

# B-systems and C-systems are equivalent

Jacopo Emmenegger  
Università degli Studi di Genova

joint work with Benedikt Ahrens, Paige Randall North  
and Egbert Rijke

Workshop on Syntax and Semantics of Type Theory  
Stockholms universitet  
20–21 May 2022

## C-systems (a.k.a. contextual categories<sup>1</sup>)

A category  $\mathcal{C}$  with a terminal object  $1$  together with structure such that:

1. the objects of  $\mathcal{C}$  can be arranged into a rooted tree  $R(\mathcal{C})$  (with root  $1$ ) that embeds into  $\mathcal{C}$ , and
2. every cospan in  $\mathcal{C}$  can be completed to a pullback if one of the arrows, say  $p$ , is coming from  $R(\mathcal{C})$ . This choice is functorial in the other arrow, and such that the chosen pullback of  $p$  is also coming from  $R(\mathcal{C})$ .

---

<sup>1</sup>Cartmell. Generalised algebraic theories and contextual categories. 1986.

## C-systems (a.k.a. contextual categories<sup>1</sup>)

A category  $\mathcal{C}$  with a terminal object  $1$  together with structure such that:

1. the objects of  $\mathcal{C}$  can be arranged into a rooted tree  $R(\mathcal{C})$  (with root  $1$ ) that embeds into  $\mathcal{C}$ , and
2. every cospan in  $\mathcal{C}$  can be completed to a pullback if one of the arrows, say  $p$ , is coming from  $R(\mathcal{C})$ . This choice is functorial in the other arrow, and such that the chosen pullback of  $p$  is also coming from  $R(\mathcal{C})$ .

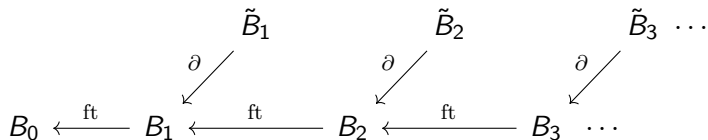
Think of:

1. The **category of contexts** of a dependent type theory. Here display maps  $(\Gamma, x : A) \rightarrow \Gamma$  are functorially stable under pullback.
2. A **category with display maps** à la Taylor  $(\mathcal{C}, \mathcal{D})$ , where the class of display maps  $\mathcal{D}$  form a rooted tree and its stability under pullback is functorial. N.B.  $\mathcal{D}$  is *not* required to be closed under composition.

<sup>1</sup>Cartmell. Generalised algebraic theories and contextual categories. 1986.

## B-systems<sup>2</sup>

A B-frame  $\mathbb{B}$ :



think of:

$A \in B_{n+1}$  as  $v_1 : \text{ft}^n(A), \dots, v_n : \text{ft}(A) \vdash A$  type

$a \in \tilde{B}_{n+1}$  as  $v_1 : \text{ft}^n(A), \dots, v_n : \text{ft}(A) \vdash a : A$ , where  $A := \partial(a)$

---

<sup>2</sup>Voevodsky. B-systems. 2014

## B-systems<sup>2</sup>

A B-frame  $\mathbb{B}$ :

$$\begin{array}{ccccccc} & & & \tilde{B}_1 & & \tilde{B}_2 & & \tilde{B}_3 & \dots \\ & & & \swarrow \partial & & \swarrow \partial & & \swarrow \partial & \\ \{*\} \cong B_0 & \xleftarrow{\text{ft}} & B_1 & \xleftarrow{\text{ft}} & B_2 & \xleftarrow{\text{ft}} & B_3 & \dots \end{array}$$

think of:

$A \in B_{n+1}$  as  $v_1 : \text{ft}^n(A), \dots, v_n : \text{ft}(A) \vdash A$  type

$a \in \tilde{B}_{n+1}$  as  $v_1 : \text{ft}^n(A), \dots, v_n : \text{ft}(A) \vdash a : A$ , where  $A := \partial(a)$

---

<sup>2</sup>Voevodsky. B-systems. 2014

## B-systems<sup>2</sup>

A B-frame  $\mathbb{B}$ :

$$\begin{array}{ccccccc} & & & \tilde{B}_1 & & \tilde{B}_2 & & \tilde{B}_3 & \dots \\ & & & \swarrow \partial & & \swarrow \partial & & \swarrow \partial & \\ \{*\} \cong B_0 & \xleftarrow{\text{ft}} & B_1 & \xleftarrow{\text{ft}} & B_2 & \xleftarrow{\text{ft}} & B_3 & \dots \end{array}$$

think of:

$A \in B_{n+1}$  as  $v_1 : \text{ft}^n(A), \dots, v_n : \text{ft}(A) \vdash A$  type

$a \in \tilde{B}_{n+1}$  as  $v_1 : \text{ft}^n(A), \dots, v_n : \text{ft}(A) \vdash a : A$ , where  $A := \partial(a)$

together with, for  $x \in \tilde{B}_{n+1}$ ,  $X \in B_n$ , homomorphisms of B-frames

$$\mathbb{B}/\partial(x) \xrightarrow{S_x} \mathbb{B}/\text{ft}\partial(x) \qquad \mathbb{B}/\text{ft}(X) \xrightarrow{W_X} \mathbb{B}/X$$

and a function  $\delta_n : B_{n+1} \rightarrow \tilde{B}_{n+2}$  for every  $n$ .

Plus a number of equations.

<sup>2</sup>Voevodsky. B-systems. 2014

# Why B-systems?



## Why B-systems?

In order to construct the initial model for a dependent type theory à la Martin-Löf:

1. From a suitable signature construct a monad of preterms  $R: \mathbf{Set} \rightarrow \mathbf{Set}$  and a left module of pretypes  $LM: \mathbf{Set} \rightarrow \mathbf{Set}$  over  $R$ .<sup>3</sup>
2. Construct a C-system  $C(R, LM)$  of preterms and pretypes.<sup>2</sup>
3. Use subsystems and quotients of C-systems<sup>4</sup> to carve out well-formed types and well-typed terms modulo a convertibility relation.

It resembles usual presentations of Martin-Löf type theory, but:

- ▶ it is done for an arbitrary signature,
- ▶ it directly produces a C-system.

---

<sup>3</sup>Voevodsky. C-system of a module over a  $Jf$ -relative monad. 2016.

<sup>4</sup>Voevodsky. Subsystems and regular quotients of C-systems. 2016.



## Why B-systems?

To address step 3 (carving out well-formed types and terms) need to develop a theory of subsystems and quotients of C-systems.

Use B-systems for this: there is a bijection<sup>5</sup> between

- ▶ subsystems of a C-system  $\mathbb{C}$ , and
- ▶ subsystems of a certain B-system  $B(\mathbb{C})$ , *i.e.* subsets of the  $B_n$ 's and  $\tilde{B}_{n+1}$ 's closed under substitution, weakening and generic elements.

and similarly for quotients.

B-systems are algebras for a monad on a category of presheaves.<sup>6</sup>

---

<sup>5</sup>Voevodsky. Subsystems and regular quotients of C-systems. 2016.

<sup>6</sup>Garner. Combinatorial structure of type dependency. 2014.

## Why B-systems?

To address step 3 (carving out well-formed types and terms) need to develop a theory of subsystems and quotients of C-systems.

Use B-systems for this: there is a bijection<sup>5</sup> between

- ▶ subsystems of a C-system  $\mathbb{C}$ , and
- ▶ subsystems of a certain B-system  $B(\mathbb{C})$ , *i.e.* subsets of the  $B_n$ 's and  $\tilde{B}_{n+1}$ 's closed under substitution, weakening and generic elements.

and similarly for quotients.

B-systems are algebras for a monad on a category of presheaves.<sup>6</sup>

### Conjecture (Voevodsky)

The categories of B-systems and C-systems are equivalent.

---

<sup>5</sup>Voevodsky. Subsystems and regular quotients of C-systems. 2016.

<sup>6</sup>Garner. Combinatorial structure of type dependency. 2014.

## Why B-systems?

To address step 3 (carving out well-formed types and terms) need to develop a theory of subsystems and quotients of C-systems.

Use B-systems for this: there is a bijection<sup>5</sup> between

- ▶ subsystems of a C-system  $\mathbb{C}$ , and
- ▶ subsystems of a certain B-system  $B(\mathbb{C})$ , *i.e.* subsets of the  $B_n$ 's and  $\tilde{B}_{n+1}$ 's closed under substitution, weakening and generic elements.

and similarly for quotients.

B-systems are algebras for a monad on a category of presheaves.<sup>6</sup>

### Theorem (Ahrens–E.–North–Rijke)

The categories of B-systems and C-systems are equivalent.

---

<sup>5</sup>Voevodsky. Subsystems and regular quotients of C-systems. 2016.

<sup>6</sup>Garner. Combinatorial structure of type dependency. 2014.

## B-systems: equations

1. Substitution distributes over substitution and over weakening:

$$S_x S_y = S_{S_x(y)} S_x \quad S_x W_X = W_{S_x(X)} S_x$$

where  $x \in \tilde{B}_{n+1}$  and  $y \in \tilde{B}_{m+n+1}$  s.t.  $\text{ft}^m \partial(y) = \partial(x)$ ,  
and it preserves generic elements

$$S_x \delta = \delta S_x$$

2. Similarly for weakening.

$$W_X S_y = S_{W_X(y)} W_X \quad W_X W_Y = W_{W_X(Y)} W_X \quad W_X \delta_n = \delta_{n+1} W_X$$

3. For  $x \in \tilde{B}_{n+1}$  and  $X \in B_{n+1}$ :

$$S_x W_{\partial(x)} = \text{id}_{\mathbb{B}/\partial(x)} \quad S_x(\delta \partial(x)) = x \quad S_{\delta(x)}(W_X/X) = \text{id}_{\mathbb{B}/X}$$

## B-systems: rooted tree

$$\{*\} \longleftarrow B_1 \xleftarrow{\text{ft}} B_2 \xleftarrow{\text{ft}} B_3 \quad \dots$$

is a rooted tree  $\mathbf{T}(\mathbb{B})$  with graph given by:

$$V := \coprod_n B_n$$

$$(m, X) \rightarrow (n, Y) \quad \text{iff} \quad m = n + 1 \text{ and } Y = \text{ft}(X).$$

$\mathbf{T}(\mathbb{B})$  encodes one-step type dependencies.

## Overview

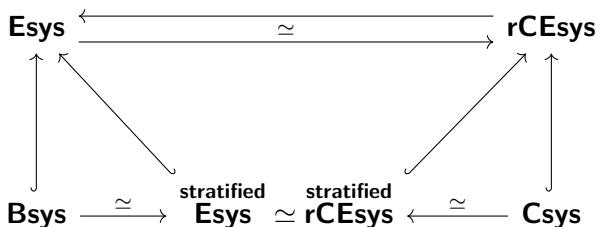
The functor  $\mathbf{Bsys} \rightarrow \mathbf{Csys}$  will construct the “syntactic category” of a B-system.

1. Define contexts and context morphisms (i.e. tuples of well-typed terms).
2. Define composition of context morphisms by means of substitution.

## Overview

The functor  $\mathbf{Bsys} \rightarrow \mathbf{Csys}$  will construct the “syntactic category” of a B-system.

1. Define contexts and context morphisms (i.e. tuples of well-typed terms).
2. Define composition of context morphisms by means of substitution.



In the top row, replace the rooted tree of single-step type dependencies with an arbitrary (strict) category of multi-step type dependencies.

## Stratification

A (strict) category with terminal object 1 is **stratified** if there is a functor  $L: \mathcal{C} \rightarrow (\mathbb{N}, \geq)$  such that

1.  $L(X) = 0$  if and only if  $X$  is the chosen terminal object 1,
2. for any  $f: X \rightarrow Y$  we have  $L(X) = L(Y)$  if and only if  $X = Y$  and  $f = \text{id}_X$ , and
3. every morphism  $f: X \rightarrow Y$  in  $\mathcal{C}$ , where  $L(X) = n + m + 1$  and  $L(Y) = n$ , has a unique factorization

$$X = X_{m+1} \xrightarrow{f_m} X_m \xrightarrow{f_{m-1}} \cdots \xrightarrow{f_1} X_1 \xrightarrow{f_0} X_0 = Y$$

where  $L(X_{i+1}) = n + i + 1 = L(X_i) + 1$ .

### Proposition

1. Being stratified is a property.



## Stratification

A (strict) category with terminal object 1 is **stratified** if there is a functor  $L: \mathcal{C} \rightarrow (\mathbb{N}, \geq)$  such that

1.  $L(X) = 0$  if and only if  $X$  is the chosen terminal object 1,
2. for any  $f: X \rightarrow Y$  we have  $L(X) = L(Y)$  if and only if  $X = Y$  and  $f = \text{id}_X$ , and
3. every morphism  $f: X \rightarrow Y$  in  $\mathcal{C}$ , where  $L(X) = n + m + 1$  and  $L(Y) = n$ , has a unique factorization

$$X = X_{m+1} \xrightarrow{f_m} X_m \xrightarrow{f_{m-1}} \cdots \xrightarrow{f_1} X_1 \xrightarrow{f_0} X_0 = Y$$

where  $L(X_{i+1}) = n + i + 1 = L(X_i) + 1$ .

### Proposition

1. Being stratified is a property.
2. The category of stratified categories is equivalent to the category of rooted trees via the free category functor.

# CE-systems

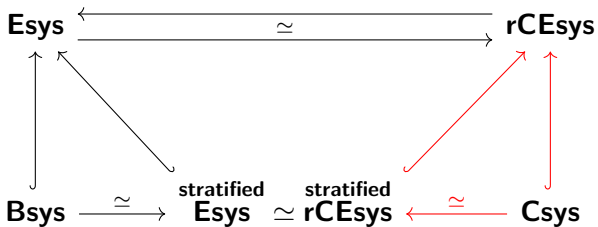
A **CE-system** consists of

1. two strict category structures  $\mathcal{F}$  and  $\mathcal{C}$  on the same set of objects  $\text{Ob}(\mathcal{F}) = \text{Ob}(\mathcal{C})$ ,
2. an identity-on-objects functor  $I: \mathcal{F} \rightarrow \mathcal{C}$  between them,
3. a chosen object  $\top$  which is terminal in  $\mathcal{F}$ , and
4. for every  $f: \Delta \rightarrow \Gamma$  in  $\mathcal{C}$  and any  $A \in \mathcal{F}/\Gamma$ , a choice of a pullback square

$$\begin{array}{ccc} \Delta.f^*A & \xrightarrow{\pi_2(f,A)} & \Gamma.A \\ I(f^*A) \downarrow & & \downarrow I(A) \\ \Delta & \xrightarrow{f} & \Gamma \end{array}$$

which is functorial in  $A$  and  $f$ .

A CE-system is **rooted** if  $I(\top) = \top$  is terminal in  $\mathcal{C}$ .



C-system:

- ▶ Functorial choice of pullbacks.
- ▶ Rooted tree of type dependency.

CE-system:

- ▶ Functorial choice of pullbacks.
- ▶ If stratified: rooted tree of type dependency.

## E-systems

An **E-system**  $\mathbb{E}$  consists of a category  $\mathcal{F}$  with a terminal object  $\square$  together with:

1. A **term structure**: for every  $A \in \mathcal{F}/\Gamma$ , a set  $T(A)$ .
2. A **substitution structure**: for every  $A \in \mathcal{F}/\Gamma$  and  $x \in T(A)$ , a functor

$$S_x: \mathcal{F}/\Gamma.A \rightarrow \mathcal{F}/\Gamma$$

with term structure: for every  $B \in \mathcal{F}/\Gamma.A$ , a function

$$T(B) \rightarrow T(S_x(B)).$$

3. A **weakening structure**: for every  $A \in \mathcal{F}/\Gamma$ , a functor with term structure  $W_A: \mathbb{E}/\Gamma \rightarrow \mathbb{E}/\Gamma.A$ .
4. A **projection structure**: for every  $A \in \mathcal{F}/\Gamma$ , an element  $\text{idtm}_A \in T(W_A(A))$ .

Satisfying equations similar to those of B-systems, plus:

$S_x$  and  $W_A$  preserve terminal objects, and

$W_A$  is functorial in  $A$ , i.e.  $W_{\text{id}_\Gamma} = \text{Id}_{\mathbb{E}/\Gamma}$  and  $W_{AP} = W_P W_A$ .

## From B-systems to E-systems

$\mathbb{B}$  a B-system.  $\mathcal{F}$  is the free category on the (graph of the) rooted tree  $\mathbf{T}(\mathbb{B})$  of single-step type dependencies.

Arrows in  $\mathcal{F}$  are of the form  $(X, k): (n + k, X) \rightarrow (n, \text{ft}^k(X))$ .

## From B-systems to E-systems

$\mathbb{B}$  a B-system.  $\mathcal{F}$  is the free category on the (graph of the) rooted tree  $\mathbf{T}(\mathbb{B})$  of single-step type dependencies.

Arrows in  $\mathcal{F}$  are of the form  $(X, k): (n+k, X) \rightarrow (n, \text{ft}^k(X))$ .

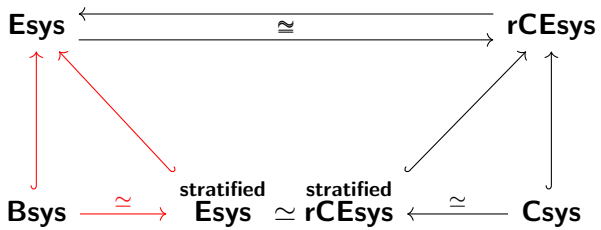
The term structure  $T(X, k)$  is defined inductively together with simultaneous substitutions  $S_t^k$  of tuples of terms  $t \in T(X, k)$ .

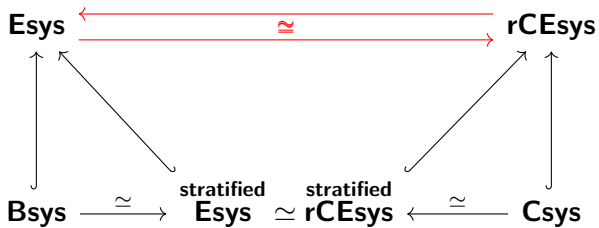
- ▶ On arrows of length 0 and 1: for  $X \in B_n$

$$\begin{aligned} T(X, 0) &:= \{*\} & S_*^0 &:= \text{id}: \mathbb{B}/X \rightarrow \mathbb{B}/X \\ T(X, 1) &:= \partial^{-1}(X) \subseteq \tilde{B}_n & S_x^1 &:= S_x: \mathbb{B}/X \rightarrow \mathbb{B}/\text{ft}(X) \end{aligned}$$

- ▶ For  $X \in B_n$  and  $k \leq n$  by induction:  
suppose that for every  $m \leq n$  and  $Y \in B_m$  we have  $T(Y, k)$  for  $k \leq m$  and  $S_s^k: \mathbb{B}/Y \rightarrow \mathbb{B}/\text{ft}^k(Y)$  for  $s \in T(Y, k)$ , then

$$T(X, k+1) := \coprod_{t \in T(\text{ft}(X), k)} T(S_t^k(X), 1) \quad S_{(t,x)}^{k+1} := S_x(S_t^k/X)$$







## From CE-systems to E-systems

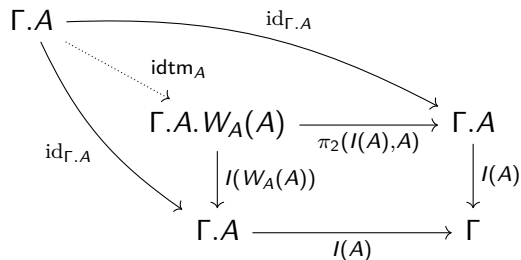
$I: \mathcal{F} \rightarrow \mathcal{C}$  a CE-system.

Define an E-system on  $\mathcal{F}$  as follows:

$$T(A) := \{x: \Gamma \rightarrow \Gamma.A \mid I(A)x = \text{id}_\Gamma\}, \quad \text{for } A \in \mathcal{F}/\Gamma$$

$$W_A := A^* : \mathcal{F}/\Gamma \rightarrow \mathcal{F}/\Gamma.A, \quad \text{for } A \in \mathcal{F}/\Gamma$$

$$S_x := x^* : \mathcal{F}/\Gamma.A \rightarrow \mathcal{F}/\Gamma, \quad \text{for } x \in T(A)$$



## From E-systems to CE-systems

$\mathbb{E}$  an E-system.

For  $A, B \in \mathcal{F}/\Gamma$ , an **internal morphism over  $\Gamma$**  is an element of

$$\text{thom}(A, B) := T(W_A(B))$$

## From E-systems to CE-systems

$\mathbb{E}$  an E-system.

For  $A, B \in \mathcal{F}/\Gamma$ , an **internal morphism over  $\Gamma$**  is an element of

$$\text{thom}(A, B) := T(W_A(B))$$

Note that, given also  $P \in \mathcal{F}/\Gamma.A$ :

$$\text{thom}(PA, B) = T(W_{PA}(B)) = T(W_P W_A(B)) = \text{thom}(P, W_A(B))$$

Indeed once we have a category of internal morphisms

$$(-) \circ A \dashv W_A$$

## From E-systems to CE-systems

$\mathbb{E}$  an E-system.

For  $A, B \in \mathcal{F}/\Gamma$ , an **internal morphism over  $\Gamma$**  is an element of

$$\text{thom}(A, B) := T(W_A(B))$$

Given  $f \in \text{thom}(A, B)$ , define **precomposition by  $f$**  as

$$\begin{array}{ccc} \mathbb{E}/\Gamma.B & \xrightarrow{f^*} & \mathbb{E}/\Gamma.A \\ & \searrow_{W_A/B} & \nearrow_{S_f} \\ & \mathbb{E}/\Gamma.A.W_A(B) & \end{array}$$

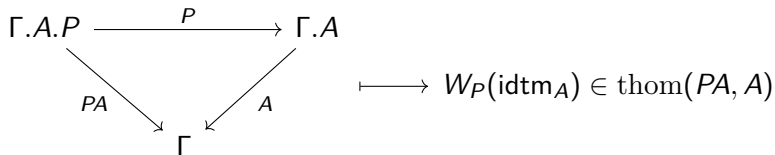
Then if  $g \in \text{thom}(B, C)$ , define  $g \circ f := f^*(g)$ .

Identities are given by  $\text{idtm}_A \in T(W_A(A)) = \text{thom}(A, A)$ .

Obtain a **category  $\mathcal{C}_{\mathbb{E}}(\Gamma)$**  of internal morphisms over  $\Gamma$ .

## From E-systems to CE-systems

Define  $I_{\mathbb{E}}^{\Gamma}: \mathcal{F}/\Gamma \rightarrow \mathcal{C}_{\mathbb{E}}(\Gamma)$  as



### Proposition

$\mathbb{E}$  and E-system. For every  $\Gamma$ ,

$$\mathcal{F}/\Gamma \xrightarrow{I_{\mathbb{E}}^{\Gamma}} \mathcal{C}_{\mathbb{E}}(\Gamma)$$

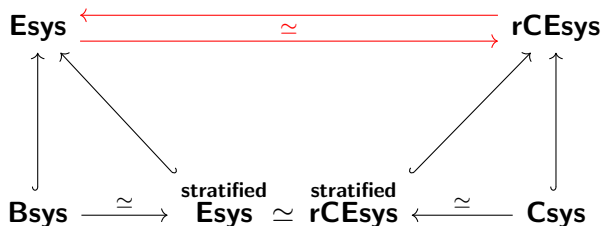
is a rooted CE-system.

The choice of pullbacks is also given algebraically using the operations of  $\mathbb{E}$ .

# Equivalence

## Theorem (Ahrens–E.–North–Rijke)

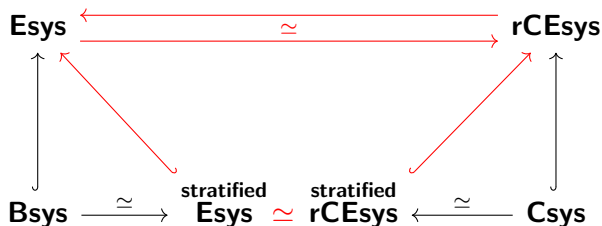
1. The functors  $\mathbf{Esys} \rightleftarrows \mathbf{rCEsys}$  give rise to an equivalence of categories.



# Equivalence

## Theorem (Ahrens–E.–North–Rijke)

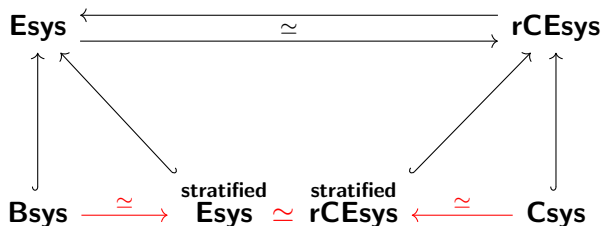
1. The functors  $\mathbf{Esys} \rightleftarrows \mathbf{rCEsys}$  give rise to an equivalence of categories.
2. The equivalence restricts to stratified systems.



# Equivalence

## Theorem (Ahrens–E.–North–Rijke)

1. The functors  $\mathbf{Bsys} \rightleftarrows \mathbf{rCEsys}$  give rise to an equivalence of categories.
2. The equivalence restricts to stratified systems.
3. It follows that  $\mathbf{Bsys} \equiv \mathbf{Csys}$ .



Details in:

*B-systems and C-systems are equivalent.* arXiv:2111.09948