathsf{comp}\ A \multimap B; \Omega; \Delta \to \Xi'; \Delta'; \Gamma'}{\mathsf{d}_1\ \Gamma; \Delta; \Xi; \Gamma_1; \Delta_1; \mathsf{comp}\ A \multimap B, \Omega \to \Xi'; \Delta'; \Gamma'}\ \mathsf{d}_1\ comp$$

9.2.4 Match Comprehension

$$\frac{\mathsf{mc}\ \Gamma; \Delta, \Delta''; \Xi_N; \Gamma_{N1}; \Delta_{N1}; \Xi, p_1; \Omega; (\Delta, p_1; \Delta''; \cdot; p; \Omega; \cdot; \cdot); \cdot; AB; \Omega_N; \Delta_N \to \Xi'; \Delta'; \Gamma'}{\mathsf{mc}\ \Gamma; \Delta, p_1, \Delta''; \Xi_N; \Gamma_{N1}; \Delta_{N1}; \cdot; p, \Omega; \cdot; \cdot; AB; \Omega_N; \Delta_N \to \Xi'; \Delta'; \Gamma'}\ \mathsf{mc}\ p\ ok\ first$$

$$\frac{\begin{array}{l}\mathsf{mc}\ \Gamma; \Delta, \Delta''; \Xi_N; \Gamma_{N1}; \Delta_{N1}; \Xi, p_1; \Omega; (\Delta, p_1; \Delta''; \Xi; p; \Omega; q, \Lambda; \Upsilon), C_1, C; P; AB; \Omega_N; \Delta_N \to \Xi'; \Delta'; \Gamma' \\ C_1 = (\Delta_{old}; \Delta'_{old}; \Xi_{old}; q; \Omega_{old}; \Lambda; \Upsilon)\end{array}}{\mathsf{mc}\ \Gamma; \Delta, p_1, \Delta''; \Xi_N; \Gamma_{N1}; \Delta_{N1}; \Xi; p, \Omega; C_1, C; P; AB; \Omega_N; \Delta_N \to \Xi'; \Delta'; \Gamma'}\ \mathsf{mc}\ p\ ok\ other\ q$$

$$\frac{\begin{array}{l}\mathsf{mc}\ \Gamma; \Delta, \Delta''; \Xi_N; \Gamma_{N1}; \Delta_{N1}; \Xi, p_1; \Omega; (\Delta, p_1; \Delta''; \Xi; p; \Omega; \Lambda; q, \Upsilon), C_1, C; P; AB; \Omega_N; \Delta_N \to \Xi'; \Delta'; \Gamma' \\ C_1 = [\Gamma_{old}; \Delta_{old}; \Xi_{old}; q; \Omega_{old}; \Lambda; \Upsilon]\end{array}}{\mathsf{mc}\ \Gamma; \Delta, p_1, \Delta''; \Xi_N; \Gamma_{N1}; \Delta_{N1}; \Xi; p, \Omega; C_1, C; P; AB; \Omega_N; \Delta_N \to \Xi'; \Delta'; \Gamma'}\ \mathsf{mc}\ p\ ok\ other\ !qC$$

$$\frac{\begin{array}{l}\mathsf{mc}\ \Gamma; \Delta, \Delta''; \Xi_N; \Gamma_{N1}; \Delta_{N1}; p_1; \Omega; (\Delta, p_1; \Delta''; \cdot; p; \Omega; \cdot; q, \Upsilon); P_1, P; AB; \Omega_N; \Delta_N \to \Xi'; \Delta'; \Gamma' \\ P_1 = [\Gamma_{old}; \Delta_N; \cdot; q; \Omega_{old}; \cdot; \Upsilon] \\ \Delta_N = \Delta, p_1, \Delta''\end{array}}{\mathsf{mc}\ \Gamma; \Delta, p_1, \Delta''; \Xi_N; \Gamma_{N1}; \Delta_{N1}; \cdot; p, \Omega; \cdot; P_1, P; AB; \Omega_N; \Delta_N \to \Xi'; \Delta'; \Gamma'}\ \mathsf{mc}\ p\ ok\ other\ !qP$$

$$\frac{\mathsf{contc}\ \Gamma; \Delta_N; \Xi_N; \Gamma_{N1}; \Delta_{N1}; C; P; \mathsf{comp}\ A \multimap B; \Omega_N \to \Xi'; \Delta'; \Gamma'}{\mathsf{mc}\ \Gamma; \Delta; \Xi_N; \Gamma_{N1}; \Delta_{N1}; \Xi; p, \Omega; C; P; \mathsf{comp}\ A \multimap B; \Omega_N; \Delta_N \to \Xi'; \Delta'; \Gamma'}\ \mathsf{mc}\ p\ fail$$

$$\frac{\mathsf{mc}\ \Gamma, \Gamma', p; \Delta_N; \Xi_N; \Gamma_{N1}; \Delta_{N1}; \cdot; \Omega; \cdot; [\Gamma'; \Delta_N; \cdot; !p; \cdot; \Omega; \cdot; \cdot]; AB; \Omega_N; \Delta_N \to \Xi'; \Delta'; \Gamma'}{\mathsf{mc}\ \Gamma, \Gamma', p; \Delta_N; \Xi_N; \Gamma_{N1}; \Delta_{N1}; \cdot; !p, \Omega; \cdot; \cdot; AB; \Omega_N; \Delta_N \to \Xi'; \Delta'; \Gamma'}\ \mathsf{mc}\ !p\ first$$

$$\dfrac{\begin{array}{l}\mathsf{mc}\ \Gamma, \Gamma', p; \Delta_N; \Xi_N; \Gamma_{N1}; \Delta_{N1}; \cdot; \Omega; [\Gamma'; \Delta_N; \cdot; !p; \cdot; \Omega; \cdot; q, \Upsilon], P_1, P; AB; \Omega_N; \Delta_N \to \Xi'; \Delta'; \Gamma' \\ P_1 = [\Gamma_{old}; \Delta_N; \cdot; !q; \Omega_{old}; \cdot; \Upsilon]\end{array}}{\mathsf{mc}\ \Gamma, \Gamma', p; \Delta_N; \Xi_N; \Gamma_{N1}; \Delta_{N1}; \cdot; !p, \Omega; \cdot; P_1, P; AB; \Omega_N; \Delta_N \to \Xi'; \Delta'; \Gamma'}\ \mathsf{mc}\ !p\ \mathit{other}\ !qP$$

$$\dfrac{\begin{array}{l}\mathsf{mc}\ \Gamma, \Gamma', p; \Delta; \Xi_N; \Gamma_{N1}; \Delta_{N1}; \Xi; \Omega; [\Gamma'; \Delta; \Xi; !p; \cdot; \Omega; \Lambda; q, \Upsilon], C_1, C; P; AB; \Omega_N; \Delta_N \to \Xi'; \Delta'; \Gamma' \\ C_1 = [\Gamma_{old}; \Delta_{old}; \Xi_{old}; !q; \Omega_{old}; \Lambda; \Upsilon]\end{array}}{\mathsf{mc}\ \Gamma, \Gamma', p; \Delta; \Xi_N; \Gamma_{N1}; \Delta_{N1}; \Xi; !p, \Omega; C_1, C; P; AB; \Omega_N; \Delta_N \to \Xi'; \Delta'; \Gamma'}\ \mathsf{mc}\ !p\ \mathit{other}\ !qC$$

$$\dfrac{\begin{array}{l}\mathsf{mc}\ \Gamma, \Gamma', p; \Delta; \Xi_N; \Gamma_{N1}; \Delta_{N1}; \Xi; \Omega; [\Gamma'; \Delta; \Xi; !p; \cdot; \Omega; \Lambda, q; \Upsilon], C_1, C; P; AB; \Omega_N; \Delta_N \to \Xi'; \Delta'; \Gamma' \\ C_1 = (\Delta_{old}; \Delta'_{old}; \Xi_{old}; q; \Omega_{old}; \Lambda; \Upsilon)\end{array}}{\mathsf{mc}\ \Gamma, \Gamma', p; \Delta; \Xi_N; \Gamma_{N1}; \Delta_{N1}; \Xi; !p, \Omega; C_1, C; P; AB; \Omega_N; \Delta_N \to \Xi'; \Delta'; \Gamma'}\ \mathsf{mc}\ !p\ \mathit{other}\ qC$$

$$\dfrac{\mathsf{mc}\ \Delta; \Xi_N; \Delta_{N1}; \Xi; X, Y, \Omega; C; P; \mathsf{comp}\ A \multimap B; \Omega_N; \Delta_N \to \Xi'; \Delta'}{\mathsf{mc}\ \Delta; \Xi_N; \Delta_{N1}; \Xi; X \otimes Y, \Omega; C; P; \mathsf{comp}\ A \multimap B; \Omega_N; \Delta_N \to \Xi'; \Delta'}\ \mathsf{mc}\ \otimes$$

$$\dfrac{\mathsf{dall}\ \Gamma; \Xi_N; \Gamma_{N1}; \Delta_{N1}; \Xi; C; P; \mathsf{comp}\ A \multimap B; \Omega_N; \Delta_N \to \Xi'; \Delta'; \Gamma'}{\mathsf{mc}\ \Gamma; \Delta; \Xi_N; \Gamma_{N1}; \Delta_{N1}; \Xi; \cdot; C; P; \mathsf{comp}\ A \multimap B; \Omega_N; \Delta_N \to \Xi'; \Delta'; \Gamma'}\ \mathsf{mc}\ \mathit{end}$$

9.2.5 Match Comprehension Continuation

$$\dfrac{\mathsf{d}_1\ \Gamma; \Delta_N; \Xi_N; \Gamma_{N1}; \Delta_{N1}; \Omega \to \Xi'; \Delta'; \Gamma'}{\mathsf{contc}\ \Gamma; \Delta_N; \Xi_N; \Gamma_{N1}; \Delta_{N1}; \cdot; \cdot; \mathsf{comp}\ A \multimap B; \Omega \to \Xi'; \Delta'; \Gamma'}\ \mathsf{contc}\ \mathit{end}$$

$$\dfrac{\mathsf{mc}\ \Gamma; \Delta; \Xi_N; \Gamma_{N1}; \Delta_{N1}; \Xi; \Omega; (\Delta, p_1; \Delta''; \Xi; p; \Omega; \Lambda; \Upsilon), C; P; AB; \Omega_N; \Delta_N \to \Xi'; \Delta'; \Gamma'}{\mathsf{contc}\ \Gamma; \Delta_N; \Xi_N; \Gamma_{N1}; \Delta_{N1}; (\Delta; p_1, \Delta''; \Xi; p; \Omega; \Lambda; \Upsilon), C; P; AB; \Omega_N \to \Xi'; \Delta'; \Gamma'}\ \mathsf{contc}\ \mathit{nextC}\ p$$

$$\dfrac{\mathsf{mc}\ \Gamma; \Delta; \Xi_N; \Gamma_{N1}; \Delta_{N1}; \Xi; \Omega; [\Gamma'; \Delta; \Xi; !p; \Omega; \Lambda; \Upsilon], C; P; AB; \Omega_N; \Delta_N \to \Xi'; \Delta'; \Gamma'}{\mathsf{contc}\ \Gamma; \Delta_N; \Xi_N; \Gamma_{N1}; \Delta_{N1}; [p_1, \Gamma'; \Delta; \Xi; !p; \Omega; \Lambda; \Upsilon], C; P; AB; \Omega_N \to \Xi'; \Delta'; \Gamma'}\ \mathsf{contc}\ \mathit{nextC}\ !p$$

$$\dfrac{\mathsf{contc}\ \Gamma; \Delta_N; \Xi_N; \Gamma_{N1}; \Delta_{N1}; C; P; AB; \Omega_N \to \Xi'; \Delta'; \Gamma'}{\mathsf{contc}\ \Gamma; \Delta_N; \Xi_N; \Gamma_{N1}; \Delta_{N1}; (\Delta; \cdot; \Xi; p; \Omega; \Lambda; \Upsilon), C; P; AB; \Omega_N \to \Xi'; \Delta'; \Gamma'}\ \mathsf{contc}\ \mathit{nextC\ empty}\ p$$

$$\dfrac{\mathsf{contc}\ \Gamma; \Delta_N; \Xi_N; \Gamma_{N1}; \Delta_{N1}; C; P; AB; \Omega_N \to \Xi'; \Delta'; \Gamma'}{\mathsf{contc}\ \Gamma; \Delta_N; \Xi_N; \Gamma_{N1}; \Delta_{N1}; [\cdot; \Delta; \Xi; !p; \Omega; \Lambda; \Upsilon], C; P; AB; \Omega_N \to \Xi'; \Delta'; \Gamma'}\ \mathsf{contc}\ \mathit{nextC\ empty}\ !p$$

$$\dfrac{\mathsf{mc}\ \Gamma; \Delta_N; \Xi_N; \Gamma_{N1}; \Delta_{N1}; \cdot; \Omega; \cdot; [\Gamma'; \Delta_N; \cdot; !p; \Omega; \cdot; \Upsilon], P; AB; \Omega_N; \Delta_N \to \Xi'; \Delta'; \Gamma'}{\mathsf{contc}\ \Gamma; \Delta_N; \Xi_N; \Gamma_{N1}; \Delta_{N1}; \cdot; [p_1, \Gamma'; \Delta_N; \cdot; !p; \Omega; \cdot; \Upsilon], P; AB; \Omega_N \to \Xi'; \Delta'; \Gamma'}\ \mathsf{contc}\ \mathit{nextP}\ !p$$

$$\dfrac{\mathsf{contc}\ \Gamma; \Delta_N; \Xi_N; \Gamma_{N1}; \Delta_{N1}; \cdot; P; AB; \Omega_N \to \Xi'; \Delta'; \Gamma'}{\mathsf{contc}\ \Gamma; \Delta_N; \Xi_N; \Gamma_{N1}; \Delta_{N1}; \cdot; [\cdot; \Delta_N; \cdot; !p; \Omega; \cdot; \Upsilon], P; AB; \Omega_N \to \Xi'; \Delta'; \Gamma'}\ \mathsf{contc}\ \mathit{nextP\ empty}\ !p$$

9.2.6 Stack Transformation

Like the previous system, we need to transform continuation stack to be able to used again. First, we remove everything except the first frame in the C stack. Next, we transform the Δ argument in the first frame of C and in every frame of P to remove recently consumed facts.

The $\mathsf{strans}\ \Xi; P; P'$ judgments transforms the P stack and has the following meaning:

Ξ : Consumed linear resources during the last application of the comprehension.

P : The P stack.

P' : The transformed P stack.

$$\frac{\mathsf{strans}\ \Xi; P; P'}{\mathsf{strans}\ \Xi; [\Gamma'; \Delta_N; \cdot; !p; \Omega; \cdot; \Upsilon], P; [\Gamma'; \Delta_N - \Xi; \cdot; !p, \Omega; \cdot; \Upsilon], P'}\ \mathsf{strans}$$

$$\frac{\mathsf{strans}\ \Xi; \cdot; \cdot}{\mathsf{strans}\ \Xi; \cdot; \cdot}\ \mathsf{strans}\ \mathit{end}$$

$$\frac{\mathsf{strans}\ \Xi; P; P' \quad \mathsf{dc}\ \Gamma; \Xi_N, \Xi; \Gamma_{N1}; \Delta_{N1}; (\Delta_x - \Xi; \Delta'' - \Xi; \cdot; p; \Omega; \cdot; \Upsilon); P'; \mathsf{comp}\ A \multimap B; \Omega_N; (\Delta_N - \Xi) \rightarrow \Xi'; \Delta'; \Gamma'}{\mathsf{dall}\ \Gamma; \Xi_N; \Gamma_{N1}; \Delta_{N1}; \Xi; (\Delta_x; \Delta''; \cdot; p; \Omega; \cdot; \Upsilon); P; \mathsf{comp}\ A \multimap B; \Omega_N; \Delta_N \rightarrow \Xi'; \Delta'; \Gamma'}\ \mathsf{dall}\ \mathit{end\ linear}$$

$$\frac{\mathsf{strans}\ \Xi; P; P' \quad \mathsf{dc}\ \Gamma; \Xi_N, \Xi; \Gamma_{N1}; \Delta_{N1}; \cdot; P'; \mathsf{comp}\ A \multimap B; \Omega_N; (\Delta_N - \Xi) \rightarrow \Xi'; \Delta'; \Gamma'}{\mathsf{dall}\ \Gamma; \Xi_N; \Gamma_{N1}; \Delta_{N1}; \Xi; \cdot; P; \mathsf{comp}\ A \multimap B; \Omega_N; \Delta_N \rightarrow \Xi'; \Delta'; \Gamma'}\ \mathsf{dall}\ \mathit{end\ empty}$$

$$\frac{\mathsf{dall}\ \Gamma; \Xi_N; \Gamma_{N1}; \Delta_{N1}; \Xi; X, C; P; \mathsf{comp}\ A \multimap B; \Omega_N; \Delta_N \rightarrow \Xi'; \Delta'; \Gamma'}{\mathsf{dall}\ \Gamma; \Xi_N; \Gamma_{N1}; \Delta_{N1}; \Xi; \_, X, C; P; \mathsf{comp}\ A \multimap B; \Omega_N; \Delta_N \rightarrow \Xi'; \Delta'; \Gamma'}\ \mathsf{dall}\ \mathit{more}$$

9.2.7 Comprehension Derivation

The dc is identical to the previous system, however it has been extended with Γ, Γ_1, C, P and Γ'.

$$\frac{\mathsf{dc}\ \Gamma; \Xi; \Gamma_1; \Delta_1, p; \Omega; C; P; \mathsf{comp}\ A \multimap B; \Omega_N; \Delta_N \rightarrow \Xi'; \Delta'; \Gamma'}{\mathsf{dc}\ \Gamma; \Xi; \Gamma_1; \Delta_1; p, \Omega; C; P; \mathsf{comp}\ A \multimap B; \Omega_N; \Delta_N \rightarrow \Xi'; \Delta'; \Gamma'}\ \mathsf{dc}\ p$$

$$\frac{\mathsf{dc}\ \Gamma; \Xi; \Gamma_1, p; \Delta_1; \Omega; C; P; \mathsf{comp}\ A \multimap B; \Omega_N; \Delta_N \rightarrow \Xi'; \Delta'; \Gamma'}{\mathsf{dc}\ \Gamma; \Xi; \Gamma_1; \Delta_1; !p, \Omega; C; P; \mathsf{comp}\ A \multimap B; \Omega_N; \Delta_N \rightarrow \Xi'; \Delta'; \Gamma'}\ \mathsf{dc}\ !p$$

$$\frac{\mathsf{dc}\ \Gamma; \Xi; \Gamma_1; \Delta_1; A, B, \Omega; C; P; \mathsf{comp}\ A \multimap B; \Omega_N; \Delta_N \rightarrow \Xi'; \Delta'; \Gamma'}{\mathsf{dc}\ \Gamma; \Xi; \Gamma_1; \Delta_1; A \otimes B, \Omega; C; P; \mathsf{comp}\ A \multimap B; \Omega_N; \Delta_N \rightarrow \Xi'; \Delta'; \Gamma'}\ \mathsf{dc}\ \otimes$$

$$\frac{\mathsf{contc}\ \Gamma; \Delta_N; \Xi; \Gamma_1; \Delta_1; C; P; \mathsf{comp}\ A \multimap B; \Omega_N \rightarrow \Xi'; \Delta'; \Gamma'}{\mathsf{dc}\ \Gamma; \Xi; \Gamma_1; \Delta_1; \cdot; C; P; \mathsf{comp}\ A \multimap B; \Omega_N; \Delta_N \rightarrow \Xi'; \Delta'; \Gamma'}\ \mathsf{dc}\ \mathit{end}$$

9.2.8 Application

Finally, we add Γ and Γ' to the main inference rules to complete the system.

$$\frac{\mathsf{m}_1\ \Gamma; \Delta; \cdot; \cdot; A; B; \cdot; R \to \Xi'; \Delta'; \Gamma'}{\mathsf{a}_1\ \Gamma; \Delta; A \multimap B; R \to \Xi'; \Delta'; \Gamma'}\ \mathsf{a}_1\ \textit{start matching}$$

$$\frac{\mathsf{a}_1\ \Gamma; \Delta; R; (\Phi, \Delta) \to \Xi'; \Delta'; \Gamma'}{\mathsf{do}_1\ \Gamma; \Delta; R, \Phi \to \Xi'; \Delta'; \Gamma'}\ \mathsf{do}_1\ \textit{rule}$$