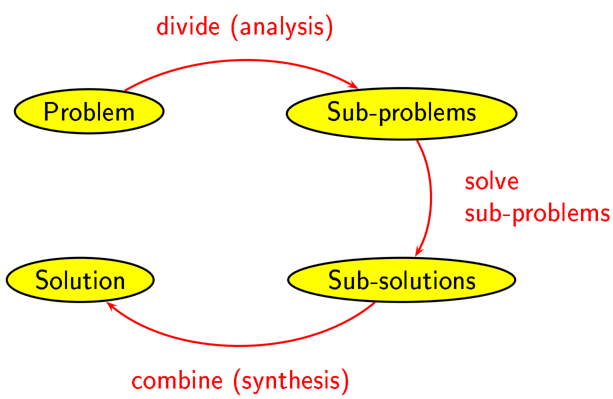
An Introduction to Relational Hylomorphisms

José N. Oliveira
Dept. Informatica
Universidade do Minho, 4700 Braga, Portugal
jno@di.uminho.pt

September 16, 2004

"How" does one specify?

General problem solving strategy? **Divide-and-conquer**:



Divide-and-conquer (formally)

Problem space Sub-problem structure

$$\begin{tikzcd} A \arrow[r, "Divide"] \arrow[d, "Solve"'] & \mathsf{F}\,A \arrow[d, "\mathsf{F}\,Solve"] \\ B & \mathsf{F}\,B \arrow[l, "Conquer"] \end{tikzcd}$$

Solution space Questions:

- What are the mathematics of $Divide$, $Conquer$, $Solve$?
- What do $(\mathsf{F}\,A)$, $(\mathsf{F}\,Solve)$ mean?

Relators

Symbol F is overloaded:

- $\mathsf{F}\,A$ means a (parametric) **datatype**, eg. $A^\star$ — `seq of A` in VDM-SL;
- $\mathsf{F}\,X$ means a **relation**

$$\begin{tikzcd} A \arrow[d, "X"'] \\ B \end{tikzcd} \qquad \begin{tikzcd} \mathsf{F}\,A \arrow[d, "\mathsf{F}\,X"] \\ \mathsf{F}\,B \end{tikzcd}$$

Example: $X^\star$ will be such that

$$l(X^\star)l' \quad \equiv \quad \mathit{len}\,l = \mathit{len}\,l' \wedge \forall i \in \mathit{inds}\,l.(l\ i)X(l'\ i)$$

Properties of relators

Every **relator** F is monotone,

$$R \subseteq S \quad \Rightarrow \quad (\mathsf{F}\,R) \subseteq (\mathsf{F}\,S)$$

and commutes with $(\cdot)$, $(\_)^\circ$ and id:

$$\begin{array}{rcl} \mathsf{F}\,(R \cdot S) & = & (\mathsf{F}\,R) \cdot (\mathsf{F}\,S) \\ \mathsf{F}\,id & = & id \\ \mathsf{F}\,(R^\circ) & = & (\mathsf{F}\,R)^\circ \end{array}$$

Terminology:

$A \xleftarrow{R} \mathsf{F}\,A$ is called an F-**algebra** $A \xrightarrow{S} \mathsf{F}\,A$ is called an F-**coalgebra**

Back to divide-and-conquer

Divide-and-conquer = relational **hylomorphism**:

$$\begin{tikzcd} A \arrow[r, "S"] \arrow[d, "X"'] & \mathsf{F}\,A \arrow[d, "\mathsf{F}\,X"] \\ B & \mathsf{F}\,B \arrow[l, "R"] \end{tikzcd}$$

that is, $X = R \cdot (\mathsf{F}\,X) \cdot S$

How do we solve this (hylo) equation for X?

An example first

```
mergeSort: seq of int -> seq of int
mergeSort(l) ==
   cases l :
      [e] -> [e] ,
      others -> let l1 ^ l2
                in set {l} be st
                        abs (len l1 - len l2) < 2 in
                lmerge(mergeSort(l1), mergeSort(l2))
   end;
```

is a **relational** hylomorphism for

$$\mathsf{F}\,X \quad = \quad \mathtt{int} + X \times X$$

In fact

$$\begin{tikzcd}
\mathtt{seq\ of\ int} \arrow[r, "S"] \arrow[d, "mergeSort"'] & \mathsf{F}\,\mathtt{seq\ of\ int} \arrow[d, "\mathsf{F}\,mergeSort"] \\
\mathtt{seq\ of\ int} & \mathsf{F}\,\mathtt{seq\ of\ int} \arrow[l, "R"]
\end{tikzcd}$$

that is,

$$mergeSort \quad = \quad R \cdot (\mathsf{F}\,mergeSort) \cdot S$$

where $R \quad = \quad [singl\ , lmerge]$, for $singl = \lambda e.[e]$

$mergeSort$ algebra and coalgebra

and S is

```
S: seq of int -> ( int | seq of int * seq of int )
S(l) ==
   cases l :
      [e] -> e ,
      others -> let l1 ^ l2
                in set {l} be st
                        abs (len l1 - len l2) < 2 in
                mk_(l1,l2)
   end;
```

Equations and fixpoints

Given an **equation** of pattern

$$x \quad = \quad f\ x$$

where $A \xleftarrow{f} A$ for some A, we will say that any **solution** to this equation — that is, any $a_0 \in A$ such that

$$a_0 \quad = \quad f\ a_0$$

is a **fixpoint** of f.

Equations versus recursion

Equation $x = f\ x$ can also be regarded as a "recursive" definition of its fixpoints, eg.

- $x = 1 + \frac{x}{2}$ is a recursive definition of number 2.

However,

- $x = \frac{x^2+3}{4}$ has two solutions (=fixpoints) 1 e 3. What are we "recursively defining" here?
- Furthermore, $x = x$ defines any object!
- Last but not least, some equations don't have any solution at all. Think eg. of $x = x + 1$ in $\mathbb{N}$.

Solving (Fixpoint) Equations I

Let $A \xleftarrow{\leq_A} A$ be a **partial order**. Then, every $a \in A$ such that

$$a \quad \leq_A f\, a$$

is said to be a post-fixpoint of f, and every $a \in A$ such that

$$a \quad \geq_A f\, a$$

is said to be a pre-fixpoint of f. Clearly,

> Every $a \in A$ which is both a pre-fixpoint and a post-fixpoint of f is a fixpoint of f.

Solving (Fixpoint) Equations II

Function $B \xleftarrow{f} A$ is **monotone** wherever

$$f \cdot \leq_A \quad \subseteq \quad \leq_B \cdot f$$

for partial orders $\leq_A$ and $\leq_B$, that is:

$$\begin{array}{cl} & f \cdot \leq_A \quad \subseteq \quad \leq_B \cdot f \\ \equiv & \quad \{ \text{ shunting} \} \\ & \leq_A \quad \subseteq \quad f^\circ \cdot \leq_B \cdot f \\ \equiv & \quad \{ \text{ going pointwise} \} \\ & a \leq_A a' \quad \Rightarrow \quad (f\, a) \leq_B (f\, a') \end{array}$$

Solving (Fixpoint) Equations III

Pointwise ordering on functions $B \xleftarrow{f,g} A$:

$$f \mathrel{\dot{\leq}_B} g \quad \equiv \quad f \quad \subseteq \quad \leq_B \cdot g$$

meaning

$$\begin{array}{rcl} f \mathrel{\dot{\leq}_B} g & \equiv & f \quad \subseteq \quad \leq_B \cdot g \\ & \equiv & \qquad \{\text{ shunting}\} \\ & & id \subseteq f^{\circ} \cdot \leq_B \cdot g \\ & \equiv & \qquad \{\text{ going pointwise}\} \\ & & \forall a . f\, a \leq_B g\, a \end{array}$$

Solving (Fixpoint) Equations IV

Lattice fixpoint theorem (Tarski 1955) for **monotone** f as above and $\leq_A$ defining a **complete lattice**:

- The set of all fixpoints of f,

  $$P = \{a \in A \mid a = f\, a\}$$

  is non-empty and $\langle P; \leq_A \rangle$ is a complete (sub)lattice.

- The least of all fixpoints ($\bigwedge P$) and the greatest one ($\bigvee P$) are as follows:

  $$\begin{array}{rcccl} \mu f & = & \bigwedge P & = & \bigwedge \{x \mid x \geq f\, x\} \\ \nu f & = & \bigvee P & = & \bigvee \{x \mid x \leq_A f\, x\} \end{array}$$

Solving relational equations

Hylo-equation $X = \underbrace{R \cdot (\mathsf{F}\, X) \cdot S}_{f\ X}$

and other relational equations such as

$$X = \underbrace{R \cup R \cdot X}_{g\ X}$$

(cf. **transitive** closure) have least solutions

$$\begin{array}{lcl} \mu f & = & [\![R, S]\!] \\ \mu g & = & R^{+} \end{array}$$

because both f, g are monotone.

Laws of the Fixpoint Calculus

Computation rule:

$$\mu f \quad = \quad f\, \mu f$$

Example: hylo-cancellation law

$$[\![R, S]\!] \quad = \quad R \cdot \mathsf{F}\, [\![R, S]\!] \cdot S$$

Rolling rule:

$$\mu(g \cdot h) \quad = \quad g(\mu(h \cdot g))$$

Example: $f = g \cdot h$ where $h\, X = R \cdot X$ and $g\, X = R \cup X$. Then

Rolling rule

$$
\begin{array}{lll}
\mu f & = & \mu(g \cdot h) \\
 & = & \quad \{ \text{ rolling rule } \} \\
 & & g(\mu(h \cdot g)) \\
 & = & \quad \{ \text{ definitions of } g, h \} \\
 & & R \cup (\mu x.(R \cdot (R \cup x)) \\
 & = & \quad \{ \ (R\cdot) \text{ is a lower-adjoint } \} \\
 & & R \cup \mu x.(R^2 \cup R \cdot x)
\end{array}
$$

Further application of this rule will "factor out" R^2, R^3, etc., In the limit, $\mu f = \bigcup_{j=1}^{\infty} R^j = R^+$.

Hylo rolling rule

Let $f = g \cdot h$ where $h\,X = \mathsf{F}\,X \cdot S$ and $g = (R\cdot)$. Then

$$
\begin{array}{lllll}
\mu f & = & \mu(g \cdot h) & = & g(\mu(h \cdot g)) \\
 & & & = & \quad \{ \text{ definitions of } g, h \} \\
 & & & & R \cdot (\mu X.(\mathsf{F}(R \cdot X) \cdot S) \\
 & & & = & \quad \{ \text{ relators } \} \\
 & & & & R \cdot (\mu X.\mathsf{F}\,R \cdot \mathsf{F}\,X \cdot S)
\end{array}
$$

that is,

$$
[\![R, S]\!] \quad = \quad R \cdot [\![\mathsf{F}\,R, S]\!]
$$

Other rules

Square rule:

$$\mu f = \mu(f^2)$$

Monotonicity:

$$f \stackrel{.}{\leq} g \quad \Rightarrow \quad \mu f \leq \mu g$$

Thus

$$[\![T, U]\!] \subseteq [\![R, S]\!] \quad \Leftarrow \quad T \subseteq R \wedge U \subseteq S$$

Other rules

Induction rule:

$$f\, x \leq x \quad \Rightarrow \quad \mu f \leq x$$

Thus

$$[\![R, S]\!] \subseteq T \quad \Leftarrow \quad R \cdot \mathsf{F}\, T \cdot S \subseteq T$$

and, in particular (**coreflexive** hylos):

$$\begin{array}{rcl} [\![R, S]\!] \subseteq id & \Leftarrow & R \cdot S \subseteq id \\ [\![R, R^\circ]\!] \subseteq id & \Leftarrow & R \text{ is simple} \end{array}$$

Last — but not least — **μ-fusion**:

μ-fusion theorem

Let

$$\begin{tikzcd} A & B \arrow[l, "f^\flat"'] \\ A \arrow[u, "g"] & B \arrow[l, "f^\flat"] \arrow[u, "h"'] \end{tikzcd}$$

- h, g be monotonic,
- $(A, \leq)$ and $(B, \sqsubseteq)$ be complete **lattices**,
- $f^\flat$ be a lower-adjoint.

Then

$$f^\flat(\mu h) = \mu g \quad \Leftarrow \quad f^\flat \cdot h = g \cdot f^\flat$$

Applications of μ-fusion theorem

Converse of a hylo

$$[\![\, S, R \,]\!]^\circ = [\![\, R^\circ, S^\circ \,]\!]$$

Proof: let $f^\flat = (\_)^\circ$ and

$$\begin{array}{rcl} h\ X & = & S \cdot \mathsf{F}\, X \cdot R \\ g\ X & = & T \cdot \mathsf{F}\, X \cdot U \end{array}$$

that is,

$$\begin{array}{l} \mu h = [\![\, S, R \,]\!] \\ \mu g = [\![\, T, U \,]\!] \end{array}$$

Proof

Then

$$
\begin{array}{ll}
 & [\![\, S, R \,]\!]^\circ = [\![\, T, U \,]\!] \\
\Leftarrow & \quad \{\ \mu\text{-fusion theorem}\ \} \\
 & (S \cdot \mathsf{F}\, X \cdot R)^\circ = T \cdot \mathsf{F}\,(X^\circ) \cdot U \\
\equiv & \quad \{\ \text{converse and } \mathsf{F} \text{ is a relator} \} \\
 & R^\circ \cdot \mathsf{F}\, X^\circ \cdot S^\circ = T \cdot \mathsf{F}\, X^\circ \cdot U \\
\Leftarrow & \quad \{\ \text{Leibnitz} \} \\
 & R^\circ = T \wedge S^\circ = U
\end{array}
$$

Hylo(cata)-fusion

$$V \cdot [\![\, S, R \,]\!] = [\![\, T, R \,]\!] \quad \Leftarrow \quad V \cdot S = T \cdot (\mathsf{F}\, V)$$

Proof: since $(V\cdot) = (V\backslash)^\flat$,

$$
\begin{array}{ll}
 & V \cdot [\![\, S, R \,]\!] = [\![\, T, R \,]\!] \\
\Leftarrow & \quad \{\ \mu\text{-fusion theorem}\ \} \\
 & V \cdot (S \cdot \mathsf{F}\, X \cdot R) = T \cdot \mathsf{F}\,(V \cdot X) \cdot R \\
\equiv & \quad \{\ \text{associative } (\cdot) \text{ and relator } \mathsf{F}\ \} \\
 & (V \cdot S) \cdot \mathsf{F}\, X \cdot R = T \cdot (\mathsf{F}\, V) \cdot (\mathsf{F}\, X) \cdot R \\
\Leftarrow & \quad \{\ \text{Leibnitz} \} \\
 & V \cdot S = T \cdot (\mathsf{F}\, V)
\end{array}
$$

Hylo(ana)-fusion

$$[\![\, S, R \,]\!] \cdot V = [\![\, S, U \,]\!] \quad \Leftarrow \quad R \cdot V = \mathsf{F}\, V \cdot U$$

Proof: $(\cdot V) = (/V)^\flat$. Then

$$\begin{array}{cl}
 & [\![\, S, R \,]\!] \cdot V = [\![\, S, U \,]\!] \\
\Leftarrow & \qquad \{ \ \mu\text{-fusion theorem } \} \\
 & (S \cdot \mathsf{F}\, X \cdot R) \cdot V = S \cdot \mathsf{F}\, (X \cdot V) \cdot U \\
\equiv & \qquad \{ \text{ associative } (\cdot) \text{ and relator } \mathsf{F} \ \} \\
 & S \cdot \mathsf{F}\, X \cdot (R \cdot V) = S \cdot (\mathsf{F}\, X) \cdot (\mathsf{F}\, V) \cdot U \\
\Leftarrow & \qquad \{ \text{ Leibnitz} \} \\
 & R \cdot V = \mathsf{F}\, V \cdot U
\end{array}$$

Examples: VDM collective types

$$\begin{tikzcd}
\mathsf{set\ of}\ A \arrow[r, "ins^\circ"] \arrow[d, "\{\!|R|\!\}"'] & 1 + A \times \mathsf{set\ of}\ A \arrow[d, "id + id \times \{\!|R|\!\}"] \\
B & 1 + A \times B \arrow[l, "R"]
\end{tikzcd}$$

that is,

$$\{\!|R|\!\} \quad = \quad [\![\, R, ins^\circ \,]\!] \qquad \text{where} \qquad ins \quad \stackrel{\text{def}}{=} \quad [\underline{\emptyset}\ , puts]$$

and ...

VDM-SL collective type set of A

```
puts[@A] :  @A * set of @A -> set of @A
puts(e,s) == {e} union s
pre not e in set s ;
```

Pointfree version (for $R = [\underline{u}\ , f]$):

```
shylo[@A,@B] : (@A*@B -> @B) * @B -> set of @A -> @B
shylo(f,u)(s) ==
      if s={} then u
      else let a in set s,
               r = s \ {a}
           in f(a,shylo[@A,@B](f,u)(r));
```

VDM-SL collective type set of A

- For `shylo(f,u)` to be a function the following must hold:

$$f(a, f(a', b)) \quad = \quad f(a', f(a, b))$$

- **Fusion** law

$$T \cdot (\![R]\!) = (\![S]\!) \quad \Leftarrow \quad T \cdot R = S \cdot (\mathsf{F}\, T)$$

  arises from hylo(cata)-fusion

- The **reflection** law holds:

$$(\![ins]\!) \quad = \quad id$$

Relational cata(ana)morphisms

Define

$$
\begin{array}{rcl}
(\!|R|\!) & = & [\![\, R, in^\circ \,]\!] \\
[\!(S)\!] & = & [\![\, in, S \,]\!]
\end{array}
$$

where $\mathsf{F}\,\mu\mathsf{F} \underset{in}{\overset{in^\circ}{\cong}} \mu\mathsf{F}$. For instance,

$$elems = (\!|ins|\!)$$

Relational cata(ana)morphisms

From

$$[\![\, S, R \,]\!]^\circ = [\![\, R^\circ, S^\circ \,]\!]$$

infer

$$
\begin{array}{rl}
 & [\!(S)\!] \\
= & [\![\, in, S \,]\!]^{\circ\circ} \\
= & [\![\, S^\circ, in^\circ \,]\!]^\circ \\
= & (\!|S^\circ|\!)^\circ
\end{array}
$$

(=ana is the converse of the cata of the converse)

Inductive coreflexives

Recall

$$[\![R, S]\!] \subseteq id \quad \Leftarrow \quad R \cdot S \subseteq id$$

which entails

$$(\![R]\!) \subseteq id \quad \Leftarrow \quad R \subseteq in$$

that is,

$$(\![in \cdot S]\!) \subseteq id \quad \Leftarrow \quad S \subseteq id$$

Example (on finite lists):

$$IsOrdered \stackrel{\mathrm{def}}{=} (\![in \cdot (id + ok)]\!)$$

Inductive coreflexives

where ok is the coreflexive induced by predicate

```
ok(a,x) == forall b in set elems x & a <= b
```

This leads to

$$\begin{array}{rcl} IsOrdered & = & [nil\ , cons \cdot ok] \cdot \\ & & (id + id \times IsOrdered) \cdot \\ & & [nil\ , cons]^{\circ} \\ & = & [nil\ , cons \cdot ok \cdot (id \times IsOrdered)] \cdot \\ & & [nil\ , cons]^{\circ} \end{array}$$

Inductive coreflexives

. . . and, finally, to

```
IsOrdered(l) ==
      if l = []
      then true
      else (forall b in set elems tl l & hd l <= b) and
           IsOrdered (tl l) ;
```

Exercise: calculate the above from $(\!| in \cdot (id + ok) |\!)$

VDM-SL data type $\texttt{map}\ A\ \texttt{to}\ B$

$$
\begin{tikzcd}
\texttt{map}\ A\ \texttt{to}\ C \arrow[r, "ins^{\circ}"] \arrow[d, "\{\!| R |\!\}"'] & 1 + (A \times C) \times \texttt{map}\ A\ \texttt{to}\ C \arrow[d, "id + id \times \langle id, \{\!| R |\!\} \rangle"] \\
B & 1 + (A \times C) \times B \arrow[l, "R"]
\end{tikzcd}
$$

leading to the following pointwise syntax:

VDM-SL data type $\texttt{map}\ A\ \texttt{to}\ B$

```
mhylo[@A,@B,@C] : (@A*@C*@B -> @B) * @B ->
                  map @A to @C ->
                  @B
mhylo(f,u)(s) ==
      if s={|->} then u
      else let a in set dom s,
               c = s(a),
               r = s <-: {a}
           in f(c,mhylo[@A,@B,@C](f,u)(r));
```

Hylos as unique solutions

A standard result of the relational calculus establishes the following condition for

$$\mu X.(R \cdot \mathsf{F}\, X \cdot S) \quad = \quad X$$

to be a **unique** solution:

- the "accessibility relation" of S is required to be inductive (cf. "well-founded" relations)
- This ensures **termination** insofar as the "size" of a sub-problem generated by S is strictly smaller than its source.
- One can perform **induction** over S.

Accessibility and membership

Accessibility relation for $\mathsf{F}\, A \xleftarrow{S} A$:

$$A \xleftarrow{\prec_S} A$$

$$\prec_S \stackrel{\text{def}}{=} \in_{\mathsf{F}} \cdot S$$

where $A \xleftarrow{\in_{\mathsf{F}}} \mathsf{F}\, A$ extends $A \xleftarrow{\in} \mathcal{P} A$ inductively over polynomial functors, as follows:

$$\begin{array}{rcl}
\in_{\mathcal{P}} & \stackrel{\text{def}}{=} & \in \\
\in_{C} & \stackrel{\text{def}}{=} & \perp \\
\in_{\lambda X.X} & \stackrel{\text{def}}{=} & id \\
\in_{\mathsf{F} \times \mathsf{G}} & \stackrel{\text{def}}{=} & (\in_{\mathsf{F}} \cdot \pi_1) \cup (\in_{\mathsf{G}} \cdot \pi_2) \\
\in_{\mathsf{F} + \mathsf{G}} & \stackrel{\text{def}}{=} & [\in_{\mathsf{F}}\ , \in_{\mathsf{G}}]
\end{array}$$

Example

Let $\mathsf{F}\,X = 1 + B \times X$. Then,

$$
\begin{array}{cl}
 & \in_{1+B\times X} \\
= & \quad \{\ \in \text{ for coproduct bifunctor } \} \\
 & [\in_1 \,, \in_{B\times X}] \\
= & \quad \{\ \in \text{ for constant and product (bi)functors } \} \\
 & [\bot \,, (\in_B \cdot \pi_1) \cup (\in_{\lambda X.X} \cdot \pi_2)] \\
= & \quad \{\ \in \text{ for constant and identity functor } \} \\
 & [\bot \,, (\bot \cdot \pi_1) \cup (id \cdot \pi_2)] \\
= & \quad \{\ \bot \text{ and } [R \,, S] = (R \cdot i_1^\circ) \cup (S \cdot i_2^\circ)\ \} \\
 & \pi_2 \cdot i_2^\circ
\end{array}
$$

Example (pointwise)

Then,

$$
\begin{array}{rcl}
\prec_S & = & \in_{1+B\times X} \cdot S \\
 & = & (\pi_2 \cdot i_2^\circ) \cdot S \\
 & = & \pi_2 \cdot (i_2^\circ \cdot S)
\end{array}
$$

meaning

$$
a' \prec_S a \quad \equiv \quad a' = \pi_2\ x \wedge (i_2\ x) S a
$$

For example, for $S = [nil \,, cons]^\circ$ on finite sequences, we get

Accessibility on finite sequences

$$
\begin{array}{rl}
 & \pi_2 \cdot (i_2^\circ \cdot [nil\ , cons]^\circ) \\
= & \pi_2 \cdot ([nil\ , cons] \cdot i_2)^\circ \\
= & \pi_2 \cdot cons^\circ
\end{array}
$$

and therefore

$$
\begin{array}{rl}
 & a' \prec_{[nil\ ,cons]^\circ} a \quad \equiv \quad a' = \pi_2(b, a') \wedge a = cons(b, a') \\
\equiv & a' \prec_{[nil\ ,cons]^\circ} cons(b, a') \\
\equiv & \mathsf{tl}\, a \prec_{[nil\ ,cons]^\circ} a
\end{array}
$$

Hylo factorization (2)

For such inductive S, we can factor $[\![\ R, S\ ]\!]$ in two components

$$
\begin{tikzcd}
\mu\mathsf{F} \arrow[r, phantom, "\cong"] \arrow[r, bend right, "in^\circ"'] & \mathsf{F}(\mu\mathsf{F}) \arrow[l, bend right, "in"']
\end{tikzcd}
$$

$$
\begin{tikzcd}
A \arrow[r, "S"] \arrow[d, "X_1"'] & \mathsf{F}\,A \arrow[d, "\mathsf{F}\,X_1"] \\
\mu\mathsf{F} \arrow[d, "X_2"'] & \mathsf{F}\,\mu\mathsf{F} \arrow[l, "in"'] \arrow[d, "\mathsf{F}\,X_2"] \\
B & \mathsf{F}\,B \arrow[l, "R"]
\end{tikzcd}
$$

$$X1 \quad = \quad in \cdot F X_1 \cdot S$$

Hylo-factorization Theorem

Using $(\!|\_|\!)$, $[\!(\_)\!]$ notation:

$$\mu X.(R \cdot \mathsf{F}\,X \cdot S) = (\!|R|\!) \cdot [\!(S)\!]$$

$$\begin{tikzcd}
A \arrow[r, "S"] \arrow[d, "{[\!(S)\!]}"'] & \mathsf{F}\,A \arrow[d, "{\mathsf{F}\,[\!(S)\!]}"] \\
\mu\mathsf{F} \arrow[d, "(\!|R|\!)"'] & \mathsf{F}\,\mu\mathsf{F} \arrow[l, "in"'] \arrow[d, "\mathsf{F}\,(\!|R|\!)"] \\
B & \mathsf{F}\,B \arrow[l, "R"]
\end{tikzcd}$$

Taking converses:

$$\mu X.(R \cdot \mathsf{F}\,X \cdot S^\circ) = (\!|R|\!) \cdot (\!|S|\!)^\circ$$

$$\begin{tikzcd}
A \arrow[d, "(\!|S|\!)^\circ"'] & \mathsf{F}\,A \arrow[l, "S"'] \arrow[d, "\mathsf{F}\,(\!|S|\!)^\circ"] \\
\mu\mathsf{F} \arrow[d, "(\!|R|\!)"'] & \mathsf{F}\,\mu\mathsf{F} \arrow[l, "in"'] \arrow[d, "\mathsf{F}\,(\!|R|\!)"] \\
B & \mathsf{F}\,B \arrow[l, "R"]
\end{tikzcd}$$

Entire /simple factorization if both R and S° are entire /simple (=S surjective /injective)

$$\begin{tikzcd}
A \arrow[d, "(\!|S|\!)^\circ"'] & \mathsf{F}\,A \arrow[l, "S"'] \arrow[d, "\mathsf{F}\,(\!|S|\!)^\circ"] \\
\mu\mathsf{F} \arrow[d, "(\!|R|\!)"'] & \mathsf{F}\,\mu\mathsf{F} \arrow[l, "in"'] \arrow[d, "\mathsf{F}\,(\!|R|\!)"] \\
B & \mathsf{F}\,B \arrow[l, "R"]
\end{tikzcd}$$

Virtual data-structuring

- Particular choice of F for sub-problem organization induces **intermediate** type $\mu\mathsf{F}$.

  This is made explicit by hylo-factorization.

- **Intermediate** data-structure saves the outcome of a "one go" **divide** step $(\!|S|\!)^\circ$ and passes it on to the **conquer** step $(\!|R|\!)$ for processing.

- In general, people **"fuse"** things very early in design, thus **virtualizing** this structure.

- Factorization helps in spec **understanding** and **classification**.

Final note on inductive relation $\prec$

Is such that the validity of a predicate ϕ can be proved by **structural induction** over it:

$$(\forall a.\ \phi\ a) \quad \Leftarrow \quad \underbrace{(\forall a.\ \phi\ a \Leftarrow (\forall c \prec a.\ \phi\ c))}_{\text{induction step}}$$

which corresponds to pointfree definition

$$\prec \backslash X \subseteq X \quad \Rightarrow \quad \top \subseteq X$$

where X generalizes ϕ such that $\phi\ a = aXb$, for some fixed b.