Index of the slides:

DTT: the exchange rule
In Jacobs (10.1) the exchange rule for DTT is stated like this:

$$\frac{\Gamma, x:\sigma, y:\tau, \Delta \vdash M:\rho}{\Gamma, y:\tau, x:\sigma, \Delta \vdash M:\rho}$$

with a side-condition: "x is not free in τ".

Let's translate this:

$$\frac{\vec{a}{:}\vec{A}, b{:}B_{\vec{a}}, c{:}C_{\vec{a}}, \vec{d}{:}\vec{D}_{\vec{a}bc} \vdash e_{\vec{a}bc\vec{d}}{:}E_{\vec{a}bc\vec{d}}}{\vec{a}{:}\vec{A}, c{:}C_{\vec{a}}, b{:}B_{\vec{a}}, \vec{d}{:}\vec{D}_{\vec{a}bc} \vdash e_{\vec{a}bc\vec{d}}{:}E_{\vec{a}bc\vec{d}}}$$

Note that if we had used $c : C_{\vec{a}b}$ instead of $c : C_{\vec{a}}$
the bottom judgment would have made no sense.

Let's make this shorter.
We can hide the annotations that indicate dependencies,
the types, and the "vector" marks:

$$\frac{\vec{a}{:}\vec{A}, b{:}B, c{:}C, \vec{d}{:}\vec{D} \vdash e{:}E}{\vec{a}{:}\vec{A}, c{:}C, b{:}B, \vec{d}{:}\vec{D} \vdash e{:}E} \qquad \frac{\vec{a}, b, c, \vec{d} \vdash e}{\vec{a}, c, b, \vec{d} \vdash e} \qquad \frac{a, b, c, d \vdash e}{a, c, b, d \vdash e}$$

It is this last form that we will use.

Exercise: rewrite the first translation with

$$\begin{array}{rcl} (\vec{a}{:}\vec{A}) & := & (a_1{:}A_1[\,], \ldots, a_n{:}A_n[a_1, ..., a_{n-1}]) \\ (\vec{d}{:}\vec{D}_{\vec{a}bc}) & := & (d_1{:}D_1[a_1, ..., a_n, b, c], \ldots, \\ & & \;d_m{:}D_m[a_1, ..., a_n, b, c, d_1, ..., d_{m-1}]) \end{array}$$

and check that the rule becomes unbearably big.

DTT: the "unpack" rule
In Jacobs (10.1, but after 10.1.2) the strong sum-elimination rule is stated as this:

$$\frac{\Gamma, z : \Sigma x : \sigma.\tau \vdash \rho : \mathsf{Type} \quad \Gamma, x : \sigma, y : \tau \vdash Q : \rho[\langle x, y\rangle / z]}{\Gamma, z : \Sigma x : \sigma.\tau \vdash (\text{unpack } z \text{ as } \langle x, y\rangle \text{ in } Q) : \rho} \text{ (strong)}$$

In an "unpack" term like

$$\text{unpack } P \text{ as } \langle x, y\rangle \text{ in } Q$$

the "unpack" binds two variables in Q, x and y, at the same time, and sets their values to the components of the (dependent) pair P.

In the presence of π and π' we can *define*:

$$(\text{unpack } P \text{ as } \langle x, y\rangle \text{ in } Q) := Q[x := \pi P, y := \pi' P].$$

Let's change the "unpack" notation one step at a time:

$$\begin{array}{ll} & \text{unpack } P \text{ as } \langle x, y\rangle \text{ in } Q \\ \Rightarrow & \text{unpack } P =: x, y \text{ in } Q \\ \Rightarrow & Q[x, y := P] \end{array}$$

Now let's rewrite the rule:

$$\frac{\Gamma, z : \Sigma x : \sigma.\tau \vdash \rho : \mathsf{Type} \quad \Gamma, x : \sigma, y : \tau \vdash Q : \rho[z := \langle x, y\rangle]}{\Gamma, z : \Sigma x : \sigma.\tau \vdash Q[x, y := z] : \rho}$$

$$\frac{\vec{a}{:}\vec{A}, p{:}(\Sigma b{:}B.C) \vdash D{:}\mathsf{Type} \quad \vec{a}{:}\vec{A}, b{:}B, c{:}C \vdash d{:}D[p := \langle b, c\rangle]}{\vec{a}{:}\vec{A}, p{:}(\Sigma b{:}B.C) \vdash d[b, c := p]{:}D}$$

$$\frac{\vec{a}{:}\vec{A}, (b, c){:}(\Sigma b{:}B.C) \vdash D{:}\mathsf{Type} \quad \vec{a}{:}\vec{A}, b{:}B, c{:}C \vdash d{:}D[(b, c) := \langle b, c\rangle]}{\vec{a}{:}\vec{A}, (b, c){:}(\Sigma b{:}B.C) \vdash d[b, c := (b, c)]{:}D}$$

$$\frac{\vec{a}, (b, c) \vdash D \quad \vec{a}, b, c \vdash d}{\vec{a}, (b, c) \vdash d[b, c := (b, c)]}$$

$$\frac{a, (b, c) \vdash D \quad a, b, c \vdash d}{a, (b, c) \vdash d[b, c := (b, c)]} \; \Sigma\mathrm{E}^+$$

$$\frac{a, (b, c) \vdash D \quad a, b, c \vdash d}{a, (b, c) \vdash d} \; \Sigma\mathrm{E}^+$$

We will use the two last forms.

DTT: structural rules

These ones are used very often:

Variable: $$\frac{a \vdash B}{a, b \vdash b}\ \mathrm{v}$$

Substitution: $$\frac{a \vdash b \quad a, b, c \vdash D}{a, c \vdash D}\ \mathrm{s} \qquad \frac{a \vdash b \quad a, b, c \vdash d}{a, c \vdash d}\ \mathrm{s}$$

Weakeking: $$\frac{a \vdash B \quad a \vdash C}{a, b \vdash C}\ \mathrm{w} \qquad \frac{a \vdash B \quad a \vdash c}{a, b \vdash c}\ \mathrm{w}$$

These ones not so much:

Conversion: $$\frac{a \vdash b \quad a \vdash B = B'}{a \vdash b'}\ \mathrm{conv}$$

Contraction: $$\frac{a, b, b', c \vdash D}{a, b, c \vdash D}\ \mathrm{contr} \qquad \frac{a, b, b', c \vdash d}{a, b, c \vdash d}\ \mathrm{contr}$$

Exchange: $$\frac{a, b, c, d \vdash E}{a, c, b, d \vdash E}\ \mathrm{exch} \qquad \frac{a, b, c, d \vdash e}{a, c, b, d \vdash e}\ \mathrm{exch}$$

Note: “Variable” is called “Projection” at [Jacobs].

DTT: type rules
We have four different type-formers:
singleton, (dependent) products, (dependent) sums, and equality.
For each one of them we have a type-building rule, an introduction rule, and elimination rules.

There are several options for elimination rules for dependent sums and equality.

In a system with "weak sums" the rule is ΣE^-.
In a system with "strong sums" the rule is ΣE^+, or, equivalently, $\pi + \pi'$.

In a system with "weak equality" the rule is EqE^-.
In a system with "strong equality" the rule is EqE^+, or, equivalently, ee+ur ("externalization of equality" plus "uniqueness of reflexivity").

Singleton: $\dfrac{}{\vdash 1}\ 1$ $\quad \dfrac{}{\vdash *}\ 1I$ $\quad \dfrac{a \vdash *'}{a \vdash *' = *}\ 1E$

Products: $\dfrac{a, b \vdash C}{a \vdash \Pi b : B.C}\ \Pi$ $\quad \dfrac{a, b \vdash c}{a \vdash b \mapsto c}\ \Pi I$ $\quad \dfrac{a \vdash b \quad a \vdash b \mapsto c}{a \vdash c}\ \Pi E$

Sums: $\dfrac{a, b \vdash C}{a \vdash \Sigma b : B.C}\ \Sigma$ $\quad \dfrac{a \vdash B \quad a, b \vdash C}{a, b, c \vdash (b, c)}\ \Sigma I$ $\quad$ (See below)

Equality: $\dfrac{a \vdash B}{a, b, b' \vdash \mathrm{W}[b = b']}\ Eq$ $\quad \dfrac{a \vdash B}{a, b \vdash (b = b)}\ EqI$ $\quad$ (See below)

$$\dfrac{a \vdash D \quad a, b, c \vdash d}{a, (b, c) \vdash d}\ \Sigma E^- \qquad \dfrac{a, b, b', c \vdash D \quad a, b, c \vdash d}{a, b, b', (b = b'), c \vdash d}\ EqE^-$$

$$\dfrac{a, (b, c) \vdash D \quad a, b, c \vdash d}{a, (b, c) \vdash d}\ \Sigma E^+ \qquad \dfrac{a, b, b', (b = b') \vdash C \quad a, b \vdash c}{a, b, b', (b = b') \vdash c}\ EqE^+$$

$$\dfrac{a \vdash b, c}{a \vdash b}\ \pi \qquad \dfrac{a \vdash b, c}{a \vdash c}\ \pi' \qquad \dfrac{a \vdash (b = b')}{a \vdash b = b'}\ ee \qquad \dfrac{a \vdash (b = b)'}{a \vdash (b = b)' = (b = b)}\ ur$$

DTT: alternate rules for strong sum

Jacobs, 10.1.3 (i):

The rules π and π' can be defined from ΣE^+:

$$\dfrac{a \vdash (b,c)}{a \vdash b}\,\pi \quad := \quad \dfrac{a \vdash (b,c) \qquad \dfrac{\dfrac{\dfrac{\dfrac{a, b \vdash C}{a \vdash \Sigma b.C}\,\Sigma \qquad a \vdash B}{a, (b,c) \vdash B}\,\text{w} \qquad \dfrac{a, b \vdash C \qquad \dfrac{a \vdash B}{a, b \vdash b}\,\text{v}}{a, b, c \vdash b}\,\text{w}}{a, (b,c) \vdash b}\,\Sigma E^+}{a \vdash b}\,\text{s}$$

$$\dfrac{a \vdash (b,c)}{a \vdash c}\,\pi' \quad := \quad \dfrac{a \vdash (b,c) \qquad \dfrac{\dfrac{\dfrac{a, b \vdash C}{\overline{a, (b,c) \vdash b}} \qquad a, b \vdash C}{a, (b,c) \vdash C}\,\text{s} \qquad \dfrac{a, b \vdash C}{a, b, c \vdash c}\,\text{v}}{a, (b,c) \vdash c}\,\Sigma E^+}{a \vdash c}\,\text{s}$$

The rule ΣE^+ can be defined from π and π':

$$\dfrac{a, (b,c) \vdash D \qquad a, b, c \vdash d}{a, (b,c) \vdash d}\,\Sigma E^+$$

$$:= \quad \dfrac{\dfrac{a, b \vdash C}{\overline{a, (b,c) \vdash c}}\,\pi' \qquad \dfrac{\dfrac{a, b \vdash C}{\overline{a, (b,c) \vdash b}}\,\pi \qquad \dfrac{\dfrac{a, b \vdash C}{a \vdash \Sigma b.C}\,\Sigma \qquad a, b, c \vdash d}{a, (b,c), b, c \vdash d}\,\text{w}}{a, (b,c), c \vdash d}\,\text{s}}{a, (b,c) \vdash d}\,\text{s}$$

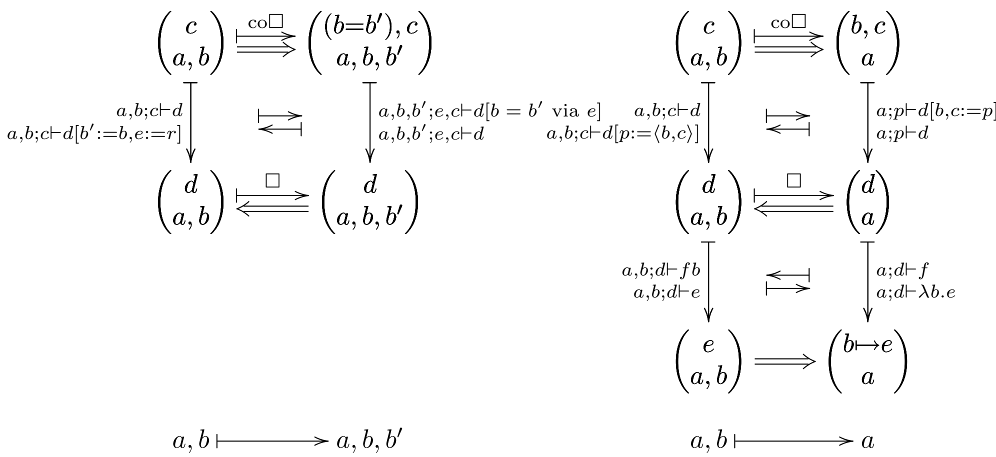
DTT: Alternate rules for strong equality

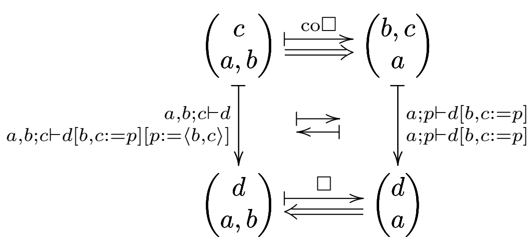
The ΣE^+ rule is equivalent to the two rules ee and ur, that say that from "witnesses of equality" we can prove external equality - i.e., that some terms are equal.
This equivalence $\Sigma E^+ \iff (\mathrm{ee}, \mathrm{ur})$ is of a different nature from the ones that we have seen before - this one uses $\beta/\epsilon/{:=}$ and lives intrinsically in the (P+T) structure - it cannot be restricted to the T-part (i.e., to the syntactical world).

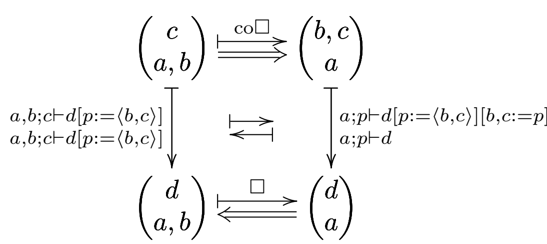
$$\begin{array}{rcl} a, b, b', e & \vdash & b \\ & \overset{\eta}{=} & b[b' := b, e := r] \text{ with } b = b' \text{ via } e \\ & = & b'[b' := b, e := r] \text{ with } b = b' \text{ via } e \\ & \overset{\eta}{=} & b' \end{array}$$

$$\begin{array}{rcl} a, b, b', e & \vdash & e \\ & \overset{\eta}{=} & e[b' := b, e := r] \text{ with } b = b' \text{ via } e \\ & = & r[b' := b, e := r] \text{ with } b = b' \text{ via } e \\ & \overset{\eta}{=} & r \end{array}$$

Adjunction diagrams







Conversions

Conventions:

β-conversions first, then η-conversions.

Underlined names are terms.

$\underline{(b{=}b)}$ is the reflexivity term.

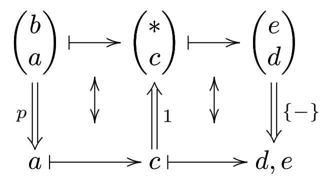
$$\begin{array}{rcl}
(\lambda b.\underline{c})\underline{b} & = & \underline{c}[\underline{b} =: b] \\
\lambda b.\underline{f}b & = & \underline{f} \\
\text{unpack } \langle \underline{b}, \underline{c} \rangle \text{ as } \langle b, c \rangle \text{ in } \underline{d} & = & \underline{d}[\underline{b} =: b, \underline{c} =: c] \\
\text{unpack } \underline{(b, c)} \text{ as } \langle b, c \rangle \text{ in } \underline{d}[\langle b, c \rangle =: (b, c)] & = & \underline{d}[\underline{(b, c)} =: (b, c)] \\
\underline{d} \text{ with } b = b \text{ via } \underline{(b{=}b)} & = & \underline{d} \\
\underline{d}[b =: b', \underline{(b{=}b)} =: (b{=}b)'] \text{ with } b' = b \text{ via } (b{=}b)' & = & \underline{d}
\end{array}$$

$$\begin{array}{rcccl}
a & \vdash & f & \stackrel{\beta}{=} & \lambda b.fb \\
a, b & \vdash & e & \stackrel{\eta}{=} & (\lambda b.e)b \\
a, b, c & \vdash & d & = & d[b, c := \langle b, c \rangle] \\
a, p & \vdash & d & = & d[p := \langle b, c \rangle][b, c := p] \\
a, b, c & \vdash & d & = & d[\text{with } b = b \text{ via } r] \\
a, b, b', e, c & \vdash & d & = & d[b' := b, e := r][\text{with } b' = b \text{ via } e]
\end{array}$$

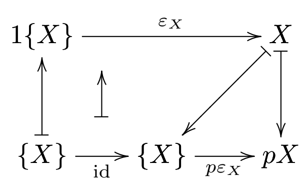
$$\frac{\Gamma, x : \sigma, x' : \sigma, \Delta \vdash \rho : \mathsf{Type} \quad \Gamma, x : \sigma, \Delta[x/x'] \vdash Q : \rho[x/x']}{\Gamma, x : \sigma, x' : \sigma, z : \mathrm{Eq}_\sigma(x, x'), \Delta \vdash (Q \text{ with } x' = x \text{ via } z) : \rho} \text{ (weak)}$$

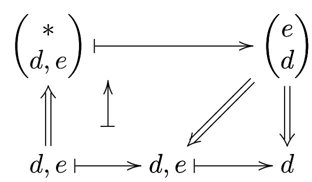
$$\frac{a, b, b', c \vdash D \quad a, b, c[b =: b'] \vdash d : D[b =: b']}{a, b, b', (b = b'), c \vdash (d \text{ with } b' = b \text{ via } (b = b')) : D} \text{ (weak)}$$

Comprehension categories with unit
Jacobs, 10.4.7 (p.616):
A fibration $p : \mathbb{E} \to \mathbb{B}$ with a terminal object functor $1 : \mathbb{B} \to \mathbb{E}$
(where we know by lemma 1.8.8 that $p \dashv 1$ and that $\eta_I = \mathrm{id}$)
is *comprehension category with unit* if 1 has a right adjoint.
We call this right adjoint $\{-\}$.

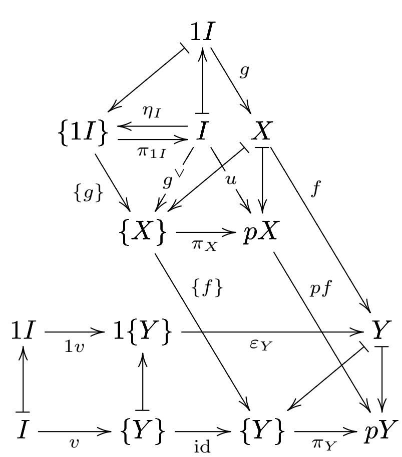


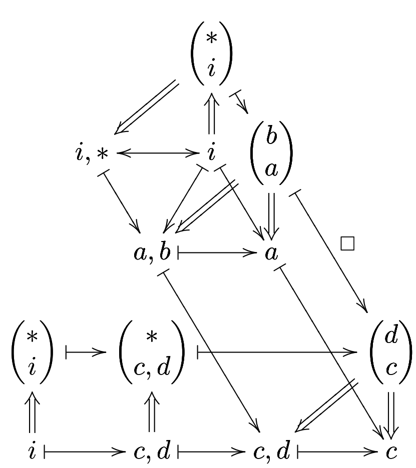
Jacobs, 10.4.7 (p.616):
Definition of the functor $\mathbb{E} \to \mathbb{B}^{\to}$:
its action on objects is $X \mapsto p\varepsilon_X$.



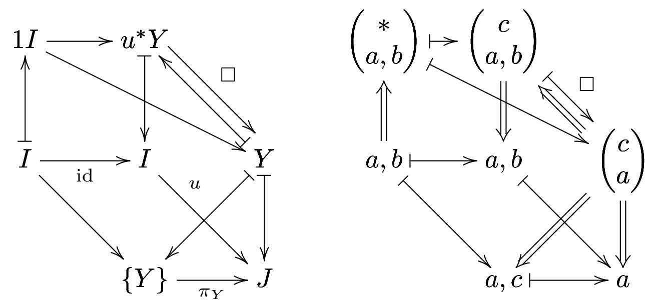


The functor $\mathbb{E} \to \mathbb{B}^{\to}$ is a comprehension category, i.e.,
it takes cartesian morphisms to pullback squares.
We want to check that the image of a cartesian morphism is a pullback.
Given two maps $i \mapsto a$ and $i \mapsto c, d$ such that
$a \mapsto c$ is well-defined, we need to construct a
mediating map $i \mapsto a, b$.

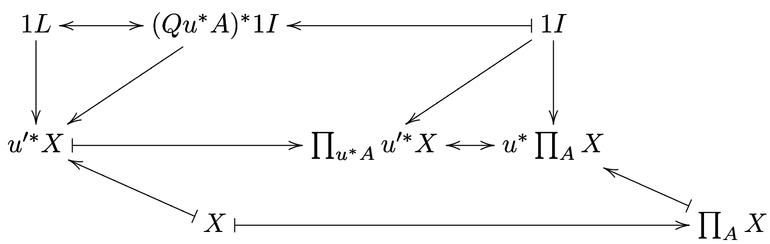


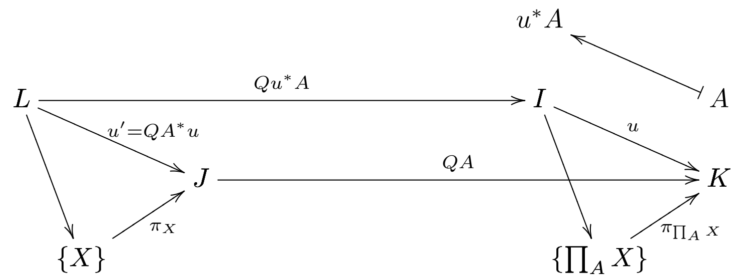


Comprehension categories with unit: a bijection
Jacobs, 10.4.9 (i):
In a CCw1, pack a morphism $u : I \to J$ in the base category, and an object Y over J. Then the vertical morphisms $1I \to u^*Y$ are in bijection with morphisms from u to π_Y in $\mathbb{B}/J$.

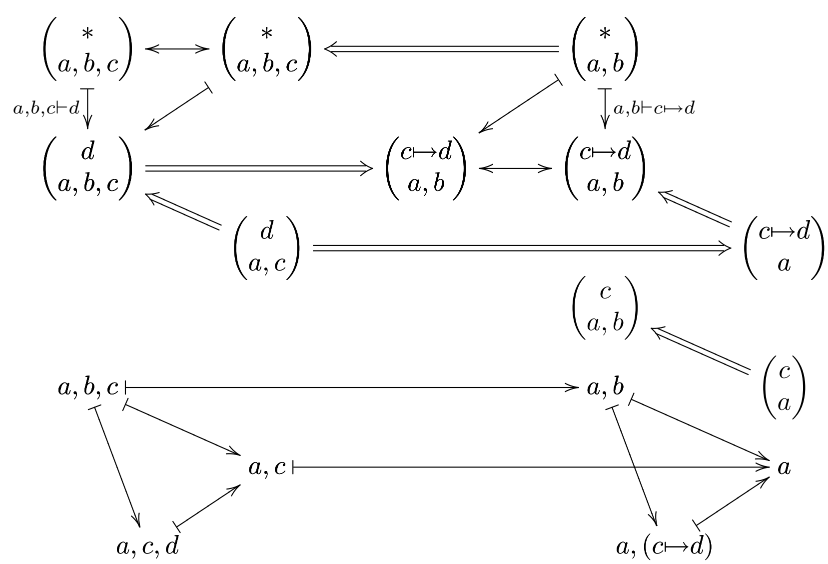


Comprehension categories with unit: big bijection
Jacobs, 10.4.9 (ii):



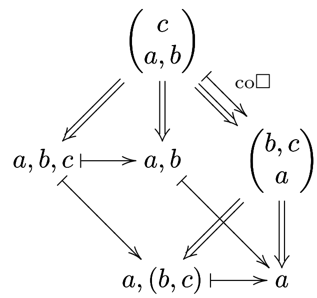


$a, c; b \vdash d$
$a; b \vdash c{\mapsto}d$

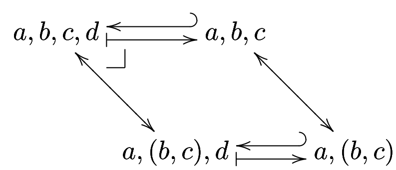


Comprehension categories with unit: three rules
Jacobs, 10.3.3:
The categorical interpretation of
the rules for dependent sums:

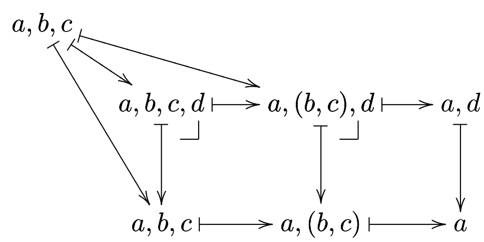
$$\frac{a, b \vdash C}{a; b, c \vdash (b, c)}\ \Sigma\mathrm{I}$$



$$\frac{a, (b, c) \vdash D \quad a, b, c \vdash d}{a, (b, c) \vdash d}\ \Sigma\mathrm{E}^{+}$$

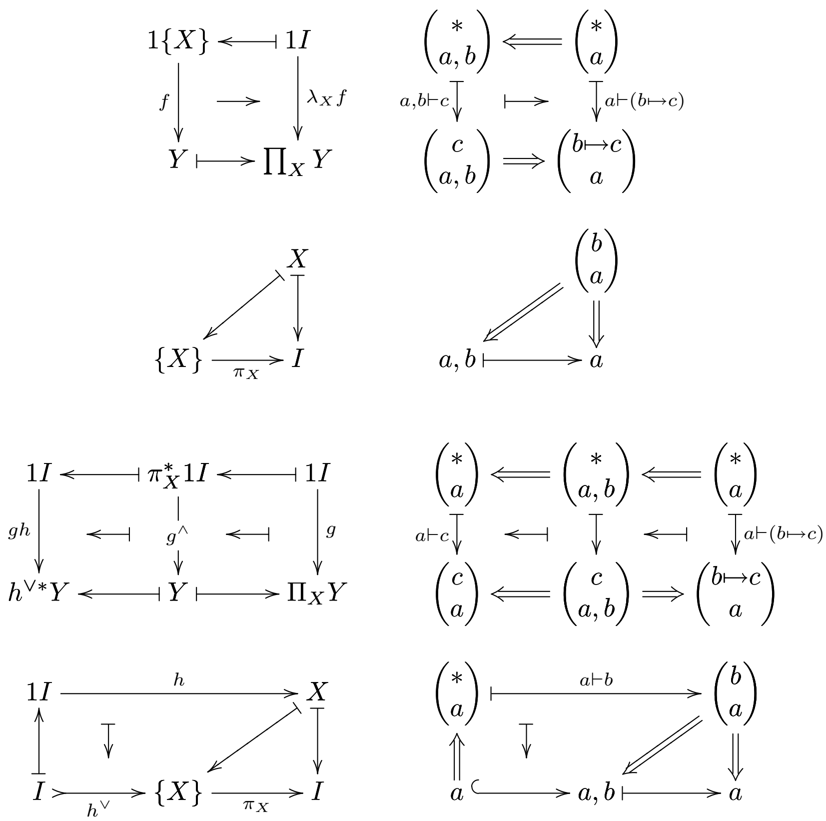


$$\frac{a \vdash D \quad a, b, c \vdash d}{a, (b, c) \vdash d}\ \Sigma\mathrm{E}^{-}$$



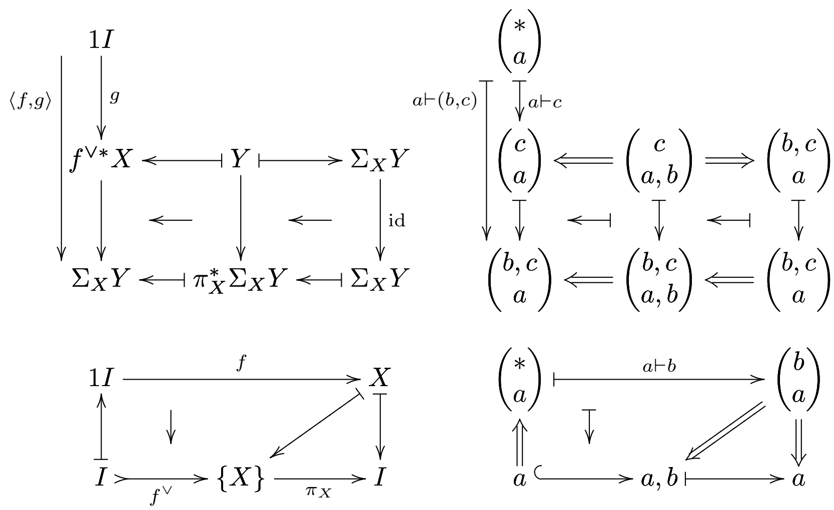
(Oops, the diagram for $\Sigma\mathrm{E}^{-}$ is wrong)

Interpreting ΠI and ΠE in a CCompC
(Jacobs, 10.5.3)

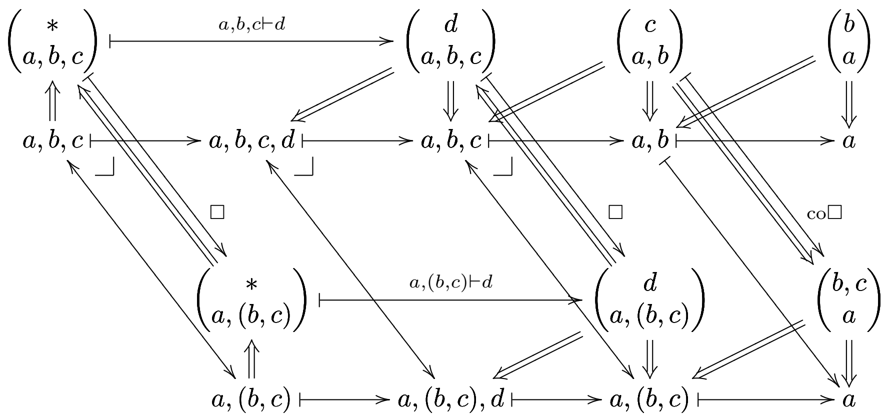


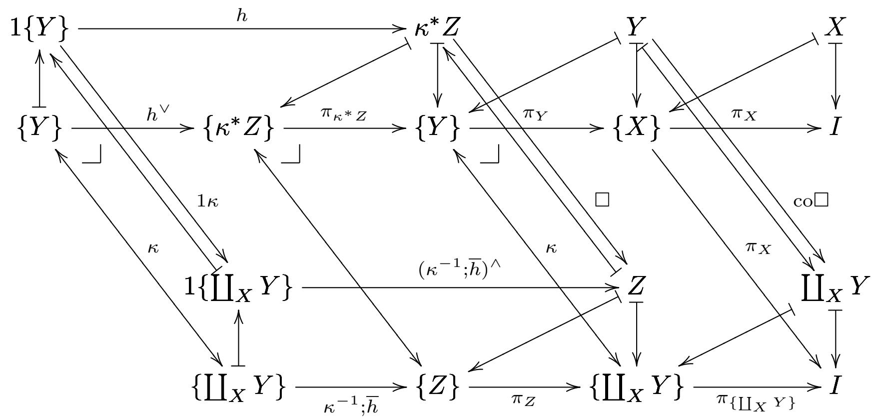
In the top left vertex of the diagram for ΠE we have omitted an iso to keep the diagram shorter: $1I \cong h^{\vee *}\pi_X^* 1I$.

Interpreting ΣI in a CCompC
(Jacobs, 10.5.3)



Interpreting ΣE^+ in a CCompC
(Jacobs, 10.5.3)





The "unpack" rule (2)

In 10.1.2 Jacobs defines (for $P : \sigma \times \tau$):

$$\begin{array}{rcl} \pi P & \stackrel{\text{def}}{=} & \text{unpack } P \text{ as } \langle x, y \rangle \text{ in } x \\ \pi' P & \stackrel{\text{def}}{=} & \text{unpack } P \text{ as } \langle x, y \rangle \text{ in } y \end{array}$$

i.e.,

$$\begin{array}{rcl} \pi P & := & x[x, y := P] \\ \pi' P & := & y[x, y := P] \end{array}$$

Rules for DTT

```
                a|-b  a|-B=B'
Conversion:     -------------
                    a|-b'

                 a|-B
Projection:     ------
                a,b|-b

                a,b,b',c|-D       a,b,b',c|-d
Contraction:    -----------       -----------
                 a,b,c|-D          a,b,c|-d

                a|-b  a,b,c|-D    a|-b  a,b,c|-d
Substitution:   --------------    --------------
                    a,c|-D          a,c|-d

                a|-B  a|-C        a|-B  a|-c
Weakening:      ----------        ----------
                  a,b|-C            a,b|-c

                a,b,c,d|-E        a,b,c,d|-e
Exchange:       ----------        ----------
                a,c,b,d|-E        a,c,b,d|-e

                Type:             Intro:          Elim:

                                                  a|-*'
Singleton:       ---               ---           -------
                 |-1               |-*           a|-*'=*

               a,b|-C             a,b|-c      a|-b  a|-b|->c
DepProds:     ---------          --------     --------------
              a|-Πb:B.C          a|-b|->c          a|-c

               a,b|-C          a|-B  a,b|-C
DepSums:      ---------        ------------    (see below)
              a|-Σb:B.C        a,b,c|-(b,c)

                a|-B               a|-B
Equality:   ---------------     ----------     (see below)
            a,b,b'|-W[b=b']     a,b|-(b=b)

Elimination rules:
                 DepSums:                  Equality:

              a|-D  a,b,c|-d         a,b,b',c|-D  a,b,c|-d
Weak:         --------------         ---------------------
                a,(b,c)|-d             a,b,b',(b=b'),c|-d

            a,(b,c)|-D  a,b,c|-d    a,b,b',(b=b')|-C  a,b|-c
Strong:     --------------------    ------------------------
                 a,(b,c)|-d            a,b,b',(b=b')|-c

             a|-b,c    a|-b,c      a|-(b=b')      a|-(b=b)'
AltStrong:   ------    ------      ---------   ---------------
              a|-b      a|-c        a|-b=b'    a|-(b=b)'=(b=b)
```