

Nondeterministic Asynchronous Dataflow in Isabelle/HOL

Rafael Castro G. Silva, Laouen Fernet and Dmitriy Traytel

Department of Computer Science
University of Copenhagen

14/05/2025

Motivation

Motivation

Context:

- Stream Processing: programs that compute (possibly) unbounded sequences of data (streams)
- A common problem in the industry
- Frameworks: Apache Flink, Google Cloud Dataflow, and Timely Dataflow



- Why use frameworks?
 - Highly Parallel
 - Low latency (output as soon as possible)
 - Incremental computing (re-uses previous computations)

Motivation

Context:

- Stream Processing: programs that compute (possibly) unbounded sequences of data (streams)
- A common problem in the industry
- Frameworks: Apache Flink, Google Cloud Dataflow, and Timely Dataflow



- Why use frameworks?
 - Highly Parallel
 - Low latency (output as soon as possible)
 - Incremental computing (re-uses previous computations)

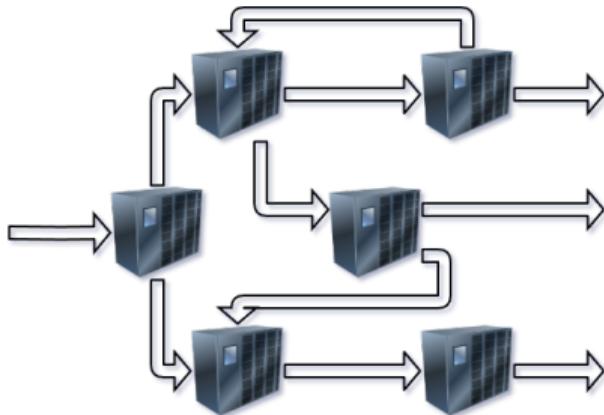
Our goal:

Mechanically Verify Timely Dataflow algorithms

A Good Foundation

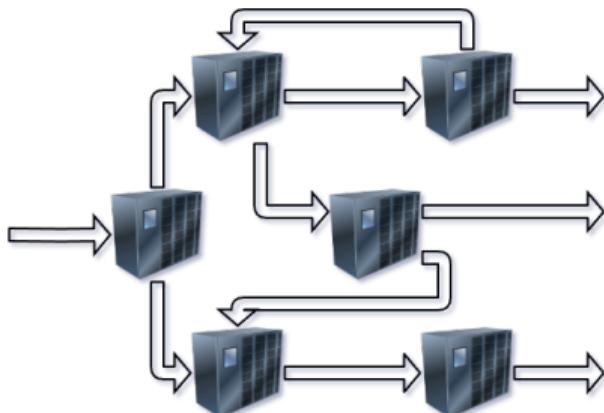
A Good Foundation

- Nondeterministic Asynchronous Dataflow

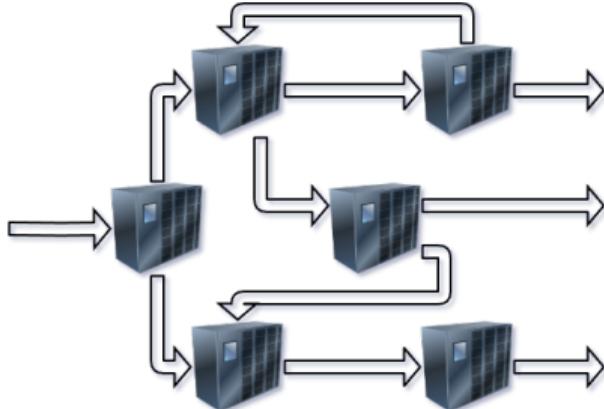


A Good Foundation

- Nondeterministic Asynchronous Dataflow
 - Dataflow: Directed graph of interconnected operators

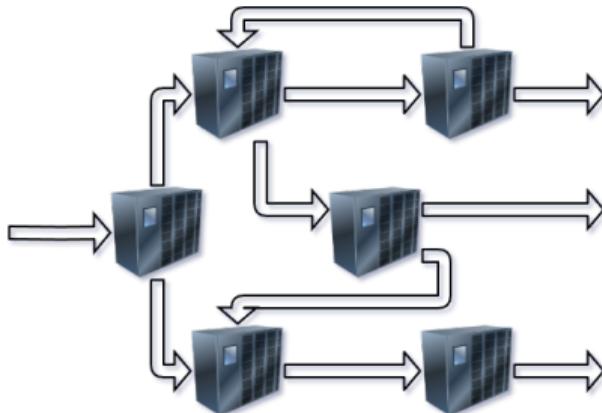


A Good Foundation



- Nondeterministic Asynchronous Dataflow
 - Dataflow: Directed graph of interconnected operators
 - Asynchronous:
 - Operators execute independently: processes without an orchestrator
 - Operators can freely communicate with the network (read/write); do silent computation steps
 - Networks are unbounded FIFO queues

A Good Foundation



- Nondeterministic Asynchronous Dataflow
 - Dataflow: Directed graph of interconnected operators
 - Asynchronous:
 - Operators execute independently: processes without an orchestrator
 - Operators can freely communicate with the network (read/write); do silent computation steps
 - Networks are unbounded FIFO queues
 - Nondeterministic:
 - Operators can make nondeterministic choices

The Algebra for Nondeterministic Asynchronous Dataflow

- Bergstra et al. presents an algebra for Nondeterministic Asynchronous Dataflow
- Primitives:
sequential and parallel composition; feedback loop...
- 52 axioms
- A process calculus instance

Network Algebra for Asynchronous Dataflow*

J.A. Bergstra^{1,2,†} C.A. Middelburg^{2,3,§} Gh. Ștefănescu^{4,‡}

¹Programming Research Group, University of Amsterdam
P.O. Box 41882, 1009 DB Amsterdam, The Netherlands

²Department of Philosophy, Utrecht University
P.O. Box 80126, 3508 TC Utrecht, The Netherlands

³Department of Network & Service Control, KPN Research
P.O. Box 421, 2260 AK Leidschendam, The Netherlands

⁴Institute of Mathematics of the Romanian Academy
P.O. Box 1-764, 70700 Bucharest, Romania

E-mail: janb@fwi.uva.nl - keesm@phil.ruu.nl - ghstef@imar.ro

Main Contributions

- A Isabelle/HOL instance of Nondeterministic Asynchronous Dataflow
 - Operators as a shallow embedding as codatatypes
 - 51 axioms proved
- Executable via code extraction to Haskell

Isabelle/HOL Preliminaries

Isabelle/HOL

- Classical higher-order logic (HOL):
Simple Typed Lambda Calculus + axiom of choice + axiom of infinity + rank-1 polymorphism

Isabelle/HOL

- Classical higher-order logic (HOL):
Simple Typed Lambda Calculus + axiom of choice + axiom of infinity + rank-1 polymorphism
- Isabelle: A generic proof assistant



- Isabelle/HOL: Isabelle's flavor of HOL

Isabelle/HOL

- Classical higher-order logic (HOL):
Simple Typed Lambda Calculus + axiom of choice + axiom of infinity + rank-1 polymorphism
- Isabelle: A generic proof assistant



- Isabelle/HOL: Isabelle's flavor of HOL

Why Isabelle/HOL?

- Codatatypes: (possibly) infinite data structures (e.g., lazy lists, streams)
- Corecursion: always eventually produces some codatatype constructor
- Coinductive predicate: infinite number of introduction rule applications
- Coinduction: reason about coinductive predicates

Operators as a Codatatype

Operators

Operators in Isabelle/HOL

```
codatatype (inputs: 'i, outputs: 'o, 'd) op =  
  Read 'i ('d ⇒ ('i, 'o, 'd) op) | Write (('i, 'o, 'd) op) 'o 'd  
  Silent ('i, 'o, 'd) op | Choice (('i, 'o, 'd) op) cset
```

Operators

Operators in Isabelle/HOL

```
codatatype (inputs: 'i, outputs: 'o, 'd) op =  
  Read 'i ('d ⇒ ('i, 'o, 'd) op) | Write (('i, 'o, 'd) op) 'o 'd  
  Silent ('i, 'o, 'd) op | Choice (('i, 'o, 'd) op) cset
```

- Type parameters:
inputs/output ports; data
- Operator's actions
- inputs/outputs:
Sets of used ports
- Possibly infinite trees

Operators

Operators in Isabelle/HOL

```
codatatype (inputs: 'i, outputs: 'o, 'd) op =  
  Read 'i ('d ⇒ ('i, 'o, 'd) op) | Write (('i, 'o, 'd) op) 'o 'd  
  Silent ('i, 'o, 'd) op | Choice (('i, 'o, 'd) op) cset
```

- Type parameters:
inputs/output ports; data
- Operator's actions
- inputs/outputs:
Sets of used ports
- Possibly infinite trees

Uncommunicative
operators

$\emptyset = \text{Choice } \{\}^c$

$\odot = \text{Silent } \odot$

$\otimes = \text{Choice } \{\otimes\}^c$

Operators

Operators in Isabelle/HOL

```
codatatype (inputs: 'i, outputs: 'o, 'd) op =  
  Read 'i ('d ⇒ ('i, 'o, 'd) op) | Write (('i, 'o, 'd) op) 'o 'd  
  Silent ('i, 'o, 'd) op | Choice (('i, 'o, 'd) op) cset
```

- Type parameters:
inputs/output ports; data
- Operator's actions
- inputs/outputs:
Sets of used ports
- Possibly infinite trees

Uncommunicative operators

$$\emptyset = \text{Choice } \{\}^c$$

$$\odot = \text{Silent } \odot$$

$$\otimes = \text{Choice } \{\otimes\}^c$$

More examples

$$\text{ex1} = \text{Choice } \{\text{Write ex1 1 42}, \emptyset\}^c$$

$$\text{ex2} = \text{Choice } \{\text{Write ex2 1 42}, \text{ex2}\}^c$$

$$\text{ex3} = \text{Choice } \{\text{Write ex3 1 42}, \text{Silent ex3}\}^c$$

An ideal equivalence relation

- Only equate operators that **behave** the same
- Useful reasoning principle

Operators Equivalences: Motivation

An ideal equivalence relation

- Only equate operators that **behave** the same
- Useful reasoning principle

- Milner's approach: Weak Bisimilarity



- Based on labeled transition systems (LTS)
- Labels (actions): Read, Write, Silent (τ)

Operators Equivalences: Label Transition System

Label Transition System

```
datatype ('i,'o,'d) IO = Inp 'i 'd | Out 'o 'd | Tau
```

inductive step where

```
step (Inp p x) (Read p f) (f x)
| step (Out q x) (Write op q x) op
| step Tau (Silent op) op
| op ∈c ops ⇒ step io op op' ⇒ step io (Choice ops) op'
```

Operators Equivalences: Weak Bisimilarity

Weakly Simulates

$$\text{estep } \text{Tau} = (\text{step } \text{Tau})^{==}$$

$$\text{estep } io = \text{step } io$$

$$\text{wstep } io = (\text{step } \text{Tau})^{**} \text{ OO } (\text{estep } io) \text{ OO } (\text{step } \text{Tau})^{**}$$

$$\text{wsim } R \text{ } op_1 \text{ } op_2 = (\forall io \text{ } op'_1. \text{ step } io \text{ } op_1 \text{ } op'_1 \longrightarrow (\exists op'_2. \text{ wstep } io \text{ } op_2 \text{ } op'_2 \wedge R \text{ } op'_1 \text{ } op'_2))$$

- A Relation R is a *weak simulation* if $\forall op_1 \text{ } op_2. \text{ } R \text{ } op_1 \text{ } op_2 \longrightarrow \text{wsim } R \text{ } op_1 \text{ } op_2$.

Operators Equivalences: Weak Bisimilarity

Weakly Simulates

$$\text{estep } \text{Tau} = (\text{step } \text{Tau})^{==}$$

$$\text{estep } io = \text{step } io$$

$$\text{wstep } io = (\text{step } \text{Tau})^{**} \text{ OO } (\text{estep } io) \text{ OO } (\text{step } \text{Tau})^{**}$$

$$\text{wsim } R \text{ } op_1 \text{ } op_2 = (\forall io \text{ } op'_1. \text{ step } io \text{ } op_1 \text{ } op'_1 \longrightarrow (\exists op'_2. \text{ wstep } io \text{ } op_2 \text{ } op'_2 \wedge R \text{ } op'_1 \text{ } op'_2))$$

- A Relation R is a *weak simulation* if $\forall op_1 \text{ } op_2. R \text{ } op_1 \text{ } op_2 \longrightarrow \text{wsim } R \text{ } op_1 \text{ } op_2$.

Weak Bisimilarity

coinductive wbisim (**infix** ≈ 40) **where**

$$\text{wsim } (\approx) \text{ } op_1 \text{ } op_2 \implies \text{wsim } (\approx) \text{ } op_2 \text{ } op_1 \implies op_1 \approx op_2$$

- R is a *weak bisimulation* when both R and its converse R^{-1} are weak simulations.
Weak bisimilarity is the largest weak bisimulation.

Operators Equivalences: Weak Bisimilarity

Weakly Simulates

$$\text{estep } \text{Tau} = (\text{step } \text{Tau})^{==}$$

$$\text{estep } io = \text{step } io$$

$$\text{wstep } io = (\text{step } \text{Tau})^{**} \text{ OO } (\text{estep } io) \text{ OO } (\text{step } \text{Tau})^{**}$$

$$\text{wsim } R \text{ } op_1 \text{ } op_2 = (\forall io \text{ } op'_1. \text{ step } io \text{ } op_1 \text{ } op'_1 \longrightarrow (\exists op'_2. \text{ wstep } io \text{ } op_2 \text{ } op'_2 \wedge R \text{ } op'_1 \text{ } op'_2))$$

- A Relation R is a *weak simulation* if $\forall op_1 \text{ } op_2. R \text{ } op_1 \text{ } op_2 \longrightarrow \text{wsim } R \text{ } op_1 \text{ } op_2$.

Weak Bisimilarity

coinductive `wbisim` (infix ≈ 40) **where**

$$\text{wsim } (\approx) \text{ } op_1 \text{ } op_2 \implies \text{wsim } (\approx) \text{ } op_2 \text{ } op_1 \implies op_1 \approx op_2$$

- R is a *weak bisimulation* when both R and its converse R^{-1} are weak simulations.
Weak bisimilarity is the largest weak bisimulation.
- \approx has a useful coinduction principle
- $\emptyset \approx \otimes \approx \odot$ and $\text{ex1} \approx \text{ex2} \approx \text{ex3}$

Asynchronous Dataflow Operators

Auxiliary definitions

Auxiliary definitions

Buffer functions

BHD $p \text{ buf} = \text{hd} (\text{buf } p)$

BTL $p \text{ buf} = \text{buf}(p := \text{tl} (\text{buf } p))$

BENQ $p \times \text{buf} = \text{buf}(p := \text{buf } p @ [x])$

$\text{buf}_1 \gg \text{buf}_2 = (\lambda p. \text{buf}_2 p @ \text{buf}_1 p)$

Auxiliary definitions

Buffer functions

BHD $p \text{ buf} = \text{hd } (\text{buf } p)$

BTL $p \text{ buf} = \text{buf}(p := \text{tl } (\text{buf } p))$

BENQ $p \times \text{buf} = \text{buf}(p := \text{buf } p @ [x])$

$\text{buf}_1 \gg \text{buf}_2 = (\lambda p. \text{buf}_2 p @ \text{buf}_1 p)$

choices function

$\text{choices } \otimes = \{\}$

Silent $op' \in_c \text{choices } op \longleftrightarrow \text{step Tau } op \text{ } op'$

$\text{choices } (\text{Choice } ops) = \bigcup_c (\text{choices } `_c \text{ } ops)$

Auxiliary definitions

Buffer functions

BHD $p \text{ buf} = \text{hd } (\text{buf } p)$

BTL $p \text{ buf} = \text{buf}(p := \text{tl } (\text{buf } p))$

BENQ $p \times \text{buf} = \text{buf}(p := \text{buf } p @ [x])$

$\text{buf}_1 \gg \text{buf}_2 = (\lambda p. \text{buf}_2 p @ \text{buf}_1 p)$

choices function

$\text{choices } \otimes = \{\}$

Silent $op' \in_c \text{choices } op \longleftrightarrow \text{step Tau } op \text{ } op'$

$\text{choices } (\text{Choice } ops) = \bigcup_c (\text{choices } `_c \text{ } ops)$

Mapping ports functions

$\text{map_op} :: ('i_1 \Rightarrow 'i_2) \Rightarrow ('o_1 \Rightarrow 'o_2) \Rightarrow ('i_1, 'o_1, 'd) \text{ op} \Rightarrow ('i_2, 'o_2, 'd) \text{ op}$

$\text{projl} :: ('a + 'b) \Rightarrow 'a$

$\text{projr} :: ('a + 'b) \Rightarrow 'b$

$\curvearrowleft :: ('a + 'b) + 'c \Rightarrow 'a + 'b + 'c$

$\curvearrowright :: 'a + 'b + 'c \Rightarrow ('a + 'b) + 'c$

Equation of the identity operator

$$\text{id_op } buf = \text{Choice}$$
$$\cup_c$$

Equation of the identity operator

$\text{id_op } buf = \text{Choice}$

$((((\lambda p. \text{Read } p (\lambda x. \text{id_op } (\text{BENQ } p \times buf)))) \backslash_c \mathfrak{U}_c)$
 \cup_c

- \mathfrak{U}_c is the set of usable ports provide by its type

Equation of the identity operator

$\text{id_op } buf = \text{Choice}$

$((((\lambda p. \text{Read } p (\lambda x. \text{id_op} (\text{BENQ } p \times buf))) \backslash_c \mathfrak{U}_c))$

\cup_c

$((\lambda p. \text{Write} (\text{id_op} (\text{BTL } p \ buf)) \ p (\text{BHD } p \ buf)) \backslash_c \{p \in_c \mathfrak{U}_c \mid buf \neq []\}))$

- \mathfrak{U}_c is the set of usable ports provide by its type

Equation of the identity operator

$\text{id_op } buf = \text{Choice}$

$(((\lambda p. \text{Read } p (\lambda x. \text{id_op} (\text{BENQ } p x buf))) \ `_c \ \mathfrak{U}_c)$

\cup_c

$((\lambda p. \text{Write} (\text{id_op} (\text{BTL } p buf)) p (\text{BHD } p buf)) \ `_c \ \{p \in_c \mathfrak{U}_c \mid buf \neq []\})$

- \mathfrak{U}_c is the set of usable ports provided by its type
- Stream delay!

Equation of the identity operator

$\text{id_op } buf = \text{Choice}$

$((\lambda p. \text{Read } p (\lambda x. \text{id_op} (\text{BENQ } p \times buf))) \ `_c \ \mathfrak{U}_c)$

\cup_c

$((\lambda p. \text{Write} (\text{id_op} (\text{BTL } p \ buf)) \ p (\text{BHD } p \ buf)) \ `_c \ \{p \in_c \mathfrak{U}_c \mid buf \ p \neq []\})$

- \mathfrak{U}_c is the set of usable ports provide by its type
- Stream delay!

Identity operator with an empty buffer

$\mathcal{I} = \text{id_op} (\lambda_. \ [])$

Composition: Preliminaries

Composition operator type

```
comp_op :: ('o1 ⇒ 'i2 option) ⇒ ('i2 ⇒ 'd list) ⇒ ('i1, 'o1, 'd) op ⇒ ('i2, 'o2, 'd) op ⇒  
      ('i1 + 'i2, 'o1 + 'o2, 'd) op
```

Composition: Preliminaries

Composition operator type

```
comp_op :: ('o1 ⇒ 'i2 option) ⇒ ('i2 ⇒ 'd list) ⇒ ('i1, 'o1, 'd) op ⇒ ('i2, 'o2, 'd) op ⇒ ('i1 + 'i2, 'o1 + 'o2, 'd) op
```

Sequential and parallel composition

$$op_1 \parallel op_2 = \text{comp_op } (\lambda_. \text{None}) (\lambda_. []) op_1 op_2$$
$$op_1 \bullet op_2 = \text{map_op projl projr } (\text{comp_op Some } (\lambda_. [])) op_1 op_2$$

Composition: equation

Composition operator

`comp_op wire buf op1 op2 =`

Composition: equation

Composition operator

`comp_op wire buf op1 op2 = Choice`

\cup_c

Composition: equation

Composition operator

`comp_op wire buf op1 op2 = Choice`

$\backslash_c \text{ choices } op_1) \cup_c$

Composition: equation

Composition operator

$\text{comp_op } \text{wire } \text{buf } op_1 \ op_2 = \text{Choice } (((\lambda \text{op}. \ \underline{\text{case}} \ \underline{\text{op}} \ \underline{\text{of}}}$
 $\text{Read } p \ f \Rightarrow \text{Read} (\text{Inl } p) (\lambda x. \ \text{comp_op } \text{wire } \text{buf } (f \ x) \ op_2)$

$\backslash_c \ \text{choices } op_1) \ \cup_c$

Composition: equation

Composition operator

$\text{comp_op } \text{wire } \text{buf } op_1 \ op_2 = \text{Choice } (((\lambda op. \underline{\text{case}} \ op \ \underline{\text{of}}}$

$\text{Read } p \ f \Rightarrow \text{Read} (\text{Inl } p) (\lambda x. \text{comp_op } \text{wire } \text{buf } (f \ x) \ op_2)$

| Write $op \ p \ x \Rightarrow (\underline{\text{case}} \ \text{wire } p \ \underline{\text{of}}$

None $\Rightarrow \text{Write} (\text{comp_op } \text{wire } \text{buf } op \ op_2) (\text{Inl } p) \ x$

| Some $q \Rightarrow \text{Silent} (\text{comp_op } \text{wire} (\text{BENQ } q \ x \ \text{buf}) \ op \ op_2))$

$\backslash_c \ \text{choices } op_1) \ \cup_c$

Composition: equation

Composition operator

$\text{comp_op } \text{wire } \text{buf } op_1 \ op_2 = \text{Choice } (((\lambda op. \underline{\text{case}} \ op \ \underline{\text{of}} \ }$

Read $p \ f \Rightarrow \text{Read} (\text{Inl } p) (\lambda x. \text{comp_op } \text{wire } \text{buf } (f \ x) \ op_2)$

| Write $op \ p \ x \Rightarrow (\underline{\text{case}} \ \text{wire } p \ \underline{\text{of}}$

None $\Rightarrow \text{Write} (\text{comp_op } \text{wire } \text{buf } op \ op_2) (\text{Inl } p) \ x$

| Some $q \Rightarrow \text{Silent} (\text{comp_op } \text{wire} (\text{BENQ } q \ x \ \text{buf}) \ op \ op_2))$

| Silent $op \Rightarrow \text{Silent} (\text{comp_op } \text{wire } \text{buf } op \ op_2)) \ \cup_c \ \text{choices } op_1 \ \cup_c$

Composition: equation

Composition operator

$\text{comp_op } \text{wire } \text{buf } op_1 \ op_2 = \text{Choice } (((\lambda op. \underline{\text{case}} \ op \ \underline{\text{of}} \ }$
 $\text{Read } p \ f \Rightarrow \text{Read} (\text{Inl } p) (\lambda x. \text{comp_op } \text{wire } \text{buf } (f \ x) \ op_2)$
| Write $op \ p \ x \Rightarrow (\underline{\text{case}} \ \text{wire } p \ \underline{\text{of}}$
 None $\Rightarrow \text{Write} (\text{comp_op } \text{wire } \text{buf } op \ op_2) (\text{Inl } p) \ x$
 Some $q \Rightarrow \text{Silent} (\text{comp_op } \text{wire } (\text{BENQ } q \ x \ \text{buf}) \ op \ op_2))$
| Silent $op \Rightarrow \text{Silent} (\text{comp_op } \text{wire } \text{buf } op \ op_2)) \ \cup_c \ \text{choices } op_1 \ \cup_c$

(choices $op_2)))$

Composition: equation

Composition operator

$\text{comp_op } \text{wire } \text{buf } op_1 \ op_2 = \text{Choice } (((\lambda op. \underline{\text{case}} \ op \ \underline{\text{of}} \ }$

$\text{Read } p \ f \Rightarrow \text{Read} (\text{Inl } p) (\lambda x. \text{comp_op } \text{wire } \text{buf } (f \ x) \ op_2)$

| Write $op \ p \ x \Rightarrow (\underline{\text{case}} \ \text{wire } p \ \underline{\text{of}}$

None \Rightarrow Write ($\text{comp_op } \text{wire } \text{buf } op \ op_2$) ($\text{Inl } p$) x

| Some $q \Rightarrow \text{Silent} (\text{comp_op } \text{wire} (\text{BENQ } q \ x \ \text{buf}) \ op \ op_2))$

| Silent $op \Rightarrow \text{Silent} (\text{comp_op } \text{wire } \text{buf } op \ op_2)) \ `_c \ \text{choices } op_1 \cup_c$

$`_c \ \text{sound_reads } \text{wire } \text{buf } (\text{choices } op_2)))$

Composition: equation

Composition operator

$\text{comp_op } \text{wire } \text{buf } op_1 \ op_2 = \text{Choice } (((\lambda op. \underline{\text{case}} \ op \ \underline{\text{of}}}$

$\text{Read } p \ f \Rightarrow \text{Read} (\text{Inl } p) (\lambda x. \text{comp_op } \text{wire } \text{buf } (f \ x) \ op_2)$

| Write $op \ p \ x \Rightarrow (\underline{\text{case}} \ \text{wire } p \ \underline{\text{of}}$

None $\Rightarrow \text{Write} (\text{comp_op } \text{wire } \text{buf } op \ op_2) (\text{Inl } p) \ x$

| Some $q \Rightarrow \text{Silent} (\text{comp_op } \text{wire} (\text{BENQ } q \ x \ \text{buf}) \ op \ op_2))$

| Silent $op \Rightarrow \text{Silent} (\text{comp_op } \text{wire } \text{buf } op \ op_2) \ \backslash_c \ \text{choices } op_1 \cup_c$

(($\lambda op. \underline{\text{case}} \ op \ \underline{\text{of}}$

| Write $op \ p \ x \Rightarrow \text{Write} (\text{comp_op } \text{wire } \text{buf } op_1 \ op) (\text{Inr } p) \ x$

$\backslash_c \ \text{sound_reads } \text{wire } \text{buf } (\text{choices } op_2)))$

Composition: equation

Composition operator

$\text{comp_op } \text{wire } \text{buf } \text{op}_1 \text{ op}_2 = \text{Choice } (((\lambda \text{op}. \underline{\text{case}} \text{ op } \underline{\text{of}}}$

$\text{Read } p \ f \Rightarrow \text{Read} (\text{Inl } p) (\lambda x. \text{comp_op } \text{wire } \text{buf } (f \ x) \ \text{op}_2)$

| Write $\text{op } p \ x \Rightarrow (\underline{\text{case}} \text{ wire } p \ \underline{\text{of}}$

None $\Rightarrow \text{Write} (\text{comp_op } \text{wire } \text{buf } \text{op } \text{op}_2) (\text{Inl } p) \ x$

| Some $q \Rightarrow \text{Silent} (\text{comp_op } \text{wire} (\text{BENQ } q \ x \ \text{buf}) \ \text{op } \text{op}_2))$

| Silent $\text{op} \Rightarrow \text{Silent} (\text{comp_op } \text{wire } \text{buf } \text{op } \text{op}_2) \ \backslash_c \ \text{choices } \text{op}_1 \cup_c$

(($\lambda \text{op}. \underline{\text{case}} \text{ op } \underline{\text{of}}$

Read $p \ f \Rightarrow \underline{\text{if }} p \in \text{ran } \text{wire}$

then $\text{Silent} (\text{comp_op } \text{wire} (\text{BTL } p \ \text{buf}) \ \text{op}_1 (f (\text{BHD } p \ \text{buf})))$

else $\text{Read} (\text{Inr } p) (\lambda x. \text{comp_op } \text{wire } \text{buf } \text{op}_1 (f \ x))$

| Write $\text{op } p \ x \Rightarrow \text{Write} (\text{comp_op } \text{wire } \text{buf } \text{op}_1 \text{ op}) (\text{Inr } p) \ x$

$\backslash_c \ \text{sound_reads } \text{wire } \text{buf } (\text{choices } \text{op}_2))$

Composition: equation

Composition operator

$\text{comp_op wire buf } op_1 \ op_2 = \text{Choice (((}\lambda op. \underline{\text{case}} \ op \ \underline{\text{of}}$

$\text{Read } p \ f \Rightarrow \text{Read} (\text{Inl } p) (\lambda x. \text{comp_op wire buf } (f \ x) \ op_2)$

| Write $op \ p \ x \Rightarrow (\underline{\text{case}} \ \text{wire } p \ \underline{\text{of}}$

None $\Rightarrow \text{Write} (\text{comp_op wire buf } op \ op_2) (\text{Inl } p) \ x$

| Some $q \Rightarrow \text{Silent} (\text{comp_op wire } (\text{BENQ } q \ x \ \text{buf}) \ op \ op_2))$

| Silent $op \Rightarrow \text{Silent} (\text{comp_op wire buf } op \ op_2) \ \backslash_c \ \text{choices } op_1 \cup_c$

(($\lambda op. \underline{\text{case}} \ op \ \underline{\text{of}}$

Read $p \ f \Rightarrow \underline{\text{if }} p \in \text{ran wire}$

then $\text{Silent} (\text{comp_op wire } (\text{BTL } p \ \text{buf}) \ op_1 (f (\text{BHD } p \ \text{buf})))$

else $\text{Read} (\text{Inr } p) (\lambda x. \text{comp_op wire buf } op_1 (f \ x))$

| Write $op \ p \ x \Rightarrow \text{Write} (\text{comp_op wire buf } op_1 \ op) (\text{Inr } p) \ x$

| Silent $op \Rightarrow \text{Silent} (\text{comp_op wire buf } op_1 \ op) \ \backslash_c \ \text{sound_reads wire buf } (\text{choices } op_2))$

Asynchronous Dataflow Operators

- foo

Asynchronous Dataflow Properties

Basic network algebra properties

$$B1: op_1 \parallel (op_2 \parallel op_3) \approx \text{map_op} \curvearrowright \curvearrowright (op_1 \parallel op_2) \parallel op_3$$

$$B2_1: op \parallel (\mathcal{I} :: (0, 0, 'd) op) \approx \text{map_op} \text{Inl} \text{Inl} op$$

$$B2_2: (\mathcal{I} :: (0, 0, 'd) op) \parallel op \approx \text{map_op} \text{Inr} \text{Inr} op$$

$$B3: (op_1 \bullet op_2) \bullet op_3 \approx op_1 \bullet (op_2 \bullet op_3)$$

$$B4_1: op \text{Inl} \bullet \mathcal{I} \approx op \text{Inl}$$

$$B4_2: \mathcal{I} \bullet \text{Inp} op \approx \text{Inp} op$$

$$B5: (op_1 \parallel op_2) \bullet (op_3 \parallel op_4) \approx (op_1 \bullet op_3) \parallel (op_2 \bullet op_4)$$

$$B6: \mathcal{I} \parallel \mathcal{I} \approx \mathcal{I}$$

$$B7: \mathcal{X} \bullet \mathcal{X} \approx \mathcal{I}$$

$$B8: (\mathcal{X} :: ('i + 0, 0 + 'i, 'd) op) \approx \text{map_op} \text{id} (\text{case_sum} \text{Inr} \text{Inl}) \mathcal{I}$$

$$B9: \mathcal{X} \approx \text{map_op} \curvearrowright \curvearrowright (\mathcal{X} \parallel \mathcal{I}) \bullet \text{map_op} \text{id} \curvearrowright (\mathcal{I} \parallel \mathcal{X})$$

$$B10: (\text{Inp} op_1 \parallel \text{Inp} op_2) \bullet \mathcal{X} \approx \mathcal{X} \bullet (op_2 \text{Inl} \parallel op_1 \text{Inl})$$

$$F1: \mathcal{I} \uparrow \approx (\mathcal{I} :: (0, 0, 'd) op)$$

$$F2: \mathcal{X} \uparrow \approx \mathcal{I}$$

$$R1: \text{Inr} \text{- inputs } op_1 \cap \text{defaults} = \{\} \Rightarrow \text{Inr} \text{- outputs } op_1 \cap \text{defaults} = \{\} \Rightarrow \\ op_2 \bullet (op_1 \uparrow) \approx ((op_2 \parallel \mathcal{I}) \bullet op_1) \uparrow$$

$$R2: \text{Inr} \text{- inputs } op_1 \cap \text{defaults} = \{\} \Rightarrow \text{Inr} \text{- outputs } op_1 \cap \text{defaults} = \{\} \Rightarrow \\ (op_1 \uparrow) \bullet op_2 \approx (op_1 \bullet (op_2 \parallel \mathcal{I})) \uparrow$$

$$R3: \text{Inr} \text{- inputs } op_2 \cap \text{defaults} = \{\} \Rightarrow \text{Inr} \text{- outputs } op_2 \cap \text{defaults} = \{\} \Rightarrow \\ op_1 \parallel (op_2 \uparrow) \approx (\text{map_op} \curvearrowright \curvearrowright (op_1 \parallel op_2)) \uparrow$$

$$R4: \text{Inr} \text{- inputs } op_1 \cap \text{defaults} = \{\} \Rightarrow \text{Inr} \text{- outputs } op_1 \cap \text{defaults} = \{\} \Rightarrow \\ \text{inputs } op_2 \cap \text{defaults} = \{\} \Rightarrow \text{outputs } op_2 \cap \text{defaults} = \{\} \Rightarrow \\ (\text{Inp} op_1 \bullet (\mathcal{I} \parallel op_2)) \uparrow \approx ((\mathcal{I} \parallel op_2) \bullet \text{Inp} op_1) \uparrow$$

$$R5: \text{Inr} \text{- inputs } op = \{\} \Rightarrow \text{Inr} \text{- outputs } op = \{\} \Rightarrow \\ \text{map_op} \text{Inl} \text{Inl} ((op :: ('i + 0, 'o + 0, 'd) op) \uparrow) \approx op$$

$$R6: \text{Inr} \text{- inputs } op = \{\} \Rightarrow \text{Inr} \text{- outputs } op = \{\} \Rightarrow \\ \text{Inr} \text{- Inl} \text{- inputs } op = \{\} \Rightarrow \text{Inr} \text{- Inl} \text{- outputs } op = \{\} \Rightarrow \\ (op \uparrow) \approx (\text{map_op} \curvearrowright \curvearrowright op) \uparrow$$

Table 1 Basic network algebra properties

Basic network algebra properties

$$B1: op_1 \parallel (op_2 \parallel op_3) \approx \text{map_op} \curvearrowright \curvearrowright (op_1 \parallel op_2) \parallel op_3$$

$$B2_1: op \parallel (\mathcal{I} :: (0, 0, 'd) op) \approx \text{map_op} \text{Inl Inl } op$$

$$B2_2: (\mathcal{I} :: (0, 0, 'd) op) \parallel op \approx \text{map_op} \text{Inr Inr } op$$

$$B3: (op_1 \bullet op_2) \bullet op_3 \approx op_1 \bullet (op_2 \bullet op_3)$$

$$B4_1: op \text{Inl} \bullet \mathcal{I} \approx op \text{Inl}$$

$$B4_2: \mathcal{I} \bullet \text{Inp} \approx \text{Inp}$$

$$B5: (op_1 \parallel op_2) \bullet (op_3 \parallel op_4) \approx (op_1 \bullet op_3) \parallel (op_2 \bullet op_4)$$

$$B6: \mathcal{I} \parallel \mathcal{I} \approx \mathcal{I}$$

$$B7: \mathcal{X} \bullet \mathcal{X} \approx \mathcal{I}$$

$$B8: (\mathcal{X} :: ('i + 0, 0 + 'i, 'd) op) \approx \text{map_op id} (\text{case_sum Inr Inl}) \mathcal{I}$$

$$B9: \mathcal{X} \approx \text{map_op} \curvearrowright \curvearrowright (\mathcal{X} \parallel \mathcal{I}) \bullet \text{map_op id} \curvearrowright (\mathcal{I} \parallel \mathcal{X})$$

$$B10: (\text{Inp} \parallel \text{Inp}) \bullet \mathcal{X} \approx \mathcal{X} \bullet (op_2 \text{Inl} \parallel op_1 \text{Inl})$$

$$F1: \mathcal{I} \uparrow \approx (\mathcal{I} :: (0, 0, 'd) op)$$

$$F2: \mathcal{X} \uparrow \approx \mathcal{I}$$

Table 1 Basic network algebra properties

$$R1: \text{Inr} \setminus \text{inputs } op_1 \cap \text{defaults} = \{\} \Rightarrow \text{Inr} \setminus \text{outputs } op_1 \cap \text{defaults} = \{\} \Rightarrow \\ op_2 \bullet (op_1 \uparrow) \approx ((op_2 \parallel \mathcal{I}) \bullet op_1) \uparrow$$

$$R2: \text{Inr} \setminus \text{inputs } op_1 \cap \text{defaults} = \{\} \Rightarrow \text{Inr} \setminus \text{outputs } op_1 \cap \text{defaults} = \{\} \Rightarrow \\ (op_1 \uparrow) \bullet op_2 \approx (op_1 \bullet (op_2 \parallel \mathcal{I})) \uparrow$$

$$R3: \text{Inr} \setminus \text{inputs } op_2 \cap \text{defaults} = \{\} \Rightarrow \text{Inr} \setminus \text{outputs } op_2 \cap \text{defaults} = \{\} \Rightarrow \\ op_1 \parallel (op_2 \uparrow) \approx (\text{map_op} \curvearrowright \curvearrowright (op_1 \parallel op_2)) \uparrow$$

$$R4: \text{Inr} \setminus \text{inputs } op_1 \cap \text{defaults} = \{\} \Rightarrow \text{Inr} \setminus \text{outputs } op_1 \cap \text{defaults} = \{\} \Rightarrow \\ \text{inputs } op_2 \cap \text{defaults} = \{\} \Rightarrow \text{outputs } op_2 \cap \text{defaults} = \{\} \Rightarrow \\ (\text{Inp}_1 \bullet (\mathcal{I} \parallel op_2)) \uparrow \approx ((\mathcal{I} \parallel op_2) \bullet \text{Inp}_1) \uparrow$$

$$R5: \text{Inr} \setminus \text{inputs } op = \{\} \Rightarrow \text{Inr} \setminus \text{outputs } op = \{\} \Rightarrow \\ \text{map_op Inl Inl} ((op :: ('i + 0, 'o + 0, 'd) op) \uparrow) \approx op$$

$$R6: \text{Inr} \setminus \text{inputs } op = \{\} \Rightarrow \text{Inr} \setminus \text{outputs } op = \{\} \Rightarrow \\ \text{Inr} \setminus \text{Inl} \setminus \text{inputs } op = \{\} \Rightarrow \text{Inr} \setminus \text{Inl} \setminus \text{outputs } op = \{\} \Rightarrow \\ (op \uparrow) \approx (\text{map_op} \curvearrowright \curvearrowright op) \uparrow$$

Basic network algebra properties

$$B1: op_1 \parallel (op_2 \parallel op_3) \approx \text{map_op} \curvearrowright \curvearrowright (op_1 \parallel op_2) \parallel op_3$$

$$B2_1: op \parallel (\mathcal{I} :: (0, 0, 'd) op) \approx \text{map_op} \text{Inl} \text{Inl} op$$

$$B2_2: (\mathcal{I} :: (0, 0, 'd) op) \parallel op \approx \text{map_op} \text{Inr} \text{Inr} op$$

$$B3: (op_1 \bullet op_2) \bullet op_3 \approx op_1 \bullet (op_2 \bullet op_3)$$

$$B4_1: op \sqsubseteq \bullet \mathcal{I} \approx op \sqsubseteq$$

$$B4_2: \mathcal{I} \bullet \sqsupseteq op \approx \sqsupseteq op$$

$$B5: (op_1 \parallel op_2) \bullet (op_3 \parallel op_4) \approx (op_1 \bullet op_3) \parallel (op_2 \bullet op_4)$$

$$B6: \mathcal{I} \parallel \mathcal{I} \approx \mathcal{I}$$

$$B7: \mathcal{X} \bullet \mathcal{X} \approx \mathcal{I}$$

$$B8: (\mathcal{X} :: ('i + 0, 0 + 'i, 'd) op) \approx \text{map_op} \text{id} (\text{case_sum} \text{Inr} \text{Inl}) \mathcal{I}$$

$$B9: \mathcal{X} \approx \text{map_op} \curvearrowright \curvearrowright (\mathcal{X} \parallel \mathcal{I}) \bullet \text{map_op} \text{id} \curvearrowright (\mathcal{I} \parallel \mathcal{X})$$

$$B10: (\sqsupseteq op_1 \parallel \sqsupseteq op_2) \bullet \mathcal{X} \approx \mathcal{X} \bullet (op_2 \sqsubseteq \parallel op_1 \sqsubseteq)$$

$$F1: \mathcal{I} \uparrow \approx (\mathcal{I} :: (0, 0, 'd) op)$$

$$F2: \mathcal{X} \uparrow \approx \mathcal{I}$$

Table 1 Basic network algebra properties

$$R1: \text{Inr} \dashv \text{inputs } op_1 \cap \text{defaults} = \{\} \Rightarrow \text{Inr} \dashv \text{outputs } op_1 \cap \text{defaults} = \{\} \Rightarrow \\ op_2 \bullet (op_1 \uparrow) \approx ((op_2 \parallel \mathcal{I}) \bullet op_1) \uparrow$$

$$R2: \text{Inr} \dashv \text{inputs } op_1 \cap \text{defaults} = \{\} \Rightarrow \text{Inr} \dashv \text{outputs } op_1 \cap \text{defaults} = \{\} \Rightarrow \\ (op_1 \uparrow) \bullet op_2 \approx (op_1 \bullet (op_2 \parallel \mathcal{I})) \uparrow$$

$$R3: \text{Inr} \dashv \text{inputs } op_2 \cap \text{defaults} = \{\} \Rightarrow \text{Inr} \dashv \text{outputs } op_2 \cap \text{defaults} = \{\} \Rightarrow \\ op_1 \parallel (op_2 \uparrow) \approx (\text{map_op} \curvearrowright \curvearrowright (op_1 \parallel op_2)) \uparrow$$

$$R4: \text{Inr} \dashv \text{inputs } op_1 \cap \text{defaults} = \{\} \Rightarrow \text{Inr} \dashv \text{outputs } op_1 \cap \text{defaults} = \{\} \Rightarrow \\ \text{inputs } op_2 \cap \text{defaults} = \{\} \Rightarrow \text{outputs } op_2 \cap \text{defaults} = \{\} \Rightarrow \\ (\sqsupseteq op_1 \bullet (\mathcal{I} \parallel op_2)) \uparrow \approx ((\mathcal{I} \parallel op_2) \bullet op_1 \sqsubseteq) \uparrow$$

$$R5: \text{Inr} \dashv \text{inputs } op = \{\} \Rightarrow \text{Inr} \dashv \text{outputs } op = \{\} \Rightarrow \\ \text{map_op} \text{Inl} \text{Inl} ((op :: ('i + 0, 'o + 0, 'd) op) \uparrow) \approx op$$

$$R6: \text{Inr} \dashv \text{inputs } op = \{\} \Rightarrow \text{Inr} \dashv \text{outputs } op = \{\} \Rightarrow \\ \text{Inr} \dashv \text{Inl} \dashv \text{inputs } op = \{\} \Rightarrow \text{Inr} \dashv \text{Inl} \dashv \text{outputs } op = \{\} \Rightarrow \\ (op \uparrow) \approx (\text{map_op} \curvearrowright \curvearrowright op) \uparrow$$

Basic network algebra properties

$$B1: op_1 \parallel (op_2 \parallel op_3) \approx \text{map_op} \curvearrowright \curvearrowright (op_1 \parallel op_2) \parallel op_3$$

$$B2_1: op \parallel (\mathcal{I} :: (0, 0, 'd) op) \approx \text{map_op} \text{Inl} \text{Inl} op$$

$$B2_2: (\mathcal{I} :: (0, 0, 'd) op) \parallel op \approx \text{map_op} \text{Inr} \text{Inr} op$$

$$B3: (op_1 \bullet op_2) \bullet op_3 \approx op_1 \bullet (op_2 \bullet op_3)$$

$$B4_1: op \sqsubseteq \bullet \mathcal{I} \approx op \sqsubseteq$$

$$B4_2: \mathcal{I} \bullet \sqsupseteq op \approx \sqsupseteq op$$

$$B5: (op_1 \parallel op_2) \bullet (op_3 \parallel op_4) \approx (op_1 \bullet op_3) \parallel (op_2 \bullet op_4)$$

$$B6: \mathcal{I} \parallel \mathcal{I} \approx \mathcal{I}$$

$$B7: \mathcal{X} \bullet \mathcal{X} \approx \mathcal{I}$$

$$B8: (\mathcal{X} :: ('i + 0, 0 + 'i, 'd) op) \approx \text{map_op} \text{id} (\text{case_sum} \text{Inr} \text{Inl}) \mathcal{I}$$

$$B9: \mathcal{X} \approx \text{map_op} \curvearrowright \curvearrowright (\mathcal{X} \parallel \mathcal{I}) \bullet \text{map_op} \text{id} \curvearrowright (\mathcal{I} \parallel \mathcal{X})$$

$$B10: (\sqsupseteq op_1 \parallel \sqsupseteq op_2) \bullet \mathcal{X} \approx \mathcal{X} \bullet (op_2 \sqsubseteq \parallel op_1 \sqsubseteq)$$

$$F1: \mathcal{I} \uparrow \approx (\mathcal{I} :: (0, 0, 'd) op)$$

$$F2: \mathcal{X} \uparrow \approx \mathcal{I}$$

Table 1 Basic network algebra properties

$$R1: \text{Inr} \dashv \text{inputs } op_1 \cap \text{defaults} = \{\} \Rightarrow \text{Inr} \dashv \text{outputs } op_1 \cap \text{defaults} = \{\} \Rightarrow \\ op_2 \bullet (op_1 \uparrow) \approx ((op_2 \parallel \mathcal{I}) \bullet op_1) \uparrow$$

$$R2: \text{Inr} \dashv \text{inputs } op_1 \cap \text{defaults} = \{\} \Rightarrow \text{Inr} \dashv \text{outputs } op_1 \cap \text{defaults} = \{\} \Rightarrow \\ (op_1 \uparrow) \bullet op_2 \approx (op_1 \bullet (op_2 \parallel \mathcal{I})) \uparrow$$

$$R3: \text{Inr} \dashv \text{inputs } op_2 \cap \text{defaults} = \{\} \Rightarrow \text{Inr} \dashv \text{outputs } op_2 \cap \text{defaults} = \{\} \Rightarrow \\ op_1 \parallel (op_2 \uparrow) \approx (\text{map_op} \curvearrowright \curvearrowright (op_1 \parallel op_2)) \uparrow$$

$$R4: \text{Inr} \dashv \text{inputs } op_1 \cap \text{defaults} = \{\} \Rightarrow \text{Inr} \dashv \text{outputs } op_1 \cap \text{defaults} = \{\} \Rightarrow \\ \text{inputs } op_2 \cap \text{defaults} = \{\} \Rightarrow \text{outputs } op_2 \cap \text{defaults} = \{\} \Rightarrow \\ (\sqsupseteq op_1 \bullet (\mathcal{I} \parallel op_2)) \uparrow \approx ((\mathcal{I} \parallel op_2) \bullet op_1 \sqsubseteq) \uparrow$$

$$R5: \text{Inr} \dashv \text{inputs } op = \{\} \Rightarrow \text{Inr} \dashv \text{outputs } op = \{\} \Rightarrow \\ \text{map_op} \text{Inl} \text{Inl} ((op :: ('i + 0, 'o + 0, 'd) op) \uparrow) \approx op$$

$$R6: \text{Inr} \dashv \text{inputs } op = \{\} \Rightarrow \text{Inr} \dashv \text{outputs } op = \{\} \Rightarrow \\ \text{Inr} \dashv \text{Inl} \dashv \text{inputs } op = \{\} \Rightarrow \text{Inr} \dashv \text{Inl} \dashv \text{outputs } op = \{\} \Rightarrow \\ (op \uparrow) \approx (\text{map_op} \curvearrowright \curvearrowright op) \uparrow$$

Properties of equality test, merge, copy, split, source and sink operators

$$B1: op_1 \parallel (op_2 \parallel op_3) \approx \text{map_op} \curvearrowright \curvearrowright (op_1 \parallel op_2) \parallel op_3$$

$$B2_1: op \parallel (\mathcal{I} :: (0, 0, 'd) op) \approx \text{map_op} \text{Inl} \text{Inl} op$$

$$B2_2: (\mathcal{I} :: (0, 0, 'd) op) \parallel op \approx \text{map_op} \text{Inr} \text{Inr} op$$

$$B3: (op_1 \bullet op_2) \bullet op_3 \approx op_1 \bullet (op_2 \bullet op_3)$$

$$B4_1: op \sqsubseteq \bullet \mathcal{I} \approx op \sqsubseteq$$

$$B4_2: \mathcal{I} \bullet \sqcup op \approx \sqcup op$$

$$B5: (op_1 \parallel op_2) \bullet (op_3 \parallel op_4) \approx (op_1 \bullet op_3) \parallel (op_2 \bullet op_4)$$

$$B6: \mathcal{I} \parallel \mathcal{I} \approx \mathcal{I}$$

$$B7: \mathcal{X} \bullet \mathcal{X} \approx \mathcal{I}$$

$$B8: (\mathcal{X} :: ('i + 0, 0 + 'i, 'd) op) \approx \text{map_op} \text{id} (\text{case_sum} \text{Inr} \text{Inl}) \mathcal{I}$$

$$B9: \mathcal{X} \approx \text{map_op} \curvearrowright \curvearrowright (\mathcal{X} \parallel \mathcal{I}) \bullet \text{map_op} \text{id} \curvearrowright (\mathcal{I} \parallel \mathcal{X})$$

$$B10: (\sqcup op_1 \parallel \sqcup op_2) \bullet \mathcal{X} \approx \mathcal{X} \bullet (op_2 \sqsubseteq \parallel op_1 \sqsubseteq)$$

$$F1: \mathcal{I} \uparrow \approx (\mathcal{I} :: (0, 0, 'd) op)$$

$$F2: \mathcal{X} \uparrow \approx \mathcal{I}$$

Table 1 Basic network algebra properties

$$R1: \text{Inr} \dashv \text{inputs } op_1 \cap \text{defaults} = \{\} \Rightarrow \text{Inr} \dashv \text{outputs } op_1 \cap \text{defaults} = \{\} \Rightarrow \\ op_2 \bullet (op_1 \uparrow) \approx ((op_2 \parallel \mathcal{I}) \bullet op_1) \uparrow$$

$$R2: \text{Inr} \dashv \text{inputs } op_1 \cap \text{defaults} = \{\} \Rightarrow \text{Inr} \dashv \text{outputs } op_1 \cap \text{defaults} = \{\} \Rightarrow \\ (op_1 \uparrow) \bullet op_2 \approx (op_1 \bullet (op_2 \parallel \mathcal{I})) \uparrow$$

$$R3: \text{Inr} \dashv \text{inputs } op_2 \cap \text{defaults} = \{\} \Rightarrow \text{Inr} \dashv \text{outputs } op_2 \cap \text{defaults} = \{\} \Rightarrow \\ op_1 \parallel (op_2 \uparrow) \approx (\text{map_op} \curvearrowright \curvearrowright (op_1 \parallel op_2)) \uparrow$$

$$R4: \text{Inr} \dashv \text{inputs } op_1 \cap \text{defaults} = \{\} \Rightarrow \text{Inr} \dashv \text{outputs } op_1 \cap \text{defaults} = \{\} \Rightarrow \\ \text{inputs } op_2 \cap \text{defaults} = \{\} \Rightarrow \text{outputs } op_2 \cap \text{defaults} = \{\} \Rightarrow \\ (\sqcup op_1 \bullet (\mathcal{I} \parallel op_2)) \uparrow \approx ((\mathcal{I} \parallel op_2) \bullet op_1 \sqsubseteq) \uparrow$$

$$R5: \text{Inr} \dashv \text{inputs } op = \{\} \Rightarrow \text{Inr} \dashv \text{outputs } op = \{\} \Rightarrow \\ \text{map_op} \text{Inl} \text{Inl} ((op :: ('i + 0, 'o + 0, 'd) op) \uparrow) \approx op$$

$$R6: \text{Inr} \dashv \text{inputs } op = \{\} \Rightarrow \text{Inr} \dashv \text{outputs } op = \{\} \Rightarrow \\ \text{Inr} \dashv \text{Inl} \dashv \text{inputs } op = \{\} \Rightarrow \text{Inr} \dashv \text{Inl} \dashv \text{outputs } op = \{\} \Rightarrow \\ (op \uparrow) \approx (\text{map_op} \curvearrowright \curvearrowright op) \uparrow$$

$$B1: op_1 \parallel (op_2 \parallel op_3) \approx \text{map_op} \curvearrowright \curvearrowright (op_1 \parallel op_2) \parallel op_3$$

$$B2_1: op \parallel (\mathcal{I} :: (0, 0, 'd) op) \approx \text{map_op} \text{Inl} \text{Inl} op$$

$$B2_2: (\mathcal{I} :: (0, 0, 'd) op) \parallel op \approx \text{map_op} \text{Inr} \text{Inr} op$$

$$B3: (op_1 \bullet op_2) \bullet op_3 \approx op_1 \bullet (op_2 \bullet op_3)$$

$$B4_1: op \sqsubseteq \bullet \mathcal{I} \approx op \sqsubseteq$$

$$B4_2: \mathcal{I} \bullet \sqcup op \approx \sqcup op$$

$$B5: (op_1 \parallel op_2) \bullet (op_3 \parallel op_4) \approx (op_1 \bullet op_3) \parallel (op_2 \bullet op_4)$$

$$B6: \mathcal{I} \parallel \mathcal{I} \approx \mathcal{I}$$

$$B7: \mathcal{X} \bullet \mathcal{X} \approx \mathcal{I}$$

$$B8: (\mathcal{X} :: ('i + 0, 0 + 'i, 'd) op) \approx \text{map_op} \text{id} (\text{case_sum} \text{Inr} \text{Inl}) \mathcal{I}$$

$$B9: \mathcal{X} \approx \text{map_op} \curvearrowright \curvearrowright (\mathcal{X} \parallel \mathcal{I}) \bullet \text{map_op} \text{id} \curvearrowright (\mathcal{I} \parallel \mathcal{X})$$

$$B10: (\sqcup op_1 \parallel \sqcup op_2) \bullet \mathcal{X} \approx \mathcal{X} \bullet (op_2 \sqsubseteq \parallel op_1 \sqsubseteq)$$

$$F1: \mathcal{I} \uparrow \approx (\mathcal{I} :: (0, 0, 'd) op)$$

$$F2: \mathcal{X} \uparrow \approx \mathcal{I}$$

Table 1 Basic network algebra properties

$$R1: \text{Inr} \dashv \text{inputs } op_1 \cap \text{defaults} = \{\} \Rightarrow \text{Inr} \dashv \text{outputs } op_1 \cap \text{defaults} = \{\} \Rightarrow \\ op_2 \bullet (op_1 \uparrow) \approx ((op_2 \parallel \mathcal{I}) \bullet op_1) \uparrow$$

$$R2: \text{Inr} \dashv \text{inputs } op_1 \cap \text{defaults} = \{\} \Rightarrow \text{Inr} \dashv \text{outputs } op_1 \cap \text{defaults} = \{\} \Rightarrow \\ (op_1 \uparrow) \bullet op_2 \approx (op_1 \bullet (op_2 \parallel \mathcal{I})) \uparrow$$

$$R3: \text{Inr} \dashv \text{inputs } op_2 \cap \text{defaults} = \{\} \Rightarrow \text{Inr} \dashv \text{outputs } op_2 \cap \text{defaults} = \{\} \Rightarrow \\ op_1 \parallel (op_2 \uparrow) \approx (\text{map_op} \curvearrowright \curvearrowright (op_1 \parallel op_2)) \uparrow$$

$$R4: \text{Inr} \dashv \text{inputs } op_1 \cap \text{defaults} = \{\} \Rightarrow \text{Inr} \dashv \text{outputs } op_1 \cap \text{defaults} = \{\} \Rightarrow \\ \text{inputs } op_2 \cap \text{defaults} = \{\} \Rightarrow \text{outputs } op_2 \cap \text{defaults} = \{\} \Rightarrow \\ (\sqcup op_1 \bullet (\mathcal{I} \parallel op_2)) \uparrow \approx ((\mathcal{I} \parallel op_2) \bullet op_1 \sqsubseteq) \uparrow$$

$$R5: \text{Inr} \dashv \text{inputs } op = \{\} \Rightarrow \text{Inr} \dashv \text{outputs } op = \{\} \Rightarrow \\ \text{map_op} \text{Inl} \text{Inl} ((op :: ('i + 0, 'o + 0, 'd) op) \uparrow) \approx op$$

$$R6: \text{Inr} \dashv \text{inputs } op = \{\} \Rightarrow \text{Inr} \dashv \text{outputs } op = \{\} \Rightarrow \\ \text{Inr} \dashv \text{Inl} \dashv \text{inputs } op = \{\} \Rightarrow \text{Inr} \dashv \text{Inl} \dashv \text{outputs } op = \{\} \Rightarrow \\ (op \uparrow) \approx (\text{map_op} \curvearrowright \curvearrowright op) \uparrow$$

Well-Behaved Operators

$$B1: op_1 \parallel (op_2 \parallel op_3) \approx \text{map_op} \curvearrowright \curvearrowright (op_1 \parallel op_2) \parallel op_3$$

$$B2_1: op \parallel (\mathcal{I} :: (0, 0, 'd) op) \approx \text{map_op} \text{Inl} \text{Inl} op$$

$$B2_2: (\mathcal{I} :: (0, 0, 'd) op) \parallel op \approx \text{map_op} \text{Inr} \text{Inr} op$$

$$B3: (op_1 \bullet op_2) \bullet op_3 \approx op_1 \bullet (op_2 \bullet op_3)$$

$$B4_1: op \sqsubseteq \bullet \mathcal{I} \approx op \sqsubseteq$$

$$B4_2: \mathcal{I} \bullet \sqsupseteq op \approx \sqsupseteq op$$

$$B5: (op_1 \parallel op_2) \bullet (op_3 \parallel op_4) \approx (op_1 \bullet op_3) \parallel (op_2 \bullet op_4)$$

$$B6: \mathcal{I} \parallel \mathcal{I} \approx \mathcal{I}$$

$$B7: \mathcal{X} \bullet \mathcal{X} \approx \mathcal{I}$$

$$B8: (\mathcal{X} :: ('i + 0, 0 + 'i, 'd) op) \approx \text{map_op} \text{id} (\text{case_sum} \text{Inr} \text{Inl}) \mathcal{I}$$

$$B9: \mathcal{X} \approx \text{map_op} \curvearrowright \curvearrowright (\mathcal{X} \parallel \mathcal{I}) \bullet \text{map_op} \text{id} \curvearrowright (\mathcal{I} \parallel \mathcal{X})$$

$$B10: (\sqsupseteq op_1 \parallel \sqsupseteq op_2) \bullet \mathcal{X} \approx \mathcal{X} \bullet (op_2 \sqsubseteq \parallel op_1 \sqsubseteq)$$

$$F1: \mathcal{I} \uparrow \approx (\mathcal{I} :: (0, 0, 'd) op)$$

$$F2: \mathcal{X} \uparrow \approx \mathcal{I}$$

$$R1: \text{Inr} \dashv \text{inputs } op_1 \cap \text{defaults} = \{\} \implies \text{Inr} \dashv \text{outputs } op_1 \cap \text{defaults} = \{\} \implies \\ op_2 \bullet (op_1 \uparrow) \approx ((op_2 \parallel \mathcal{I}) \bullet op_1) \uparrow$$

$$R2: \text{Inr} \dashv \text{inputs } op_1 \cap \text{defaults} = \{\} \implies \text{Inr} \dashv \text{outputs } op_1 \cap \text{defaults} = \{\} \implies \\ (op_1 \uparrow) \bullet op_2 \approx (op_1 \bullet (op_2 \parallel \mathcal{I})) \uparrow$$

$$R3: \text{Inr} \dashv \text{inputs } op_2 \cap \text{defaults} = \{\} \implies \text{Inr} \dashv \text{outputs } op_2 \cap \text{defaults} = \{\} \implies \\ op_1 \parallel (op_2 \uparrow) \approx (\text{map_op} \curvearrowright \curvearrowright (op_1 \parallel op_2)) \uparrow$$

$$R4: \text{Inr} \dashv \text{inputs } op_1 \cap \text{defaults} = \{\} \implies \text{Inr} \dashv \text{outputs } op_1 \cap \text{defaults} = \{\} \implies \\ \text{inputs } op_2 \cap \text{defaults} = \{\} \implies \text{outputs } op_2 \cap \text{defaults} = \{\} \implies \\ (\sqsupseteq op_1 \bullet (\mathcal{I} \parallel op_2)) \uparrow \approx ((\mathcal{I} \parallel op_2) \bullet op_1 \sqsubseteq) \uparrow$$

$$R5: \text{Inr} \dashv \text{inputs } op = \{\} \implies \text{Inr} \dashv \text{outputs } op = \{\} \implies \\ \text{map_op} \text{Inl} \text{Inl} ((op :: ('i + 0, 'o + 0, 'd) op) \uparrow) \approx op$$

$$R6: \text{Inr} \dashv \text{inputs } op = \{\} \implies \text{Inr} \dashv \text{outputs } op = \{\} \implies \\ \text{Inr} \dashv \text{Inl} \dashv \text{inputs } op = \{\} \implies \text{Inr} \dashv \text{Inl} \dashv \text{outputs } op = \{\} \implies \\ (op \uparrow) \approx (\text{map_op} \curvearrowright \curvearrowright op) \uparrow$$

Table 1 Basic network algebra properties

Related Work

Related Work

- A Isabelle/HOL instance of Nondeterministic Asynchronous Dataflow
 - Operators as a shallow embedding as codatatypes
 - 51 axioms proved
- Executable via code extraction to Haskell

Conclusion

Conclusion

- Isabelle/HOL has a good tool set to formalize and reason about stream processing:
 - Codatatypes, coinductive predicates, corecursion with friends, reasoning up to friends (congruence),
 - Coinduction up to congruence principle is automatically derived for codatatypes (but not for coinductive principles)
 - Next step: Feedback loop

Questions, comments and suggestions