A Reinvestigation of Filinski's Symmetric Lambda Calculus

– Continuations, Duality, Classical Logic, but no Categories –

Kenichi Asai

Ochanomizu University (Tokyo, Japan)

If you have difficulty remembering the name of the university,

ocha	=	green tea	
no	=	of	i.e., water of green tea
mizu	=	water	

April 12, 2010

History

- Griffin [POPL'90] showed that the control operator $\mathcal{C}$ has type $\neg\neg A \rightarrow A$.
- Parigot [LPAR'92] introduced $\lambda\mu$-calculus that corresponds to classical natural deduction.
- Curien and Herbelin [ICFP'00] introduced $\overline{\lambda}\mu\tilde{\mu}$-calculus based on sequent calculus that has expression/continuation duality and CBV/CBN duality.
- Wadler [ICFP '03, RTA '05] introduced the Dual Calculus with clean syntax and CBV/CBN duality.

History

- Filinski [1989] introduced symmetric lambda calculus (SLC).
- Griffin [POPL'90] showed that the control operator $\mathcal{C}$ has type $\neg\neg A \to A$.
- Parigot [LPAR'92] introduced $\lambda\mu$-calculus that corresponds to classical natural deduction.
- Curien and Herbelin [ICFP'00] introduced $\overline{\lambda}\mu\tilde{\mu}$-calculus based on sequent calculus that has expression/continuation duality and CBV/CBN duality.
- Wadler [ICFP '03, RTA '05] introduced the Dual Calculus with clean syntax and CBV/CBN duality.

History

- Griffin [POPL'90]: Shortly before the deadline, the work of Filinski was brought to my attention. His work may provide a "deep reason" for the correspondence described in this paper.
- Curien and Herbelin [ICFP'00]: an earlier attempt in this direction [=CBV/CBN duality] can be found in Filinski.
- Wadler [ICFP '03]: Filinski was the first to suggest that CBV might be dual to CBN in the presence of continuations. Filinski's formulation lacks any connection with logic.
- Wadler [RTA '05]: A line of work, including Filinski, Griffin, ..., has led to a startling conclusion: CBV is de Morgan dual of CBN.

Filinski's duality

Expressions produce data.

Continuations consume data.

$$\mathbf{0} \longrightarrow A \longrightarrow B \longrightarrow \mathbf{1}$$

CBV/CBN duality naturally follows from expression/continuation duality:

CBV evaluates expressions first.

CBN evaluates continuations first.

Filinski's duality

Expressions produce data.
Functions transform data.
Continuations consume data.

$$\mathbf{0} \longrightarrow A \longrightarrow B \longrightarrow \mathbf{1}$$

CBV/CBN duality naturally follows from expression/continuation duality:

CBV evaluates expressions first.
CBN evaluates continuations first.

SLC: Syntax

A configuration is either $\langle\, e \mid c \,\rangle$ or $\langle\, e \mid f \mid c \,\rangle$, where:

$$
\begin{array}{rccl}
\text{expression} & e & ::= & \circ_T \mid x \mid (e, e) \mid \lceil f \rceil \mid e \uparrow f \\
\text{function} & f & ::= & g \mid x \Rightarrow e \mid (x_1, x_2) \Rightarrow e \mid \lceil g \rceil \Rightarrow e \mid \overline{e} \\
 & & & h \mid c \Leftarrow y \mid c \Leftarrow (y_1, y_2) \mid c \Leftarrow \lfloor h \rfloor \mid \underline{c} \\
\text{continuation} & c & ::= & \bullet_T \mid y \mid (c, c) \mid \lfloor f \rfloor \mid f \downarrow c
\end{array}
$$

$$
\begin{array}{rl}
\text{Example:} & \langle\, 1 \uparrow x_1 \Rightarrow x_1 + 2 \uparrow x_2 \Rightarrow x_2 * 4 \mid \bullet_{\mathsf{int}} \,\rangle \\
\rightsquigarrow & \langle\, 1 \uparrow x_1 \Rightarrow x_1 + 2 \mid x_2 \Rightarrow x_2 * 4 \mid \bullet_{\mathsf{int}} \,\rangle \\
\rightsquigarrow & \langle\, 1 \uparrow x_1 \Rightarrow x_1 + 2 \mid x_2 \Rightarrow x_2 * 4 \downarrow \bullet_{\mathsf{int}} \,\rangle \\
\rightsquigarrow & \langle\, 1 \mid \color{red}{x_1 \Rightarrow x_1 + 2} \mid x_2 \Rightarrow x_2 * 4 \downarrow \bullet_{\mathsf{int}} \,\rangle \\
\rightsquigarrow & \langle\, 1 + 2 \mid x_2 \Rightarrow x_2 * 4 \downarrow \bullet_{\mathsf{int}} \,\rangle \\
\rightsquigarrow & \langle\, 3 \mid x_2 \Rightarrow x_2 * 4 \downarrow \bullet_{\mathsf{int}} \,\rangle \\
\rightsquigarrow & \langle\, 3 \mid x_2 \Rightarrow x_2 * 4 \mid \bullet_{\mathsf{int}} \,\rangle \\
\rightsquigarrow^* & \langle\, 12 \mid \bullet_{\mathsf{int}} \,\rangle
\end{array}
$$

SLC: Syntax

A configuration is either $\langle\, e \mid c \,\rangle$ or $\langle\, e \mid f \mid c \,\rangle$, where:

$$\begin{array}{rrcl}
\text{expression} & e & ::= & \circ_T \mid x \mid (e, e) \mid \lceil f \rceil \mid e \uparrow f \\
\text{function} & f & ::= & g \mid x \Rightarrow e \mid (x_1, x_2) \Rightarrow e \mid \lceil g \rceil \Rightarrow e \mid \overline{e} \\
 & & & h \mid c \Leftarrow y \mid c \Leftarrow (y_1, y_2) \mid c \Leftarrow \lfloor h \rfloor \mid \underline{c} \\
\text{continuation} & c & ::= & \bullet_T \mid y \mid (c, c) \mid \lfloor f \rfloor \mid f \downarrow c
\end{array}$$

Example: Felleisen's $\mathcal{C}$ operator:

$$\mathcal{C} \;\equiv\; (\lceil g \rceil \Rightarrow \lceil y \Leftarrow \_ \rceil \uparrow g) \downarrow \bullet_\perp \Leftarrow y$$

Felleisen's $\mathcal{C}$ operator

$$
\begin{array}{rrcl}
\text{value} & V & ::= & x \mid \lambda x.\, M \mid \mathcal{C} \\
\text{term} & M & ::= & V \mid M\, M' \\
\text{evaluation context} & E & ::= & [] \mid E\, M \mid V\, M \\
\text{reduction rules} & E\,[\,(\lambda x.\, M)\, V\,] & \rightsquigarrow & E\,[\,M[V/x]\,] \\
 & E\,[\,\mathcal{C}\, V\,] & \rightsquigarrow & V\,(\lambda x.\, \mathcal{A}\,(E\,[\,x\,])) \\
\text{where} \quad & \mathcal{A}\, M & \equiv & \mathcal{C}\,(\lambda \_.\, M)
\end{array}
$$

Example execution:

$$
\begin{array}{rl}
 & 2 + \mathcal{C}\,(\lambda k.\, 4 * (k\, 1)) \\
\rightsquigarrow & (\lambda k.\, 4 * (k\, 1))\,(\lambda x.\, \mathcal{A}\,(2 + x)) \\
\rightsquigarrow & 4 * ((\lambda x.\, \mathcal{A}\,(2 + x))\, 1) \\
\rightsquigarrow & 4 * (\mathcal{A}\,(2 + 1)) \\
\rightsquigarrow & 2 + 1 \\
\rightsquigarrow & 3
\end{array}
$$

Felleisen's $\mathcal{C}$ operator in SLC

$$\mathcal{C} \quad \equiv \quad (\lceil g \rceil \Rightarrow \lceil y \Leftarrow \_ \rceil \uparrow g) \downarrow \bullet_{\perp} \Leftarrow y$$

$$2 + \mathcal{C}\,(\lambda k.\,4 * (k\,1))$$

$$\begin{array}{ll}
& \langle\, \lceil \lceil k \rceil \Rightarrow 1 \uparrow k \uparrow x_2 \Rightarrow 4 * x_2 \rceil \uparrow \mathcal{C} \uparrow x \Rightarrow 2 + x \mid \bullet_{\mathsf{int}} \,\rangle \\
\rightsquigarrow^* & \langle\, \lceil \lceil k \rceil \Rightarrow \cdots \rceil \mid \mathcal{C} \mid (x \Rightarrow 2 + x) \downarrow \bullet_{\mathsf{int}} \,\rangle \\
\rightsquigarrow & \langle\, \lceil \lceil k \rceil \Rightarrow \cdots \rceil \mid (\lceil g \rceil \Rightarrow \lceil (x \Rightarrow 2 + x) \downarrow \bullet_{\mathsf{int}} \Leftarrow _ \rceil \uparrow g) \downarrow \bullet_{\perp} \,\rangle \\
\rightsquigarrow & \langle\, \lceil \lceil k \rceil \Rightarrow \cdots \rceil \mid \lceil g \rceil \Rightarrow \lceil (x \Rightarrow 2 + x) \downarrow \bullet_{\mathsf{int}} \Leftarrow _ \rceil \uparrow g \mid \bullet_{\perp} \,\rangle \\
\rightsquigarrow & \langle\, \lceil (x \Rightarrow 2 + x) \downarrow \bullet_{\mathsf{int}} \Leftarrow _ \rceil \uparrow (\lceil k \rceil \Rightarrow \cdots) \mid \bullet_{\perp} \,\rangle \\
\rightsquigarrow & \langle\, \lceil (x \Rightarrow 2 + x) \downarrow \bullet_{\mathsf{int}} \Leftarrow _ \rceil \mid \lceil k \rceil \Rightarrow \cdots \mid \bullet_{\perp} \,\rangle \\
\rightsquigarrow & \langle\, 1 \uparrow ((x \Rightarrow 2 + x) \downarrow \bullet_{\mathsf{int}} \Leftarrow _) \uparrow x_2 \Rightarrow 4 * x_2 \mid \bullet_{\perp} \,\rangle \\
\rightsquigarrow^* & \langle\, 1 \mid ((x \Rightarrow 2 + x) \downarrow \bullet_{\mathsf{int}} \Leftarrow _) \mid x_2 \Rightarrow 4 * x_2 \downarrow \bullet_{\perp} \,\rangle \\
\rightsquigarrow & \langle\, 1 \mid (x \Rightarrow 2 + x) \downarrow \bullet_{\mathsf{int}} \,\rangle \\
\rightsquigarrow^* & \langle\, 3 \mid \bullet_{\mathsf{int}} \,\rangle
\end{array}$$

Reduction rules (non-deterministic)

$$
\begin{array}{rrcl}
(pop) & \langle\, e \uparrow f \mid c \,\rangle & \rightsquigarrow & \langle\, e \mid f \mid c \,\rangle \\
(push) & \langle\, e \mid f \mid c \,\rangle & \rightsquigarrow & \langle\, e \mid f \downarrow c \,\rangle \\
(left) & \langle\, (e_1, e_2) \mid c \,\rangle & \rightsquigarrow & \langle\, e_1 \mid x_1 \Rightarrow (x_1, e_2) \mid c \,\rangle \\
(right) & \langle\, (e_1, e_2) \mid c \,\rangle & \rightsquigarrow & \langle\, e_2 \mid x_2 \Rightarrow (e_1, x_2) \mid c \,\rangle \\
(exc) & \langle\, e \mid \overline{e'} \mid c \,\rangle & \rightsquigarrow & \langle\, e' \mid \lceil g \rceil \Rightarrow e \uparrow g \mid c \,\rangle \\
(\beta) & \langle\, e \mid x \Rightarrow e' \mid c \,\rangle & \rightsquigarrow & \langle\, e'[e/x] \mid c \,\rangle \\
(\beta_p) & \langle\, (e_1, e_2) \mid (x_1, x_2) \Rightarrow e' \mid c \,\rangle & \rightsquigarrow & \langle\, e'[e_1/x_1, e_2/x_2] \mid c \,\rangle \\
(\beta_f) & \langle\, \lceil f \rceil \mid \lceil g \rceil \Rightarrow e' \mid c \,\rangle & \rightsquigarrow & \langle\, e'[f/g] \mid c \,\rangle \\
(\overline{\beta_f}) & \langle\, e \mid c' \Leftarrow \lfloor h \rfloor \mid \lfloor f \rfloor \,\rangle & \rightsquigarrow & \langle\, e \mid c'[f/h] \,\rangle \\
(\overline{\beta_p}) & \langle\, e \mid c' \Leftarrow (y_1, y_2) \mid (c_1, c_2) \,\rangle & \rightsquigarrow & \langle\, e \mid c'[c_1/y_1, c_2/y_2] \,\rangle \\
(\overline{\beta}) & \langle\, e \mid c' \Leftarrow y \mid c \,\rangle & \rightsquigarrow & \langle\, e \mid c'[c/y] \,\rangle \\
(\overline{exc}) & \langle\, e \mid \underline{c'} \mid c \,\rangle & \rightsquigarrow & \langle\, e \mid h \downarrow c \Leftarrow \lfloor h \rfloor \mid c' \,\rangle \\
(\overline{right}) & \langle\, e \mid (c_1, c_2) \,\rangle & \rightsquigarrow & \langle\, e \mid (c_1, y_2) \Leftarrow y_2 \mid c_2 \,\rangle \\
(\overline{left}) & \langle\, e \mid (c_1, c_2) \,\rangle & \rightsquigarrow & \langle\, e \mid (y_1, c_2) \Leftarrow y_1 \mid c_1 \,\rangle \\
(\overline{push}) & \langle\, e \mid f \mid c \,\rangle & \rightsquigarrow & \langle\, e \uparrow f \mid c \,\rangle \\
(\overline{pop}) & \langle\, e \mid f \downarrow c \,\rangle & \rightsquigarrow & \langle\, e \mid f \mid c \,\rangle
\end{array}
$$

Types

$$
\begin{array}{rcll}
S & ::= & +T & \text{type of expressions} \\
 & | & \left\{ {+A \rightarrow +B \atop \neg A \leftarrow \neg B} \right. & \text{type of functions} \\
 & | & \neg T & \text{type of continuations}
\end{array}
$$

$$T, A, B \; ::= \; \bot \mid \top \mid X \mid A \wedge B \mid A \vee B \mid A \rightarrow B \mid A - B$$

Negation is represented as $A \rightarrow \bot$ or $\top - A$.

$$
\begin{array}{ccccccc}
 & & +A & +A \rightarrow +B & & & \\
 & & \vdots & \vdots & & & \\
\mathbf{0} & \longrightarrow & A & \longrightarrow & B & \longrightarrow & \mathbf{1} \\
 & & & \vdots & & \vdots & \\
 & & & \neg A \leftarrow \neg B & & \neg B &
\end{array}
$$

Type system

$$\frac{}{\Gamma \vdash \circ_T : +T}\ \mathtt{TInit} \qquad \frac{}{\Gamma, \mathsf{var} : S \vdash \mathsf{var} : S}\ \mathtt{TVar} \qquad \frac{}{\Gamma \vdash \bullet_T : \neg T}\ \overline{\mathtt{TInit}}$$

$$\frac{\Gamma \vdash e_1 : +A \quad \Gamma \vdash e_2 : +B}{\Gamma \vdash (e_1, e_2) : +(A \wedge B)}\ \mathtt{TAnd} \qquad \frac{\Gamma \vdash c_1 : \neg A \quad \Gamma \vdash c_2 : \neg B}{\Gamma \vdash (c_1, c_2) : \neg(A \vee B)}\ \overline{\mathtt{TOr}}$$

$$\frac{\Gamma, x : +A \vdash e : +B}{\Gamma \vdash x \Rightarrow e : \{\substack{+A \to +B \\ \neg A \leftarrow \neg B}}\ \mathtt{TFun} \qquad \frac{\Gamma, y : \neg B \vdash c : \neg A}{\Gamma \vdash c \Leftarrow y : \{\substack{+A \to +B \\ \neg A \leftarrow \neg B}}\ \overline{\mathtt{TFun}}$$

$$\frac{\Gamma \vdash f : \{\substack{+A \to +B \\ \neg A \leftarrow \neg B} \quad \Gamma \vdash e : +A}{\Gamma \vdash e \uparrow f : +B}\ \mathtt{TApp} \qquad \frac{\Gamma \vdash f : \{\substack{+A \to +B \\ \neg A \leftarrow \neg B} \quad \Gamma \vdash c : \neg B}{\Gamma \vdash f \downarrow c : \neg A}\ \overline{\mathtt{TApp}}$$

$$\frac{\Gamma \vdash f : \{\substack{+A \to +B \\ \neg A \leftarrow \neg B}}{\Gamma \vdash \lceil f \rceil : +(A \to B)}\ \mathtt{TFClo} \qquad \frac{\Gamma \vdash f : \{\substack{+A \to +B \\ \neg A \leftarrow \neg B}}{\Gamma \vdash \lfloor f \rfloor : \neg(A - B)}\ \overline{\mathtt{TFClo}}$$

$$\frac{\Gamma \vdash e : +(A \to B)}{\Gamma \vdash \overline{e} : \{\substack{+A \to +B \\ \neg A \leftarrow \neg B}}\ \mathtt{TCFun} \qquad \frac{\Gamma \vdash c : \neg(A - B)}{\Gamma \vdash \underline{c} : \{\substack{+A \to +B \\ \neg A \leftarrow \neg B}}\ \overline{\mathtt{TCFun}}$$

$$\frac{\vdash e : +T \quad \vdash c : \neg T}{\vdash \langle e \mid c \rangle}\ \mathtt{TProg1} \qquad \frac{\vdash e : +A \quad \vdash f : \{\substack{+A \to +B \\ \neg A \leftarrow \neg B} \quad \vdash c : \neg B}{\vdash \langle e \mid f \mid c \rangle}\ \mathtt{TProg2}$$

Theorems

- Progress
  If a configuration is well-typed, it can take one more step, or the configuration is of the form $\langle\, \boldsymbol{v} \mid \bullet_{\boldsymbol{T}} \,\rangle$ or $\langle\, \circ_{\boldsymbol{T}} \mid \boldsymbol{k} \,\rangle$.
- Preservation
  If a configuration is well-typed and can take a step, the next configuration is also well-typed.
- Termination for CBV and CBN
  The execution terminates under CBV or CBN evaluation strategy. (The proof uses logical predicate arguments.)
- Translations to and from the Dual Calculus preserve equations.
  We can define equation-preserving translations.

Classical Programming

$\mathcal{C}$ has type $\left\{ {+((A \to \bot) \to \bot) \to +A \atop \neg((A \to \bot) \to \bot) \leftarrow \neg A} \right.$.

- It eliminates double negation.
- It corresponds to proof by contradiction.
- Give me a term of type $((A \to \bot) \to \bot)$.
- In other words, assume that f is a proof that A is false; from this assumption, give me a way to show contradiction. For example, if $A = B \to B$, then $\lceil f \rceil \Rightarrow \lceil x \Rightarrow x \rceil \uparrow f$.
- Then, I will give you a term of type A: $\lceil \lceil f \rceil \Rightarrow \lceil x \Rightarrow x \rceil \uparrow f \rceil \uparrow \mathcal{C}$.

Classical Programming

$$
\begin{array}{rcl}
\mathcal{C} & \equiv & (\lceil g \rceil \Rightarrow \lceil y \Leftarrow \_ \rceil \uparrow g) \downarrow \bullet_{\perp} \Leftarrow y \\
\lceil y \Leftarrow \_ \rceil & : & A \to \perp \\
g & : & (A \to \perp) \to \perp
\end{array}
$$

$$
\begin{array}{rl}
& \langle \lceil \lceil f \rceil \Rightarrow \lceil x \Rightarrow x \rceil \uparrow f \rceil \uparrow \mathcal{C} \mid \bullet_{B\to B} \rangle \\
\rightsquigarrow & \langle \lceil \lceil f \rceil \Rightarrow \lceil x \Rightarrow x \rceil \uparrow f \rceil \mid \mathcal{C} \mid \bullet_{B\to B} \rangle \\
\rightsquigarrow & \langle \lceil \lceil f \rceil \Rightarrow \lceil x \Rightarrow x \rceil \uparrow f \rceil \mid (\lceil g \rceil \Rightarrow \lceil \bullet_{B\to B} \Leftarrow \_ \rceil \uparrow g) \downarrow \bullet_{\perp} \rangle \\
\rightsquigarrow & \langle \lceil \lceil f \rceil \Rightarrow \lceil x \Rightarrow x \rceil \uparrow f \rceil \mid \lceil g \rceil \Rightarrow \lceil \bullet_{B\to B} \Leftarrow \_ \rceil \uparrow g \mid \bullet_{\perp} \rangle \\
\rightsquigarrow & \langle \lceil \bullet_{B\to B} \Leftarrow \_ \rceil \uparrow (\lceil f \rceil \Rightarrow \lceil x \Rightarrow x \rceil \uparrow f) \mid \bullet_{\perp} \rangle \\
\rightsquigarrow & \langle \lceil \bullet_{B\to B} \Leftarrow \_ \rceil \mid \lceil f \rceil \Rightarrow \lceil x \Rightarrow x \rceil \uparrow f \mid \bullet_{\perp} \rangle \\
\rightsquigarrow & \langle \lceil x \Rightarrow x \rceil \uparrow (\bullet_{B\to B} \Leftarrow \_) \mid \bullet_{\perp} \rangle \\
\rightsquigarrow & \langle \lceil x \Rightarrow x \rceil \mid \bullet_{B\to B} \Leftarrow \_ \mid \bullet_{\perp} \rangle \\
\rightsquigarrow & \langle \lceil x \Rightarrow x \rceil \mid \bullet_{B\to B} \rangle
\end{array}
$$

Challenge

Find a (classical) type A with the following properties:

- It is hard to prove A directly.
- It is easy to show contradiction assuming $A \to \bot$.

Classical Programming

$$\circ_\top \uparrow ((x \Rightarrow \lceil y_1 \Leftarrow \_\rceil) \downarrow y_2 \Leftarrow (y_1, y_2)) : +(A \vee (A \rightarrow \bot)).$$

- For any type A, you can assume either A or $A \rightarrow \bot$ without knowing if A actually holds or not.
- You can use this fact by providing two futures.
- One for when A is true (y_1).
- The other for when A is false (y_2).
- The computation first assumes A is false.
- If A turns out to be true, the other future is invoked.

Challenge

Find a (classical) type A with the following properties:

- It is hard to prove A directly.
- It is easy to show A assuming $B \lor (B \rightarrow \bot)$ for some B whose truthhood is not obvious.

Excluded Middle

An irrational number raised by another irrational number can be a rational number.

- $\sqrt{2}^{\sqrt{2}}$ is either rational or irrational.
- If it it rational, the proposition holds.
- If it is irrational, we have: $(\sqrt{2}^{\sqrt{2}})^{\sqrt{2}} = \sqrt{2}^2 = 2$.

A program either halts or diverges.

- If it halts, do something.
- If it diverges, do another thing.

What are their computational contents?

Summary

- Filinski's SLC naturally contains familiar notions: λ-calculus, control operators, and evaluation contexts.
- Expression/continuation duality explains behavior of continuations nicely.
- It is formalized as a programming language.
- It has connection with logic.

- Could lead to classical programming?
- Delimited continuations?

Non-deterministic reduction

CBV:

$$\begin{array}{llll} & \langle\, 1 \uparrow \bullet_{\mathsf{int}} \Leftarrow y \,|\, x \Rightarrow 2 \downarrow \bullet_{\mathsf{int}} \,\rangle & & (\lambda x.\, 2)\,(\mathcal{C}\,(\lambda k.\, 1)) \\ \rightsquigarrow & \langle\, 1 \,|\, \bullet_{\mathsf{int}} \Leftarrow y \,|\, x \Rightarrow 2 \downarrow \bullet_{\mathsf{int}} \,\rangle & \rightsquigarrow & (\lambda k.\, 1)\,(\lambda x.\, \mathcal{A}\,((\lambda x.\, 2)\; x)) \\ \rightsquigarrow & \langle\, 1 \,|\, \bullet_{\mathsf{int}} \,\rangle & \rightsquigarrow & 1 \end{array}$$

CBN:

$$\begin{array}{llll} & \langle\, 1 \uparrow \bullet_{\mathsf{int}} \Leftarrow y \,|\, x \Rightarrow 2 \downarrow \bullet_{\mathsf{int}} \,\rangle & & (\lambda x.\, 2)\,(\mathcal{C}\,(\lambda k.\, 1)) \\ \rightsquigarrow & \langle\, 1 \uparrow \bullet_{\mathsf{int}} \Leftarrow y \,|\, x \Rightarrow 2 \,|\, \bullet_{\mathsf{int}} \,\rangle & & \\ \rightsquigarrow & \langle\, 2 \,|\, \bullet_{\mathsf{int}} \,\rangle & \rightsquigarrow & 2 \end{array}$$