Testing Properties of Dataflow Program Operators

Zhihong Xu
U. of Nebraska

Martin Hirzel
IBM Research

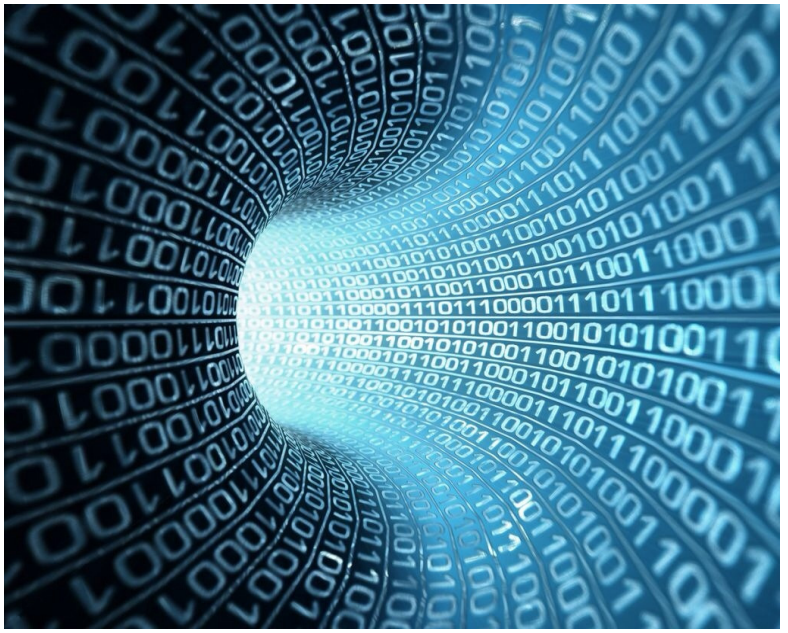
Gregg Rothermel
U. of Nebraska

Kun-Lung Wu
IBM Research

Presentation at ASE, November 2013

“Big Data”

- We need to:
  - Sort data
  - Merge data
  - Order data
  - Extract data



Dataflow Systems

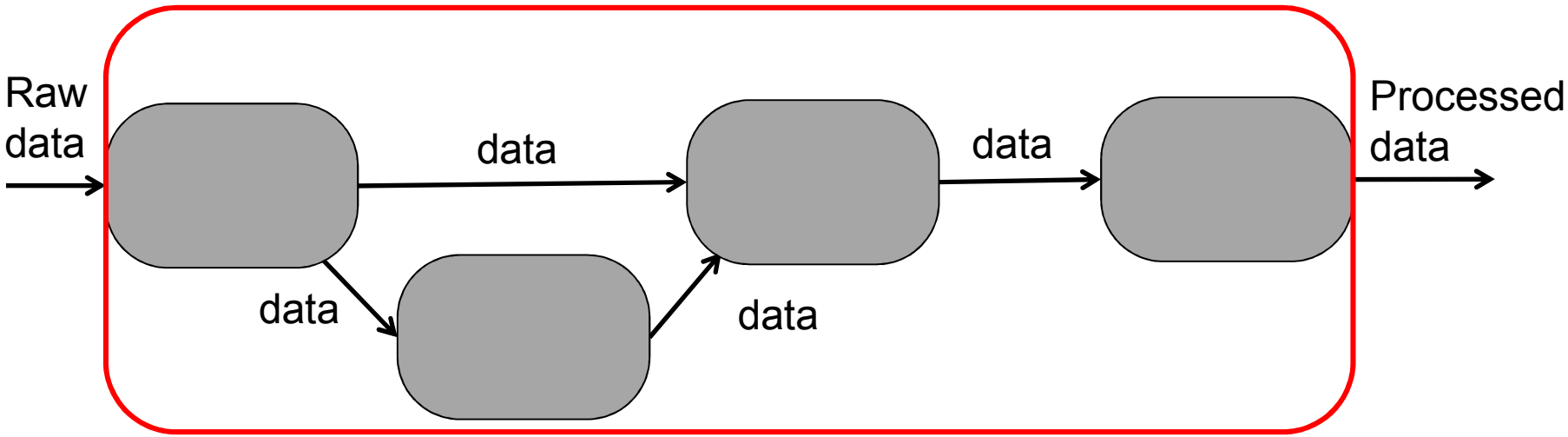
- MapReduce: “batch” approach
- IBM’s InfoSphere Streams: streaming processing approach

Dataflow Programming Languages

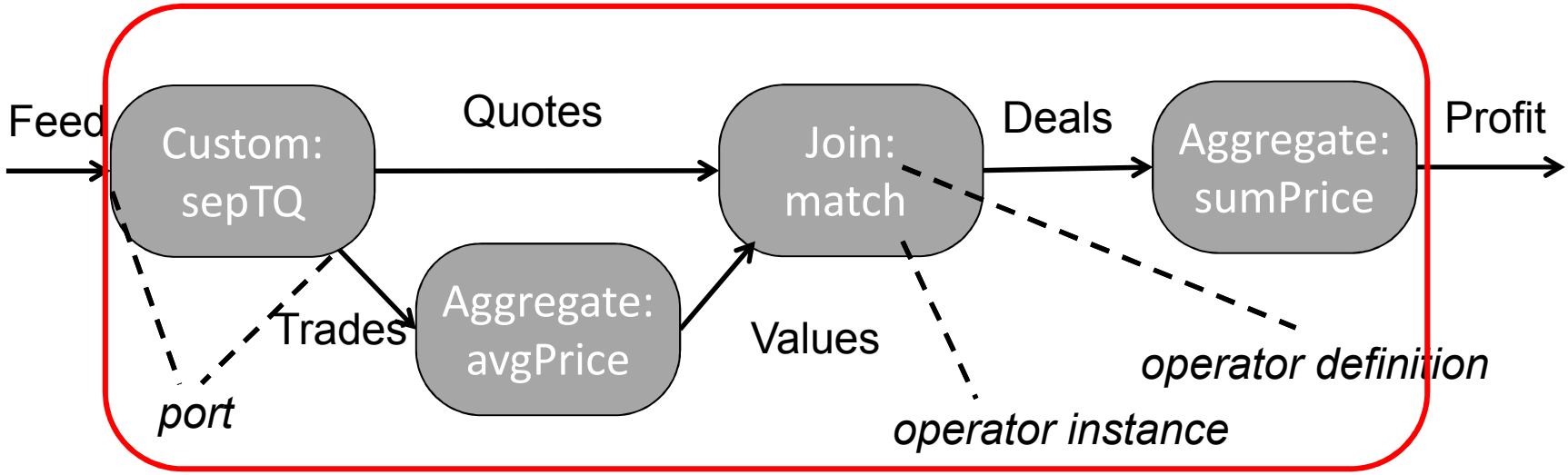
- MapReduce
  - Pig Latin, FlumeJava
- IBM's InfoSphere Streams
  - Streams processing language (SPL)

Dataflow Programs

- Dataflow programs are written in dataflow programming languages and represented as graphs of data streams and operators



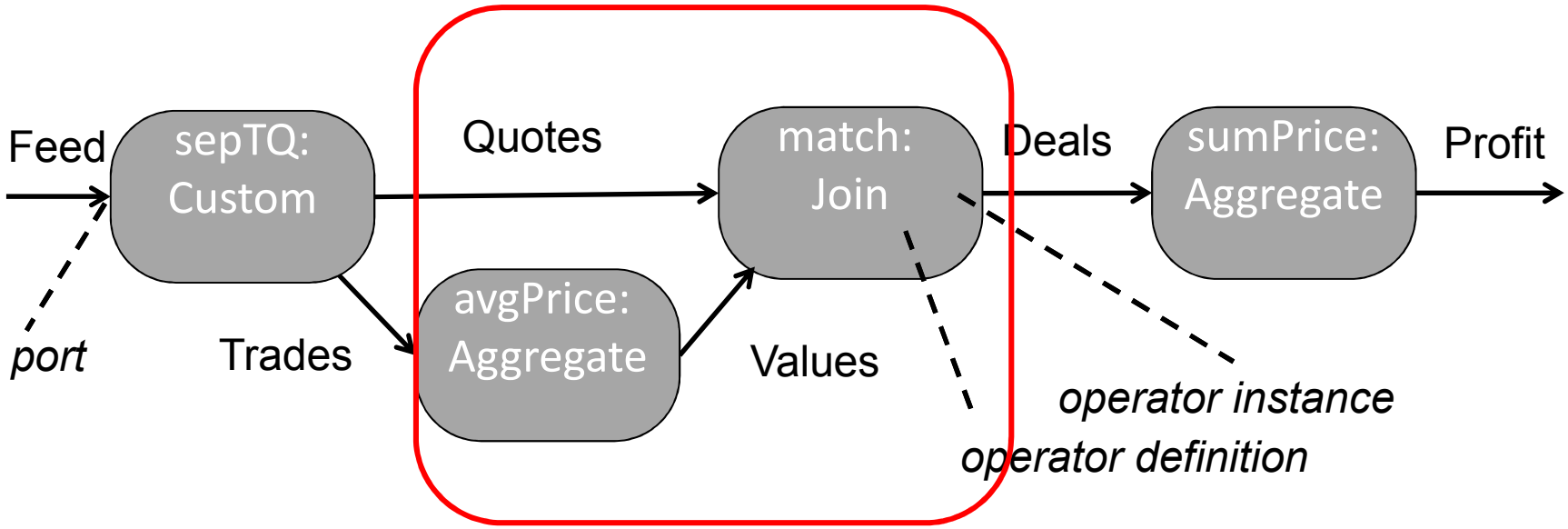
Dataflow Program Operators



Dataflow Program Operator Properties

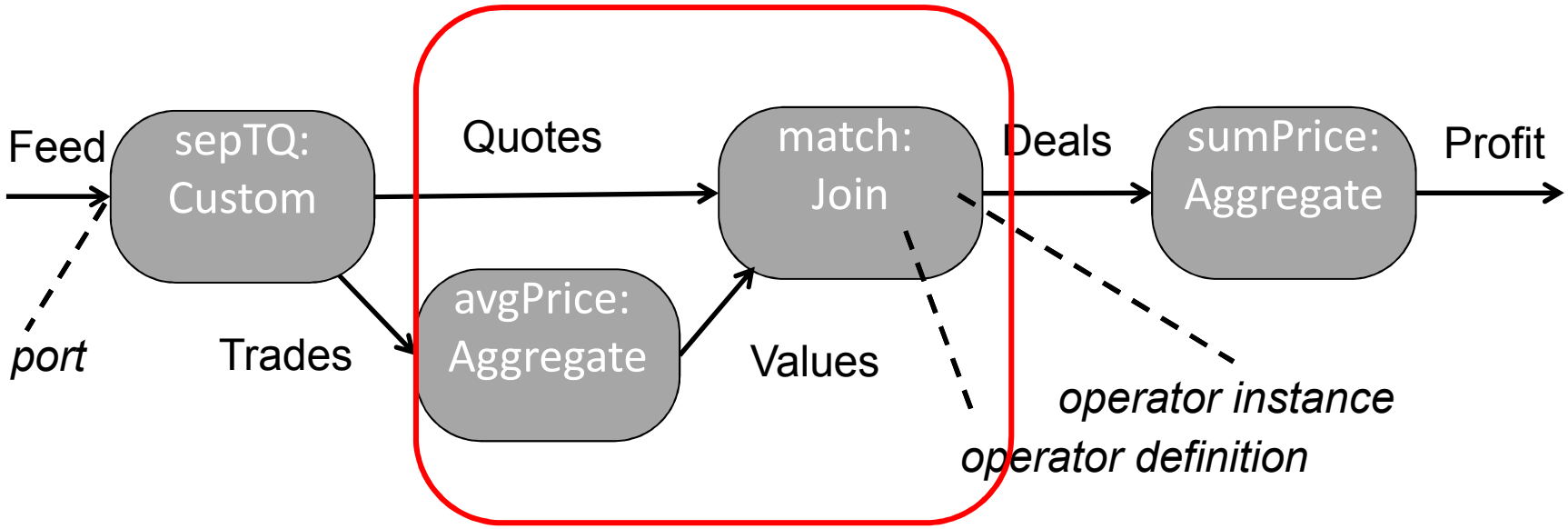
- Non-determinism
  - Does the same sequence of inputs lead to different sequences of outputs?
- Selectivity
  - How many outputs are generated for each input?
- Statefulness
  - Is current output determined by historical inputs?
- Blocking
  - When called, can an operator fail to return
- Non-commutativity
  - Does order of inputs in the same count-based window affect output?
- Partition-interference
  - Do inputs associated with "key" c_2 affect outputs associated with "key" c_1?

Problems: Case 1



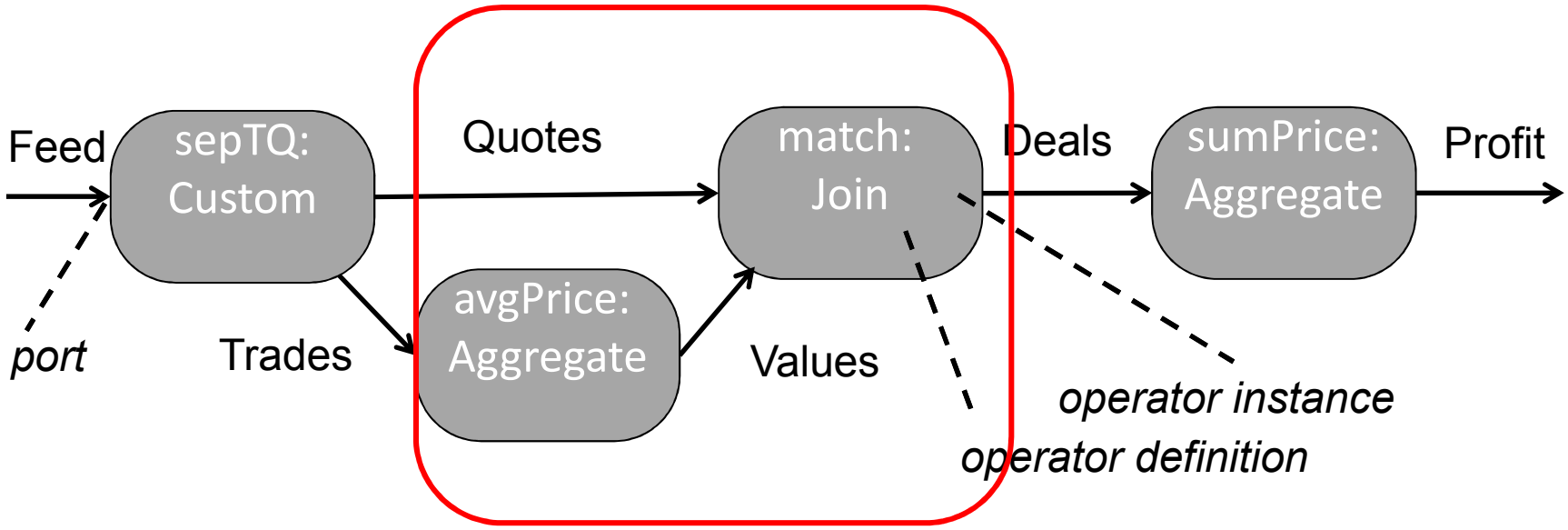
Match : commutative, or blocking or stateful

Problems: Case 1



Match : non-commutative and non-blocking and stateless

Problems: Case 1



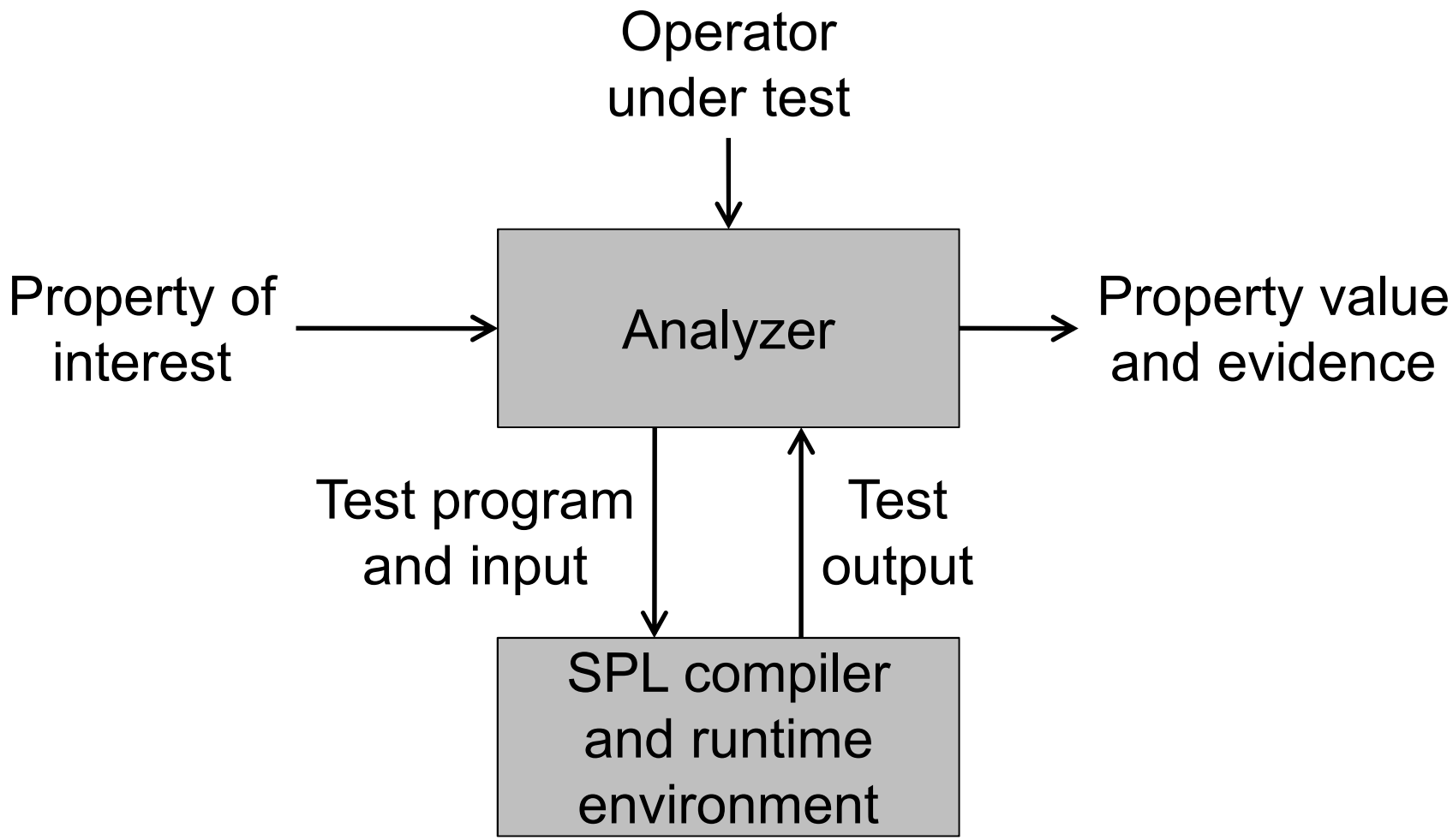
Match : non-commutative and non-blocking and stateless

→ Non-deterministic

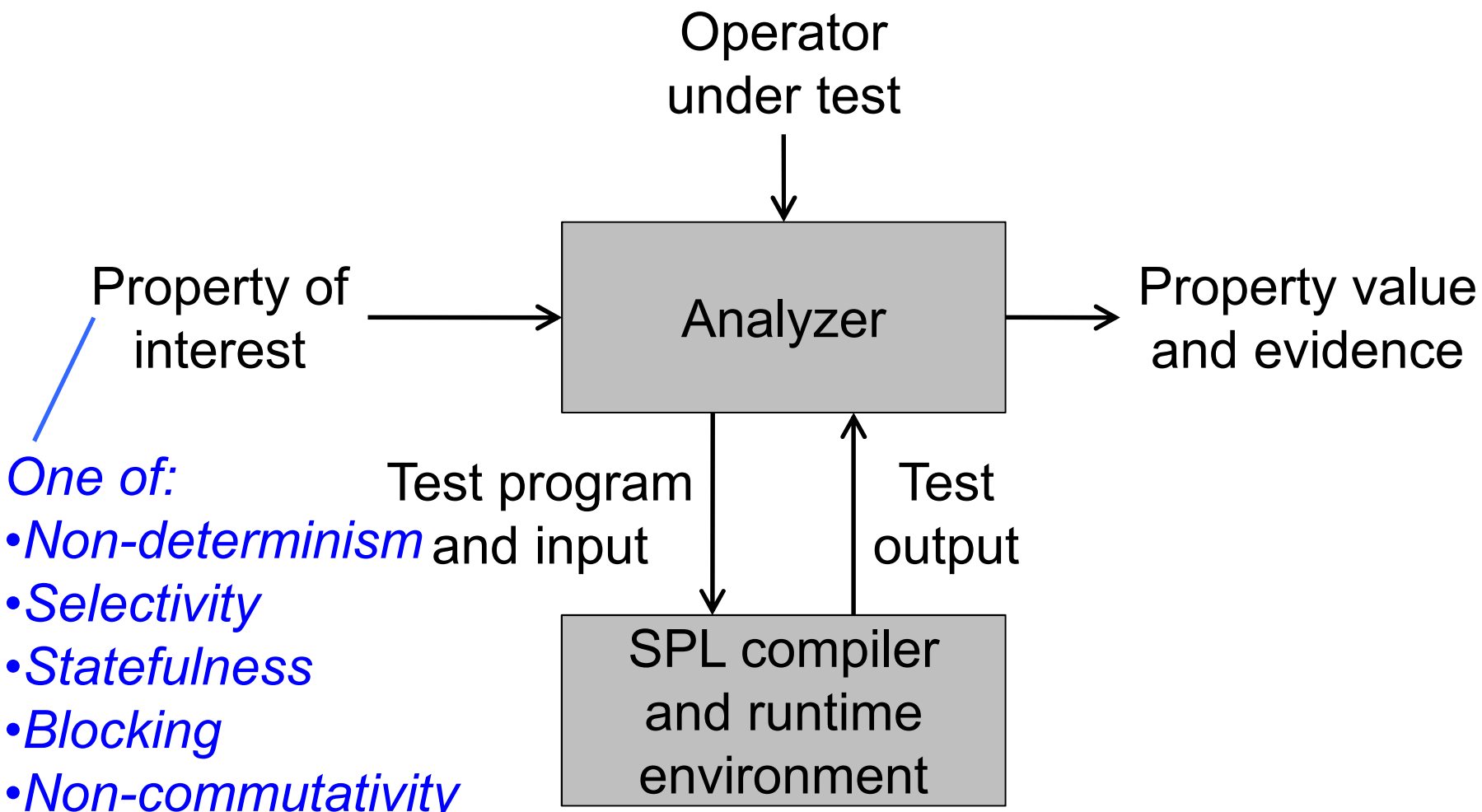
Problems: Case 2

- MapReduce programming model assumes:
  - Map operator: stateless
  - Reduce operator used in combine stage: partition-isolated
- However, languages do not check on those ☹
  - If the properties do not hold, MapReduce may yield unpredictable results

Main Idea: Operator Properties



Main Idea: Operator Properties



Test-Based Dynamic Analysis

Algorithm Algorithm for checking properties

```
evidence = null
initialize(property)
while evidence==null ∧¬(techniqueTimeLimit reached) do
    test = generateTest(property)
    evidence = checkProperty(property, test)
end while
return evidenceValue(property, evidence)
```

Test-Based Dynamic Analysis

FUNCTION IMPLEMENTATIONS FOR PROPERTIES

Property	*initialize*	*generateTest*	*checkProperty*	*evidenceValue*
Non-determinism		Return two lists l_1 and l_2 of triples $\langle delay, d, p\rangle$ with the same values for d and p, but different *delay* values.	Run $l_{1,2}$ with d, p values as inputs, timed by the *delay* values. If the outputs differ, return $\langle l_1, l_2\rangle$, else return **null**.	If *evidence* $\neq$ **null**, return "definitely non-deterministic", else return "potentially deterministic".
Selectivity	Set *selectiveEvidence* = **null**.	Return list l of pairs $\langle d, p\rangle$.	Run l. If any firing has > 1 output data items, return l. If any firing has 0 output data items, set *selectiveEvidence*=l. Return **null**.	If *evidence* $\neq$ **null**, return "definitely prolific", else if *selectiveEvidence* $\neq$ **null**, return "potentially selective", else return "potentially one-to-one".
Blocking		Return list l of pairs $\langle d, p\rangle$, where all p values are the port-under-test.	Run l. If any firing stalls for *blockingTimeLimit*, return l, else return **null**.	If *evidence* $\neq$ **null**, return "potentially blocking", else return "potentially non-blocking".
Statefulness		Return list l of pairs $\langle d, p\rangle$, where all d values are the same for simplicity.	Run l. If any two firings produce different outputs, return l, else return **null**.	If *evidence* $\neq$ **null**, return "definitely stateful", else return "potentially stateless".
Non-commutativity	Generate and run tests l of increasing length until a window punctuation is generated. Let w be the length of such a list.	Generate list l of w pairs $\langle d, p\rangle$; return permutations of l.	Run each permutation. If any two permutations produce different outputs, return them, else return **null**.	If *evidence* $\neq$ **null**, return "definitely non-commutative", else return "potentially commutative".
Partition-interference		Generate list l_1 of pairs $\langle d, p\rangle$ with the same $read(d, k)$ for all d. Create l_2 containing elements from l_1 interspersed with data items with different keys.	Run l_1 and l_2. If any of the outputs for corresponding data items differ, return $\langle l_1, l_2\rangle$, else return **null**.	If *evidence* $\neq$ **null**, return "definitely partition-interfering", else return "potentially partition-isolated".

Test-Based Dynamic Analysis

FUNCTION IMPLEMENTATIONS FOR PROPERTIES

Property	*initialize*	*generateTest*	*checkProperty*	*evidenceValue*
Non-determinism		Return two lists l_1 and l_2 of triples $\langle delay, d, p\rangle$ with the same values for d and p, but different *delay* values.	Run $l_{1,2}$ with d, p values as inputs, timed by the *delay* values. If the outputs differ, return $\langle l_1, l_2\rangle$, else return **null**.	If *evidence* $\neq$ **null**, return "definitely non-deterministic", else return "potentially deterministic".
Selectivity	Set *selectiveEvidence* = **null**.	Return list l of pairs $\langle d, p\rangle$.	Run l. If any firing has > 1 output data items, return l. If any firing has 0 output data items, set *selectiveEvidence*=l. Return **null**.	If *evidence* $\neq$ **null**, return "definitely prolific", else if *selectiveEvidence* $\neq$ **null**, return "potentially selective", else return "potentially one-to-one".
Blocking		Return list l of pairs $\langle d, p\rangle$, where all p values are the port-under-test.	Run l. If any firing stalls for *blockingTimeLimit*, return l, else return **null**.	If *evidence* $\neq$ **null**, return "potentially blocking", else return "potentially non-blocking".
Statefulness		Return list l of pairs $\langle d, p\rangle$, where all d values are the same for simplicity.	Run l. If any two firings produce different outputs, return l, else return **null**.	If *evidence* $\neq$ **null**, return "definitely stateful", else return "potentially stateless".
Non-commutativity	Generate and run tests l of increasing length until a window punctuation is generated. Let w be the length of such a list.	Generate list l of w pairs $\langle d, p\rangle$; return permutations of l.	Run each permutation. If any two permutations produce different outputs, return them, else return **null**.	If *evidence* $\neq$ **null**, return "definitely non-commutative", else return "potentially commutative".
Partition-interference		Generate list l_1 of pairs $\langle d, p\rangle$ with the same $read(d, k)$ for all d. Create l_2 containing elements from l_1 interspersed with data items with different keys.	Run l_1 and l_2. If any of the outputs for corresponding data items differ, return $\langle l_1, l_2\rangle$, else return **null**.	If *evidence* $\neq$ **null**, return "definitely partition-interfering", else return "potentially partition-isolated".

Test Generation Techniques

- Base techniques
  - Random: just choose inputs randomly
  - Pool: pick constants used in code
  - Pair: systematically cover “pool” combinations
- Two reuse approaches
  - Use tests from one property for another or not
- Two mutation approaches
  - Try small variations, e.g., ±1, or not
- Hybrid technique

Test Generation Techniques

- Base techniques
  - Random: just choo inputs randomly
  - Pool: pick constants used in code
  - Pair: systematically co er "pool" combinations
- Two reuse approaches
  - Use tests from one property for another or not
- Two mutation approaches
  - Try small variations, e.g., ±1, no
- Hybrid technique

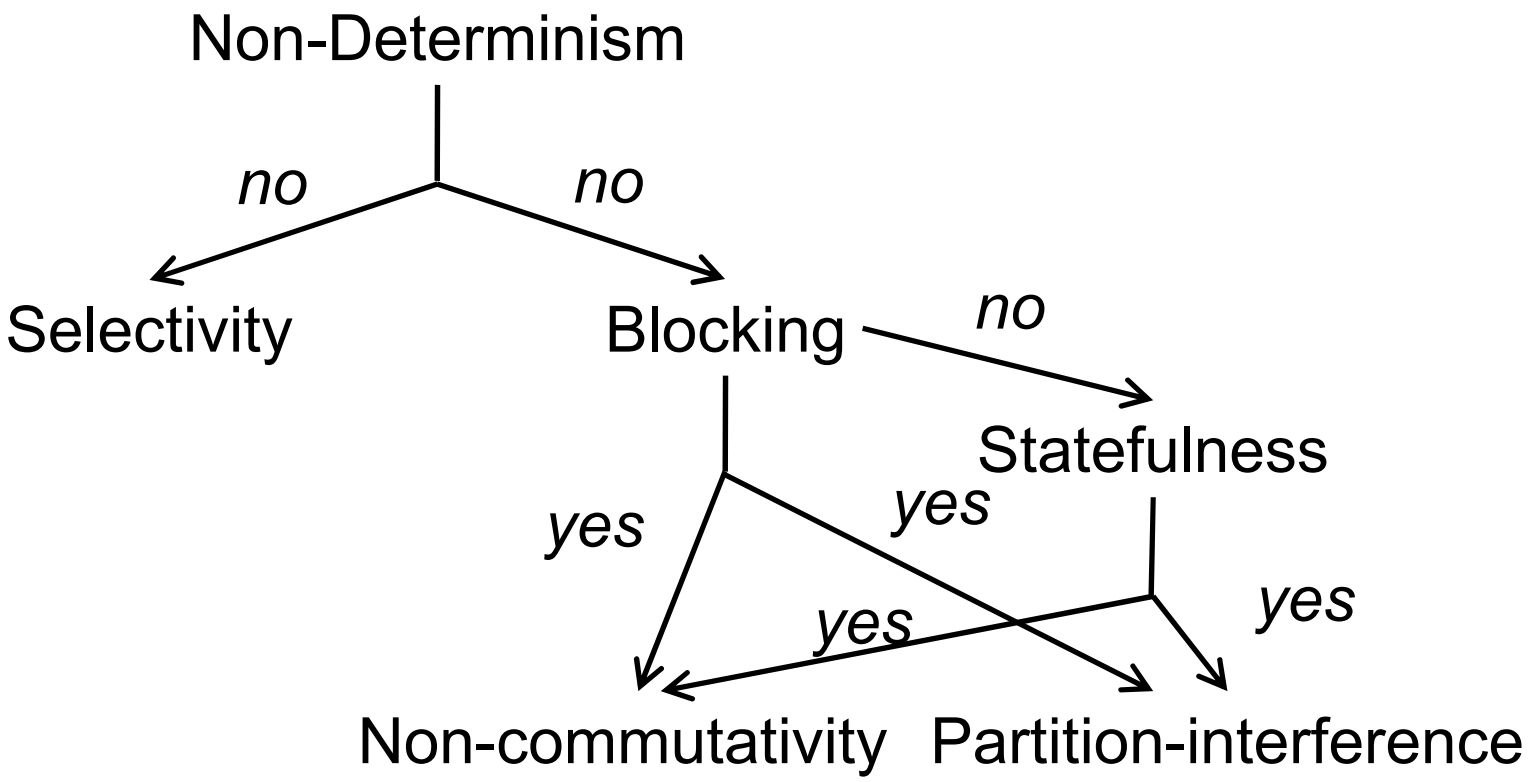
3 × 2 × 2 = 12

Test Generation Techniques

- Base techniques
  - Random: just choo inputs randomly
  - Pool: pick constants used in code
  - Pair: systematically co er "pool" combinations
- Two reuse approaches
  - Use tests from one property for another or not
- Two mutation approaches
  - Try small variations, e.g., ±1, no
- Hybrid technique

3 × 2 × 2 = 12

Testing Order



Empirical Study

- Research questions
  - RQ1: How do the test generation techniques fare, in terms of precision and recall, in testing the six properties considered
  - RQ2: How do the test generation techniques fare, in terms of efficiency, in testing the six properties considered

Object of Analysis

- IBM's InfoSphere Streams
- 56 operators
- 4 in Java and 6 in C++
- 145.7 lines of code on average

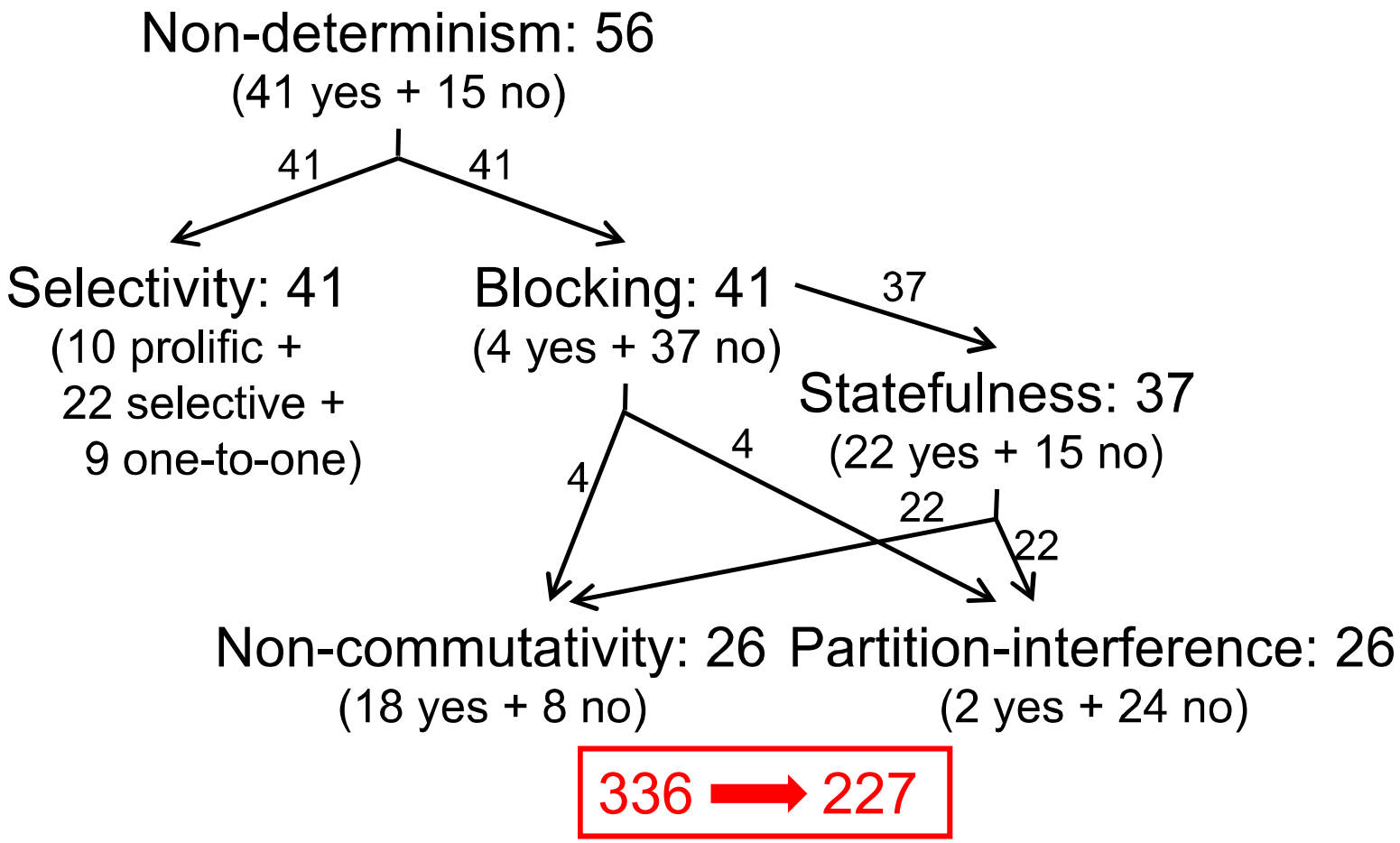
Independent and Dependent Variables

- Independent variables
  - 13 test generation techniques
- Dependent variables
  - Recall
    - For properties where there is evidence how often do we conclude correctly?
  - Precision
    - For properties where there is no evidence how often do we conclude correctly?
  - Time to find evidence
    - Average time to find evidence, or timeout

Effectiveness

Efficiency

Applying Test Order



Results: Effectiveness

- All techniques achieved 100% precision in all cases

Results: Effectiveness

RECALL

Property	Without Reuse						With Reuse						Hybrid
	Without Mutation			With Mutation			Without Mutation			With Mutation			
	Rand	Pool	Pair	Rand	Pool	Pair	Rand	Pool	Pair	Rand	Pool	Pair	
Non-determinism	48.7	53.3	52.0	24.0	40.7	36.7	-	-	-	-	-	-	***74.7***
Selectivity	75.3	93.4	91.6	91.9	***100.0***	***100.0***	84.7	93.8	93.1	93.8	***100.0***	***100.0***	***100.0***
Blocking	***83.3***	***83.3***	***83.3***	***83.3***	***83.3***	***83.3***	***83.3***	***83.3***	***83.3***	***83.3***	***83.3***	***83.3***	***83.3***
Statefulness	68.2	72.7	72.7	72.7	81.8	81.8	77.3	81.8	81.8	81.8	***86.4***	***86.4***	***86.4***
Non-commutativity	35.0	71.3	73.8	35.0	72.5	77.5	32.5	71.3	75.0	38.8	70.0	72.5	***90.0***
Partition-interference	80.0	93.3	88.8	79.6	90.8	91.7	83.8	90.4	88.8	83.8	88.8	89.6	***94.6***

- Tested 56 operator instances for determinism, subsets for other properties
- Timeout = 3 minutes

Results: Effectiveness

RECALL

Property	Without Reuse						With Reuse						Hybrid
	Without Mutation			With Mutation			Without Mutation			With Mutation			
	Rand	Pool	Pair	Rand	Pool	Pair	Rand	Pool	Pair	Rand	Pool	Pair	
Non-determinism	48.7	53.3	52.0	24.0	40.7	36.7	-	-	-	-	-	-	***74.7***
Selectivity	75.3	93.4	91.6	91.9	***100.0***	***100.0***	84.7	93.8	93.1	93.8	***100.0***	***100.0***	***100.0***
Blocking	***83.3***	***83.3***	***83.3***	***83.3***	***83.3***	***83.3***	***83.3***	***83.3***	***83.3***	***83.3***	***83.3***	***83.3***	***83.3***
Statefulness	68.2	72.7	72.7	72.7	81.8	81.8	77.3	81.8	81.8	81.8	***86.4***	***86.4***	***86.4***
Non-commutativity	35.0	71.3	73.8	35.0	72.5	77.5	32.5	71.3	75.0	38.8	70.0	72.5	***90.0***
Partition-interference	80.0	93.3	88.8	79.6	90.8	91.7	83.8	90.4	88.8	83.8	88.8	89.6	***94.6***

- Tested 56 operator instances for determinism, subsets for other properties
- Timeout = 3 minutes
- Applied each technique 10 times to each operator

Results: Effectiveness

Property	Recall												
	Without Reuse						With Reuse						
	Without Mutation			With Mutation			Without Mutation			With Mutation			Hybrid
	Rand	Pool	Pair	Rand	Pool	Pair	Rand	Pool	Pair	Rand	Pool	Pair	
Non-determinism	48.7	53.3	52.0	24.0	40.7	36.7	-	-	-	-	-	-	***74.7***
Selectivity	75.3	93.4	91.6	91.9	***100.0***	***100.0***	84.7	93.8	93.1	93.8	***100.0***	***100.0***	***100.0***
Blocking	***83.3***	***83.3***	***83.3***	***83.3***	***83.3***	***83.3***	***83.3***	***83.3***	***83.3***	***83.3***	***83.3***	***83.3***	***83.3***
Statefulness	68.2	72.7	72.7	72.7	81.8	81.8	77.3	81.8	81.8	81.8	***86.4***	***86.4***	***86.4***
Non-commutativity	35.0	71.3	73.8	35.0	72.5	77.5	32.5	71.3	75.0	38.8	70.0	72.5	***90.0***
Partition-interference	80.0	93.3	88.8	79.6	90.8	91.7	83.8	90.4	88.8	83.8	88.8	89.6	***94.6***

- Tested 56 operator instances for determinism, subsets for other properties
- Timeout = 3 minutes
- Applied each technique 10 times to each operator

Results: Effectiveness

RECALL

Property	Without Reuse						With Reuse						Hybrid
	Without Mutation			With Mutation			Without Mutation			With Mutation			
	Rand	Pool	Pair	Rand	Pool	Pair	Rand	Pool	Pair	Rand	Pool	Pair	
Non-determinism	48.7	53.3	52.0	24.0	40.7	36.7	-	-	-	-	-	-	***74.7***
Selectivity	75.3	93.4	91.6	91.9	***100.0***	***100.0***	84.7	93.8	93.1	93.8	***100.0***	***100.0***	***100.0***
Blocking	***83.3***	***83.3***	***83.3***	***83.3***	***83.3***	***83.3***	***83.3***	***83.3***	***83.3***	***83.3***	***83.3***	***83.3***	***83.3***
Statefulness	68.2	72.7	72.7	72.7	81.8	81.8	77.3	81.8	81.8	81.8	***86.4***	***86.4***	***86.4***
Non-commutativity	35.0	71.3	73.8	35.0	72.5	77.5	32.5	71.3	75.0	38.8	70.0	72.5	***90.0***
Partition-interference	80.0	93.3	88.8	79.6	90.8	91.7	83.8	90.4	88.8	83.8	88.8	89.6	***94.6***

- Tested 56 operator instances for determinism, subsets for other properties
- Timeout = 3 minutes
- Applied each technique 10 times to each operator

Results: Effectiveness

RECALL

Property	Without Reuse						With Reuse						Hybrid
	Without Mutation			With Mutation			Without Mutation			With Mutation			
	Rand	Pool	Pair	Rand	Pool	Pair	Rand	Pool	Pair	Rand	Pool	Pair	
Non-determinism	48.7	53.3	52.0	24.0	40.7	36.7	-	-	-	-	-	-	***74.7***
Selectivity	75.3	93.4	91.6	91.9	***100.0***	***100.0***	84.7	93.8	93.1	93.8	***100.0***	***100.0***	***100.0***
Blocking	***83.3***	***83.3***	***83.3***	***83.3***	***83.3***	***83.3***	***83.3***	***83.3***	***83.3***	***83.3***	***83.3***	***83.3***	***83.3***
Statefulness	68.2	72.7	72.7	72.7	81.8	81.8	77.3	81.8	81.8	81.8	***86.4***	***86.4***	***86.4***
Non-commutativity	35.0	71.3	73.8	35.0	72.5	77.5	32.5	71.3	75.0	38.8	70.0	72.5	***90.0***
Partition-interference	80.0	93.3	88.8	79.6	90.8	91.7	83.8	90.4	88.8	83.8	88.8	89.6	***94.6***

- Tested 56 operator instances for determinism, subsets for other properties
- Timeout = 3 minutes
- Applied each technique 10 times to each operator

Results: Effectiveness

RECALL

Property	Without Reuse						With Reuse						Hybrid
	Without Mutation			With Mutation			Without Mutation			With Mutation			
	Rand	Pool	Pair	Rand	Pool	Pair	Rand	Pool	Pair	Rand	Pool	Pair	
Non-determinism	48.7	53.3	52.0	24.0	40.7	36.7	-	-	-	-	-	-	***74.7***
Selectivity	75.3	93.4	91.6	91.9	***100.0***	***100.0***	84.7	93.8	93.1	93.8	***100.0***	***100.0***	***100.0***
Blocking	***83.3***	***83.3***	***83.3***	***83.3***	***83.3***	***83.3***	***83.3***	***83.3***	***83.3***	***83.3***	***83.3***	***83.3***	***83.3***
Statefulness	68.2	72.7	72.7	72.7	81.8	81.8	77.3	81.8	81.8	81.8	***86.4***	***86.4***	***86.4***
Non-commutativity	35.0	71.3	73.8	35.0	72.5	77.5	32.5	71.3	75.0	38.8	70.0	72.5	***90.0***
Partition-interference	80.0	93.3	88.8	79.6	90.8	91.7	83.8	90.4	88.8	83.8	88.8	89.6	***94.6***

- Tested 56 operator instances for determinism, subsets for other properties
- Timeout = 3 minutes
- Applied each technique 10 times to each operator

Results: Effectiveness

RECALL

Property	Without Reuse						With Reuse						Hybrid
	Without Mutation			With Mutation			Without Mutation			With Mutation			
	Rand	Pool	Pair	Rand	Pool	Pair	Rand	Pool	Pair	Rand	Pool	Pair	
Non-determinism	48.7	53.3	52.0	24.0	40.7	36.7	-	-	-	-	-	-	***74.7***
Selectivity	75.3	93.4	91.6	91.9	***100.0***	***100.0***	84.7	93.8	93.1	93.8	***100.0***	***100.0***	***100.0***
Blocking	***83.3***	***83.3***	***83.3***	***83.3***	***83.3***	***83.3***	***83.3***	***83.3***	***83.3***	***83.3***	***83.3***	***83.3***	***83.3***
Statefulness	68.2	72.7	72.7	72.7	81.8	81.8	77.3	81.8	81.8	81.8	***86.4***	***86.4***	***86.4***
Non-commutativity	35.0	71.3	73.8	35.0	72.5	77.5	32.5	71.3	75.0	38.8	70.0	72.5	***90.0***
Partition-interference	80.0	93.3	88.8	79.6	90.8	91.7	83.8	90.4	88.8	83.8	88.8	89.6	***94.6***

- Tested 56 operator instances for determinism, subsets for other properties
- Timeout = 3 minutes
- Applied each technique 10 times to each operator

Results: Effectiveness

RECALL

Property	Without Reuse						With Reuse						Hybrid
	Without Mutation			With Mutation			Without Mutation			With Mutation			
	Rand	Pool	Pair	Rand	Pool	Pair	Rand	Pool	Pair	Rand	Pool	Pair	
Non-determinism	48.7	53.3	52.0	24.0	40.7	36.7	-	-	-	-	-	-	***74.7***
Selectivity	75.3	93.4	91.6	91.9	***100.0***	***100.0***	84.7	93.8	93.1	93.8	***100.0***	***100.0***	***100.0***
Blocking	***83.3***	***83.3***	***83.3***	***83.3***	***83.3***	***83.3***	***83.3***	***83.3***	***83.3***	***83.3***	***83.3***	***83.3***	***83.3***
Statefulness	68.2	72.7	72.7	72.7	81.8	81.8	77.3	81.8	81.8	81.8	***86.4***	***86.4***	***86.4***
Non-commutativity	35.0	71.3	73.8	35.0	72.5	77.5	32.5	71.3	75.0	38.8	70.0	72.5	***90.0***
Partition-interference	80.0	93.3	88.8	79.6	90.8	91.7	83.8	90.4	88.8	83.8	88.8	89.6	***94.6***

- Tested 56 operator instances for determinism, subsets for other properties
- Timeout = 3 minutes
- Applied each technique 10 times to each operator

Results: Effectiveness

RECALL

Property	Without Reuse						With Reuse						Hybrid
	Without Mutation			With Mutation			Without Mutation			With Mutation			
	Rand	Pool	Pair	Rand	Pool	Pair	Rand	Pool	Pair	Rand	Pool	Pair	
Non-determinism	48.7	53.3	52.0	24.0	40.7	36.7	-	-	-	-	-	-	***74.7***
Selectivity	75.3	93.4	91.6	91.9	***100.0***	***100.0***	84.7	93.8	93.1	93.8	***100.0***	***100.0***	***100.0***
Blocking	***83.3***	***83.3***	***83.3***	***83.3***	***83.3***	***83.3***	***83.3***	***83.3***	***83.3***	***83.3***	***83.3***	***83.3***	***83.3***
Statefulness	68.2	72.7	72.7	72.7	81.8	81.8	77.3	81.8	81.8	81.8	***86.4***	***86.4***	***86.4***
Non-commutativity	35.0	71.3	73.8	35.0	72.5	77.5	32.5	71.3	75.0	38.8	70.0	72.5	***90.0***
Partition-interference	80.0	93.3	88.8	79.6	90.8	91.7	83.8	90.4	88.8	83.8	88.8	89.6	***94.6***

- Tested 56 operator instances for determinism, subsets for other properties
- Timeout = 3 minutes
- Applied each technique 10 times to each operator

Results: Effectiveness

RECALL

Property	Without Reuse						With Reuse						Hybrid
	Without Mutation			With Mutation			Without Mutation			With Mutation			
	Rand	Pool	Pair	Rand	Pool	Pair	Rand	Pool	Pair	Rand	Pool	Pair	
Non-determinism	48.7	53.3	52.0	24.0	40.7	36.7	-	-	-	-	-	-	***74.7***
Selectivity	75.3	93.4	91.6	91.9	***100.0***	***100.0***	84.7	93.8	93.1	93.8	***100.0***	***100.0***	***100.0***
Blocking	***83.3***	***83.3***	***83.3***	***83.3***	***83.3***	***83.3***	***83.3***	***83.3***	***83.3***	***83.3***	***83.3***	***83.3***	***83.3***
Statefulness	68.2	72.7	72.7	72.7	81.8	81.8	77.3	81.8	81.8	81.8	***86.4***	***86.4***	***86.4***
Non-commutativity	35.0	71.3	73.8	35.0	72.5	77.5	32.5	71.3	75.0	38.8	70.0	72.5	***90.0***
Partition-interference	80.0	93.3	88.8	79.6	90.8	91.7	83.8	90.4	88.8	83.8	88.8	89.6	***94.6***

- Tested 56 operator instances for determinism, subsets for other properties
- Timeout = 3 minutes
- Applied each technique 10 times to each operator
- Most effective: hybrid

Results: Effectiveness

RECALL

Property	Without Reuse						With Reuse						Hybrid
	Without Mutation			With Mutation			Without Mutation			With Mutation			
	Rand	Pool	Pair	Rand	Pool	Pair	Rand	Pool	Pair	Rand	Pool	Pair	
Non-determinism	48.7	53.3	52.0	24.0	40.7	36.7	-	-	-	-	-	-	***74.7***
Selectivity	75.3	93.4	91.6	91.9	***100.0***	***100.0***	84.7	93.8	93.1	93.8	***100.0***	***100.0***	***100.0***
Blocking	***83.3***	***83.3***	***83.3***	***83.3***	***83.3***	***83.3***	***83.3***	***83.3***	***83.3***	***83.3***	***83.3***	***83.3***	***83.3***
Statefulness	68.2	72.7	72.7	72.7	81.8	81.8	77.3	81.8	81.8	81.8	***86.4***	***86.4***	***86.4***
Non-commutativity	35.0	71.3	73.8	35.0	72.5	77.5	32.5	71.3	75.0	38.8	70.0	72.5	***90.0***
Partition-interference	80.0	93.3	88.8	79.6	90.8	91.7	83.8	90.4	88.8	83.8	88.8	89.6	***94.6***

- Tested 56 operator instances for determinism, subsets for other properties
- Timeout = 3 minutes
- Applied each technique 10 times to each operator
- Most effective: hybrid

Results: Effectiveness

RECALL

Property	Without Reuse						With Reuse						Hybrid
	Without Mutation			With Mutation			Without Mutation			With Mutation			
	Rand	Pool	Pair	Rand	Pool	Pair	Rand	Pool	Pair	Rand	Pool	Pair	
Non-determinism	48.7	53.3	52.0	24.0	40.7	36.7	-	-	-	-	-	-	***74.7***
Selectivity	75.3	93.4	91.6	91.9	***100.0***	***100.0***	84.7	93.8	93.1	93.8	***100.0***	***100.0***	***100.0***
Blocking	***83.3***	***83.3***	***83.3***	***83.3***	***83.3***	***83.3***	***83.3***	***83.3***	***83.3***	***83.3***	***83.3***	***83.3***	***83.3***
Statefulness	68.2	72.7	72.7	72.7	81.8	81.8	77.3	81.8	81.8	81.8	***86.4***	***86.4***	***86.4***
Non-commutativity	35.0	71.3	73.8	35.0	72.5	77.5	32.5	71.3	75.0	38.8	70.0	72.5	***90.0***
Partition-interference	80.0	93.3	88.8	79.6	90.8	91.7	83.8	90.4	88.8	83.8	88.8	89.6	***94.6***

- Tested 56 operator instances for determinism, subsets for other properties
- Timeout = 3 minutes
- Applied each technique 10 times to each operator
- Most effective: hybrid

Results: Effectiveness

RECALL

Property	Without Reuse						With Reuse						Hybrid
	Without Mutation			With Mutation			Without Mutation			With Mutation			
	Rand	Pool	Pair	Rand	Pool	Pair	Rand	Pool	Pair	Rand	Pool	Pair	
Non-determinism	48.7	53.3	52.0	24.0	40.7	36.7	-	-	-	-	-	-	***74.7***
Selectivity	75.3	93.4	91.6	91.9	***100.0***	***100.0***	84.7	93.8	93.1	93.8	***100.0***	***100.0***	***100.0***
Blocking	***83.3***	***83.3***	***83.3***	***83.3***	***83.3***	***83.3***	***83.3***	***83.3***	***83.3***	***83.3***	***83.3***	***83.3***	***83.3***
Statefulness	68.2	72.7	72.7	72.7	81.8	81.8	77.3	81.8	81.8	81.8	***86.4***	***86.4***	***86.4***
Non-commutativity	35.0	71.3	73.8	35.0	72.5	77.5	32.5	71.3	75.0	38.8	70.0	72.5	***90.0***
Partition-interference	80.0	93.3	88.8	79.6	90.8	91.7	83.8	90.4	88.8	83.8	88.8	89.6	***94.6***

- Tested 56 operator instances for determinism, subsets for other properties
- Timeout = 3 minutes
- Applied each technique 10 times to each operator
- Most effective: hybrid

Results: Efficiency

EFFICIENCY

Property	Without Reuse						With Reuse						Hybrid
	Without Mutation			With Mutation			Without Mutation			With Mutation			
	Rand	Pool	Pair	Rand	Pool	Pair	Rand	Pool	Pair	Rand	Pool	Pair	
Non-determinism	114.9	109.8	118.7	146.0	127.8	136.3	-	-	-	-	-	-	***78.8***
Selectivity	149.2	134.9	134.8	136.5	126.2	126.8	129.8	126.3	126.7	130.0	125.1	125.4	***124.3***
Blocking	32.4	32.3	32.4	32.1	32.1	32.1	32.0	***31.9***	***31.9***	32.0	32.0	32.0	***31.9***
Statefulness	62.2	53.4	47.5	55.5	41.6	36.2	45.2	37.4	32.3	48.8	40.4	37.7	***25.8***
Non-commutativity	126.0	63.6	55.3	125.8	56.6	48.4	127.7	61.5	56.1	120.8	59.2	58.3	***23.5***
Partition-interference	58.6	39.8	43.4	53.0	38.0	38.0	55.3	47.8	51.4	56.7	53.0	56.5	***18.3***

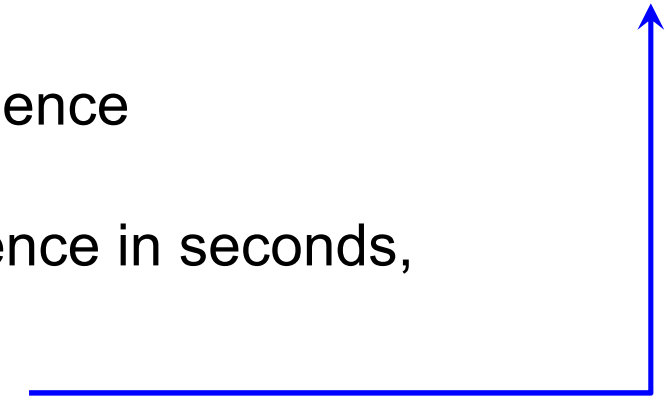
- Tested only operators for which ∃ evidence
- Timeout = 3 minutes
- Efficiency = average time to find evidence in seconds

Results: Efficiency

EFFICIENCY

Property	Without Reuse						With Reuse						Hybrid
	Without Mutation			With Mutation			Without Mutation			With Mutation			
	Rand	Pool	Pair	Rand	Pool	Pair	Rand	Pool	Pair	Rand	Pool	Pair	
Non-determinism	114.9	109.8	118.7	146.0	127.8	136.3	-	-	-	-	-	-	***78.8***
Selectivity	149.2	134.9	134.8	136.5	126.2	126.8	129.8	126.3	126.7	130.0	125.1	125.4	***124.3***
Blocking	32.4	32.3	32.4	32.1	32.1	32.1	32.0	***31.9***	***31.9***	32.0	32.0	32.0	***31.9***
Statefulness	62.2	53.4	47.5	55.5	41.6	36.2	45.2	37.4	32.3	48.8	40.4	37.7	***25.8***
Non-commutativity	126.0	63.6	55.3	125.8	56.6	48.4	127.7	61.5	56.1	120.8	59.2	58.3	***23.5***
Partition-interference	58.6	39.8	43.4	53.0	38.0	38.0	55.3	47.8	51.4	56.7	53.0	56.5	***18.3***

- Tested only operators for which ∃ evidence
- Timeout = 3 minutes
- Efficiency = average time to find evidence in seconds, or reach time out
- Most efficient: hybrid



Results: Efficiency

EFFICIENCY

Property	Without Reuse						With Reuse						Hybrid
	Without Mutation			With Mutation			Without Mutation			With Mutation			
	Rand	Pool	Pair	Rand	Pool	Pair	Rand	Pool	Pair	Rand	Pool	Pair	
Non-determinism	114.9	109.8	118.7	146.0	127.8	136.3	-	-	-	-	-	-	***78.8***
Selectivity	149.2	134.9	134.8	136.5	126.2	126.8	129.8	126.3	126.7	130.0	125.1	125.4	***124.3***
Blocking	32.4	32.3	32.4	32.1	32.1	32.1	32.0	***31.9***	***31.9***	32.0	32.0	32.0	***31.9***
Statefulness	62.2	53.4	47.5	55.5	41.6	36.2	45.2	37.4	32.3	48.8	40.4	37.7	***25.8***
Non-commutativity	126.0	63.6	55.3	125.8	56.6	48.4	127.7	61.5	56.1	120.8	59.2	58.3	***23.5***
Partition-interference	58.6	39.8	43.4	53.0	38.0	38.0	55.3	47.8	51.4	56.7	53.0	56.5	***18.3***

- Tested only operators for which ∃ evidence
- Timeout = 3 minutes
- Efficiency = average time to find evidence in seconds, or reach time out
- Most efficient: hybrid

Summary

- Overall, the dynamic approach is effective and efficient
- Test generation approaches
  - Pool and pair were better than random
- Mutation strategy
  - Not always better
- Reuse strategy
  - Not always better

Conclusions

- Defined 6 properties
- Developed a testing approach for testing dataflow program operators on properties and an efficient testing order
- Did an empirical study: results on 56 operators showing effective + efficient
- Ideas have been applied on map and reduce functions for MapReduce applications (IISWC 2013)