# Bisimulation-Based Process Algebra in Higher-Order GSOS

Master's Thesis

Florian Guthmann

Friedrich-Alexander-Universität Erlangen-Nürnberg

2024-12-10

# Process Algebra

- model the behaviour of concurrent systems
- CCS (Milner), CSP (Hoare), $\pi$-calculus
- Bisimilarity can serve as the semantic property distinguishing processes

## Key Property

Bisimilarity is a Congruence

# CCS[1]

A process algebra featuring

- the deadlocked process $\emptyset$

[1] Milner, *A Calculus of Communicating Systems*.

# CCS[1]

A process algebra featuring

- the deadlocked process $\varnothing$
- action prefixing $\alpha \centerdot P$

[1] Milner, *A Calculus of Communicating Systems*.

# CCS[1]

A process algebra featuring

- the deadlocked process $\varnothing$
- action prefixing $\alpha \mathbin{.} P$
- nondeterministic choice $P + Q$

[1] Milner, *A Calculus of Communicating Systems*.

# CCS[1]

A process algebra featuring

- the deadlocked process $\varnothing$
- action prefixing $\alpha \,.\, P$
- nondeterministic choice $P + Q$
- parallel composition $P \mid Q$

[1] Milner, *A Calculus of Communicating Systems*.

# CCS[1]

A process algebra featuring

- the deadlocked process $\varnothing$
- action prefixing $\alpha\,.\,P$
- nondeterministic choice $P + Q$
- parallel composition $P \mid Q$
- action renaming $P[\varphi]$

[1] Milner, *A Calculus of Communicating Systems*.

# CCS[1]

A process algebra featuring

- the deadlocked process $\varnothing$
- action prefixing $\alpha \,.\, P$
- nondeterministic choice $P + Q$
- parallel composition $P \mid Q$
- action renaming $P[\varphi]$
- action restriction $P \setminus L$

[1] Milner, *A Calculus of Communicating Systems*.

# CCS[1]

A process algebra featuring

- the deadlocked process $\varnothing$
- action prefixing $\alpha \mathbin{.} P$
- nondeterministic choice $P + Q$
- parallel composition $P \mid Q$
- action renaming $P[\varphi]$
- action restriction $P \setminus L$
- recursion $\mathrm{fix}\, X.\, P$

[1] Milner, *A Calculus of Communicating Systems*.

# Preliminaries

- a set of actions $\mathcal{A}$ with $\tau \in \mathcal{A}$
- an involution $\overline{\cdot}\colon \mathcal{A} \setminus \{\tau\} \to \mathcal{A} \setminus \{\tau\}$ $\quad (\overline{\overline{\alpha}} = \alpha)$
- $\mathrm{Ren}(\mathcal{A}) \coloneqq \left\{\varphi\colon \mathcal{A} \setminus \{\tau\} \to \mathcal{A} \setminus \{\tau\} \mid \forall\, \alpha \in \mathcal{A}.\, \varphi(\overline{\alpha}) = \overline{\varphi(\alpha)}\right\}$
- a set $\mathcal{V}$ of variables

# Syntax of CCS

$$P, Q \coloneqq \varnothing \mid X \mid \alpha \,.\, P \mid P + Q \mid P \mid Q \mid P \setminus L \mid P[\varphi] \mid \mathrm{fix}\, X.\, P$$

where

# Syntax of CCS

$$P, Q \coloneqq \varnothing \;\Big|\; X \;\Big|\; \alpha \,.\, P \;\Big|\; P + Q \;\Big|\; P \mid Q \;\Big|\; P \setminus L \;\Big|\; P[\varphi] \;\Big|\; \mathrm{fix}\, X.\, P$$

where

- $X \in \mathcal{V}$
- $\alpha \in \mathcal{A}$
- $L \subseteq \mathcal{A} \setminus \{\tau\}$
- $\varphi \in \mathrm{Ren}(\mathcal{A})$

# To Steal an Example

$$\mathsf{CM} \coloneqq \mathrm{fix}\, CM.\, \mathrm{coin} \,.\, \overline{\mathrm{coffee}} \,.\, CM$$

$$\mathsf{CS} \coloneqq \mathrm{fix}\, CS.\, \overline{\mathrm{pub}} \,.\, \overline{\mathrm{coin}} \,.\, \mathrm{coffee} \,.\, CS$$

$$\mathsf{UNI} \coloneqq (\mathsf{CM} \mid \mathsf{CS}) \setminus \{\mathrm{coin}, \mathrm{coffee}\}$$

# To Steal an Example

Synchronization via handshake

$$\mathsf{CM} := \mathrm{fix}\, CM.\, \mathrm{coin} \centerdot \overline{\mathrm{coffee}} \centerdot CM$$

$$\mathsf{CS} := \mathrm{fix}\, CS.\, \overline{\mathrm{pub}} \centerdot \overline{\mathrm{coin}} \centerdot \mathrm{coffee} \centerdot CS$$

$$\mathsf{UNI} := (\mathsf{CM} \mid \mathsf{CS}) \setminus \{\mathrm{coin}, \mathrm{coffee}\}$$

# Operational Semantics of CCS

$$\frac{}{\alpha \,.\, P \xrightarrow{\alpha} P} \; \text{act}$$

$$\frac{P \xrightarrow{\alpha} P'}{P + Q \xrightarrow{\alpha} P'} \; \text{sum}_\text{l} \qquad \frac{Q \xrightarrow{\alpha} Q'}{P + Q \xrightarrow{\alpha} Q'} \; \text{sum}_\text{r}$$

$$\frac{P \xrightarrow{\alpha} P'}{P \mid Q \xrightarrow{\alpha} P' \mid Q} \; \text{par}_\text{l} \qquad \frac{Q \xrightarrow{\alpha} Q'}{P \mid Q \xrightarrow{\alpha} P \mid Q'} \; \text{par}_\text{r}$$

$$\frac{P \xrightarrow{\alpha} P' \qquad Q \xrightarrow{\overline{\alpha}} Q'}{P \mid Q \xrightarrow{\tau} P' \mid Q'} \; \text{sync}$$

$$\frac{P \xrightarrow{\alpha} P' \qquad \alpha, \overline{\alpha} \notin L}{P \setminus L \xrightarrow{\alpha} P'} \; \text{res} \qquad \frac{P \xrightarrow{\alpha} P'}{P\,[\varphi] \xrightarrow{\varphi(\alpha)} P'\,[\varphi]} \; \text{ren}$$

$$\frac{P\;[\mathrm{fix}\, X.\, P/X] \xrightarrow{\alpha} P'}{\mathrm{fix}\, X.\, P \xrightarrow{\alpha} P'} \; \text{fix}$$

# Guarded terms

## Definition ( guarded )

A term $P$ is *guarded* if all variable occurences in $P$ are only in subterms of the form $\alpha \centerdot Q$.

$$\text{fix}\, X.\, \alpha \centerdot X \quad \checkmark$$

$$\text{fix}\, Y.\, (\text{fix}\, X.\, \alpha \centerdot X) + Y$$

# Strong Bisimulation

## Definition

$R \subseteq \mathrm{Proc} \times \mathrm{Proc}$ is a *strong bisimulation* iff


$$P \sim Q :\Longleftrightarrow \exists\, R.\, (P, Q) \in R \wedge R \text{ is a bisimulation}$$

# Strong Bisimulation

## Definition

$R \subseteq \mathrm{Proc} \times \mathrm{Proc}$ is a *strong bisimulation* iff


$$P \sim Q :\Longleftrightarrow \exists\, R.\, (P, Q) \in R \wedge R \text{ is a bisimulation}$$

$$P + Q \sim Q + P, \quad P + P \sim P, \quad P + \varnothing \sim P$$

$$(P + Q) \mid S \not\sim (P \mid S) + (Q \mid S)$$

# Bisimulation as a Congruence

We want $\sim$ to be a $\Sigma$-congruence, i.e. for all $f \in \overline{\Sigma}$

$$X_1 \sim Y_1, \ldots X_{\mathrm{ar}(f)} \sim Y_{\mathrm{ar}(f)} \implies f(X_1, \ldots, X_{\mathrm{ar}(f)}) \sim f(Y_1, \ldots, Y_{\mathrm{ar}(f)})$$

Proofs of this in the presence of fixpoints are rather *involved* and *fragile*[2]

[2] Amadio and Curien, *Domains and Lambda-Calculi*.

# de Bruijn Indices

- We set $\mathcal{V} := \mathbb{N}$
- A substitution is a function $\sigma\colon n \to \mathrm{Proc}$
- $\mathrm{fix}$ always binds $0$, we write $\mathrm{fix}\ P$ for $\mathrm{fix}\,\mathit{0}.\,P$
- Bound variables count the number of binders to “jump over”:

$$\mathrm{fix}\ \alpha \mathbin{.} \mathrm{fix}\ (\beta \mathbin{.} 0 + \gamma \mathbin{.} 1)$$

# GSOS[3]

$$\frac{\left(x_{i_j} \xrightarrow{\alpha_j} y_{i_j}\right)_j \qquad \left(x_{i_k} \stackrel{\alpha_k}{\not\longrightarrow}\right)_k}{f(x_1, \ldots, x_{\mathrm{ar}(f)}) \xrightarrow{\alpha} t}$$

[3] Bloom, Istrail, and Meyer, “Bisimulation can’t be traced”.

# GSOS[3]

$$\frac{\left(x_{i_j} \xrightarrow{\alpha_j} y_{i_j}\right)_j \qquad \left(x_{i_k} \overset{\alpha_k}{\not\longrightarrow}\right)_k}{f(x_1, \ldots, x_{\mathrm{ar}(f)}) \xrightarrow{\alpha} t}$$

where

- $f \in \Sigma$
- $1 \leq i_j, i_k \leq \mathrm{ar}(f)$
- $x_{i_j}, y_{i_j}$ are distinct
- $t$ contains only variables from $x_{i_j}, y_{i_j}$

[3]Bloom, Istrail, and Meyer, "Bisimulation can't be traced".

# First-Order Abstract GSOS[4]

A categorical framework:

- A signature functor $\Sigma\colon \mathcal{C} \to \mathcal{C}$
- A behaviour functor $B\colon \mathcal{C} \to \mathcal{C}$

A GSOS rule corresponds to a natural transformation

$$\varrho\colon \Sigma(\mathrm{Id} \times B) \Rightarrow B\Sigma^{\star}$$

[4] Turi and Plotkin, "Towards a mathematical operational semantics".

# Signature

$$
\begin{aligned}
\overline{\Sigma} :=\ & \mathbb{N} \\
& \cup \{\alpha \bullet (\,\cdot\,)/1 \mid \alpha \in \mathcal{A}\} \\
& \cup \{+/2, \mid /2\} \\
& \cup \{(\,\cdot\,)\,[\varphi]\,/1 \mid \varphi \in \mathrm{Ren}(\mathcal{A})\} \\
& \cup \{(\,\cdot\,) \setminus L/1 \mid L \subseteq \mathcal{A}\}
\end{aligned}
$$

$$
\Sigma\, X = \coprod_{f \in \overline{\Sigma}} X^{\mathrm{ar}(f)}
$$

# Abstract GSOS

- A signature functor $\Sigma X = \coprod_{f \in \overline{\Sigma}} X^{\mathrm{ar}(f)}$
- A behaviour functor $B X = \mathcal{P}_{\omega 1}(\mathcal{A} \times X)$

$$\frac{P \xrightarrow{\alpha} P'}{P \mid Q \xrightarrow{\alpha} P' \mid Q}$$

$$\varrho_X \colon \Sigma(X \times B X) \to B\Sigma^\star X$$

$$\varrho_X((P, b_P) \mid (Q, b_Q)) = \{(\alpha, (P' \mid Q)) \mid (\alpha, P') \in b_P\}$$

$$\cup \ldots$$

# Abstract GSOS

- A signature functor $\Sigma X = \coprod_{f \in \overline{\Sigma}} X^{\mathrm{ar}(f)}$
- A behaviour functor $B\,X = \mathcal{P}_{\omega 1}\,(\mathcal{A} \times X)$

$$\frac{P \xrightarrow{\alpha} P'}{P \mid Q \xrightarrow{\alpha} P' \mid Q}$$

$$\varrho_X \colon \Sigma(X \times B\,X) \to B\Sigma^\star\,X$$

$$\varrho_X((P, b_P) \mid (Q, b_Q)) = \{(\alpha, (P' \mid Q)) \mid (\alpha, P') \in b_P\}$$
$$\cup \ldots$$

# Abstract GSOS

- A signature functor $\Sigma X = \coprod_{f \in \overline{\Sigma}} X^{\mathrm{ar}(f)}$
- A behaviour functor $B X = \mathcal{P}_{\omega 1}(\mathcal{A} \times X)$

$$\frac{P \xrightarrow{\alpha} P'}{P \mid Q \xrightarrow{\alpha} P' \mid Q}$$

$$\varrho_X \colon \Sigma(X \times B X) \to B\Sigma^\star X$$

$$\varrho_X((P, b_P) \mid (Q, b_Q)) = \{(\alpha, (P' \mid Q)) \mid (\alpha, P') \in b_P\} \cup \ldots$$

# Operational Model

We obtain a coalgebra $(\mu\Sigma, \gamma)$:

$$\begin{array}{ccccc}
\Sigma\,\mu\Sigma & \xrightarrow{\iota} & \mu\Sigma & \xrightarrow{\gamma} & B\,\mu\Sigma \\
\downarrow{\scriptstyle \Sigma\langle\,\mathrm{id},\gamma\rangle} & & & & \uparrow{\scriptstyle B\hat{\iota}} \\
\Sigma(\mu\Sigma \times B\,\mu\Sigma) & & \xrightarrow{\varrho_{\mu\Sigma}} & & B\Sigma^{\star}\,\mu\Sigma
\end{array}$$

# Operational Model

By initiality of $(\mu\Sigma, \iota)$

We obtain a coalgebra $(\mu\Sigma, \gamma)$:

$$\begin{array}{ccccc}
\Sigma\,\mu\Sigma & \xrightarrow{\iota} & \mu\Sigma & \xrightarrow{\gamma} & B\,\mu\Sigma \\
\downarrow{\scriptstyle \Sigma\langle \mathrm{id},\gamma\rangle} & & & & \uparrow{\scriptstyle B\hat{\iota}} \\
\Sigma(\mu\Sigma \times B\,\mu\Sigma) & & \xrightarrow{\varrho_{\mu\Sigma}} & & B\Sigma^{\star}\,\mu\Sigma
\end{array}$$

# Abstract Behaviour

$$\begin{array}{ccccc} \Sigma\,\mu\Sigma & \xrightarrow{\iota} & \mu\Sigma & \xrightarrow{\gamma} & B\,\mu\Sigma \\ \downarrow{\scriptstyle \Sigma[\![-]\!]_\varrho} & & \downarrow{\scriptstyle [\![-]\!]_\varrho} & & \downarrow{\scriptstyle B[\![-]\!]_\varrho} \\ \Sigma\,\nu B & \xrightarrow{\alpha} & \nu B & \xrightarrow{\tau} & B\,\nu B \end{array}$$

We obtain strong bisimulation:

$$P \sim Q :\Longleftrightarrow [\![P]\!]_\varrho = [\![Q]\!]_\varrho$$

# Compositionality for Free

## Proposition

$\sim\ \subseteq \mu\Sigma \times \mu\Sigma$ *is a* $\Sigma$*-congruence.*

# Why $\mathrm{fix}$ poses a problem

$$\frac{P\ [\mathrm{fix}\,X.\,P/X] \xrightarrow{\alpha} P'}{\mathrm{fix}\,X.\,P \xrightarrow{\alpha} P'}\ \textsf{fix}$$

# Higher-Order Abstract GSOS[5]

Extend first-order abstract GSOS to allow for higher-order languages:

- a signature functor $\Sigma\colon \mathcal{C} \to \mathcal{C}$
- a mixed-variance behaviour bifunctor $B\colon \mathcal{C}^{\mathrm{op}} \times \mathcal{C} \to \mathcal{C}$

## Definition

A *higher-order GSOS law* is a family

$$\varrho_{X,Y}\colon \Sigma(X \times B(X, Y)) \to B(X, \Sigma^{\star}(X + Y))$$

*dinatural* in $X$ and natural in $Y$.

[5] Goncharov, Milius, Schröder, Tsampas, and Urbat, "Higher-Order Mathematical Operational Semantics".

# Alternative fixpoint semantics

$$\frac{P\ [\text{fix}\ P/0] \xrightarrow{\alpha} P'}{\text{fix}\ P \xrightarrow{\alpha} P'}\ \textsf{fix}$$

$$\frac{P \xrightarrow{\alpha}_{\text{fix'}} P'}{\text{fix}\ P \xrightarrow{\alpha}_{\text{fix'}} P'\ [\text{fix}\ P/0]}\ \textsf{fix'}$$

# Unguarded Case (Standard Semantics)[6]

$$P := \mathrm{fix}\ (0 \mid \alpha \bullet \varnothing + \overline{\alpha} \bullet \varnothing)$$

[6] ☞ `counter-example`

# Unguarded Case (Standard Semantics)[6]

$$P \coloneqq \mathrm{fix}\ (0 \mid \alpha \mathbin{.} \varnothing + \overline{\alpha} \mathbin{.} \varnothing)$$

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{\alpha \mathbin{.} \varnothing \xrightarrow{\alpha} \varnothing}\ \textsf{act}}{\alpha \mathbin{.} \varnothing + \overline{\alpha} \mathbin{.} \varnothing \xrightarrow{\alpha} \varnothing}\ \textsf{sum}_\textsf{l}}{P \mid \alpha \mathbin{.} \varnothing + \overline{\alpha} \mathbin{.} \varnothing \xrightarrow{\tau} P \mid \varnothing}\ \textsf{comp}_\textsf{r}}{P \xrightarrow{\alpha} P \mid \varnothing}\ \textsf{fix} \qquad \dfrac{\dfrac{}{\overline{\alpha} \mathbin{.} \varnothing \xrightarrow{\overline{\alpha}} \varnothing}\ \textsf{act}}{\alpha \mathbin{.} \varnothing + \overline{\alpha} \mathbin{.} \varnothing \xrightarrow{\overline{\alpha}} \varnothing}\ \textsf{sum}_\textsf{r}}{\dfrac{P \mid \alpha \mathbin{.} \varnothing + \overline{\alpha} \mathbin{.} \varnothing \xrightarrow{\tau} P \mid \varnothing \mid \varnothing}{P \xrightarrow{\tau} P \mid \varnothing \mid \varnothing}\ \textsf{fix}}\ \textsf{sync}$$

[6] `counter-example`

# Unguarded Case (Alternative Semantics)

$$P \coloneqq \text{fix } 0 \mid \alpha \centerdot \varnothing + \overline{\alpha} \centerdot \varnothing$$

$$\dfrac{\dfrac{0 \overset{\alpha}{\not\longrightarrow}_{\text{fix}'}}{0 \mid \alpha \centerdot \varnothing + \overline{\alpha} \centerdot \varnothing \xrightarrow{\tau}_{\text{fix}'}}\ \textsf{sync}}{\text{fix } 0 \mid \alpha \centerdot \varnothing + \overline{\alpha} \centerdot \varnothing \xrightarrow{\tau}_{\text{fix}'}}\ \textsf{fix'}$$

## Lemma ( step-subst )

*Let $P, P'$ be terms, $\sigma$ a substitution.*

$$P \xrightarrow{\alpha} P' \implies (P\,\sigma) \xrightarrow{\alpha} (P'\,\sigma)$$

## Lemma ( subst-fix-swap )

*Let $P$ be a guarded term, $T, Q$ arbitrary terms. If $(P\ [T/0]) \xrightarrow{\alpha} Q$ then there exists a $Q'$ such that*

$$\exists\, Q'.\, P \xrightarrow{\alpha} Q' \wedge Q = (Q'\ [T/0])$$

# Equivalence of the operational semantics

**Theorem ( 🐓 fix⇔fix' )**

*Let $P$ be a guarded term. Then for all $Q$, $\alpha$:*

$$P \xrightarrow{\alpha} Q \iff P \xrightarrow{\alpha}_{\mathrm{fix}'} Q$$

## Proof.

"$\Rightarrow$" by induction on the derivation using ☞ `subst-fix-swap` in the $\mathrm{fix}$ case

"$\Leftarrow$" by induction on the derivation using ☞ `step-subst` in the $\mathrm{fix}'$ case

$\square$

# Indexing with Substitutions

## Problem

$$\frac{P \xrightarrow{\alpha}_{\text{fix'}} P'}{\text{fix } P \xrightarrow{\alpha}_{\text{fix'}} P' \; [\text{fix } P/0]} \text{ fix'}$$

# Indexing with Substitutions

## Problem

$$\frac{P \xrightarrow{\alpha}_{\text{fix'}} P'}{\text{fix } P \xrightarrow{\alpha}_{\text{fix'}} P' \; [\text{fix } P/0]} \text{ fix'}$$

## Idea

Define a set of rules with labels $\mathcal{A} \times \mathrm{Subst}(n, m)$ such that

$$P \xrightarrow{\alpha}_{\sigma} P'' \iff (\exists\, P'.\, P \xrightarrow{\alpha} P' \wedge P'' = P'\,\sigma)$$

# Indexing with Substitutions

$$\frac{}{\alpha\,.\,P \xrightarrow{\alpha}_{\sigma} P\sigma}\ \mathsf{act}$$

$$\frac{P \xrightarrow{\alpha}_{\sigma} P'}{P \mid Q \xrightarrow{\alpha}_{\sigma} P' \mid Q\sigma}\ \mathsf{par_l} \qquad \frac{Q \xrightarrow{\alpha}_{\sigma} Q'}{P \mid Q \xrightarrow{\alpha}_{\sigma} P\sigma \mid Q'}\ \mathsf{par_r}$$

$$\frac{P \xrightarrow{\alpha}_{(\sigma\circ\ [\mathrm{fix\ \ P}/0])} P'}{\mathrm{fix}\ P \xrightarrow{\alpha}_{\sigma} P'}\ \mathsf{fix}$$

# Indexing with Substitutions

**Proposition (** `subst-step⇔fix'` **)**

*It does indeed hold that*

$$P \xrightarrow{\alpha}_{\sigma} P'' \iff (\exists\, P'.\, P \xrightarrow{\alpha} P' \wedge P'' = P'\,\sigma)$$

# Indexing with Lists

## Problem

Indexing by arbitrary substitutions is not necessary; We only need substitutions of the form $[\text{fix } P/0]$.

## Solution

We index by a list of $\mathrm{X}+1$

# Indexing with Lists

$$\frac{P[\square, \xi] = P'}{(\text{fix } P)[\xi] = \text{fix } P'} \ \mathsf{fix_{sub}}$$

$$\frac{P_i \neq \square}{i\,[P_0, P_1, \ldots, P_n] = \text{fix } P} \ \mathsf{name_1}$$

$$\frac{P_i = \square}{i\,[P_0, P_1, \ldots, P_n] = i} \ \mathsf{name_2}$$

$$\frac{P[\xi] = P'}{\alpha \bullet P \xrightarrow{\alpha}_{\xi} P'} \ \mathsf{act}$$

$$\frac{P \xrightarrow{\alpha}_{\mathrm{P,P_0,\ldots,P_n}} P'}{\text{fix } P \xrightarrow{\alpha}_{\mathrm{P_0,\ldots,P_n}} P'} \ \mathsf{fix}$$

# Example

$$\dfrac{\dfrac{\dfrac{P' = \mathit{1}[\alpha \,.\, \mathit{1}, \text{fix } \alpha \,.\, \mathit{1}] = \text{fix fix } \alpha \,.\, \mathit{1}}{\alpha \,.\, 1 \xrightarrow{\alpha}_{\alpha\bullet\mathit{1},\text{fix } \alpha\bullet\mathit{1}} P'}\ \text{act}}{\text{fix } \alpha \,.\, \mathit{1} \xrightarrow{\alpha}_{\text{fix } \alpha\bullet\mathit{1}} \text{fix fix } \alpha \,.\, \mathit{1}}\ \text{fix}}{\text{fix fix } \alpha \,.\, \mathit{1} \xrightarrow{\alpha}_{\varepsilon} \text{fix fix } \alpha \,.\, \mathit{1}}\ \text{fix}$$

# Fixing a behaviour

$$B(X, Y) = Y^{(X+1)^*} \times \mathcal{P}_{\omega 1}\left(\mathcal{A} \times Y^{(X+1)^*}\right)$$

# Defining $\varrho$

$$\varrho_{X,Y} \colon \Sigma(X \times B(X, Y)) \to B(X, \Sigma^\star(X + Y))$$

# Defining $\varrho$

$$\begin{aligned}\varrho_{X,Y} \colon \Sigma(X \times B(X, Y)) \to {} & (\Sigma^\star(X + Y))^{(X+1)^*} \times \\ & \mathcal{P}_{\omega 1}\left(A \times (\Sigma^\star(X + Y))^{(X+1)^*}\right)\end{aligned}$$

$$\varrho_{X,Y} = \langle \varrho^1_{X,Y}, \varrho^2_{X,Y} \rangle$$

# Defining $\varrho^1_{X,Y}$

$$
\begin{aligned}
\varrho^1_{X,Y} \colon \Sigma(X \times B(X,Y)) &\to (\Sigma^\star(X+Y))^{(X+1)^*} \\
\varnothing &\mapsto \lambda\, \_ .\, \varnothing \\
m &\mapsto \lambda\, P_0, P_1, \ldots, P_n . \begin{cases} \mathrm{fix}\ P & m \le n, P_m = \mathrm{inl}\ P \\ m & \text{otherwise} \end{cases} \\
\alpha \bullet (P, \sigma_P, \_) &\mapsto \lambda\, \xi .\, \alpha \bullet \mathrm{inr}(\sigma_P\, \xi) \\
\mathrm{fix}\ (P, \sigma_P, \_) &\mapsto \lambda\, \xi .\, \mathrm{fix}\ \mathrm{inr}(\sigma_P(\mathrm{inr}\ \top, \xi)) \\
&\vdots
\end{aligned}
$$

# Defining $\varrho^2_{X,Y}$

$$
\begin{aligned}
\varrho^2_{X,Y} \colon \Sigma(X \times B(X,Y)) &\to \mathcal{P}_{\omega 1}\left(\mathcal{A} \times (\Sigma^\star(X+Y))^{X+1^*}\right) \\
\varnothing &\mapsto \emptyset \\
m &\mapsto \emptyset \\
\alpha \bullet (P, \sigma_P, \_) &\mapsto \{(\alpha, \eta_Y \circ \mathrm{inr} \circ \sigma_P)\} \\
\mathrm{fix}\ (P, \sigma_P, b_P) &\mapsto \{(\alpha, \lambda \xi .\, \sigma(\mathrm{inl} \top, \xi)) \mid (\alpha, \sigma) \in b_P\} \\
&\vdots
\end{aligned}
$$

# Operational Model

A $B(\mu\Sigma, -)$-coalgebra $\iota^{\clubsuit} \colon \mu\Sigma \to B(\mu\Sigma, \mu\Sigma)$

$$
\begin{array}{ccc}
\Sigma\,\mu\Sigma & \xrightarrow{\qquad\qquad\qquad\iota\qquad\qquad\qquad} & \mu\Sigma \\
\Big\downarrow{\scriptstyle \Sigma\langle\, \mathrm{id}, \iota^{\clubsuit} \rangle} & & \Big\downarrow{\scriptstyle \langle\, \mathrm{id}, \iota^{\clubsuit} \rangle} \\
\Sigma(\mu\Sigma \times B(\mu\Sigma, \mu\Sigma)) \xrightarrow{\langle \iota \circ \Sigma\,\mathrm{fst}, \varrho_{\mu\Sigma,\mu\Sigma} \rangle} & \mu\Sigma \times B(\mu\Sigma, \Sigma^{\star}(\mu\Sigma + \mu\Sigma)) & \xrightarrow{\mathrm{id} \times B(\mathrm{id}, \hat{\iota} \circ \Sigma^{\star} \nabla)} \mu\Sigma \times B(\mu\Sigma, \mu\Sigma)
\end{array}
$$

# Operational Model

A $B(\mu\Sigma, -)$-coalgebra $\iota^{\clubsuit}\colon \mu\Sigma \to B(\mu\Sigma, \mu\Sigma)$

$$\begin{array}{ccc} \Sigma\,\mu\Sigma & \xrightarrow{\qquad\qquad\iota\qquad\qquad} & \mu\Sigma \\ \big\downarrow{\scriptstyle \Sigma\langle \mathrm{id},\iota^{\clubsuit}\rangle} & & \big\downarrow{\scriptstyle \langle \mathrm{id},\iota^{\clubsuit}\rangle} \\ \Sigma(\mu\Sigma\times B(\mu\Sigma,\mu\Sigma)) \xrightarrow{\langle \iota\circ\Sigma\,\mathrm{fst},\varrho_{\mu\Sigma,\mu\Sigma}\rangle} \mu\Sigma\times B(\mu\Sigma,\Sigma^{\star}(\mu\Sigma+\mu\Sigma)) & \xrightarrow{\mathrm{id}\times B(\mathrm{id},\hat{\iota}\circ\Sigma^{\star}\nabla)} & \mu\Sigma\times B(\mu\Sigma,\mu\Sigma) \end{array}$$

$\hat{\iota}\colon \Sigma^{\star}\,\mu\Sigma \to \mu\Sigma$

# Operational Model

A $B(\mu\Sigma, -)$-coalgebra $\iota^{\clubsuit}\colon \mu\Sigma \to B(\mu\Sigma, \mu\Sigma)$

$$\begin{array}{ccc} \Sigma\,\mu\Sigma & \xrightarrow{\quad\iota\quad} & \mu\Sigma \\ \big\downarrow{\scriptstyle \Sigma\langle \mathrm{id},\iota^{\clubsuit}\rangle} & & \big\downarrow{\scriptstyle \langle \mathrm{id},\iota^{\clubsuit}\rangle} \\ \Sigma(\mu\Sigma \times B(\mu\Sigma,\mu\Sigma)) \xrightarrow{\langle \iota\circ\Sigma\,\mathrm{fst},\, \varrho_{\mu\Sigma,\mu\Sigma}\rangle} \mu\Sigma \times B(\mu\Sigma, \Sigma^{\star}(\mu\Sigma + \mu\Sigma)) & \xrightarrow{\mathrm{id}\times B(\mathrm{id},\hat{\iota}\circ\Sigma^{\star}\nabla)} & \mu\Sigma \times B(\mu\Sigma,\mu\Sigma) \end{array}$$

$\hat{\iota}\colon \Sigma^{\star}\,\mu\Sigma \to \mu\Sigma$

$\nabla_X\colon X + X \to X$

# Strong Bisimulation

$$
\begin{array}{ccc}
\sim & \longrightarrow & \mu\Sigma \\
\downarrow & \lrcorner & \downarrow{\scriptstyle \text{coit } \iota^{\clubsuit}} \\
\mu\Sigma & \xrightarrow{\text{coit } \iota^{\clubsuit}} & \nu\gamma.\, B(\mu\Sigma, \gamma)
\end{array}
$$

## Proposition

$\sim$ *is a* $\Sigma$*-congruence*

# Formalizing in Agda

`https://wwwcip.cs.fau.de/~oc45ujef/ma/ccs/`

- de Bruijn indices inspired by PLFA[7]
- type of processes is a family $\mathrm{Proc}\colon \mathbb{N} \to$ Set
- $P\colon \mathrm{Proc}\ n$ means $P$ is a term with at most $n$ free variables

[7] Wadler, Kokke, and Siek, *Programming Language Foundations in Agda*.

Amadio, Roberto M. and Pierre-Louis Curien. *Domains and Lambda-Calculi*. Cambridge Tracts in Theoretical Computer Science. Cambridge University Press, 1998.

Barr, Michael. “Coequalizers and free triples”. In: *Mathematische Zeitschrift* 116 (1970), pp. 307–322.

Bloom, Bard, Sorin Istrail, and Albert R. Meyer. “Bisimulation can’t be traced”. In: *J. ACM* 42.1 (Jan. 1995), pp. 232–268. ISSN: 0004-5411. DOI: 10.1145/200836.200876. URL: https://doi.org/10.1145/200836.200876.

Goncharov, Sergey, Stefan Milius, Lutz Schröder, Stelios Tsampas, and Henning Urbat. “Higher-Order Mathematical Operational Semantics”. In: (2024).

Milner, Robin. *A Calculus of Communicating Systems*. 1980. DOI: https://doi.org/10.1007/3-540-10235-3.

Turi, D. and G. Plotkin. “Towards a mathematical operational semantics”. In: *Proceedings of Twelfth Annual IEEE Symposium on Logic in Computer Science*. 1997, pp. 280–291. DOI: 10.1109/LICS.1997.614955.

# Intermission: Free Algebra[8]

## Definition (Free $\Sigma$-Algebra)

Let $\Sigma\colon \mathcal{C} \to \mathcal{C}$, $X \in \mathrm{Ob}\,(\mathcal{C})$. A free $\Sigma$-Algebra of $X$ is a $\Sigma$-Algebra $(\Sigma^\star X, \iota_X)$ with a morphism $X\colon \Sigma^\star X$ such that

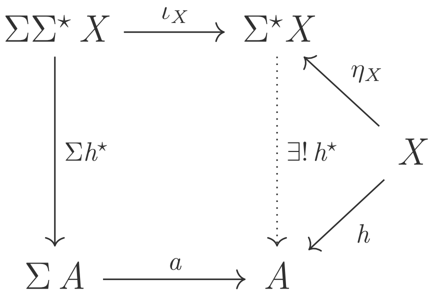
$$
\begin{array}{ccc}
\Sigma\Sigma^\star X & \xrightarrow{\ \iota_X\ } & \Sigma^\star X \\
 & & \nwarrow{\scriptstyle \eta_X} \\
 & & \qquad X
\end{array}
$$

[8] Barr, "Coequalizers and free triples".

# Intermission: Free Algebra[8]

## Definition (Free $\Sigma$-Algebra)

Let $\Sigma\colon \mathcal{C} \to \mathcal{C}$, $X \in \mathrm{Ob}\,(\mathcal{C})$. A free $\Sigma$-Algebra of $X$ is a $\Sigma$-Algebra $(\Sigma^\star X, \iota_X)$ with a morphism $X\colon \Sigma^\star X$ such that

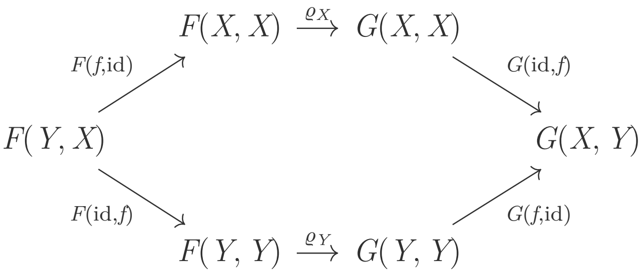



[8] Barr, "Coequalizers and free triples".

# Intermission: Dinatural Transformations

## Definition

Let $F, G\colon \mathcal{C}^{\mathrm{op}} \times \mathcal{C} \to \mathcal{D}$. A *dinatural transformation* $\varrho\colon F \stackrel{\bullet}{\Rightarrow} G$ is a family $\varrho_X\colon F(X, X) \to G(X, X)$ of morphisms such that




for all $f \in \mathrm{Hom}_{\mathcal{C}}(X, Y)$

# Bialgebras

## Definition

A $\varrho$-bialgebra $(X, a, b)$ consist of

- a $\Sigma$-algebra $(X, a\colon \Sigma\, X \to X)$
- a $B$-coalgebra $(X, b\colon X \to B\, X)$

such that

$$\begin{array}{ccccc} \Sigma\, X & \xrightarrow{\quad a \quad} & X & \xrightarrow{\; b \;} & B\, X \\ \Big\downarrow{\scriptstyle \Sigma\langle \mathrm{id}, b\rangle} & & & & \Big\uparrow{\scriptstyle B\hat{a}} \\ \Sigma(X \times B\, X) & & \xrightarrow{\qquad \varrho_X \qquad} & & B\Sigma^\star\, X \end{array}$$

# Bialgebras

## Definition

A $\varrho$-bialgebra $(X, a, b)$ consist of

- a $\Sigma$-algebra $(X, a\colon \Sigma\, X \to X)$
- a $B$-coalgebra $(X, b\colon X \to B\, X)$

such that

Eilenberg-Moore algebra induced by $a$

$$\begin{array}{ccccc}
\Sigma\, X & \xrightarrow{\;a\;} & X & \xrightarrow{\;b\;} & B\, X \\
\Big\downarrow{\scriptstyle \Sigma\langle \mathrm{id}, b\rangle} & & & & \Big\uparrow{\scriptstyle B\hat{a}} \\
\Sigma(X \times B\, X) & & \xrightarrow{\quad \varrho_X \quad} & & B\Sigma^{\star}\, X
\end{array}$$

# Denotational Model

If $B$ has a final coalgebra $(\nu B, \tau)$, we obtain the *final* $\varrho$-bialgebra $(\nu B, \alpha, \tau)$:

$$
\begin{array}{ccccc}
\Sigma\,\nu B & \xrightarrow{\quad\alpha\quad} & \nu B & \xrightarrow{\tau} & B\,\nu B \\
\Big\downarrow{\scriptstyle \Sigma\langle\,\mathrm{id},\tau\rangle} & & & & \Big\uparrow{\scriptstyle B\hat{\alpha}} \\
\Sigma(\nu B \times B\,\nu B) & & \xrightarrow{\qquad\varrho_{\nu B}\qquad} & & B\Sigma^{\star}\,\nu B
\end{array}
$$

# Denotational Model

If $B$ has a final coalgebra $(\nu B, \tau)$, we obtain the *final* $\varrho$-bialgebra $(\nu B, \alpha, \tau)$: