Appendix A

MFP-I/0203: Addenda to the Lectures Notes

A.1 Elements of the Fixpoint Calculus

A.1.1 Basic definitions

Definition 1 (Poset) *A poset $(A, \leq_A)$ is a set A equipped with a partial ordering $\leq_A$, that is, a relation $\leq_A \subseteq A \times A$ which is reflexive, transitive and antisymmetric.*
□

Definition 2 (Pre/post-fixpoints) *Let $A \xleftarrow{f} A$ be a (endo) function on poset $(A, \leq_A)$. Then*

- *every $a \in A$ such that*

$$a \quad \leq_A \; f\, a \tag{A.1}$$

is said to be a post-fixpoint *of f.*

- *every $a \in A$ such that*

$$a \quad \geq_A \; f\, a \tag{A.2}$$

is said to be a pre-fixpoint *of f.*

- *every $a \in A$ which is both a pre-fixpoint and a post-fixpoint of f is said to be a* fixpoint *of f and is such that*

$$a \quad = \quad f\, a \tag{A.3}$$

holds.

□

Examples:

- Given endofunction

$$\begin{array}{rcl} f : [0,10] & \to & [0,10] \\ x & \rightsquigarrow & 10-x \end{array}$$

  one very easily checks that 5 is a fixpoint of f, since $f\,5 = 10 - 5 = 5$.

- Let $R \subseteq P \times P$ be a relation on nonempty P in

$$x = R \cup R \circ x \tag{A.4}$$

  Define

$$f\,x = R \cup R \circ x \tag{A.5}$$

  on poset $(\mathcal{P}(P \times P), \subseteq)$. Then

  - $P \times P$ is an example of a pre-fixpoint of f ($P \times P$ is the largest relation in the poset).
  - $\emptyset$ and R are examples of post-fixpoints of f. In fact, $\emptyset \subseteq R$ and $R \subseteq R \cup R^2$.

Clearly, every fixpoint $a = f\,a$ can be regarded as a "solution" to equation

$$x = f\,x \tag{A.6}$$

But one can also regard this equation as a "recursive" definition of its fixpoints. For instance, recall equation (2.3)

$$x = 1 + \frac{x}{2}$$

The fact that 2 is a fixpoint of this equation can be rephrased to: *"$x = 1 + \frac{x}{2}$" is a recursive definition of number* 2.

However, the following equation

$$x = \frac{x^2 + 3}{4}$$

admits two solutions (fixpoints) 1 e 3. What are we "recursively defining" here? The 1 or the 3? Furthermore, equation

$$x = x$$

defines any object! By contrast, some equations don't have any solution at all. Think *e.g.* of

$$x = x + 1$$

in $\mathbb{N}$. So, in this case, our recursive equation defines... nothing!

A.1.2 Computing fixpoints

Definition 3 (Monotone functions) *A function* $B \xleftarrow{f} A$ *from poset* $(A, \leq_A)$ *to poset* $(B, \leq_B)$ *is said to be* monotone *iff*

$$\forall a, a' \in A : a \leq_A a' \Rightarrow (f\, a) \leq_B (f\, a')$$

holds.
□

Definition 4 (Ordering on functions) *Given two functions* $B \xleftarrow{f} A$ *and* $B \xleftarrow{g} A$ *from poset* $(A, \leq_A)$ *to poset* $(B, \leq_B)$ *define*

$$f \leq g \quad \stackrel{\text{def}}{=} \quad \forall a \in A : (f\, a) \leq_B (g\, a) \tag{A.7}$$

□

Theorem 1 (Lattice Fixpoints) *[Tarski 1955]*
Let

- $A \xleftarrow{f} A$ *be a monotone function on a* complete lattice $(A; \leq)$*;*
- P *be the set of all fixpoints of* f*,* i.e.

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

Then

- P *is non-empty and* $(P; \leq)$ *is a complete (sub)lattice.*
- *In particular, the least of all fixpoints* $(\bigwedge P)$ *and the greatest one* $(\bigvee P)$ *are as follows:*

$$\bigwedge P = \bigwedge \{x \mid x \geq f\, x\} \tag{A.8}$$
$$\bigvee P = \bigvee \{x \mid x \leq f\, x\} \tag{A.9}$$

We define:

$$\mu f \stackrel{\text{def}}{=} \bigwedge P \tag{A.10}$$
$$\nu f \stackrel{\text{def}}{=} \bigvee P \tag{A.11}$$

□

In the sequel we shall be focussing on *least* fixpoints.

A.2 2001.12.06 — Laws of the Fixpoint Calculus

Computation rule:

$$\mu f \quad = \quad f\,\mu f \tag{A.12}$$

Rolling rule:

$$\mu(g \cdot f) \quad = \quad g(\mu(f \cdot g)) \tag{A.13}$$

Square rule:

$$\mu f = \mu(f^2) \tag{A.14}$$

Monotonicity:

$$\mu f \leq \mu g \quad \Leftarrow \quad f \leq g \tag{A.15}$$

Induction rule:

$$\mu f \leq x \quad \Leftarrow \quad f\,x \leq x \tag{A.16}$$

A.2.1 Illustration

Let $f\,x = 1 + \frac{x}{2}$. Successive application of the computation rule (A.12) leads to:

$$\begin{aligned} \mu f &= 1 + \frac{\mu f}{2} \\ &= 1 + \frac{(1 + \frac{\mu f}{2})}{2} = 1 + \frac{1}{2} + \frac{\mu f}{4} \\ &\vdots \\ &= \sum_{i=1}^{n} \frac{1}{2^n} + \frac{\mu f}{2^{n+1}} \end{aligned}$$

In the limit ($n \to \infty$), we get $\frac{\mu f}{2^{n+1}} = 0$ and therefore $\mu f = \sum_{i=1}^{\infty} \frac{1}{2^n} = 2$.

The rolling rule (A.13) can be applied decomposing $f = g \cdot h$ for $h\,x = \frac{x}{2}$ and $g\,x = 1 + x$. Then

$$\begin{aligned} \mu f &= \mu(g \cdot h) = g(\mu(h \cdot g)) \\ &= 1 + \mu x.\frac{1 + x}{2} \end{aligned}$$

whereby $x = \frac{1+x}{2}$ has solution 1.

The aother rules enable us to reason inequationally. For instante, fact $1 + \frac{x}{2} \leq 2 + \frac{x}{2}$, for all x, and monotonicity (A.15) enables us to say that $\mu x.(1 + \frac{x}{2}) = 2$ is smaller than $\mu x.(2 + \frac{x}{2}) = 4$.

Similar intuition can be gathered from (A.5), providing evidence that $\mu f = R^+$ (transitive closure of R).

For instance (rolling rule), we can decompose f into $g \cdot h$ where $h\,x = R \cdot x$ and $g\,x = R \cup x$. Then

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

Further application of this rule will "factor out" R^2, R^3, *etc.*, leaving a "smaller and smaller" fixpoint to be calculated. In the limit, one gets $\mu f = \bigcup_{j=1}^{\infty} R^j = R^+$.

A.2.2 Inductive datatypes "are" fixpoints

Recall

$$X = \underbrace{1 + A \times X}_{\mathsf{F}\,X} \tag{A.17}$$

- The "=" symbol in equation (A.17) should be understood as "$\cong$"
- F should be understood as a *functor*
- So any solution X_0 to the equation should carry along an algebra in and its inverse out thus providing evidence of the required *isomorphism*:

$$X_0 \underset{in}{\overset{out}{\cong}} 1 + A \times X_0$$

For instance,

$$A^\star \underset{in=[\,\underline{]},cons\,]}{\overset{out}{\cong}} 1 + A \times A^\star$$

where out is the obvious inverse of in.

- The $\leq$-ordering corresponds to right-invertibility:

$$A \underset{f}{\overset{r}{\leq}} B \quad \text{that is} \quad \{ \; f \cdot r = id_A \tag{A.18}$$

In general: For F a polynomial functor, equation $X \cong \mathsf{F}\, X$

- admits a standard solution — its **least fixpoint** solution

$$\mu\mathsf{F} \underset{in}{\overset{out}{\cong}} \mathsf{F}\,\mu\mathsf{F} \tag{A.19}$$

  Example:

$$\mu X.1 + A \times X \quad = \quad A^\star$$

  — where $\mu X.\mathsf{F}\, X$ abbreviates $\mu(\lambda X.\mathsf{F}\, X)$.

- $\mu\mathsf{F}$ is *initial* among all other F-structures — that is to say, for a given $(A,\ A \xleftarrow{\alpha} \mathsf{F}\, A\ )$, arrow k in

$$\begin{array}{ccc} \mu\mathsf{F} & \xleftarrow{in} & \mathsf{F}\,\mu\mathsf{F} \\ {\scriptstyle k}\downarrow & & \downarrow{\scriptstyle \mathsf{F}\,k} \\ A & \xleftarrow[\alpha]{} & \mathsf{F}\, A \end{array}$$

  is unique, recall **universal property**:

$$k = (\!|\alpha|\!) \quad \Leftrightarrow \quad k \cdot in = \alpha \cdot \mathsf{F}\, k$$

A.2.3 Application of the Fixpoint Calculus to datatypes

Computation rule:

$$\mu\mathsf{F} \underset{in}{\overset{out}{\cong}} \mathsf{F}\,(\mu\mathsf{F}) \tag{A.20}$$

cf. (A.19).

Rolling rule:

$$\mu(\mathsf{G}\cdot\mathsf{F}) \underset{in_{\mathsf{G}\cdot\mathsf{F}}\cdot\mathsf{G}(\mathsf{F}\ in_{\mathsf{G}\cdot\mathsf{F}})}{\overset{(\!|\mathsf{G}\ in_{\mu(\mathsf{F}\cdot\mathsf{G})}|\!)}{\cong}} \mathsf{G}(\mu(\mathsf{F}\cdot\mathsf{G})) \tag{A.21}$$

cf.

$$\begin{array}{ccc} \mu(\mathsf{G}\cdot\mathsf{F}) & \xleftarrow{in_{\mu(\mathsf{G}\cdot\mathsf{F})}} & (\mathsf{G}\cdot\mathsf{F})(\mu(\mathsf{G}\cdot\mathsf{F})) \\ {\scriptstyle (\!|g|\!)}\downarrow & & \downarrow{\scriptstyle (\mathsf{G}\cdot\mathsf{F})(\!|g|\!)} \\ \mathsf{G}(\mu(\mathsf{F}\cdot\mathsf{G})) & \underset{g=\mathsf{G}\ in_{\mu(\mathsf{F}\cdot\mathsf{G})}}{\xleftarrow{\qquad}} & (\mathsf{G}\cdot\mathsf{F})(\mathsf{G}(\mu(\mathsf{F}\cdot\mathsf{G}))) \end{array}$$

Example: Let

$$\begin{aligned} \mathsf{F}\,X &= 1 + X \\ \mathsf{G}\,X &= A \times X \end{aligned}$$

Then

$$\begin{aligned} (\mathsf{F}\cdot\mathsf{G})X &= 1 + A \times X \\ (\mathsf{G}\cdot\mathsf{F})X &= A \times (1 + X) \cong A + A \times X \end{aligned}$$

where $(\mathsf{G}\cdot\mathsf{F})$ involves natural isomorphism $k = \langle [\ id, \pi_1\ ], (! + \pi_2)\rangle$. Then

$$\begin{aligned} \mu(\mathsf{F}\cdot\mathsf{G}) &= A^\star \\ \mu(\mathsf{G}\cdot\mathsf{F}) &= A^+ \end{aligned}$$

The rolling rule will state the obvious fact that

$$A^+ \cong A \times A^\star$$

holds, that is

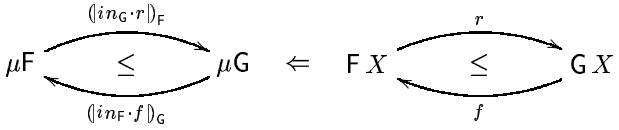
$$\begin{array}{ccccc} \mu(\mathsf{G}\cdot\mathsf{F}) = A^+ & \xleftarrow{\qquad in_{A^+} \qquad} & A \times (1 + A^+) & \xleftarrow[k]{} & A + A \times A^+ \\ {\scriptstyle f}\downarrow & & \downarrow{\scriptstyle id\times(id+f)} & & \downarrow{\scriptstyle id+id\times f} \\ A \times A^\star & \xleftarrow{id\times in_{A^\star}} & A \times (1 + A \times A^\star) & \xleftarrow{k} & A + A \times (A \times A^\star) \end{array} \tag{A.22}$$

for $k = \langle [\ id, \pi_1\ ], (! + \pi_2)\rangle$.

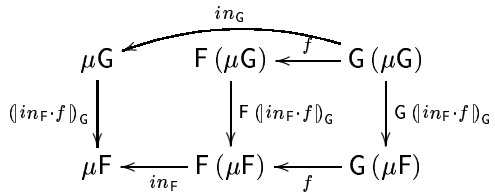
Exercise 1.1 Concerning (A.22), show that $f = \langle f_1 , f_2\rangle$ where f_1 is the "head" function and f_2 is the "tail" function on A^+.

□

Monotonicity:



cf. diagram



Let us see an example of application, whereby possibly empty sequences are represented by non-empty ones — $\mu\mathsf{F} = A^\star$ and $\mu\mathsf{G} = A^+$, for $1 \leq A$:

$$A^\star \underset{[\,[\,],cons\,]}{\overset{\cong}{\longleftarrow}} 1 + A \times A^\star$$

$$A^+ \underset{[\,sing,cons\,]}{\overset{\cong}{\longleftarrow}} A + A \times A^+$$

where $sing\, a = [a]$.

A.3 Mutual recursion

Consider mutually-dependent f and g as follows:

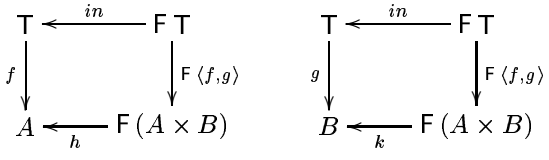
```
f: nat -> nat
f(n) == if n = 0 then 0 else g(n - 1);

g: nat -> nat
g(n) == if n = 0 then 1 else f(n - 1) + g(n - 1);
```

How we reason about mutually-dependent functions?

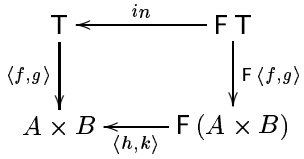
The situation is handled by the so-called *mutual-recursion law*, also called "Fokkinga law":

$$\begin{array}{l} f \cdot in = h \cdot \mathsf{F}\,\langle f, g\rangle \\ \qquad\wedge \\ g \cdot in = k \cdot \mathsf{F}\,\langle f, g\rangle \end{array} \quad \Rightarrow \quad \langle f, g\rangle = (\!|\langle h, k\rangle|\!) \tag{A.23}$$

In terms of diagrams: from



we get



Proof:

$$
\begin{aligned}
& \langle f, g\rangle \cdot in = \langle h, k\rangle \cdot \mathsf{F}\,\langle f, g\rangle \\
\equiv & \quad \{ \text{ by } \times\text{-fusion (1.24)}\} \\
& \langle f, g\rangle \cdot in = \langle h \cdot \mathsf{F}\,\langle f, g\rangle, k \cdot \mathsf{F}\,\langle f, g\rangle\rangle \\
\equiv & \quad \{ \text{ by hypothesis}\} \\
& \langle f, g\rangle \cdot in = \langle f \cdot in, g \cdot in\rangle \\
\equiv & \quad \{ \text{ by (reverse) } \times\text{-fusion (1.24)}\} \\
& \langle f, g\rangle \cdot in = \langle f, g\rangle \cdot in \\
\equiv & \quad \{ \text{ equality is reflexive}\} \\
& TRUE
\end{aligned}
$$

Applying this to the above pair of f and g:

$$
\begin{aligned}
f \cdot [\, \underline{0}, suc\,] &= [\, \underline{0}, g\,] \\
g \cdot [\, \underline{0}, suc\,] &= [\, \underline{1}, + \cdot \langle f, g\rangle\,]
\end{aligned}
$$

The mutual dependence can be made more explicit by forcing

$$
\begin{aligned}
f \cdot [\, \underline{0}, suc\,] &= [\, \underline{0}, \pi_2 \cdot \langle f, g\rangle\,] \\
g \cdot [\, \underline{0}, suc\,] &= [\, \underline{1}, + \cdot \langle f, g\rangle\,]
\end{aligned}
$$

The underlying inductive type is

$$
\mathbb{N}_0 \;\cong\; \underbrace{1 + \mathbb{N}_0}_{\mathsf{F}\,\mathbb{N}_0} \tag{A.24}
$$

which is such that $\mathsf{F}\,f = id + f$. So we can write

$$
\begin{aligned}
f \cdot in &= [\, \underline{0}, \pi_2\,] \cdot \mathsf{F}\,\langle f, g\rangle \\
g \cdot in &= [\, \underline{1}, +\,] \cdot \mathsf{F}\,\langle f, g\rangle
\end{aligned}
$$

So we identify $h = [\ \underline{0}, \pi_2\ ]$ and $k = [\ \underline{1}, +\ ]$ therefore obtaining

$$\begin{array}{ll} & \langle f, g\rangle \\ = & \quad \{\text{ Fokkinga law}\} \\ & (\!|\langle [\ \underline{0}, \pi_2\ ], [\ \underline{1}, +\ ]\rangle|\!) \\ = & \quad \{\text{ exchange law}\} \\ & (\!|[\ \langle \underline{0}, \underline{1}\rangle, \langle \pi_2, +\rangle\ ]|\!) \end{array}$$

which is easily converted into VDM-SL as follows:

```
fg: nat -> nat
fg(n) == if n = 0 then mk_(0,1)
         else let p=fg(n - 1)
              in mk_(p.#2,p.#1 + p.#2);
```

A.3.1 Example

Checking a list-invariant which ensures that a (non-empty) list is ordered:

$$\begin{array}{l} ordered:\ A^+ \longrightarrow 2 \\ ordered\,[a] = TRUE \\ ordered\,(cons(a, l)) = a > (Max\,l) \wedge (ordered\,l) \end{array}$$

Assuming *singl* $a = [a]$ we can depict $ordered$ as follows:

$$\begin{array}{ccc} A^+ & \xleftarrow{[\ singl, cons\ ]} & A + A \times A^+ \\ {\scriptstyle ordered}\downarrow & & \downarrow{\scriptstyle id+id\times\langle Max, ordered\rangle} \\ 2 & \xleftarrow[[\ \underline{TRUE}, \alpha\ ]]{} & A + A \times (A \times 2) \end{array}$$

where

$$\alpha(a, (m, b)) \stackrel{\text{def}}{=} a > m \wedge b$$

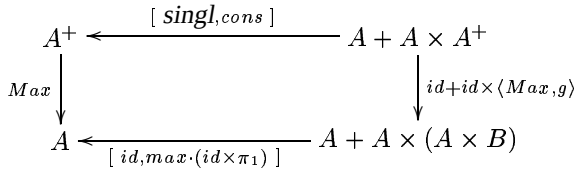
and where

$$Max = (\!|[\ id, max\ ]|\!)$$

cf.

$$\begin{array}{ccc} A^+ & \xleftarrow{[\ singl, cons\ ]} & A + A \times A^+ \\ {\scriptstyle Max}\downarrow & & \downarrow{\scriptstyle id+id\times Max} \\ A & \xleftarrow[[\ id, max\ ]]{} & A + A \times A \end{array}$$

It is easy to check that the equation implicit in this diagram is the same as the one implicit in



for any $A^+ \xrightarrow{g} B$. For $B = 2$ and $g = ordered$ we are in position to apply Fokkinga's law and to obtain:

$$\begin{aligned}\langle Max, ordered\rangle &= (\!|\langle [\, id, max \cdot (id \times \pi_1)\,], [\, \underline{TRUE}, \alpha\,]\rangle|\!)\\ &= \quad \{\ \textit{exchange law}\ (1.47)\}\\ &\quad (\!|[\, \langle id, \underline{TRUE}\rangle, \langle max \cdot (id \times \pi_1), \alpha\rangle\,]|\!)\end{aligned}$$

Of course, $ordered = \pi_2 \cdot \langle Max, ordered\rangle$. Calling aux to the above synthesized catamorphism, we end up with the following realization of $ordered$:

$$\begin{array}{lcll} ordered\, l & = & let & (a,b) = aux\, l\\ & & in & b\end{array}$$

where

$$\begin{array}{l} aux : A^+ \longrightarrow A \times 2\\ ordered\,[a] = (a, TRUE)\\ ordered\,(cons(a,l)) = \begin{array}{lll} let & (m,b) = aux\, l\\ in & (max(a,m), (a > m \wedge b))\end{array}\end{array}$$

A.3.2 "Banana-split": a corollary of the mutual-recursion law

Let $h = i \cdot \mathsf{F}\,\pi_1$ and $k = j \cdot \mathsf{F}\,\pi_2$ in (A.23). Then

$$\begin{aligned} & f \cdot in = (i \cdot \mathsf{F}\,\pi_1) \cdot \mathsf{F}\,\langle f, g\rangle\\ \equiv & \quad \{\ \text{composition is associative and } \mathsf{F} \text{ is a functor}\}\\ & f \cdot in = i \cdot \mathsf{F}\,(\pi_1 \cdot \langle f, g\rangle)\\ \equiv & \quad \{\ \text{by } \times\text{-cancellation } (1.20)\}\\ & f \cdot in = i \cdot \mathsf{F}\, f\\ \equiv & \quad \{\ \text{by cata-cancellation}\}\\ & f = (\!|i|\!)\end{aligned}$$

Similarly, from $k = j \cdot \mathsf{F}\,\pi_2$ we get

$$g = (\!|j|\!)$$

Then, from (A.23), we get

$$\langle (\!|i|\!), (\!|j|\!) \rangle = (\!| \langle i \cdot \mathsf{F}\,\pi_1, j \cdot \mathsf{F}\,\pi_2 \rangle |\!)$$

that is

$$\langle (\!|i|\!), (\!|j|\!) \rangle = (\!| (i \times j) \cdot \langle \mathsf{F}\,\pi_1, \mathsf{F}\,\pi_2 \rangle |\!) \tag{A.25}$$

by (reverse) $\times$-absorption (1.25).

This law provides us with a very useful tool for "parallel loop" inter-combination: "loops" $(\!|i|\!)$ and $(\!|j|\!)$ are fused together into a single "loop" $(\!|(i \times j) \cdot \langle \mathsf{F}\,\pi_1, \mathsf{F}\,\pi_2 \rangle|\!)$. The need for this kind of calculation arises very often. Consider, for instance, the function which computes the average of a non-empty list of natural numbers:

$$average \stackrel{\text{def}}{=} (/) \cdot \langle sum, length \rangle$$

Both sum and $length$ are $\mathbb{N}^+$ catamorphisms:

$$suma = (\!| [\, id, + \,] |\!)$$
$$length = (\!| [\, \underline{1}, suc \cdot \pi_2 \,] |\!)$$

Function $average$ will do two independent traversals of the argument list before division $(/)$ takes place. Banana-split fuses such two traversals into a single one, thus leading to a function which: (a) runs twice as fast (b) can be converted into a *while loop* by introduction of accummulation parameters (such as seen above).

Exercise 1.2 Apply the banana-split law to the following definition of the $unzip$ function:

$$unzip \stackrel{\text{def}}{=} \langle {\pi_1}^\star, {\pi_2}^\star \rangle$$

Extend $unzip$ to binary trees and repeat the exercise.
□

A.4 Paramorphisms

Consider the standard definition of the factorial function (in VDM-SL notation):

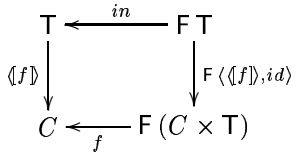
```
fac : nat -> nat
fac(n) == if n = 0 then 1 else fac(n-1) * n
```

The pattern of recursion of this function — usually known as *primitive recursion* — is somewhat more elaborate than that of a catamorphism over $\mathbb{N}$.

Note than it can be captured by the following diagram:

$$\begin{array}{ccc} \mathbb{N} & \xleftarrow{[\,\underline{0}, suc\,]} & 1 + \mathbb{N} \\ {\scriptstyle fac}\downarrow & & \downarrow{\scriptstyle id + \langle fac, id \rangle} \\ \mathbb{N} & \xleftarrow[[\,\underline{1}, mul \cdot (id \times suc)\,]]{} & 1 + (\mathbb{N} \times \mathbb{N}) \end{array}$$

Function fac is a particular instance of a so-called *paramorphism*. In general, a paramorphism of some f relative to functor F, is the unique morphism $(\!|f|\!)$ which is such that



that is, we have the following universal property:

$$h = (\!|f|\!) \quad \equiv \quad h \cdot in = f \cdot \mathsf{F}\,\langle h, id\rangle$$

A.4.1 Examples of paramorphisms

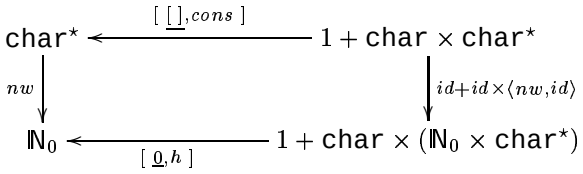
From above we can express the factorial function as a paramorphism:

$$fac = (\!|[\,\underline{1}, mul \cdot (id \times suc)\,]|\!)$$

A less straightforward example is that of a function nw — *cf.* `wc -w` in LINUX— counting the number of words in text (`seq of char`):

```
nw : seq of char -> nat
nw(s) == if s = [] then 0
         else if not sep(hd s) and sepahead(tl s)
              then nw(tl s) + 1 else nw(tl s) ;

sepahead: seq of char -> bool
sepahead(s) == (s = []) or sep(hd s) ;

sep : char -> bool
sep(c) == c  = ' ' or c  = '\n' or c  = '\t' ;
```

This is list-paramorphism



where

```
h : char * (nat * seq of char) -> nat
h(c,mk_(i,s)) == if not sep(c) and sepahead(s) then i + 1 else i ;
```

A.4.2 Properties of paramorphisms

1. Clearly, every catamorphism can be expressed by a paramorphism:

$$(\!|f|\!) = (\![f \cdot \mathsf{F}\,\pi_1]\!) \tag{A.26}$$

Proof:

$$\begin{array}{ll} & h = (\![f \cdot \mathsf{F}\,\pi_1]\!) \equiv h \cdot in = f \cdot \mathsf{F}\,\pi_1 \cdot \mathsf{F}\,\langle h, id\rangle \\ = & \quad \{ \text{ functor versus composition (2.45),} \times\text{-cancellation} \} \\ & h = (\![f \cdot \mathsf{F}\,\pi_1]\!) \equiv h \cdot in = f \cdot \mathsf{F}\,h \\ = & \quad \{ \text{ cata-universal} \} \\ & (\!|f|\!) = (\![f \cdot \mathsf{F}\,\pi_1]\!) \end{array}$$

2. Conversely, every paramorphism can be expressed (indirectly) in terms of a catamorphism:

$$(\![h]\!) = \pi_1 \cdot (\!|\langle h, in \cdot \mathsf{F}\,\pi_2\rangle|\!) \tag{A.27}$$

Proof: let g be id in the mutual-recursion law, leading to $f = (\![h]\!)$. Then the equation for $g = id$ is

$$id \cdot in = k \cdot \mathsf{F}\,\langle f, id\rangle$$

and this is satisfied for $k = in \cdot \mathsf{F}\,\pi_2$.

So $\langle f, g\rangle = (\!|\langle h, in \cdot \mathsf{F}\,\pi_2\rangle|\!)$ and $f = (\![h]\!) = \pi_1 \cdot (\!|\langle h, in \cdot \mathsf{F}\,\pi_2\rangle|\!)$.

3. PARA-REFLECTION:

$$id = (\![in \cdot \mathsf{F}\,\pi_1]\!) \tag{A.28}$$

By cata-reflection (2.58) this can be regarded as an instance of (A.26) above.

4. PARA-FUSION:

$$h \cdot (\![f]\!) = (\![g]\!) \quad \Leftarrow \quad h \cdot f = g \cdot \mathsf{F}(h \times id) \tag{A.29}$$

Example of aplication

By (A.27) the factorial function can be expressed by the projection of a catamorphism:

$$\begin{array}{ll} & fac = \pi_1 \cdot (\!|\langle [\,\underline{1}, mul \cdot (id \times suc)\,], in \cdot (id + \pi_2)\rangle|\!) \\ \equiv & \quad \{ +\text{-absorption (1.41)} \} \\ & fac = \pi_1 \cdot (\!|\langle [\,\underline{1}, mul \cdot (id \times suc)\,], [\,\underline{0}, suc \cdot \pi_2\,]\rangle|\!) \\ = & \quad \{ \textit{exchange law} \text{ (1.47)} \} \\ & fac = \pi_1 \cdot (\!|[\,\langle \underline{1}, \underline{0}\rangle, \langle mul \cdot (id \times suc), suc \cdot \pi_2\rangle\,]|\!) \end{array}$$

This will lead to the following VDM-SL:

```
fac : nat -> nat
fac(n)==facaux(n).#1;

facaux: nat -> nat*nat
facaux(n) == if n=0 then mk_(1,0)
             else let p = facaux(n-1),
                      a = p.#1,
                      b = p.#2
                  in mk_(a * (b + 1), b + 1);
```