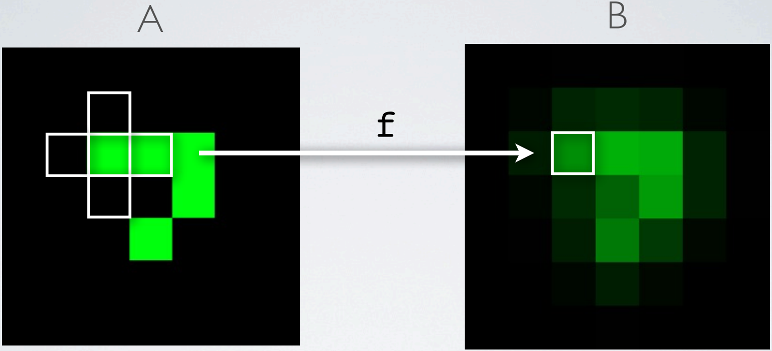
Complexity bounds from abstract categorical models (of containers)

Dominic Orchard
Imperial College London

April 11, 2015

Stencil Computations




```
for (i=0; i<N; i++)
    for (j=0; j<M; j++)
        B[i][j] = f(A[i][j],   A[i-1][j], A[i+1][j],
                    A[i][j-1], A[i][j+1]);
```

(Explicit) complexity differences

```
for i = 1..n
 for j = 1 .. m
   B(i, j) = f(A, i, j)
for i = 1 .. n
 for j = 1 .. m
   C(i, j) = g(B, i, j)
```

$$\in \mathcal{O}([f]nm + [g]nm)$$

vs

```
for i = 1..n
 for j = 1..m
   for u = 1..n
    for v = 1 .. m
      B(u, v) = f(A, u, v)
   C(i, j) = g(B, i, j)
```

$$\in \mathcal{O}([f][g](nm)^2)$$

Complexity bounds and optimisations

Hypothesis

The axioms of useful categorical structures imply program optimisations.

i.e., *orient* axiom $f \equiv g$ as rewrite $f \rightsquigarrow g$ which is guaranteed to not make the program asymptotically slower.

Definition

Given two programs f and g which are equivalent ($f \equiv g$) then the rewrite $f \rightsquigarrow g$ is an optimisation iff, for input size n:

$$[g]_n \in \mathcal{O}([f]_n)$$

(Endo)functors

- Model element-wise (point-wise) data structure traversals
- Object mapping $\mathrm{D} : \mathbb{C} \to \mathbb{C}$ and morphism mapping:

$$\frac{f : A \to B}{\mathrm{D}f : \mathrm{D}A \to \mathrm{D}B}$$

- with two axioms:

$$[\text{F1}]\ \ \mathrm{D}id_A \equiv id_{\mathrm{D}A} \qquad\qquad [\text{F2}]\ \ \mathrm{D}(g \circ f) \equiv \mathrm{D}g \circ \mathrm{D}f$$

Example (Lists)

Object mapping is data type $[\,] : * \to *$ and morphism mapping $map : \forall a, b.(a \to b) \to ([a] \to [b])$.

Finite containers

A data type D with only strictly positive occurences of A in DA.
Comes equipped with a natural transformation:

$$\mathsf{size}_A : \mathrm{D}A \to \mathbb{N}$$

Naturality means:

$$\begin{tikzcd}
\mathrm{D}A \arrow[r, "\mathsf{size}_A"] \arrow[d, "\mathrm{D}f"'] & \mathbb{N} \\
\mathrm{D}B \arrow[ur, "\mathsf{size}_B"'] &
\end{tikzcd}$$

Useful: **functor lifting produces a size-preserving function**

Implicit complexity of (container) functors

Q: For some f, what is the complexity of $\mathrm{D}f$?
A: $[\mathrm{D}f]_n \in \Omega(n[f]_1)$.

Proof.

- Size naturality, $\mathrm{size}_B \circ \mathrm{D}f = \mathrm{size}_A$, means $|inp.| = |outp.| = n$.

 *apply f to one element and copy n times* $\therefore [\mathrm{D}f]_n \in \Omega(n + [f]_1)$

- $[F1]$ $\mathrm{D}id_A \equiv id_{\mathrm{D}A}$, thus f must be applied to each element

 *define* $\mathrm{D}f = id$ **or** *if* $f = id$ *return input otherwise do above*

- Parametricity [see, *e.g.*, Reynolds]
  $\forall a, b,\ f : a \to b$ then $\mathrm{D}f : \mathrm{D}a \to \mathrm{D}b$, therefore $f \neq id$ is undecideable (due to infinite domains)

 ...

A slight refinement...

$[\mathrm{D}f]_n \in \Omega(n[f]_1)$.

Consider:

$$map\ (map\ (*2))\ [[1,2,3],[4,5],[6,7]] = [[2,4,6],[8,10],[12,14]]$$

Use a notion of *structure size*, with bounds:

- $n[\Omega(m)]$ is a structure of size n with elements at least size m

Proposition

For any discretely finite container D*, the morphism mapping operation has lower bound complexity:*

$$[\mathrm{D}f]_{n[\Omega(m)]} \in \Omega(n[f]_m)$$

Constant factors and suggested optimisations

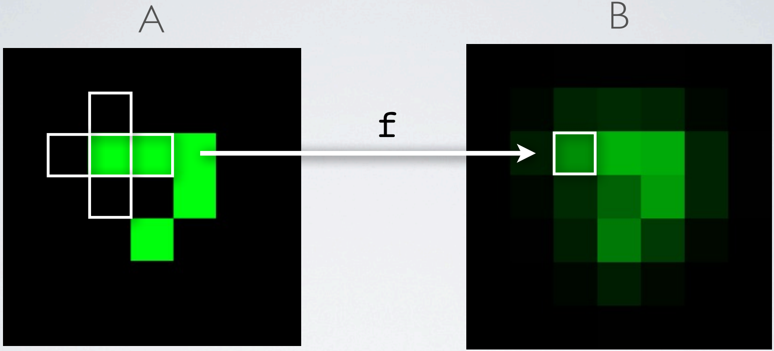
A common operation $\uparrow_A : A \to \mathrm{D}A$ *promotion*, natural in A:

$$\begin{tikzcd} A \arrow[r, "f"] \arrow[d, "\uparrow_A"'] & B \arrow[d, "\uparrow_B"] \\ \mathrm{D}A \arrow[r, "\mathrm{D}f"'] & \mathrm{D}B \end{tikzcd}$$

- *Parametricity*: for some $x : A$ then $\exists m \;.\; \mathrm{size}_A(\uparrow_A \; x) = m$.
- Thus, $[\uparrow_A]_n \in \mathcal{O}(1)$.

- $\therefore [\uparrow_B \circ f]_n \in \mathcal{O}([f]_n)$ and $[\mathrm{D}f \circ \uparrow_A]_n \in \Omega(m[f]_n) \in \Omega([f]_n)$
- Since m is a constant, no asymptotic improvment.
- But suggestion that $(\mathrm{D}f \circ \uparrow_A) \rightsquigarrow (\uparrow_B \circ f)$.

Stencil Computations



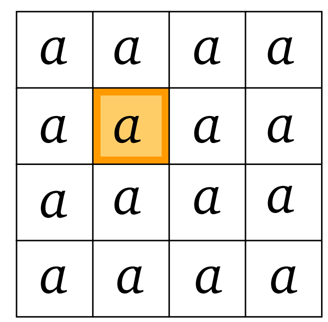

```
for (i=0; i<N; i++)
    for (j=0; j<M; j++)
        B[i][j] = f(A[i][j],   A[i-1][j], A[i+1][j],
                   A[i][j-1], A[i][j+1]);
```

Comonads - context-wise application

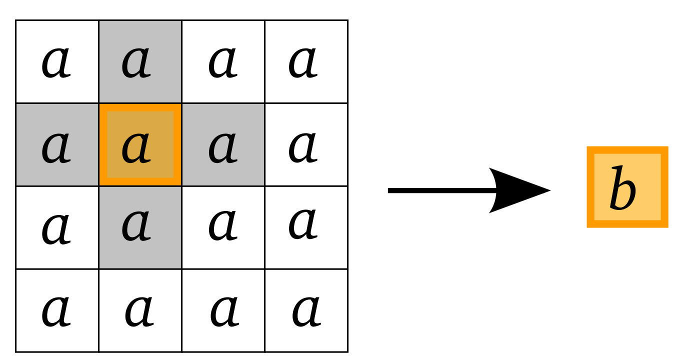
$$\textbf{functor}\,\frac{f : A \to B}{\mathsf{D}f : \mathsf{D}A \to \mathsf{D}B} \qquad \textbf{comonad}\,\frac{g : \mathsf{D}A \to B}{g^{\dagger} : \mathsf{D}A \to \mathsf{D}B}$$

Example comonad: Array

Array is an array with a *cursor*



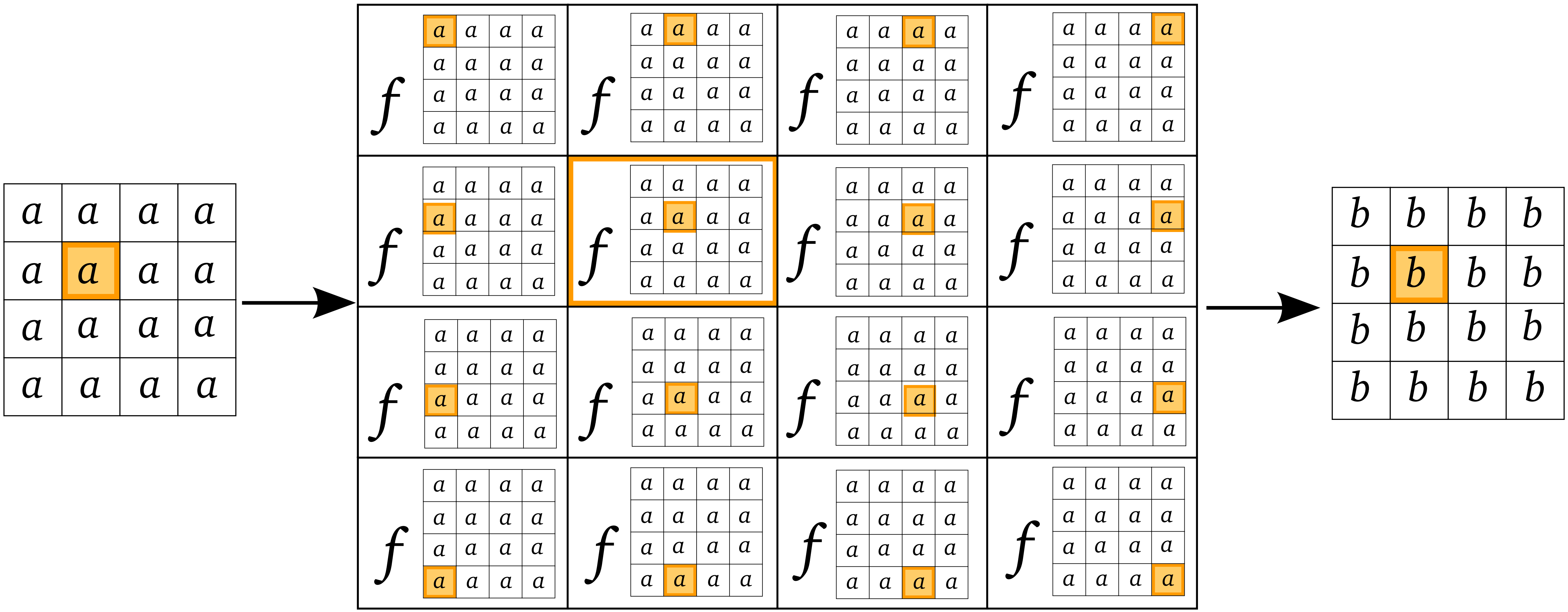
$$f : \mathbf{Array}\, a \rightarrow b$$



[see "Ypnos: Declarative, Parallel Structured Grid Programming", Orchard, Bolingbroke, Mycroft'10]

Example comonad: Array

$$(\_)^{\dagger} : (\mathbf{Array}\, a \rightarrow b) \rightarrow (\mathbf{Array}\, a \rightarrow \mathbf{Array}\, b)$$



Generalised-map on arrays (e.g. convolution)

Comonads

(Co)unit

$$\epsilon : Da \rightarrow a$$

Extract the value at the "current context"

$$\epsilon : \mathbf{Array}\, a \rightarrow a$$

$$\begin{array}{|c|c|c|c|}\hline a & a & a & a \\ \hline a & \boxed{a} & a & a \\ \hline a & a & a & a \\ \hline a & a & a & a \\ \hline \end{array} \longrightarrow \boxed{a}$$

Comonads - context-wise application

- Provides a model for *gathers*/*context-dependent* traversals
- Comprises object mapping $\mathsf{D} : \mathbb{C} \to \mathbb{C}$ and *extension*:

$$\frac{g : \mathsf{D}A \to B}{g^\dagger : \mathsf{D}A \to \mathsf{D}B}$$

- *extract* operation $\varepsilon_A : \mathsf{D}A \to A$ (current context)
- with three axioms:

$$[\text{C1}]\ \ \varepsilon^\dagger \equiv id \qquad [\text{C2}]\ \ \varepsilon \circ f^\dagger \equiv f \qquad [\text{C3}]\ \ g^\dagger \circ f^\dagger \equiv (g \circ f^\dagger)^\dagger$$

Axiom [C3], associativity

$$\dfrac{\dfrac{f : \mathsf{D}A \to B}{f^\dagger : \mathsf{D}A \to \mathsf{D}B} \qquad \dfrac{g : \mathsf{D}B \to C}{g^\dagger : \mathsf{D}B \to \mathsf{D}C}}{g^\dagger \circ f^\dagger : \mathsf{D}A \to \mathsf{D}C} \equiv \dfrac{\dfrac{\dfrac{f : \mathsf{D}A \to B}{f^\dagger : \mathsf{D}A \to \mathsf{D}B} \qquad g : \mathsf{D}B \to C}{g \circ f^\dagger : \mathsf{D}A \to C}}{(g \circ f^\dagger)^\dagger : \mathsf{D}A \to \mathsf{D}C}$$

Compare:

```
for i = 1..n
 for j = 1 .. m
   B(i, j) = f (A, i, j)
for i = 1 .. n
 for j = 1 .. m
   C(i, j) = g(B, i, j)
```

vs

```
for i = 1..n
 for j = 1..m
   for u = 1..n
    for v = 1 .. m
      B(u, v) = f (A, u, v)
   C(i, j) = g (B, i, j)
```

Q: Is $(g \circ f^\dagger)^\dagger \rightsquigarrow g^\dagger \circ f^\dagger$ always asymptotically better?

Implicit complexity of (container) comonads

Q: For some f, what is the complexity of $f^\dagger$?
A: $[f^\dagger]_n \in \mathcal{O}(P_n + nQ_n[f]_n)$

(P_n to reach elements, Q_n extra times to apply f)

Proof.

- Comonads are size/shape preserving[1]: $\mathsf{size}_B \circ f^\dagger = \mathsf{size}_A$ means $|inp.| = |outp.| = n$.

 *apply f to one element and copy n times* $\therefore [\mathrm{D}f]_n \in \mathcal{O}(P_1 + [f]_1 + n)$

- $[C1]\ \varepsilon_A^\dagger \equiv id_{\mathrm{D}A}$, thus f must be applied to each element

 *let* $f^\dagger = id$ **or** *if* $f = id$ *return input otherwise do above*

$\square$

[1] *A Notation for Comonads* (Orchard, Mycroft 2012)

Implicit complexity of (container) comonads

Q: For some f, what is the complexity of $f^\dagger$?

A: $[f^\dagger]_n \in \mathcal{O}(P_n + nQ_n[f]_n)$.

Proof.

- Parametricity - $\forall a, b,\ f : \mathrm{D}a \to b$ then $f^\dagger : \mathrm{D}a \to \mathrm{D}b$ and therefore $f \neq id$ is undecideable

Pass (asympotitcally) larger $x : \mathrm{D}A$ *to* $f : \mathrm{D}A \to B$ *at each context.*

- $[C2]$ $\varepsilon \circ f^\dagger = f$ therefore $\varepsilon \circ \mathrm{size}^\dagger_A = \mathrm{size}_A$. $\therefore$ size preserved at current context.

Pass (asympotitcally) larger $\mathrm{D}A$ *at non current contexts.*

- By $[C3]$, $(\mathrm{sum} \circ \mathrm{size}^\dagger_A)^\dagger = \mathrm{sum}^\dagger \circ \mathrm{size}^\dagger_A$. Therefore, extension size preserving at all contexts

...

Axiom [C3], associativity

$$\dfrac{\dfrac{f : \mathsf{D}A \to B}{f^\dagger : \mathsf{D}A \to \mathsf{D}B} \qquad \dfrac{g : \mathsf{D}B \to C}{g^\dagger : \mathsf{D}B \to \mathsf{D}C}}{g^\dagger \circ f^\dagger : \mathsf{D}A \to \mathsf{D}C} \equiv \dfrac{\dfrac{\dfrac{f : \mathsf{D}A \to B}{f^\dagger : \mathsf{D}A \to \mathsf{D}B} \quad g : \mathsf{D}B \to C}{g \circ f^\dagger : \mathsf{D}A \to C}}{(g \circ f^\dagger)^\dagger : \mathsf{D}A \to \mathsf{D}C}$$

Proposition

Axiom [C3] can be oriented as $(g \circ f^\dagger)^\dagger \rightsquigarrow g^\dagger \circ f^\dagger$ guaranteeing an asymptotic improvement.

Proof.

$$\begin{aligned}
[g^\dagger \circ f^\dagger]_n &\in \mathcal{O}(P_n + nQ_n[g]_n + nQ_n[f]_n) \\
[(g \circ f^\dagger)^\dagger]_n &\in \mathcal{O}(P_n + nQ_n([g]_n + P_n + nQ_n[f]_n)) \\
&\in \mathcal{O}(P_n + nQ_n[g]_n + (nQ_n)^2[f]_n + nQ_nP_n)
\end{aligned}$$

$\square$

Conclusions & further work

- From axioms and parametricity, conditions for asymptotic optimisations
- Sometimes only a 'constant' factor
- Todo: Formalise proofs further (see Reynolds)
- Todo: Tighter bounds via (bounded) linear typing:

  $$(-)^{\dagger} : !_n(!_1 \mathrm{D}A \to B) \to (\mathrm{D}_n A \to \mathrm{D}_n B)$$

  implies $[f^{\dagger}]_n \in \mathcal{O}(P_n + n[f]_n)$
- Deeper categorical treament (adjunctions, monads)

Thanks!

Backup slides

Upper bounds are more useful

Proposition

There exists terms P_n and $Q_n \geq 1$, parameterised by n, such that:

$$[\mathrm{D}f]_{n[\mathcal{O}(m)]} \in \mathcal{O}(P_n + n\, Q_n [f]_m) \tag{1}$$

Proof.

Follows from lower-bound: at least n uses of f (at size at most m) with possible additional overhead:

- P_n accounts for time traversing the container to reach the leaves (the elements) and
- Q_n accounts for any extraneous applications of f beyond the linear (in n) use.

□

For naturality

Given two functors D, G and natural transformation $\eta_A : \mathrm{D}A \to \mathrm{G}A$:

$$\begin{tikzcd}
\mathrm{D}A \arrow[r, "\mathrm{D}f"] \arrow[d, "\eta_A"'] & \mathrm{D}B \arrow[d, "\eta_B"] \\
\mathrm{G}A \arrow[r, "\mathrm{G}f"'] & \mathrm{G}B
\end{tikzcd}$$

Let $\mathrm{size}_A(\eta_A\, x) = k(\mathrm{size}_A\, x)$. Then:

$$[\eta_B \circ \mathrm{D}f]_{n[\mathcal{O}(m)]} \in \mathcal{O}([\eta]_n + P_n + nQ_n[f]_m)$$
$$[\mathrm{G}f \circ \eta_A]_{n[\mathcal{O}(m)]} \in \mathcal{O}([\eta]_n + P_{k(n)} + k(n)Q_{k(n)}[f]_m)$$

Therefore, if $n \in \mathcal{O}(k(n))$ then $(\mathrm{G}f \circ \eta_A) \rightsquigarrow (\eta_B \circ \mathrm{D}f)$, otherwise the converse.