Bloom Filter Based Routing for Content-Based Publish/Subscribe

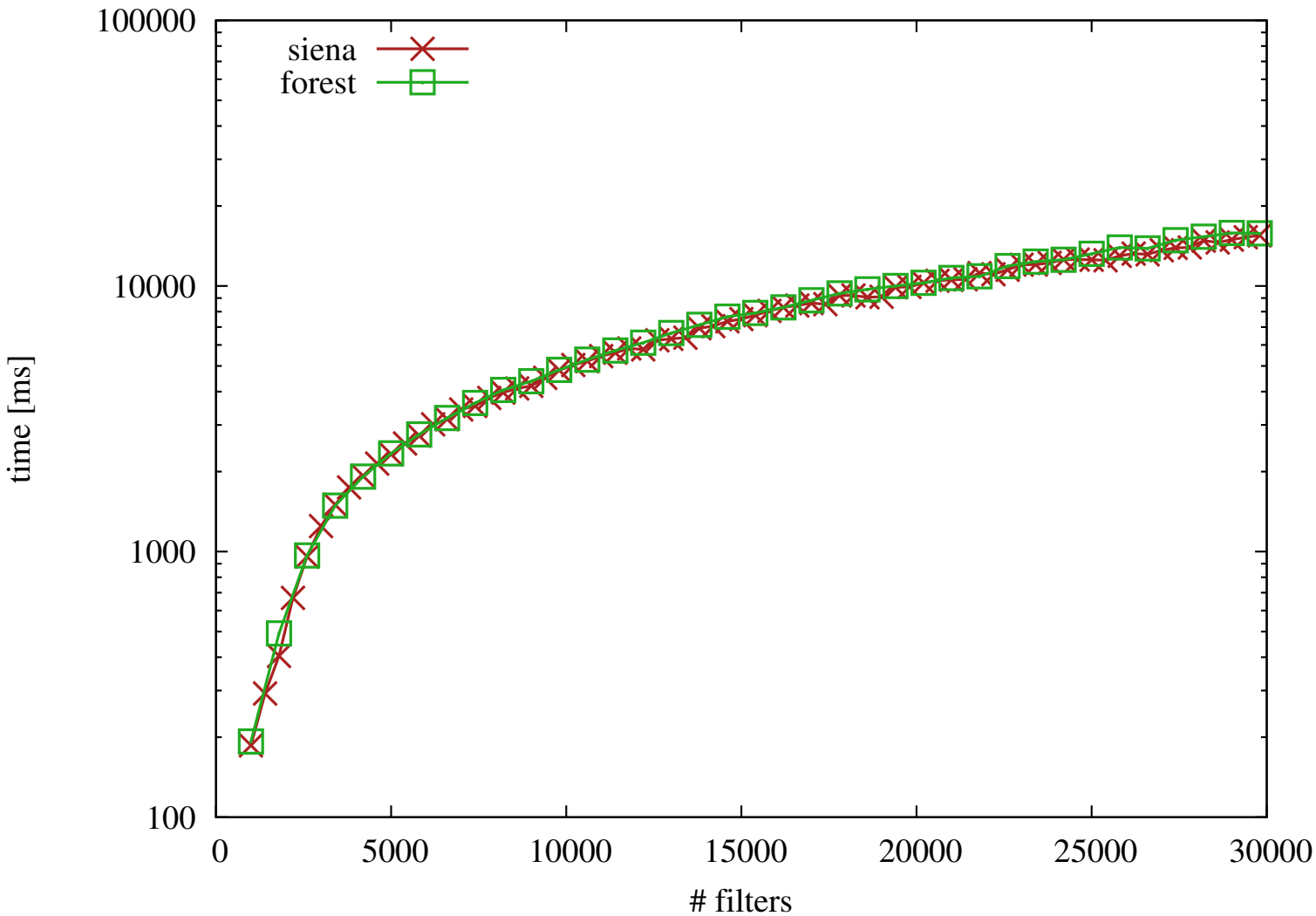
DEBS 2008

Zbigniew Jerzak and Christof Fetzer

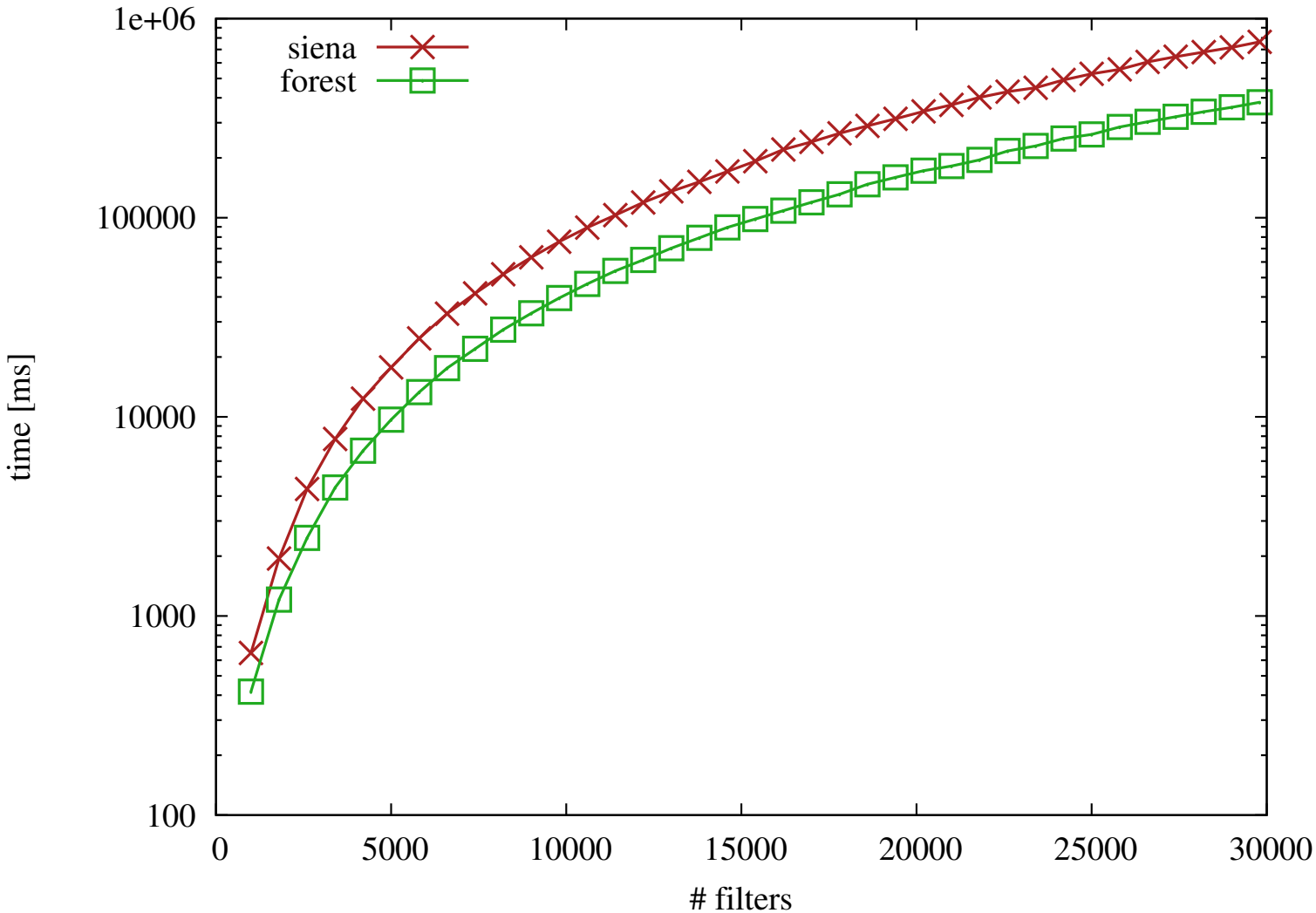
Systems Engineering Group, Dresden University of Technology

Wednesday 2nd July, 2008

Event forwarding ([CRW01]: siena, [TK06]: forest)



Subscription routing ([CRW01]: siena, [TK06]: forest)



Motivation

- Efficient End-to-End transmission delays
- More flexible edge routing approach [JF07]
- A flexible communication substrate for event processing systems

Our Contributions

- New routing structures and algorithms:
  - applicable for both traditional and edge routing
  - `sbsposet` – predicate storage and coverage
  - `sbstree` – management of disjunctions of predicates conjunctions
- Content summarisation using Bloom filters [Blo70]
  - not imposing any limitation on the type and content of the events/subscriptions
  - sparse Bloom filters for space complexity

Our Contributions

- New routing structures and algorithms:
  - applicable for both traditional and edge routing
  - `sbsposet` – predicate storage and coverage
  - `sbstree` – management of disjunctions of predicates conjunctions
- Content summarisation using Bloom filters [Blo70]
  - not imposing any limitation on the type and content of the events/subscriptions
  - sparse Bloom filters for space complexity

Publish/Subscribe Model

- Decoupled [EFGK03] communication...
- ...between publishers and subscribers via routers
- ...using subscriptions (conjunction of predicates)
- ...and events (disjunction of predicates)
- ...based on their content

Publish/Subscribe Model

- Decoupled [EFGK03] communication...
- ...between publishers and subscribers via routers
- ...using subscriptions (conjunction of predicates)
- ...and events (disjunction of predicates)
- ...based on their content

Publish/Subscribe Model

- Decoupled [EFGK03] communication...
- ...between publishers and subscribers via routers
- ...using subscriptions (conjunction of predicates)
  - $\{\underbrace{\text{movie} = \text{"star wars"}}_{p_0()} \text{ AND } \underbrace{\text{price} < 15}_{p_1()}\}$
- ...and events (disjunction of predicates)
- ...based on their content

Publish/Subscribe Model

- Decoupled [EFGK03] communication...
- ...between publishers and subscribers via routers
- ...using subscriptions (conjunction of predicates)
- ...and events (disjunction of predicates)
  - $\left\{ \underbrace{\text{movie}}_{\text{an}_0} = \underbrace{\text{"batman"}}_{\text{av}_0} \text{ OR } \underbrace{\text{price}}_{\text{an}_1} = \underbrace{15.5}_{\text{av}_1} \right\}$
- ...based on their content

Publish/Subscribe Model

- Decoupled [EFGK03] communication...
- ...between publishers and subscribers via routers
- ...using subscriptions (conjunction of predicates)
- ...and events (disjunction of predicates)
- ...based on their content

Event forwarding

An event e matches a subscription s ($e \precsim s$)

$$\forall_{p \in s} \quad \exists_{\{an,av\} \in e} : \quad p(\{an, av\}) = \text{true} \tag{1}$$

A broker contains $|\mathbb{S}|$ subscriptions:

$$\forall_{s \in \mathbb{S}} : \quad \text{execute Equation 1} \tag{2}$$

Approach

An event e matches a subscription s ($e \precsim s$)

$$\forall_{p \in s} \quad \exists_{\{an,av\} \in e} : \quad p(\{an, av\}) = \text{true}$$

Evaluate predicate functions

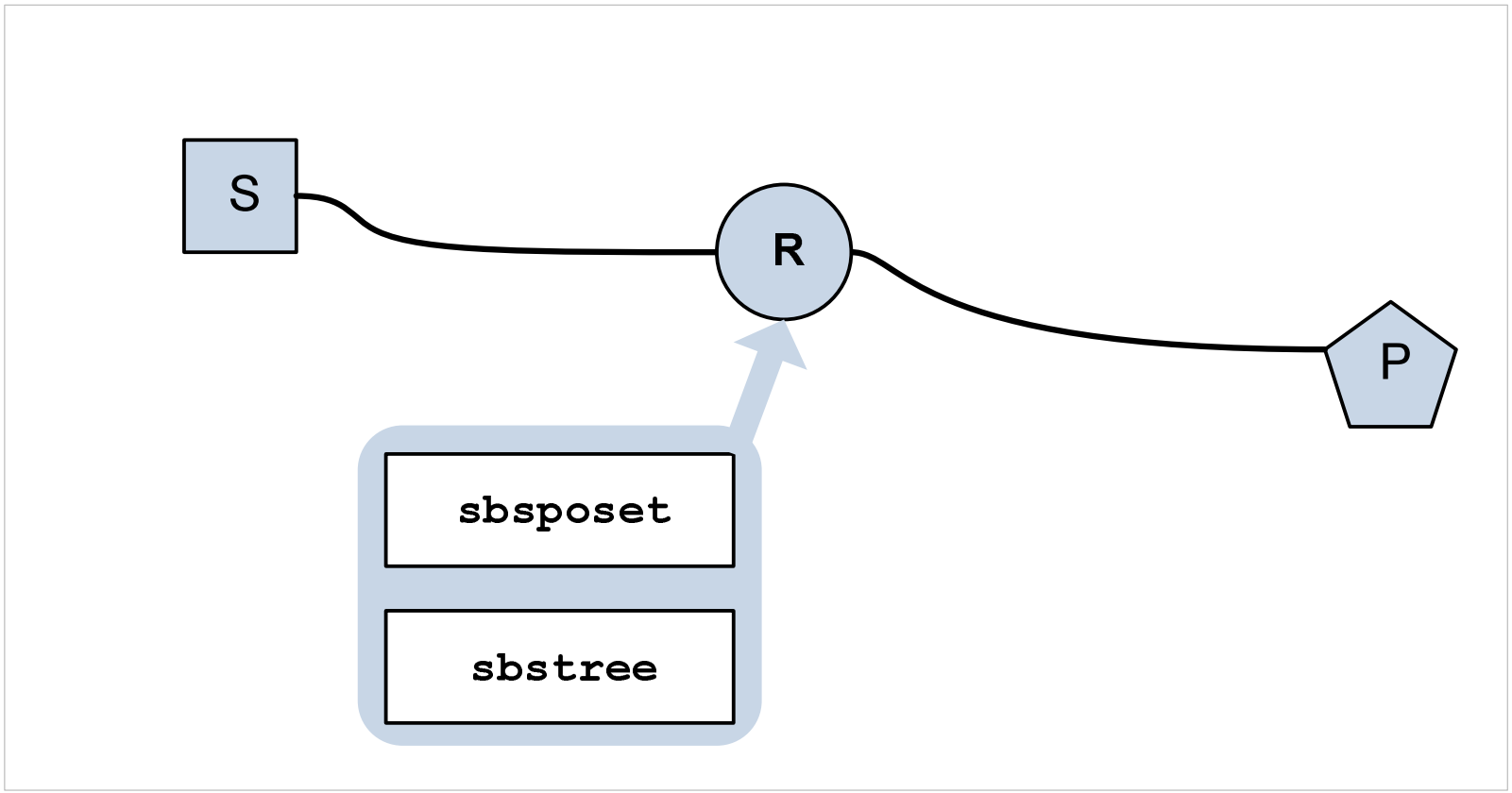
Approach

An event e matches a subscription s ($e \precsim s$)

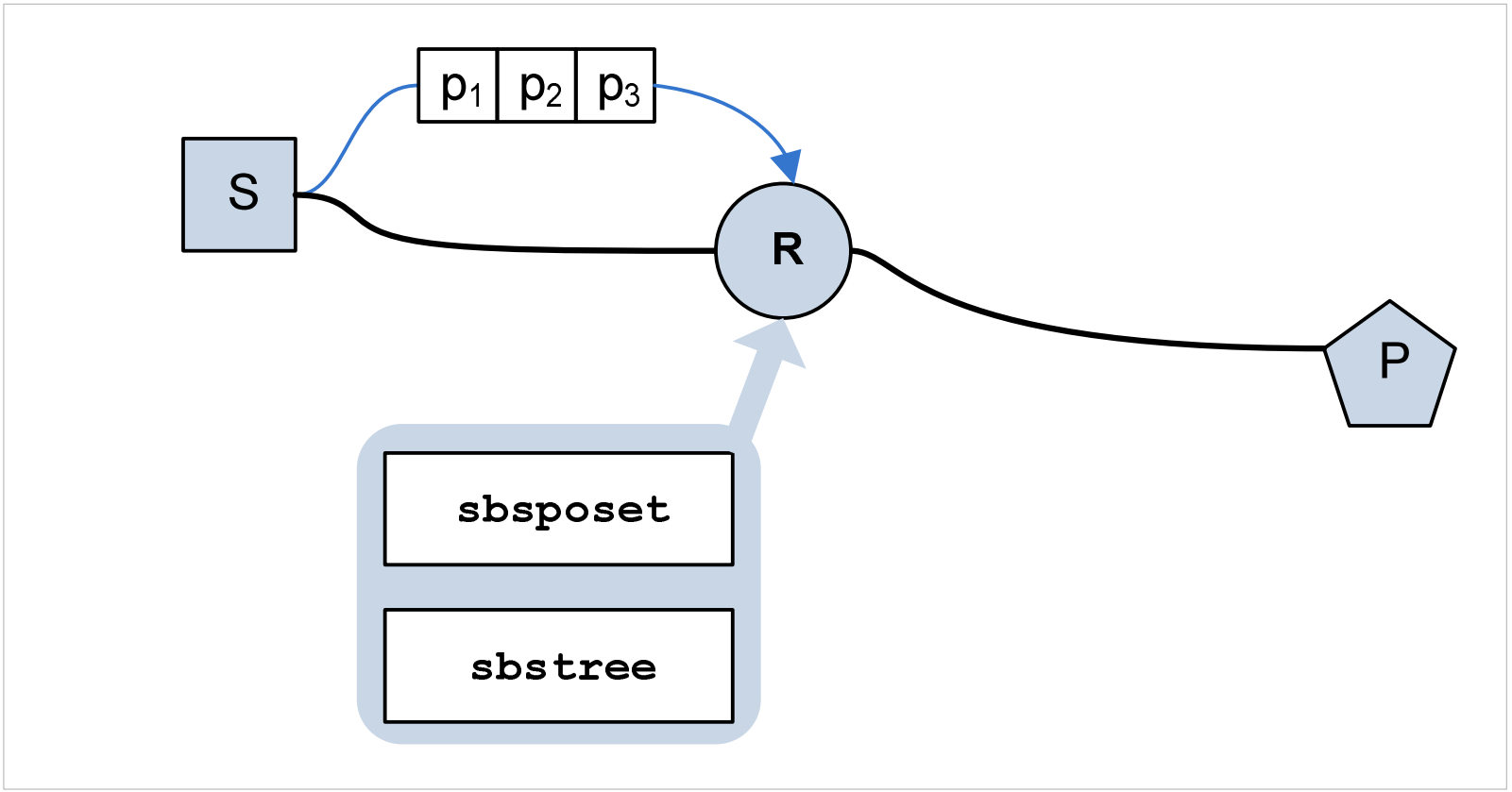
$$\forall_{p \in s} \quad \exists_{\{an,av\} \in e} : \quad p(\{an, av\}) = \text{true}$$

Calculate of disjunction of conjunctions of predicates

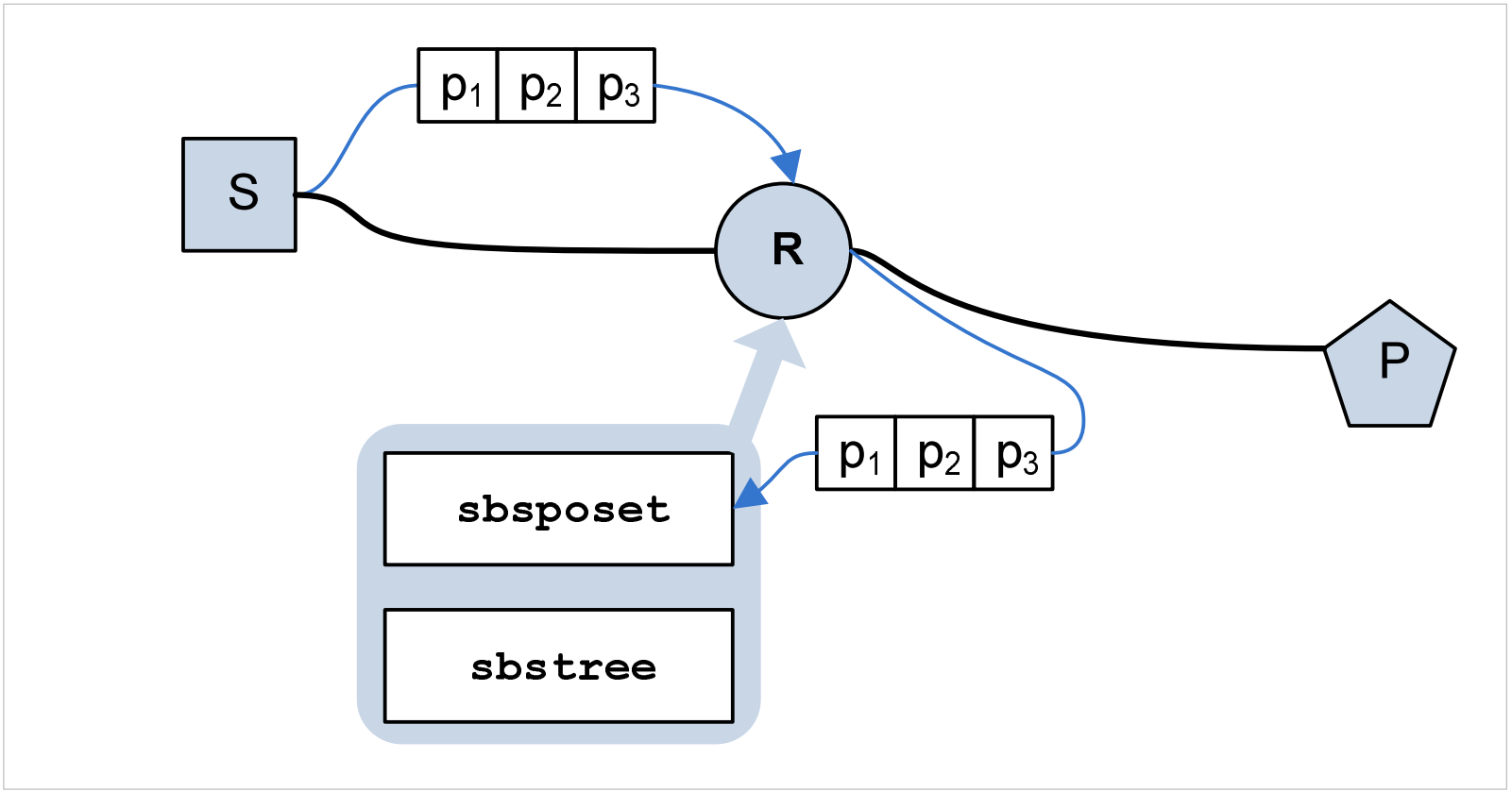
Overview: Subscription Routing



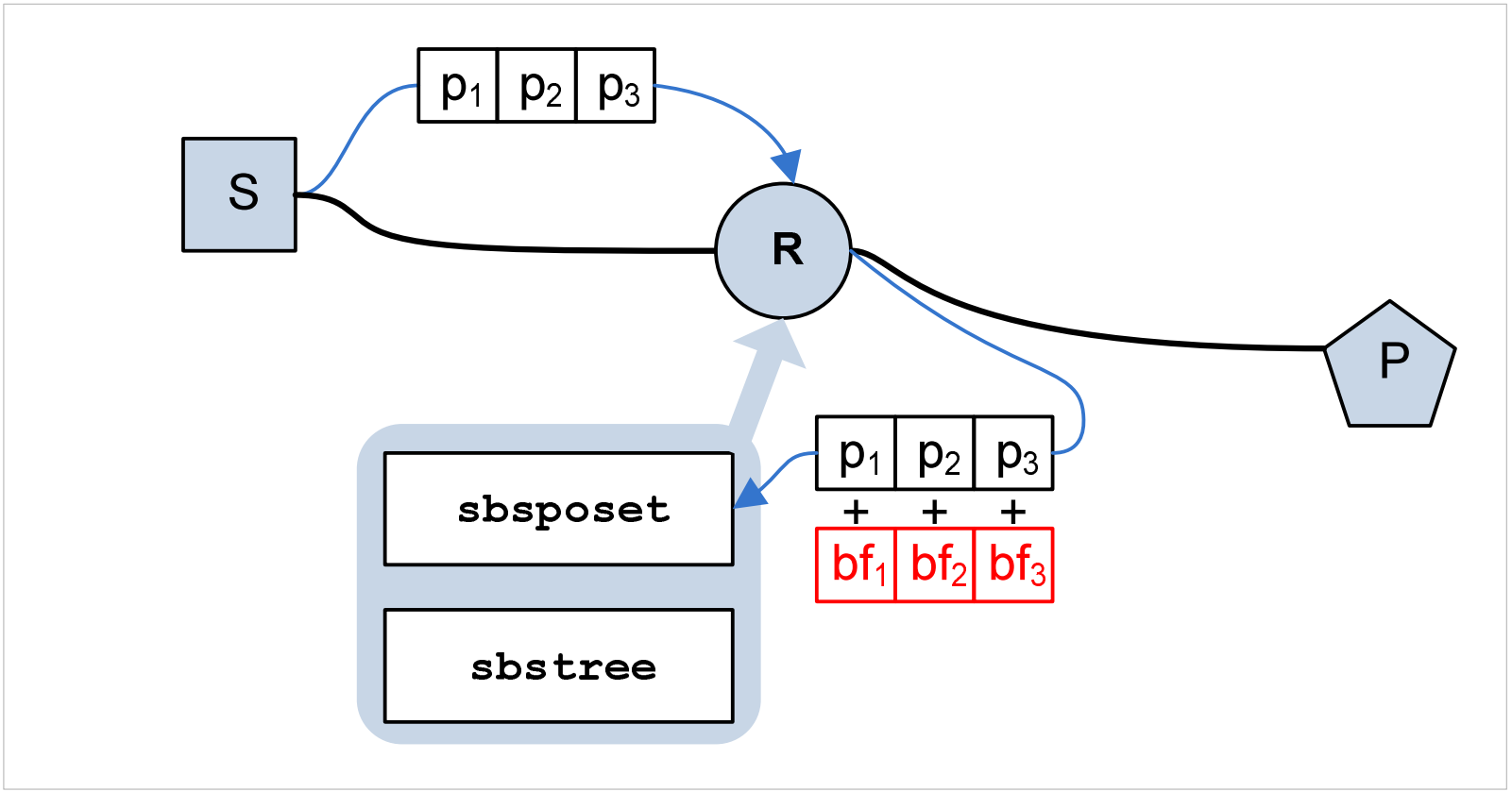
Overview: Subscription Routing



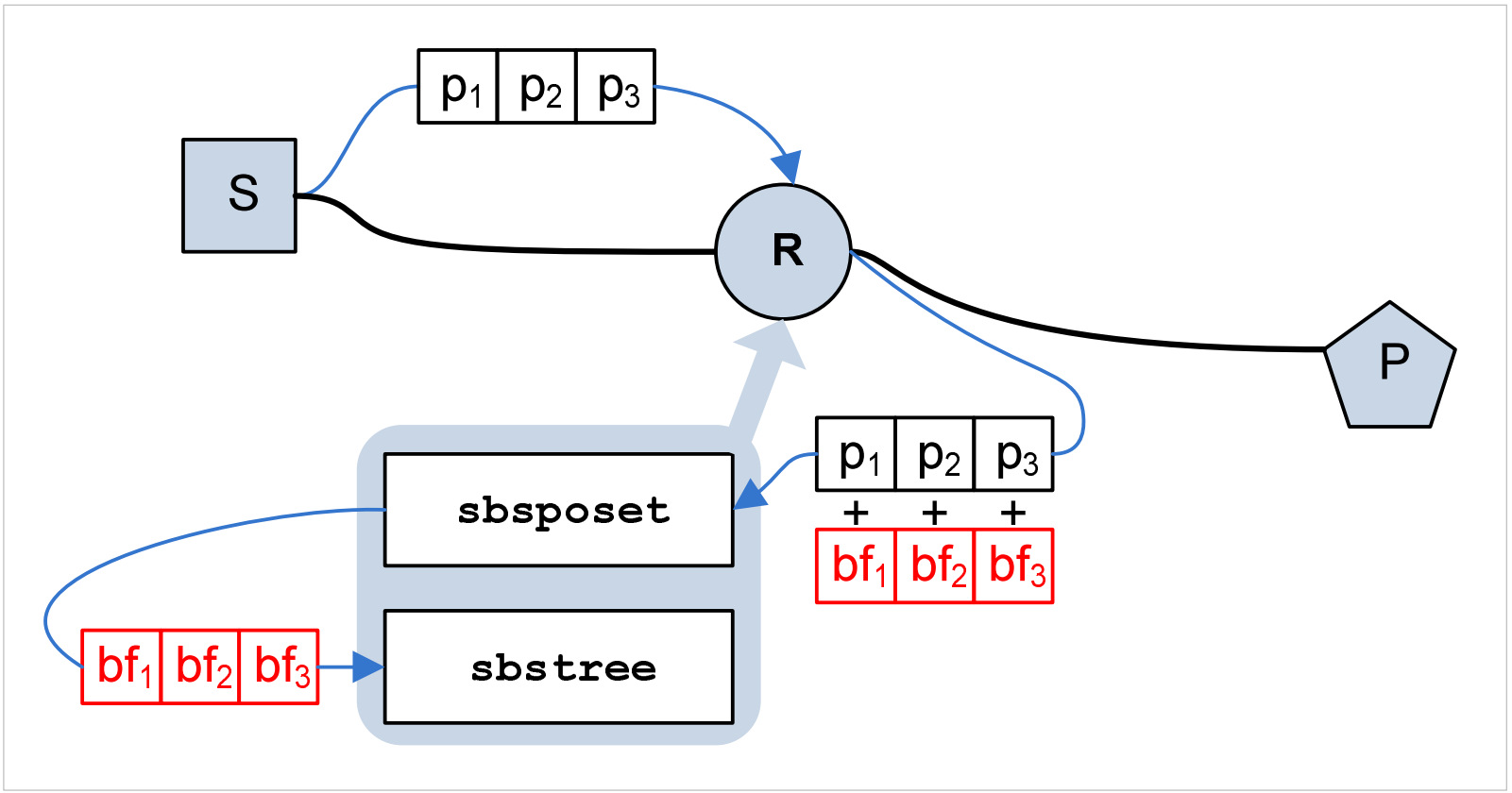
Overview: Subscription Routing



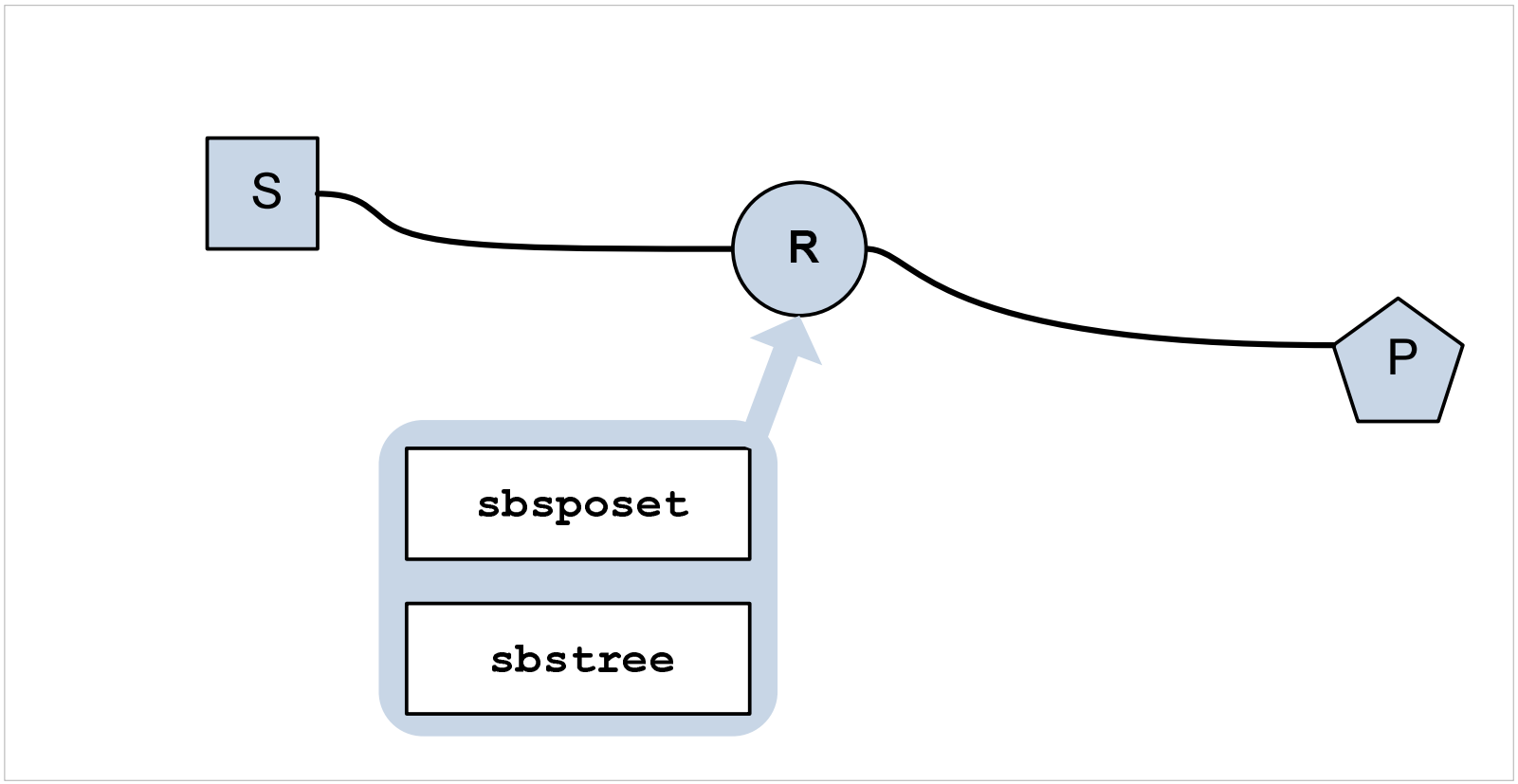
Overview: Subscription Routing



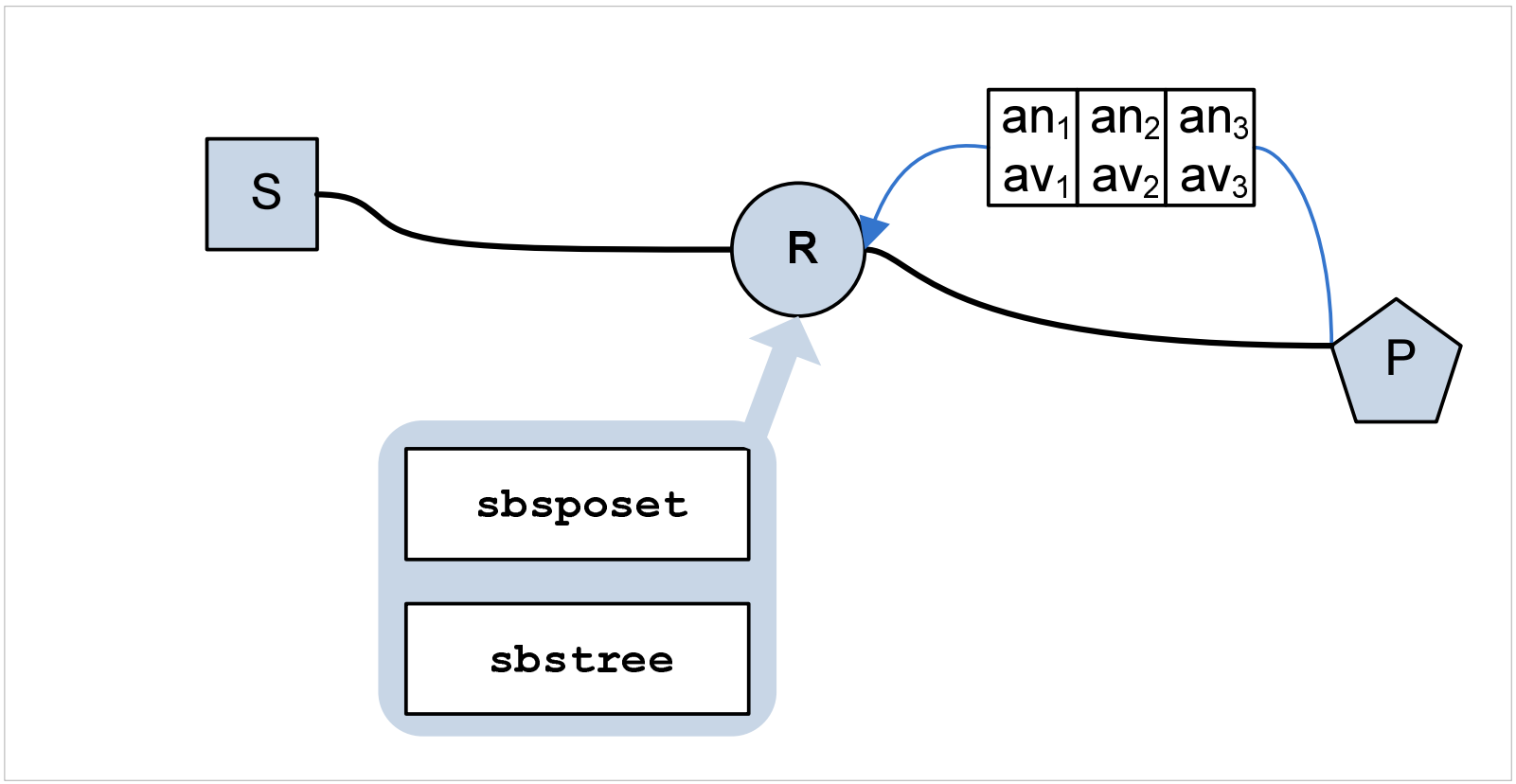
Overview: Subscription Routing



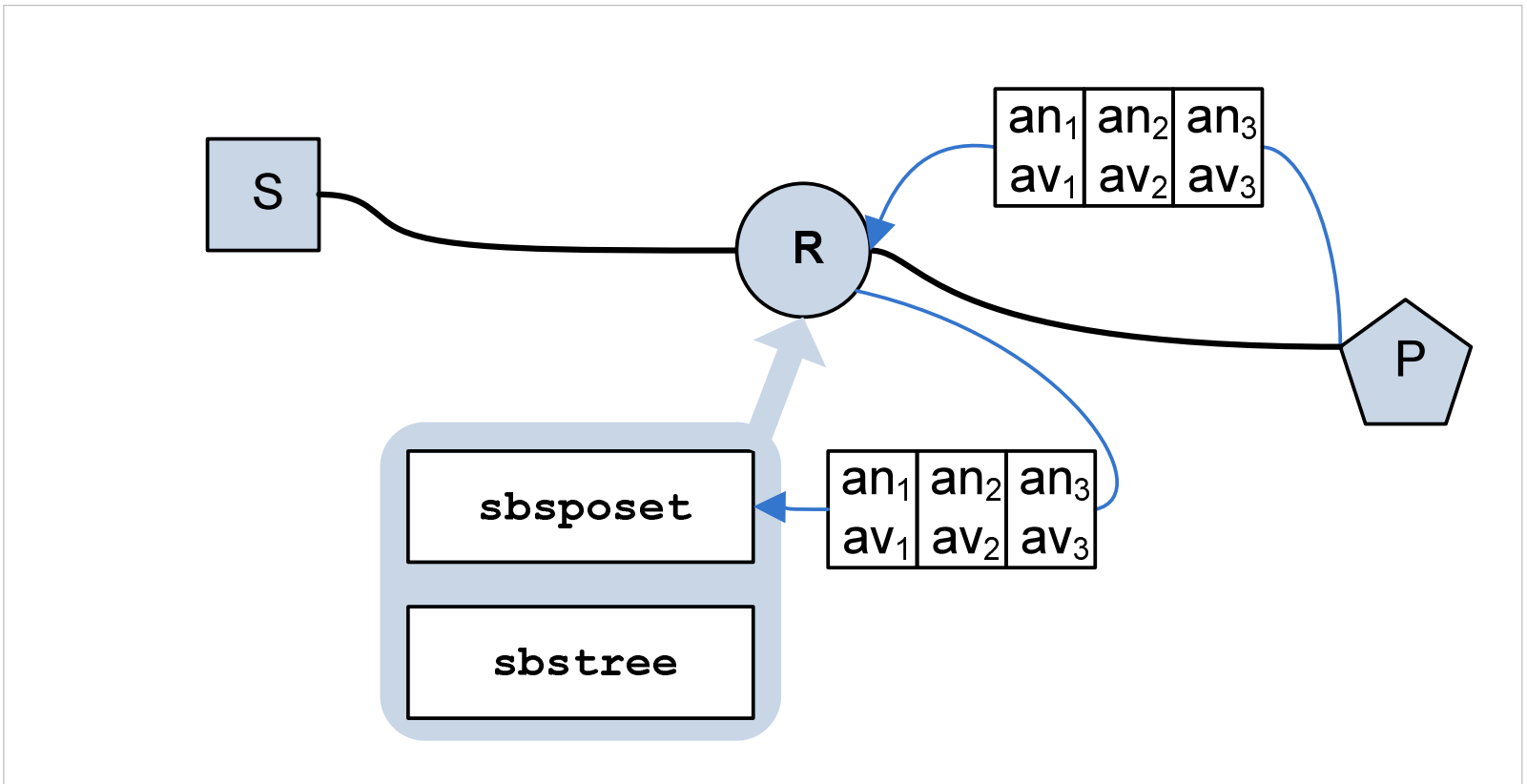
Overview: Event Forwarding



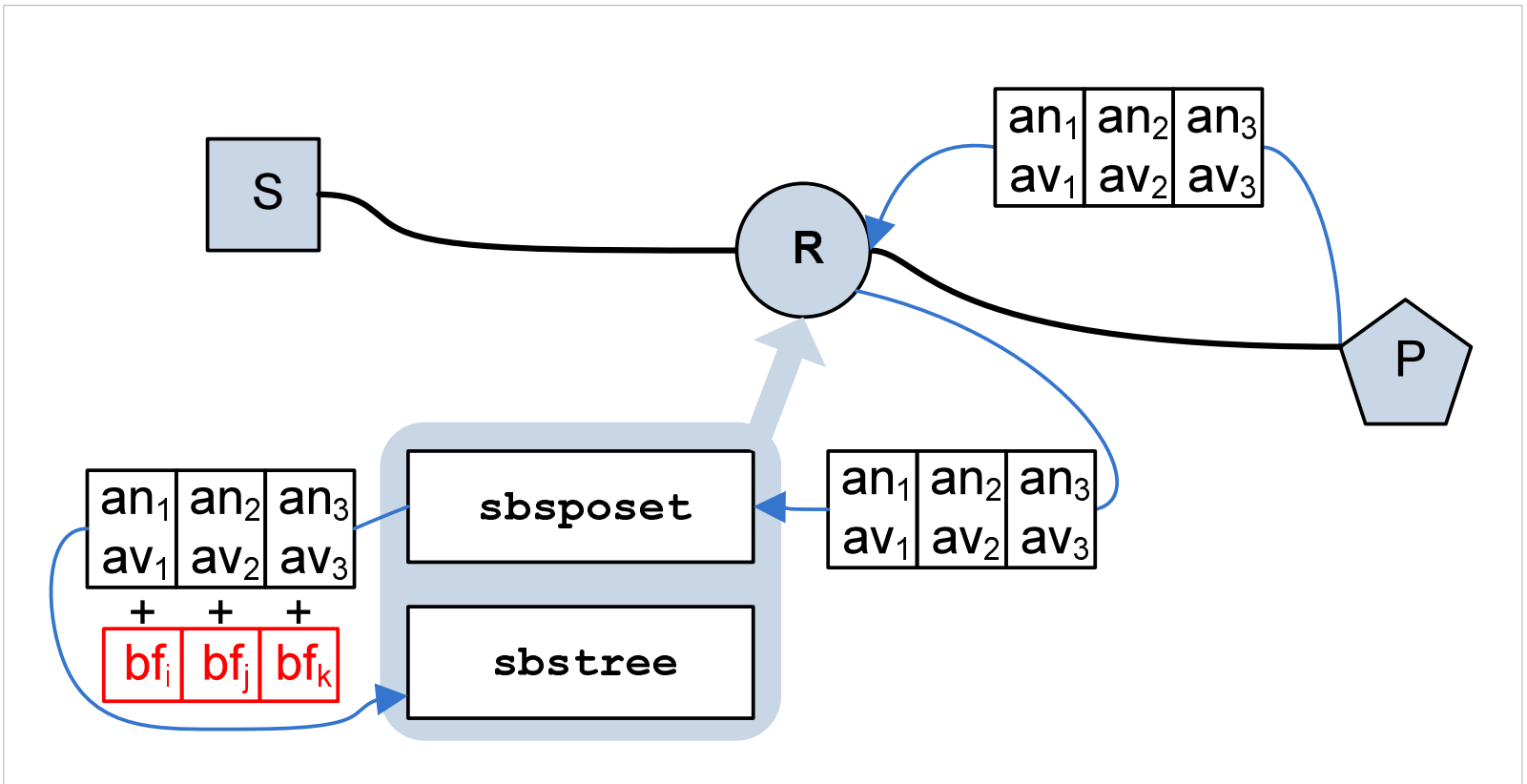
Overview: Event Forwarding



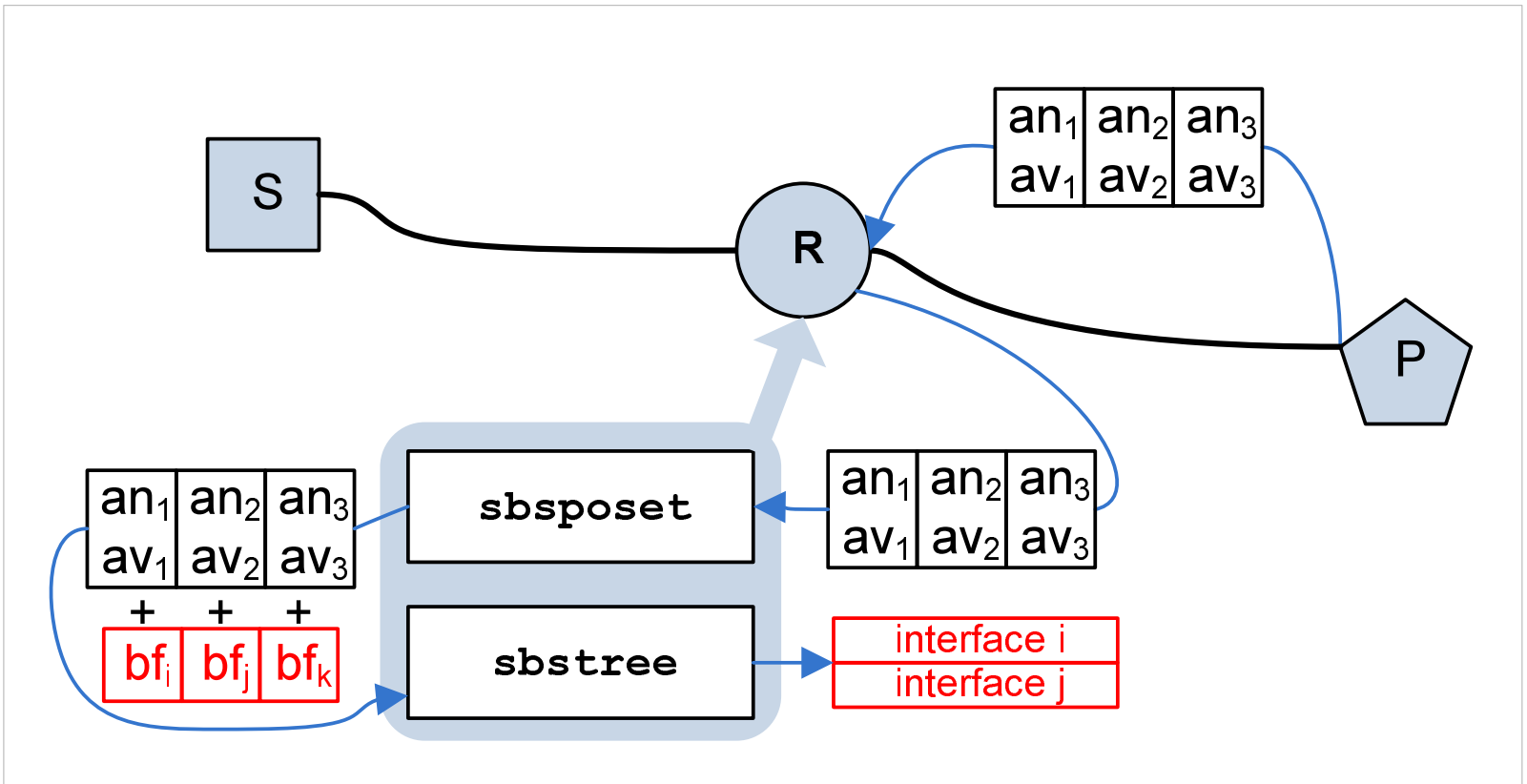
Overview: Event Forwarding



Overview: Event Forwarding



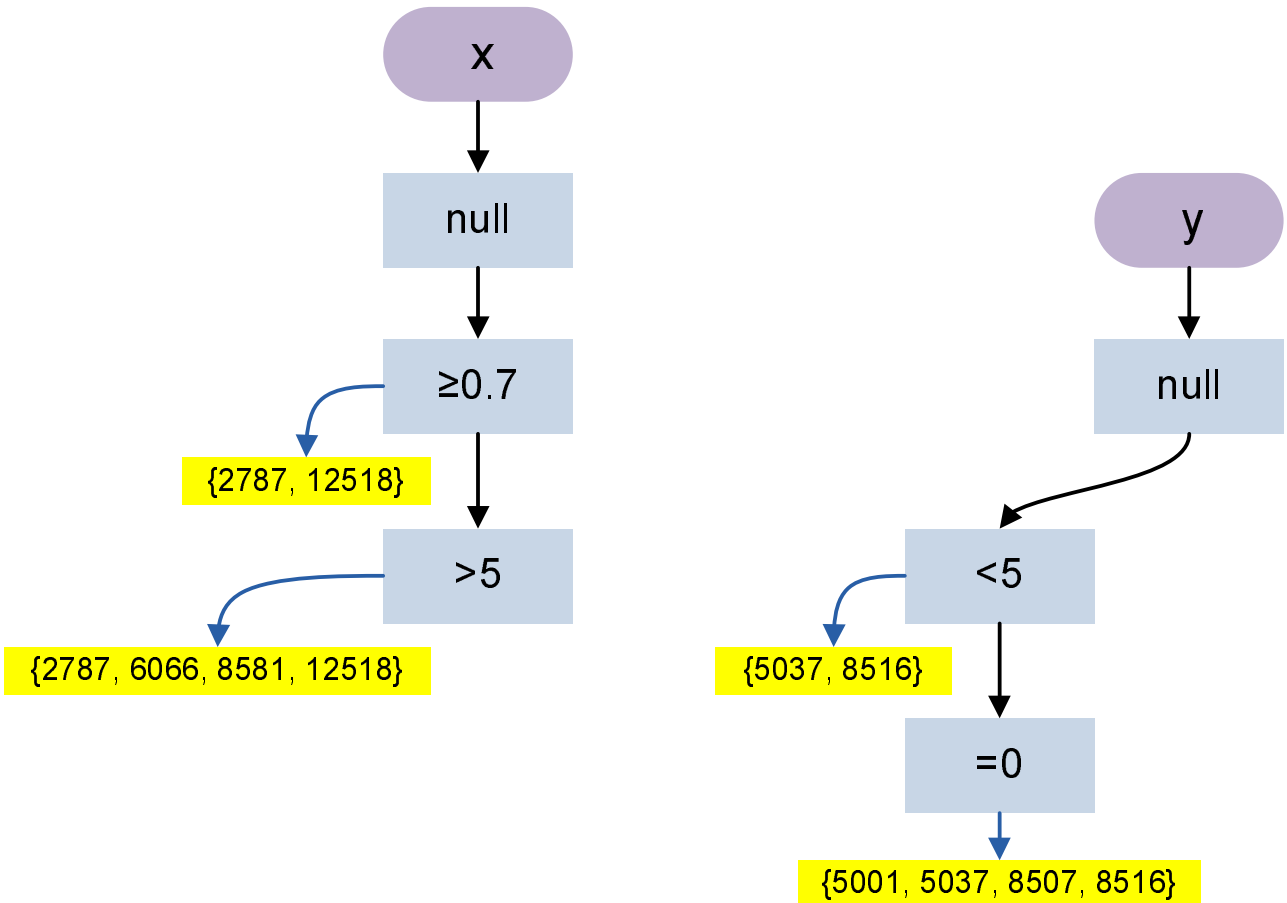
Overview: Event Forwarding



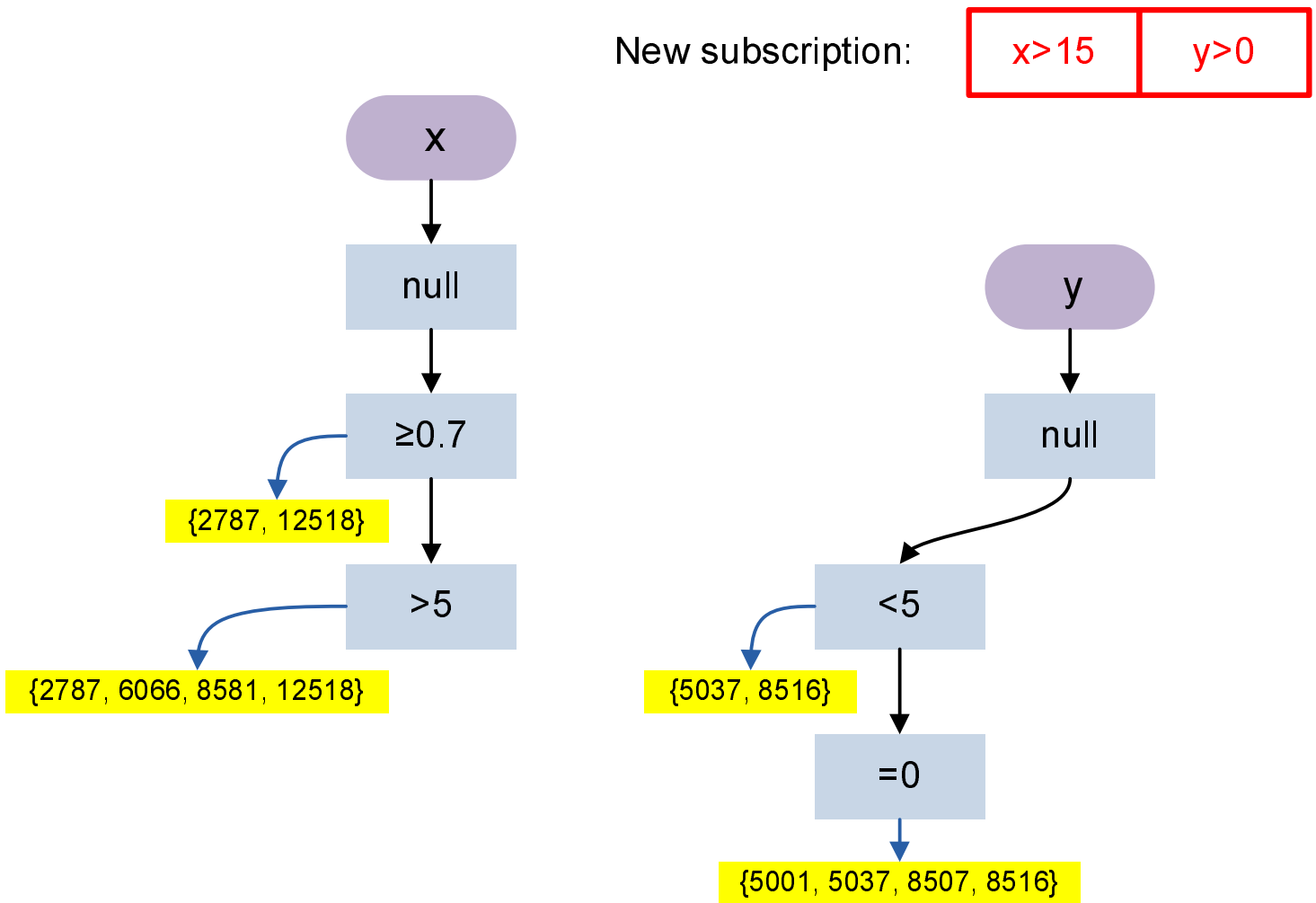
SBSPoset

- Stores subscriptions as:
  - disjunction of single predicates
  - ordered by the coverage relation
- Every predicate is assigned a Bloom filter
  - which summarizes its content
  - and that of covering predicates
- No limitations on the expressiveness of the subscription/event language

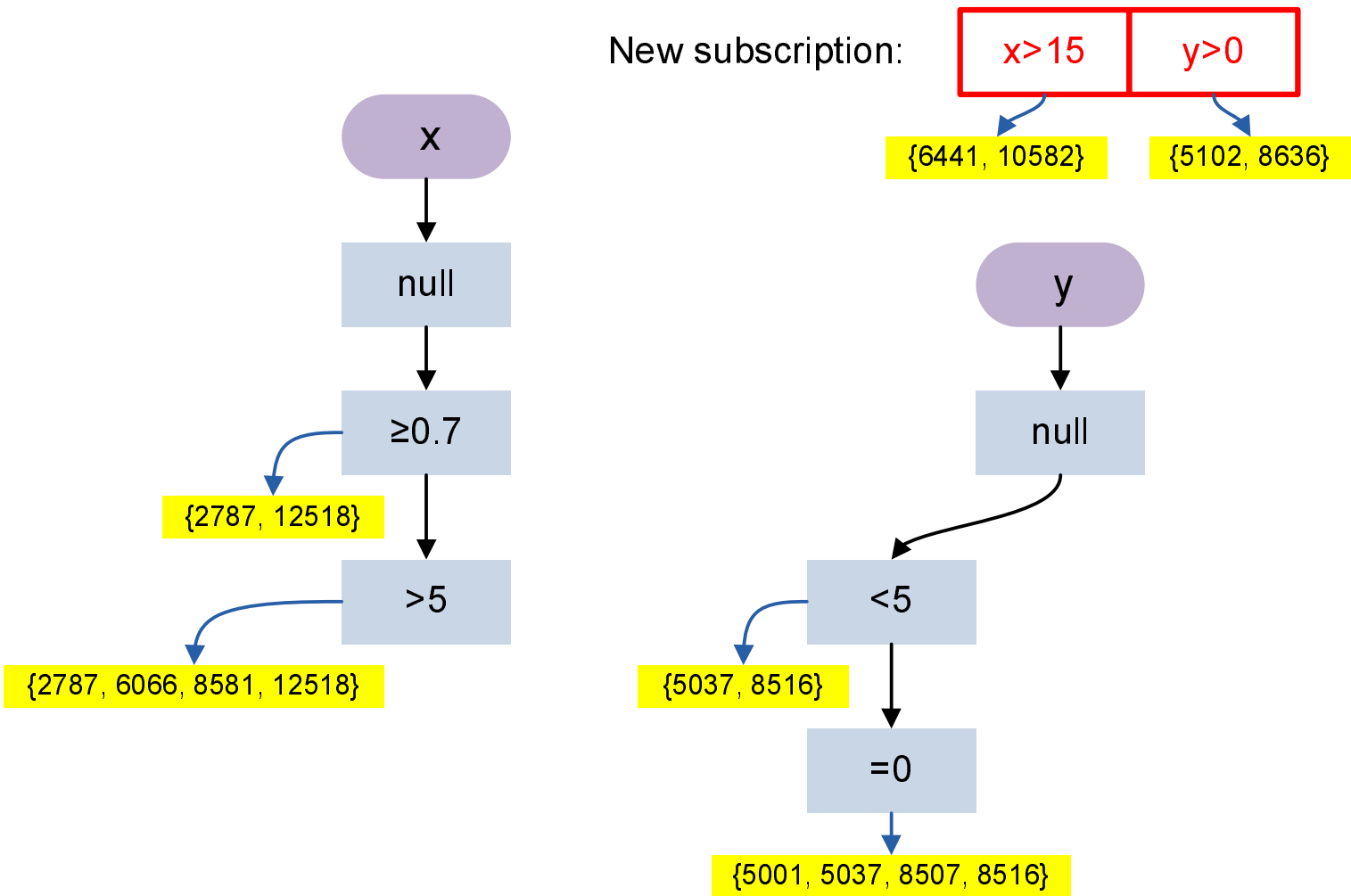
SBSPoset: subscription storage



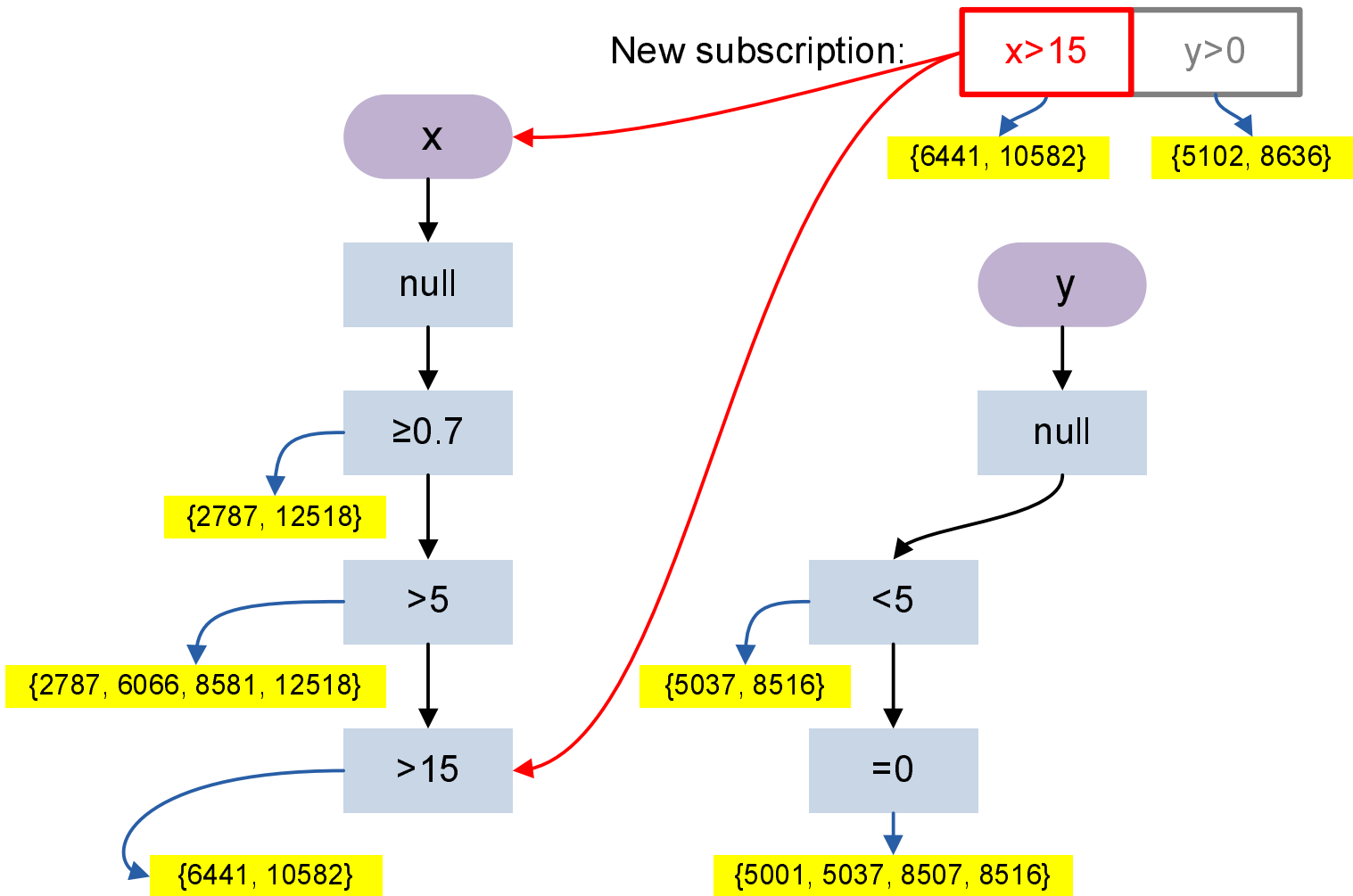
SBSPoset: subscription storage



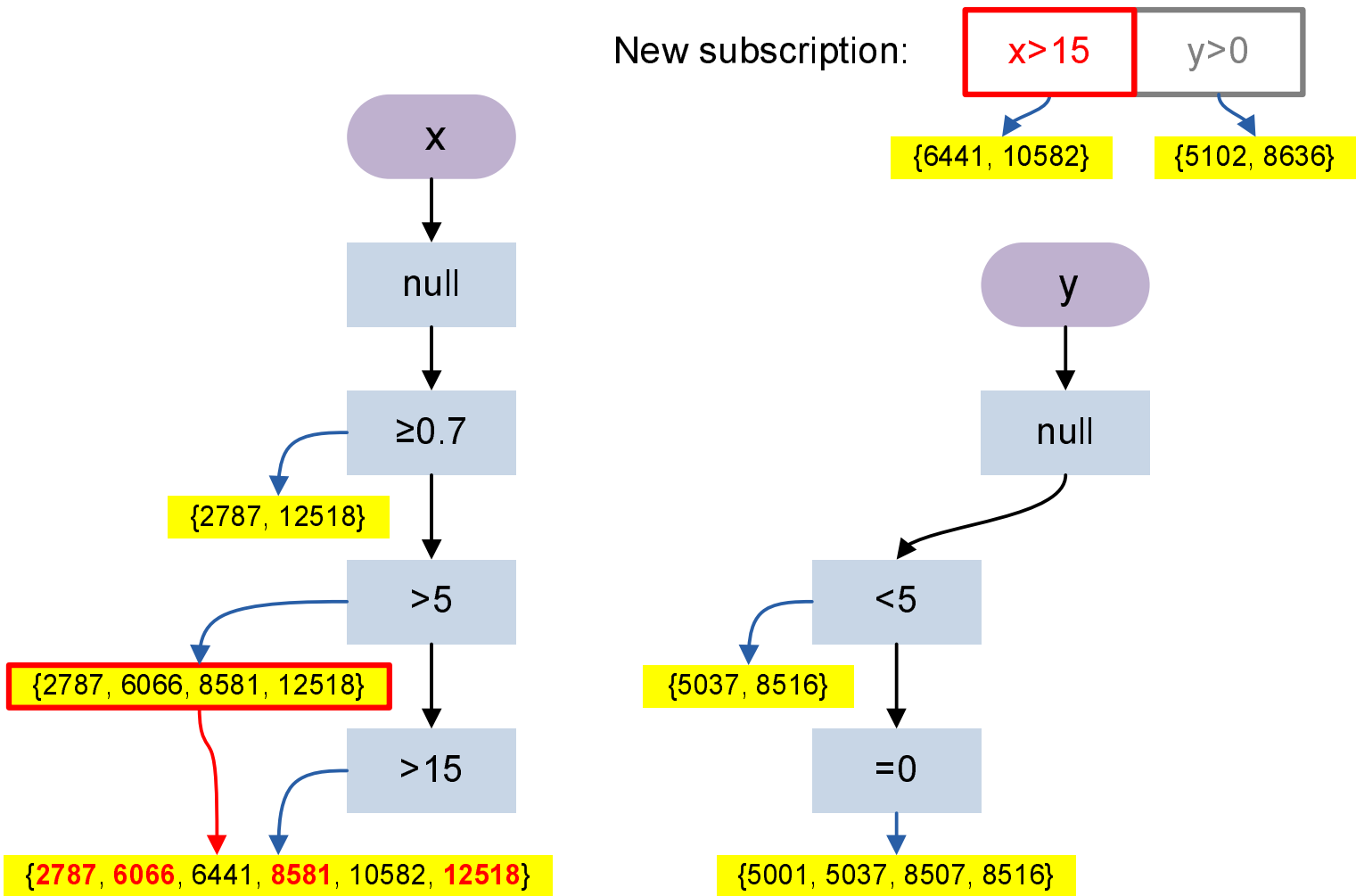
SBSPoset: subscription storage



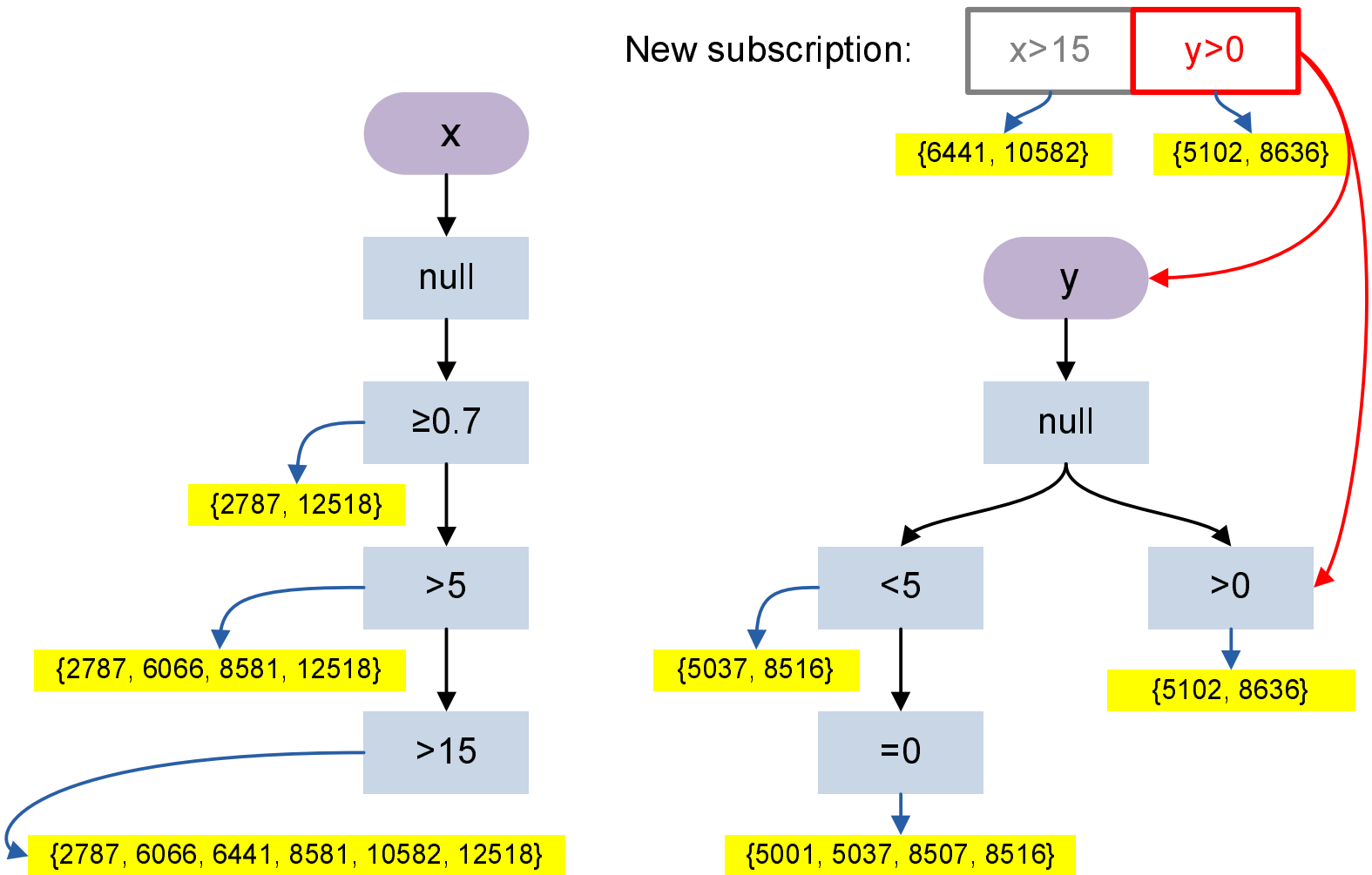
SBSPoset: subscription storage



SBSPoset: subscription storage



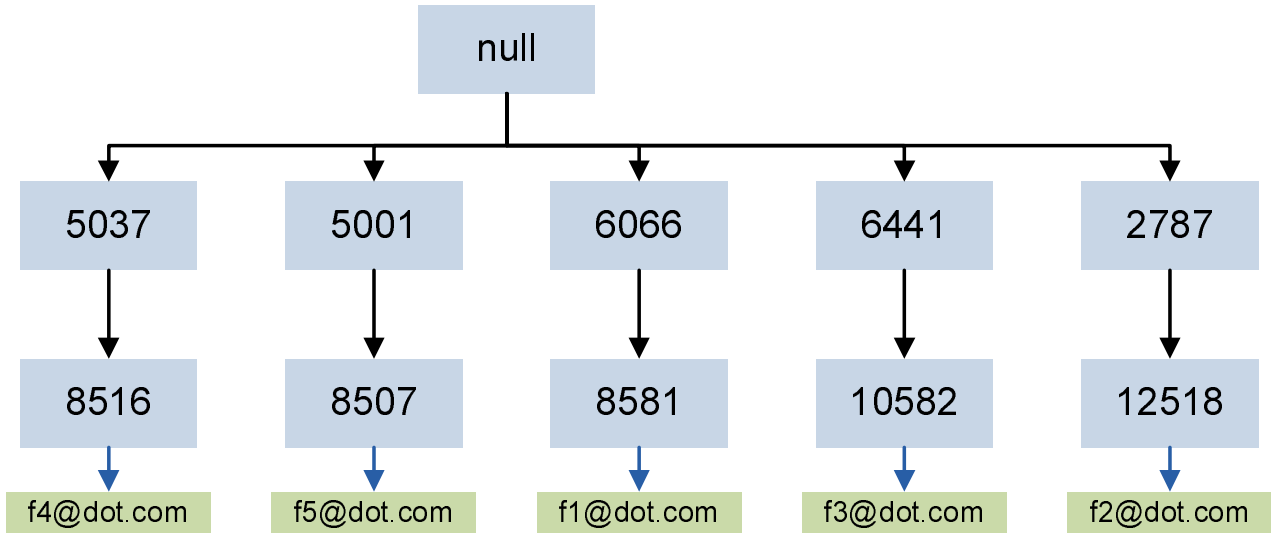
SBSPoset: subscription storage

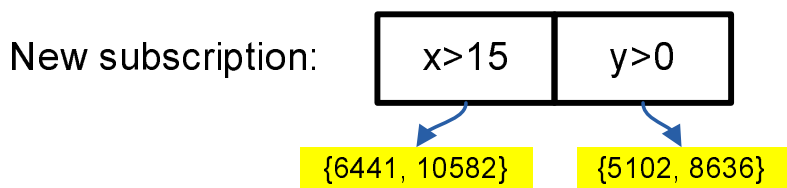


SBSTree

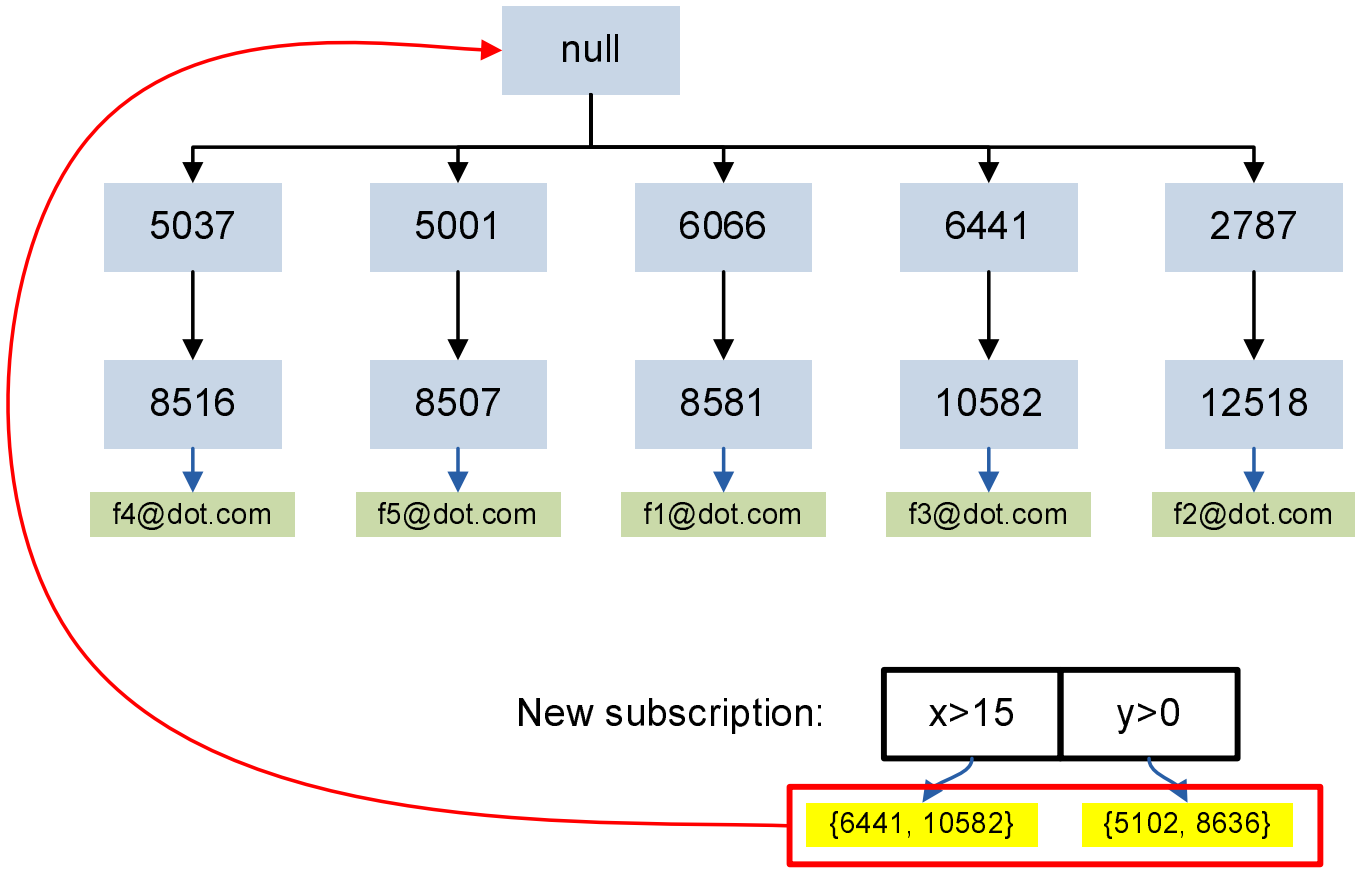
- Represents the disjunction of conjunctions of subscriptions' predicates:
  - stores only Bloom filters

SBSTree: subscription storage

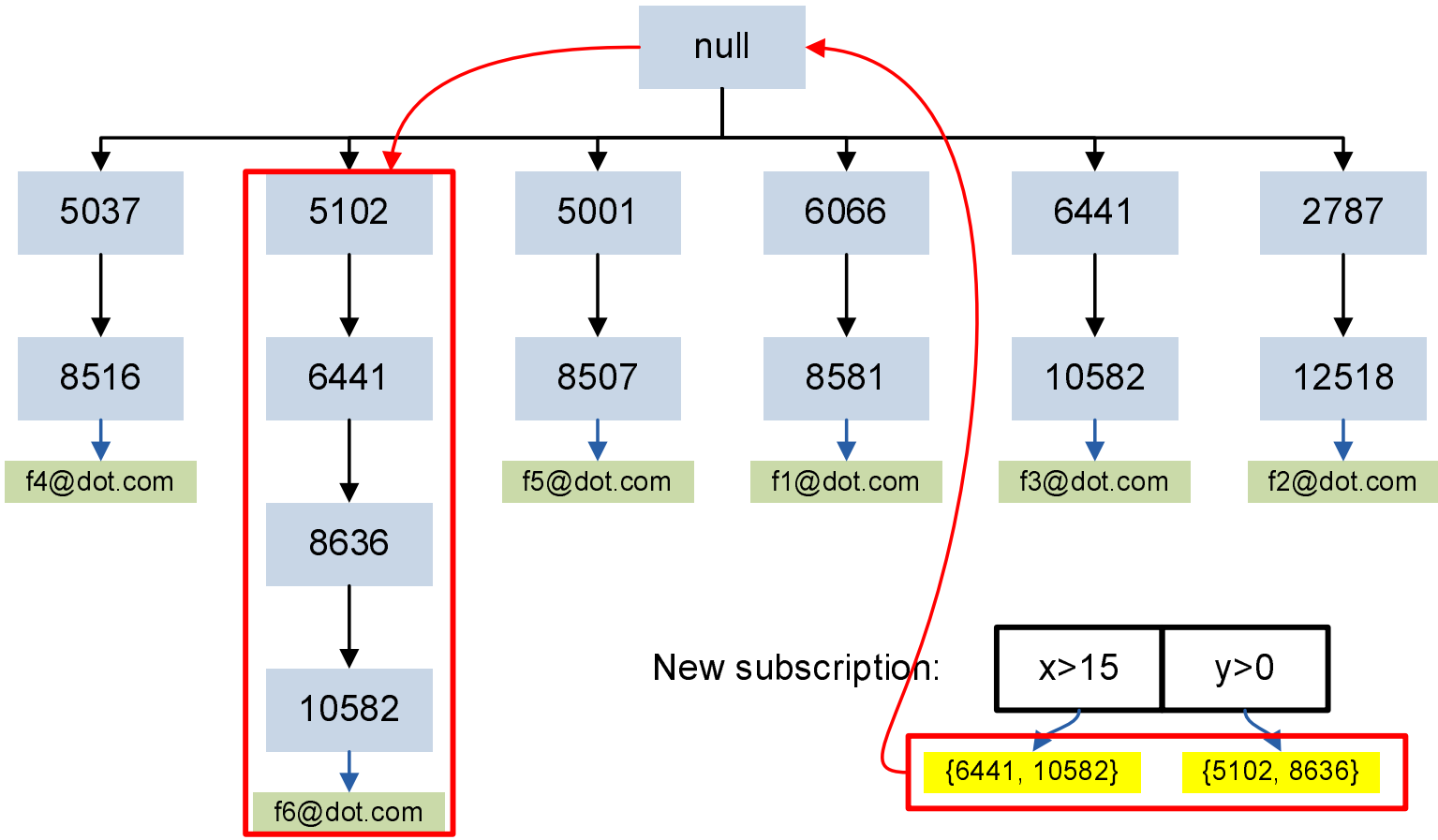




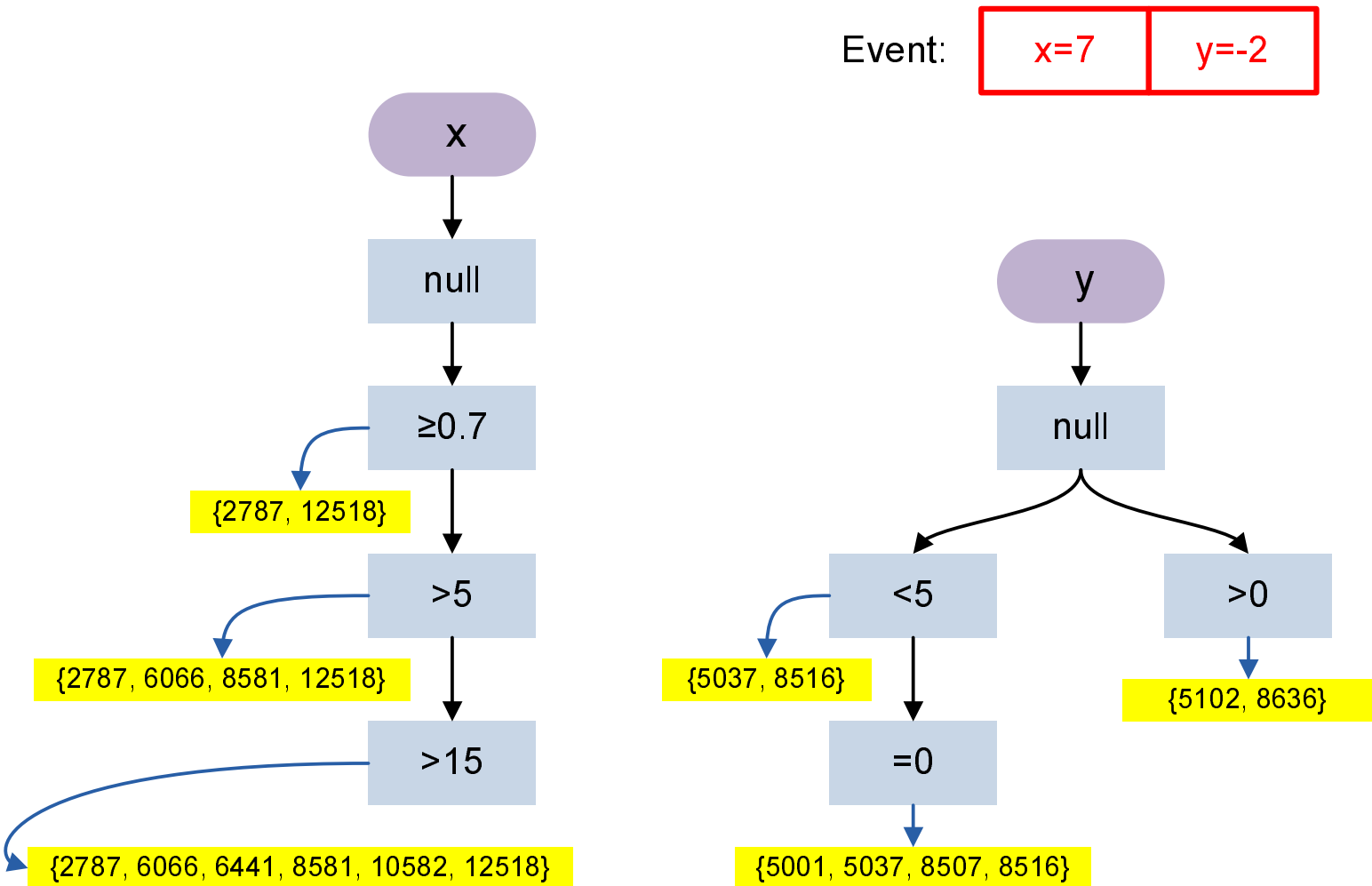
SBSTree: subscription storage



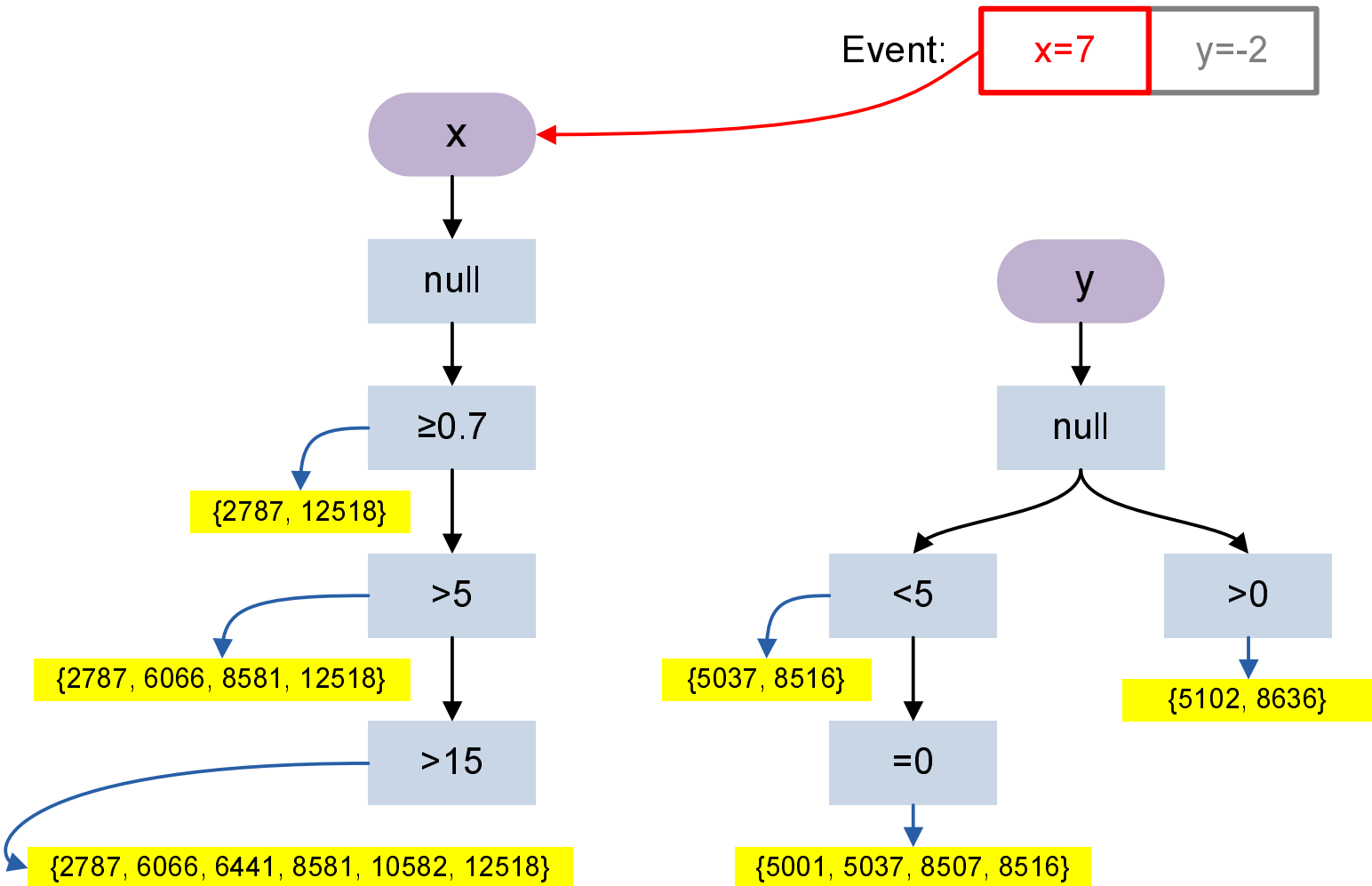
SBSTree: subscription storage



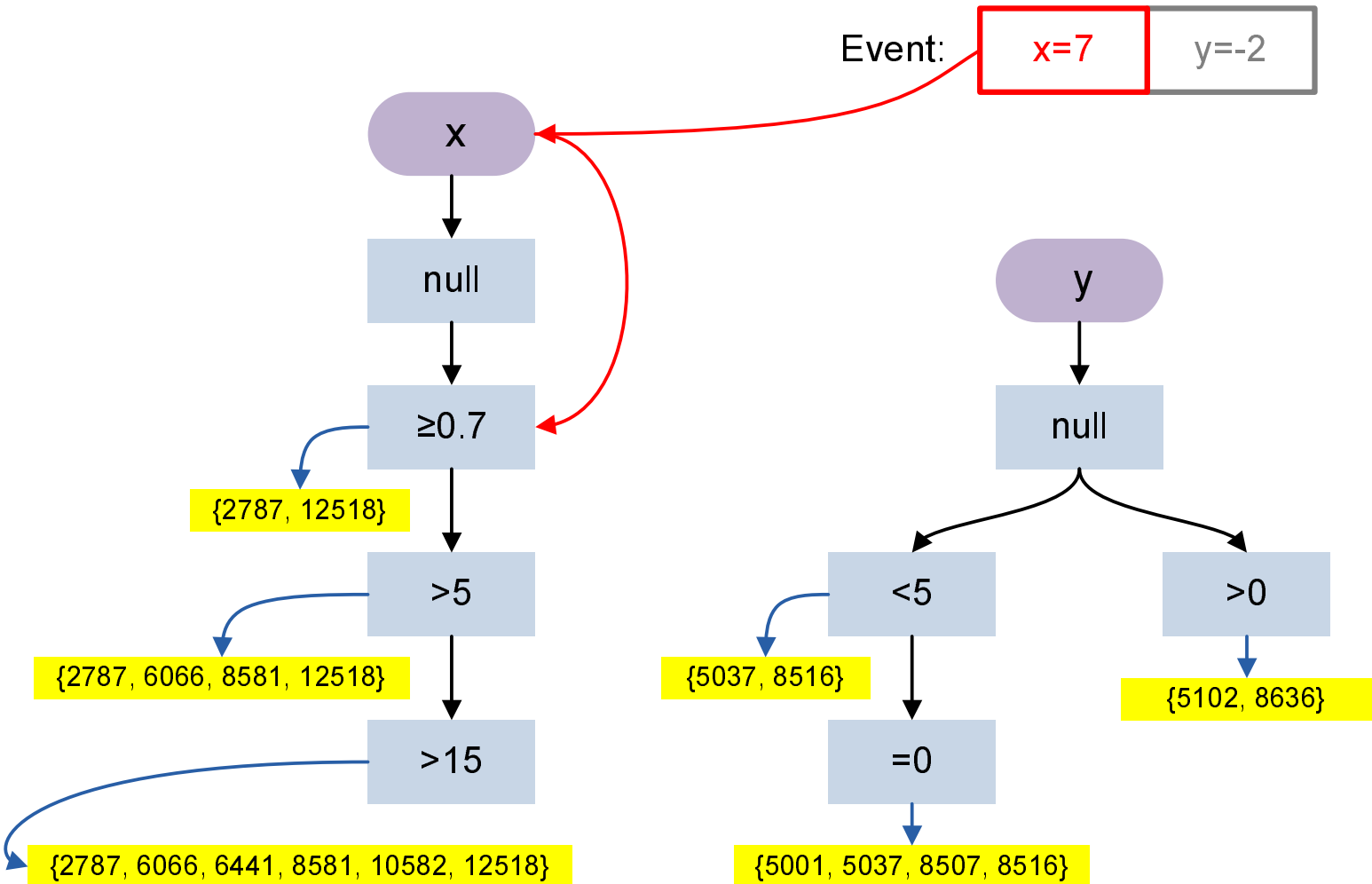
SBSPoset: event forwarding



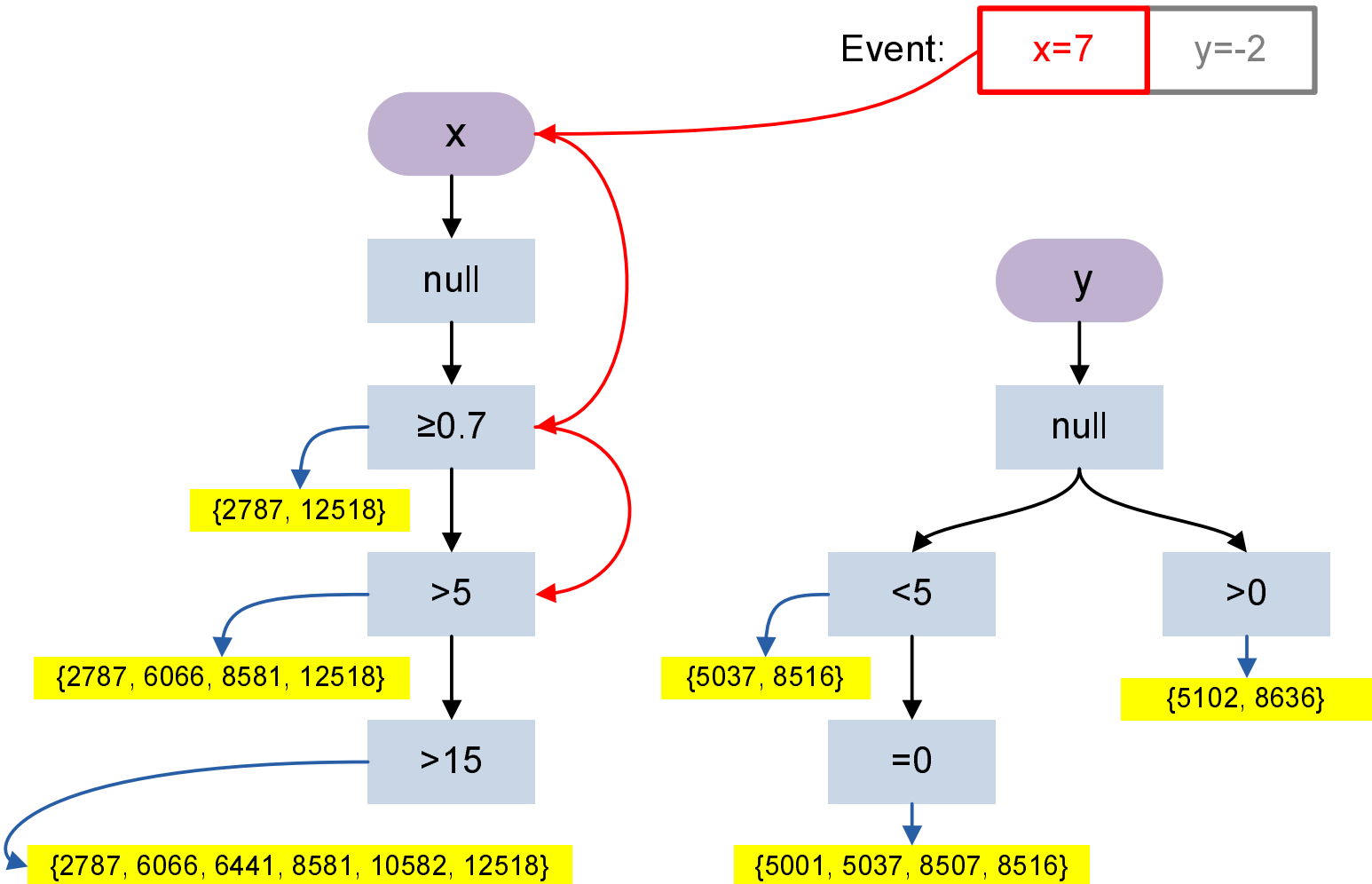
SBSPoset: event forwarding



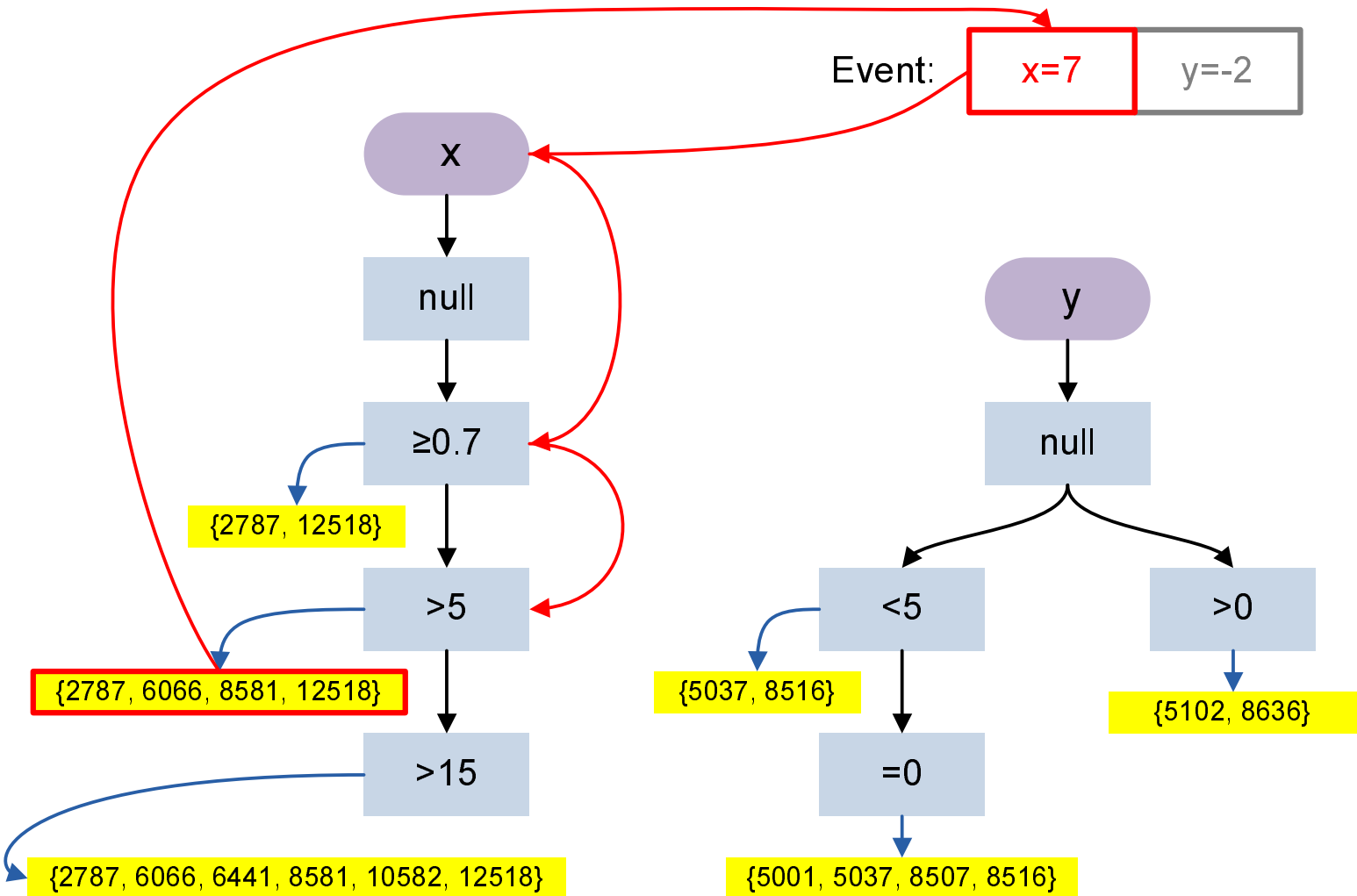
SBSPoset: event forwarding



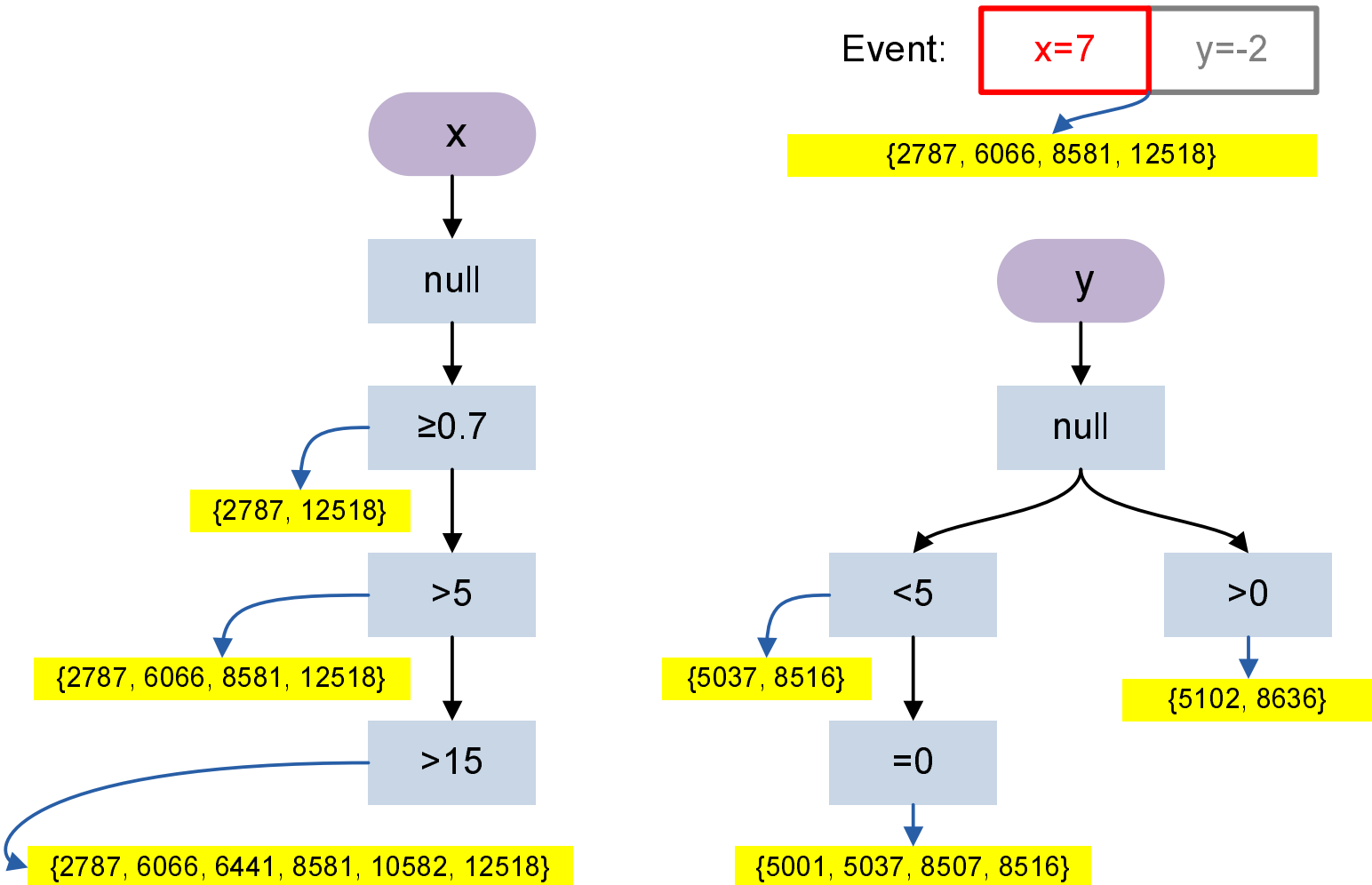
SBSPoset: event forwarding



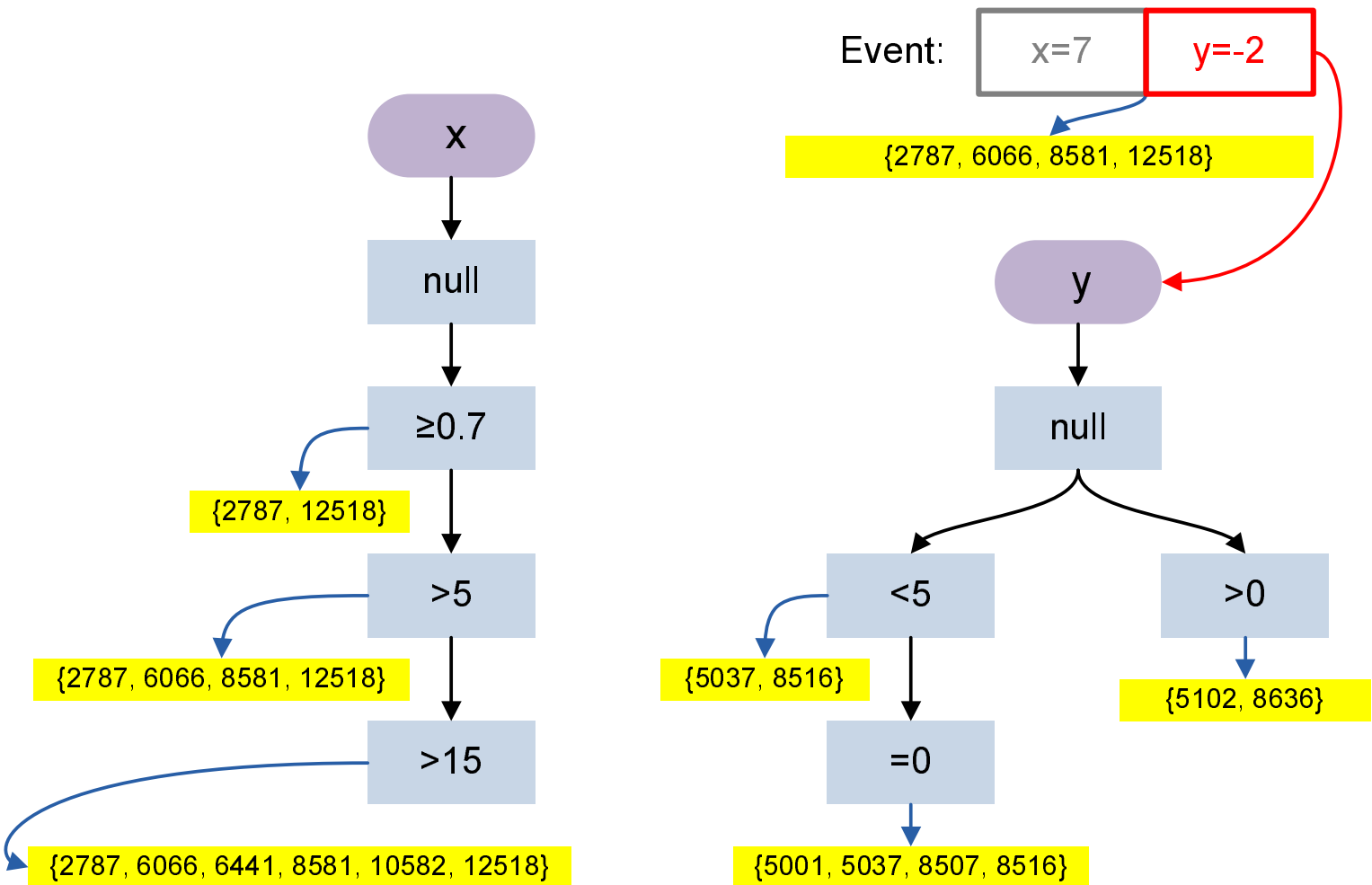
SBSPoset: event forwarding



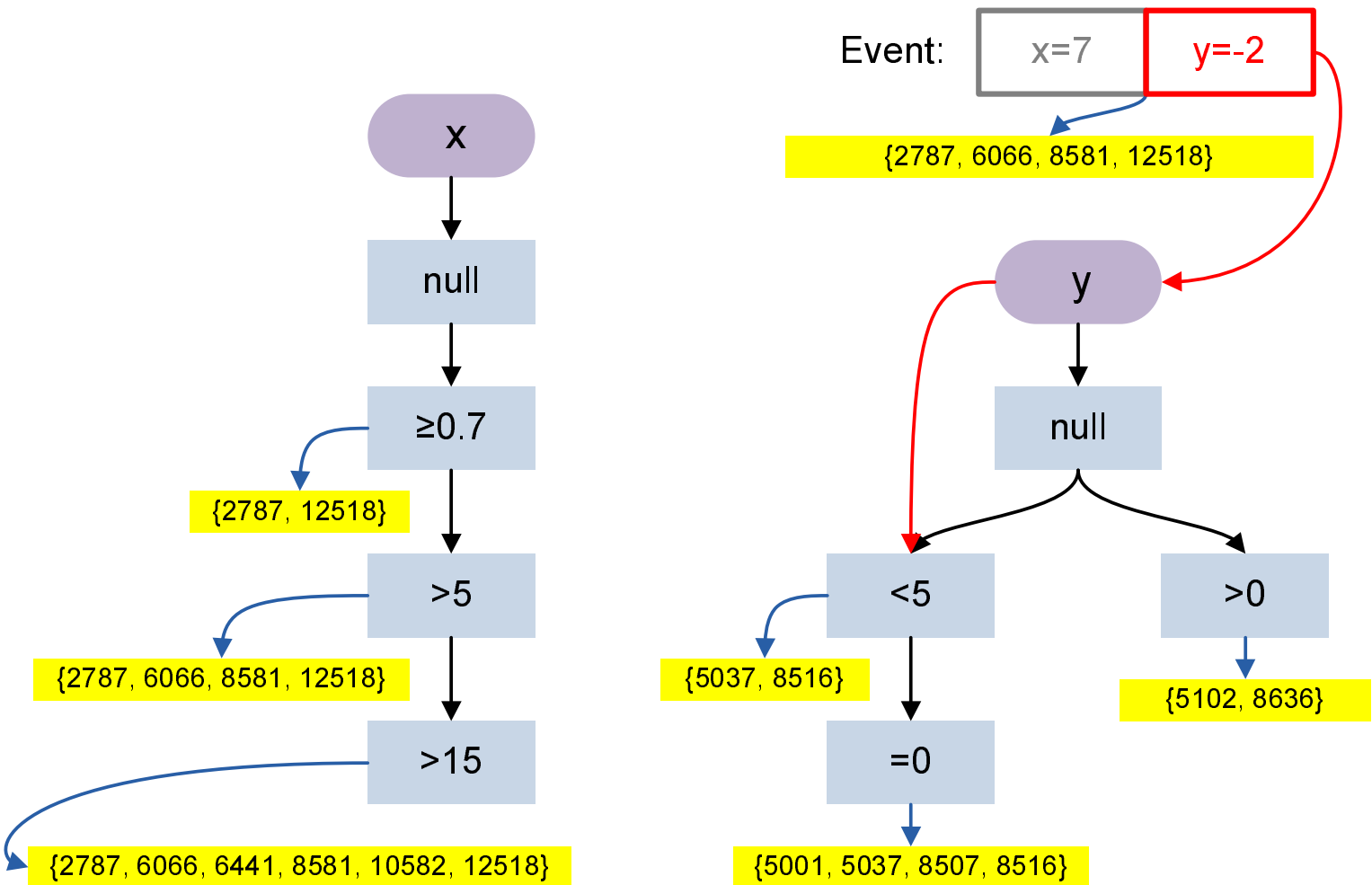
SBSPoset: event forwarding



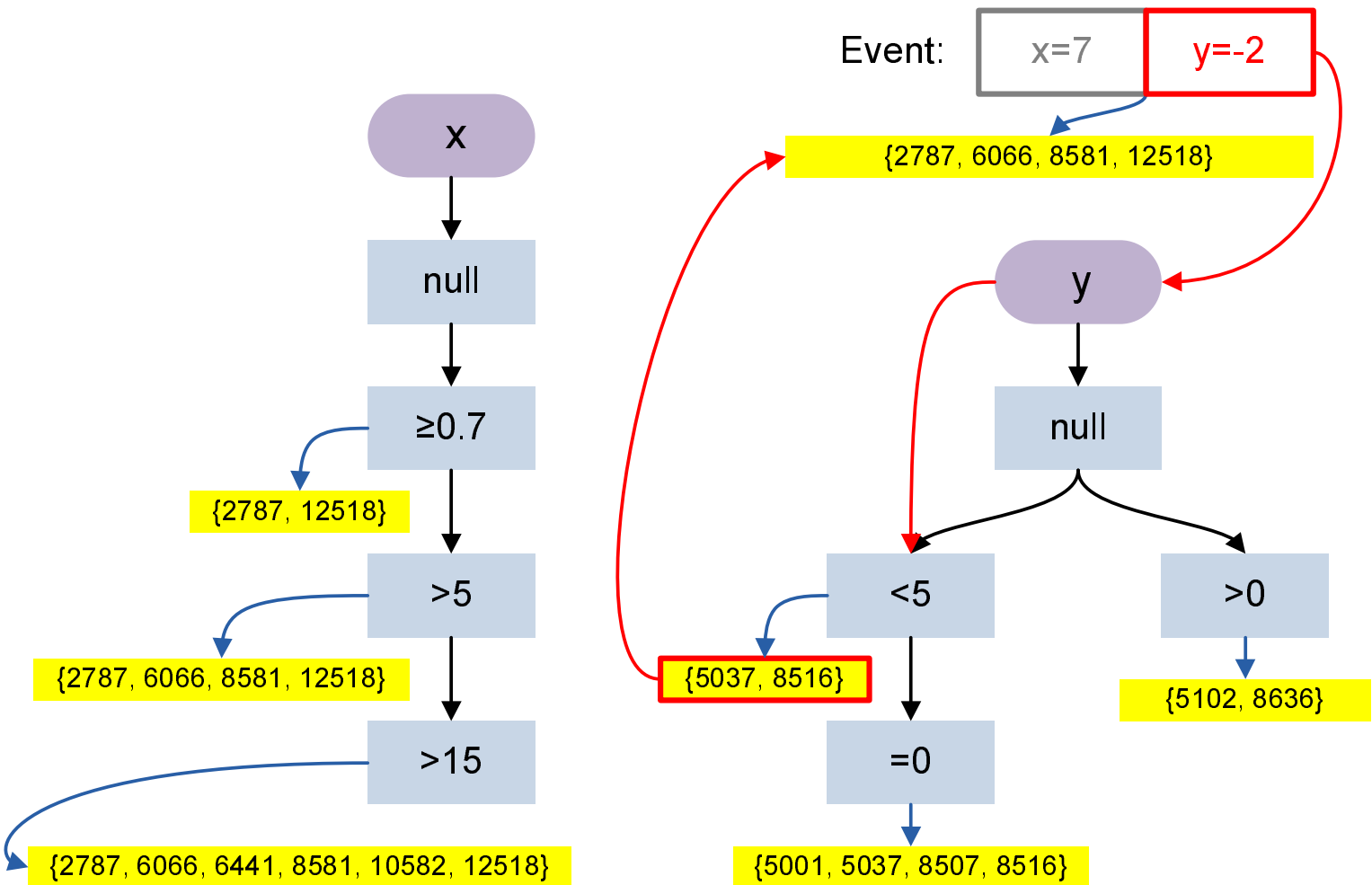
SBSPoset: event forwarding



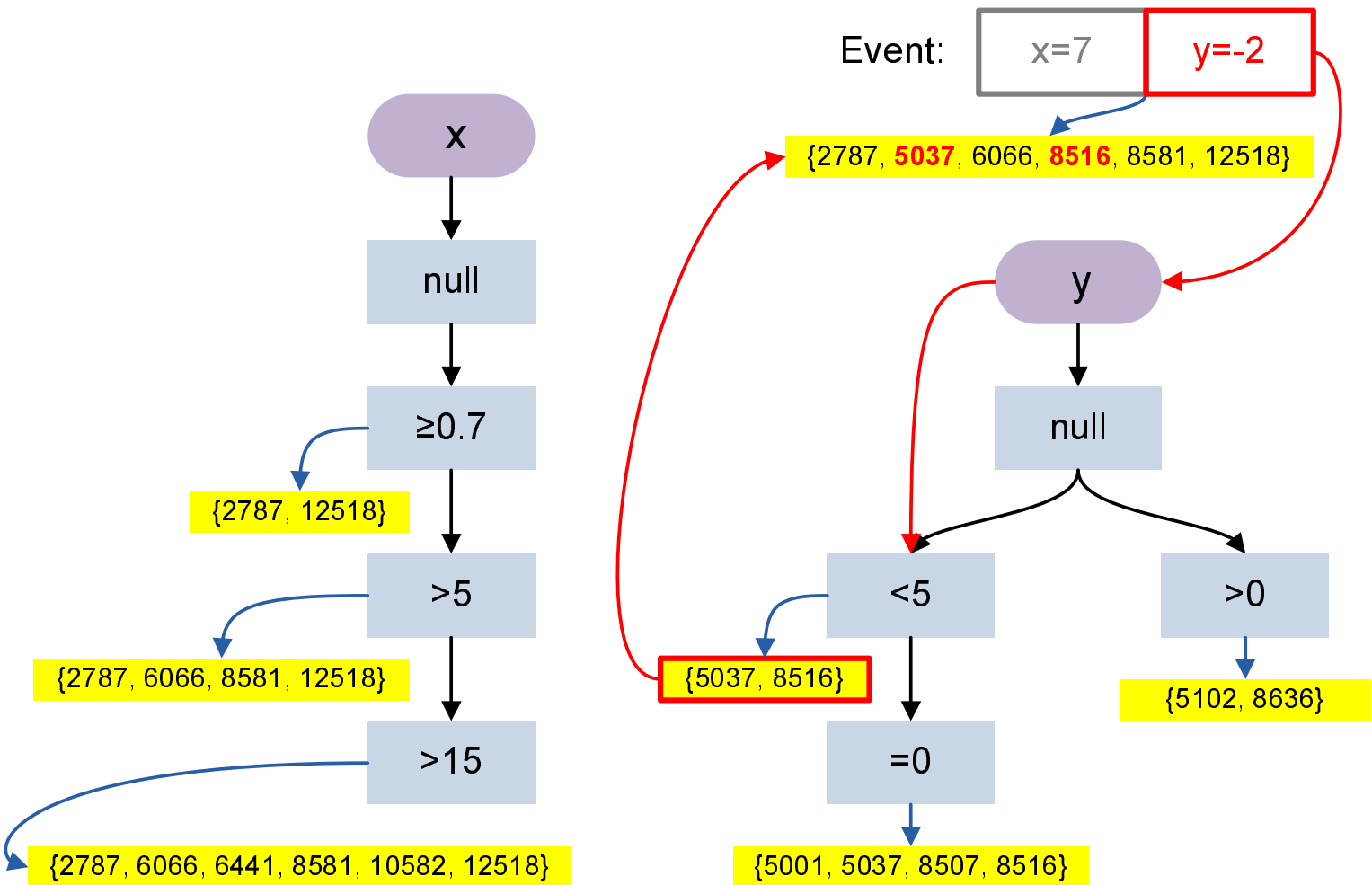
SBSPoset: event forwarding



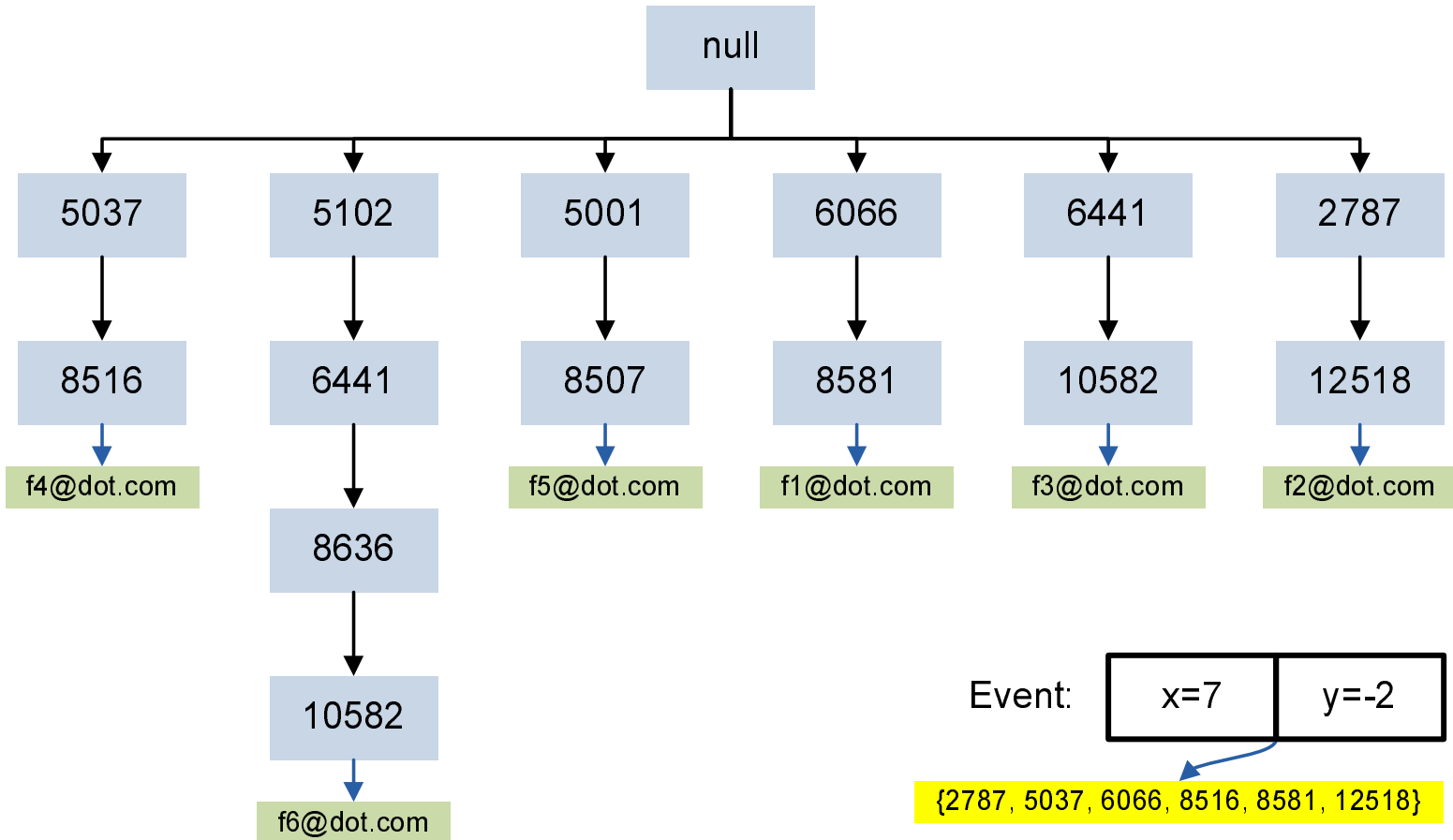
SBSPoset: event forwarding



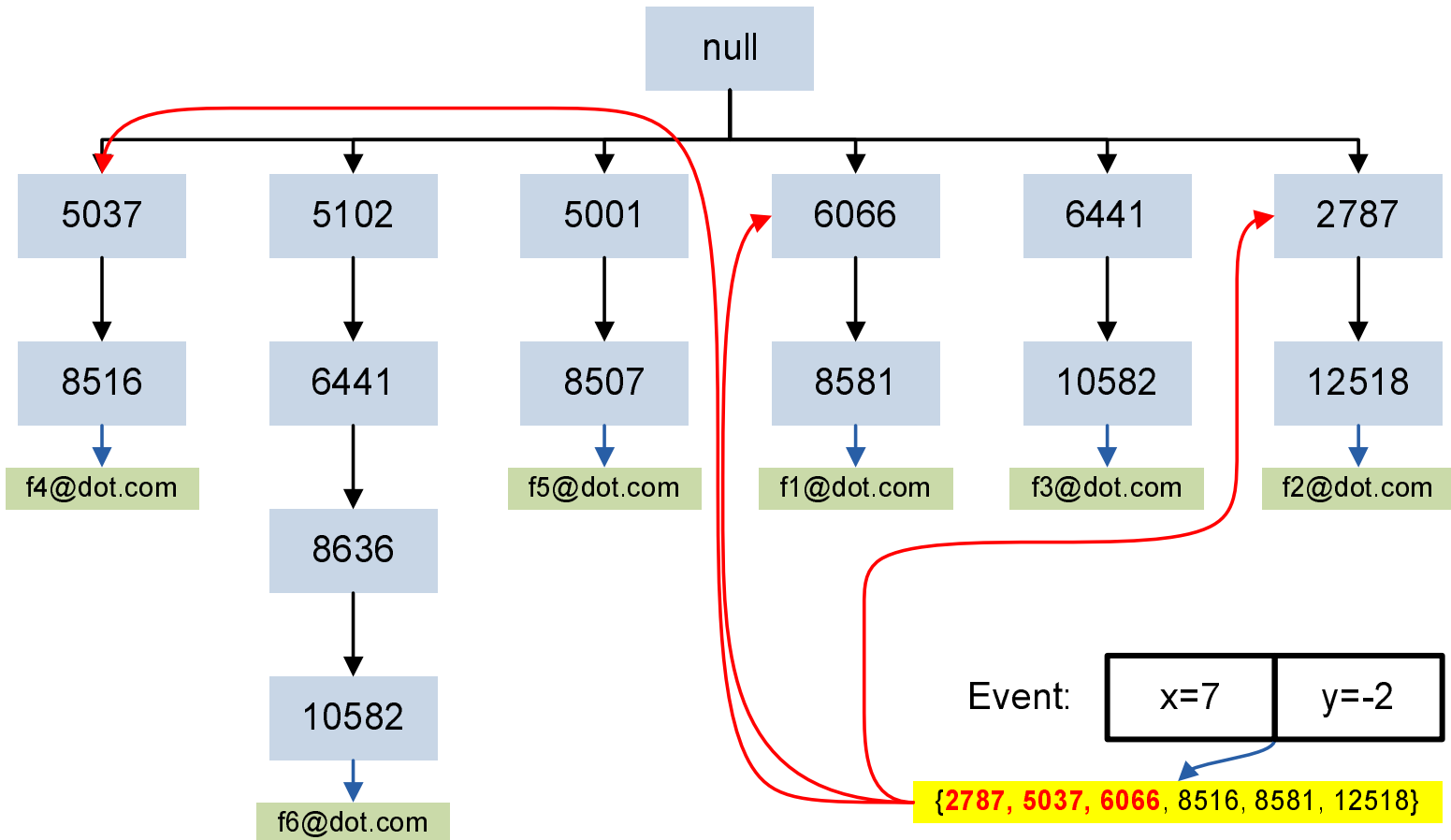
SBSPoset: event forwarding



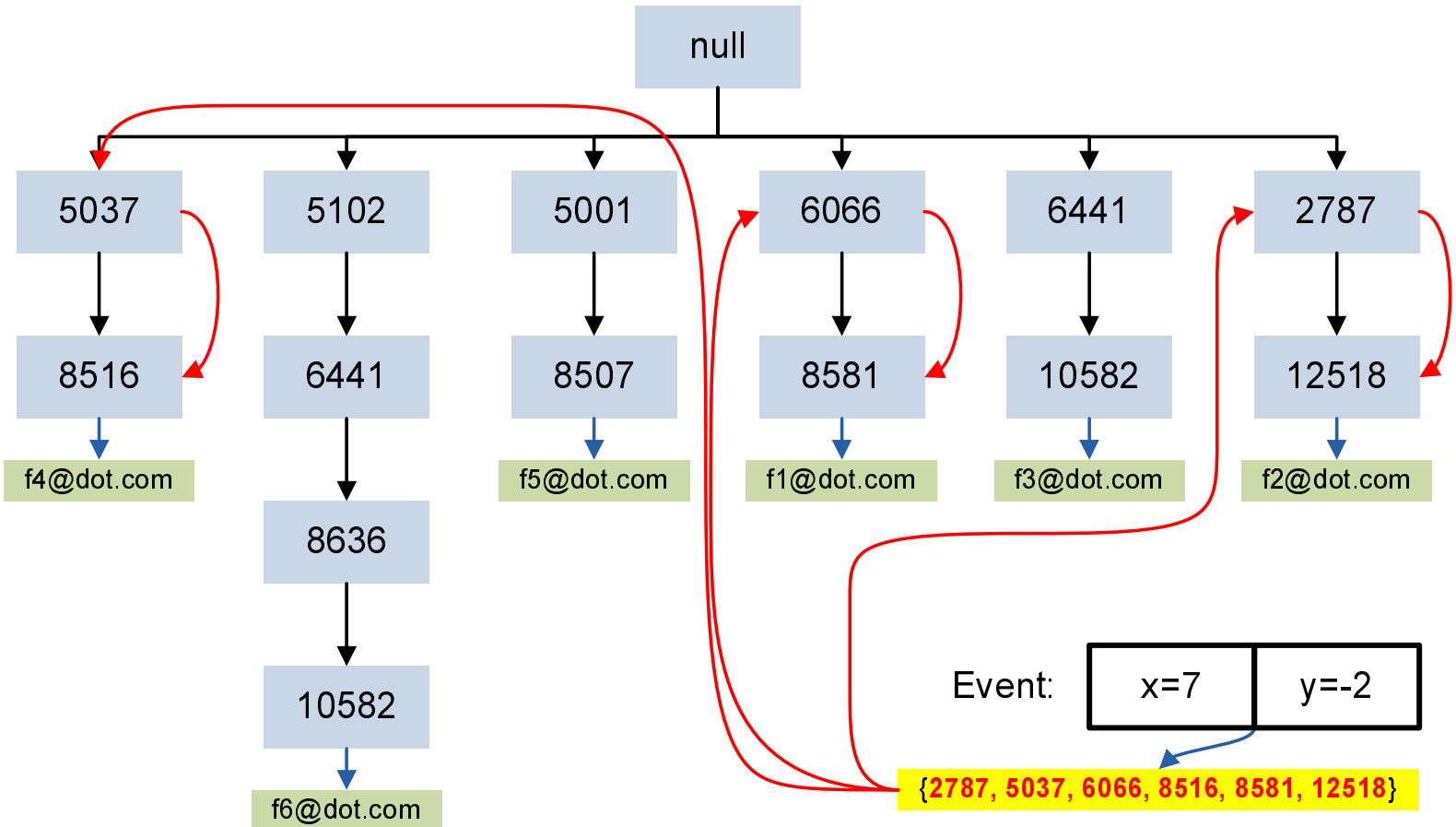
SBSTree: event forwarding



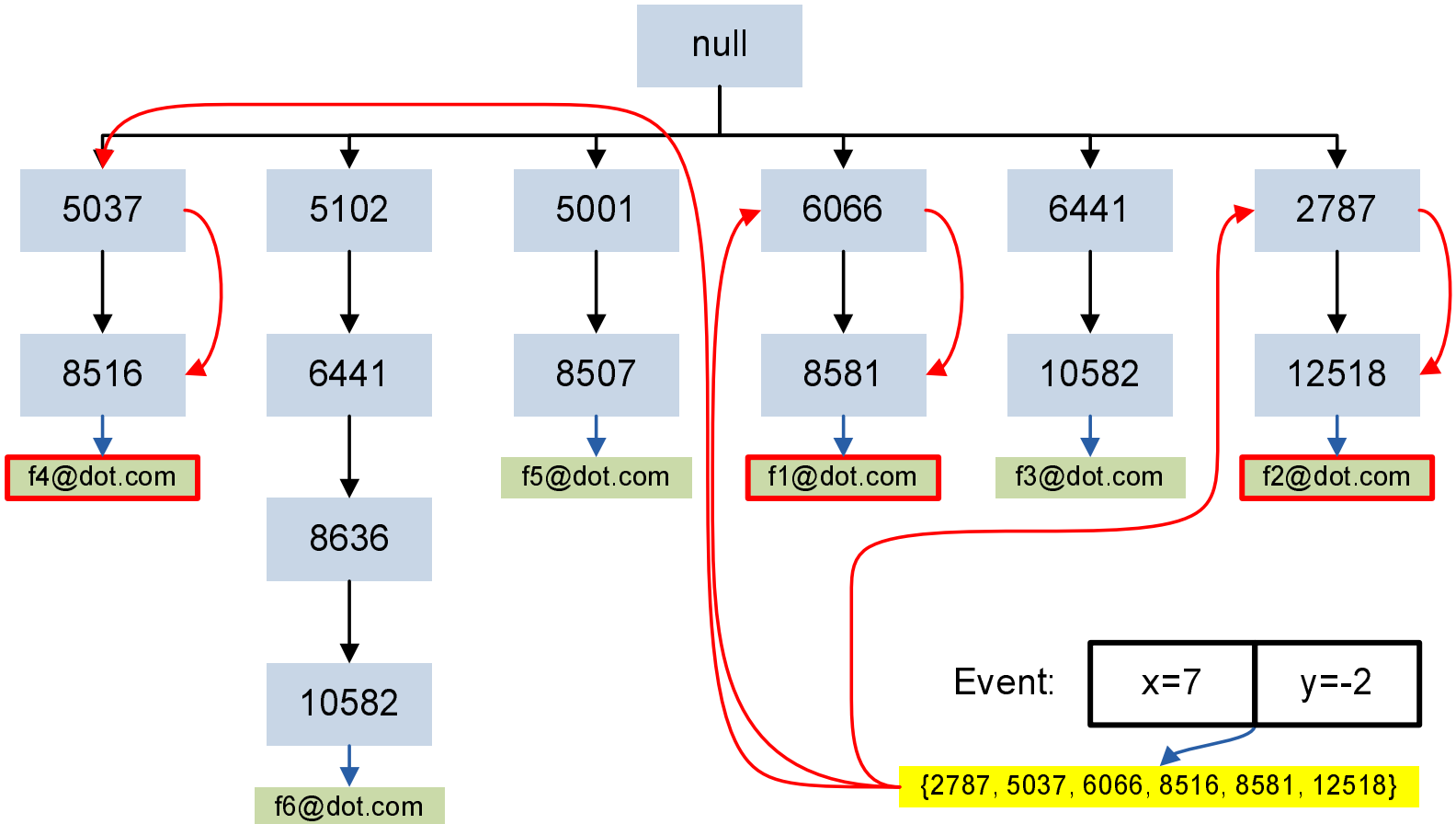
SBSTree: event forwarding



SBSTree: event forwarding



SBSTree: event forwarding



Evaluation Environment

- SIENA 1.5.4
  - http://www.inf.unisi.ch/carzaniga/siena/
- Apache Mina
- Stochastic Simulation in Java (SSJ) library
- BloomFilter

Evaluation Environment

- SIENA 1.5.4
- Apache Mina
  - http://mina.apache.org/
  - ver. 2.0.0-M1 as of 24th Jan 2008
- Stochastic Simulation in Java (SSJ) library
- BloomFilter

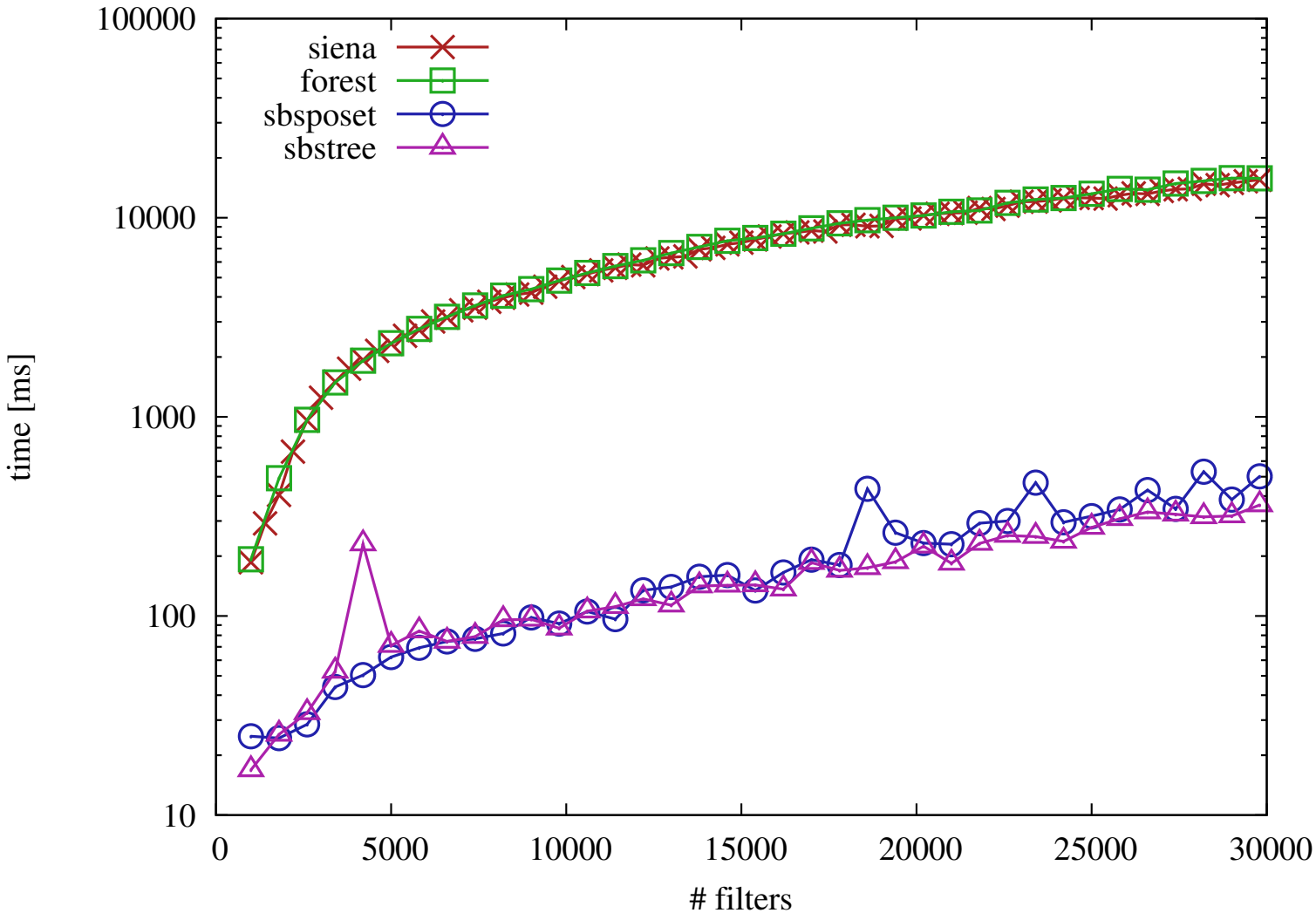
Evaluation Environment

- SIENA 1.5.4
- Apache Mina
- Stochastic Simulation in Java (SSJ) library
  - `umontreal.iro.lecuyer.randvar.*`
  - `umontreal.iro.lecuyer.rng.*`
- BloomFilter

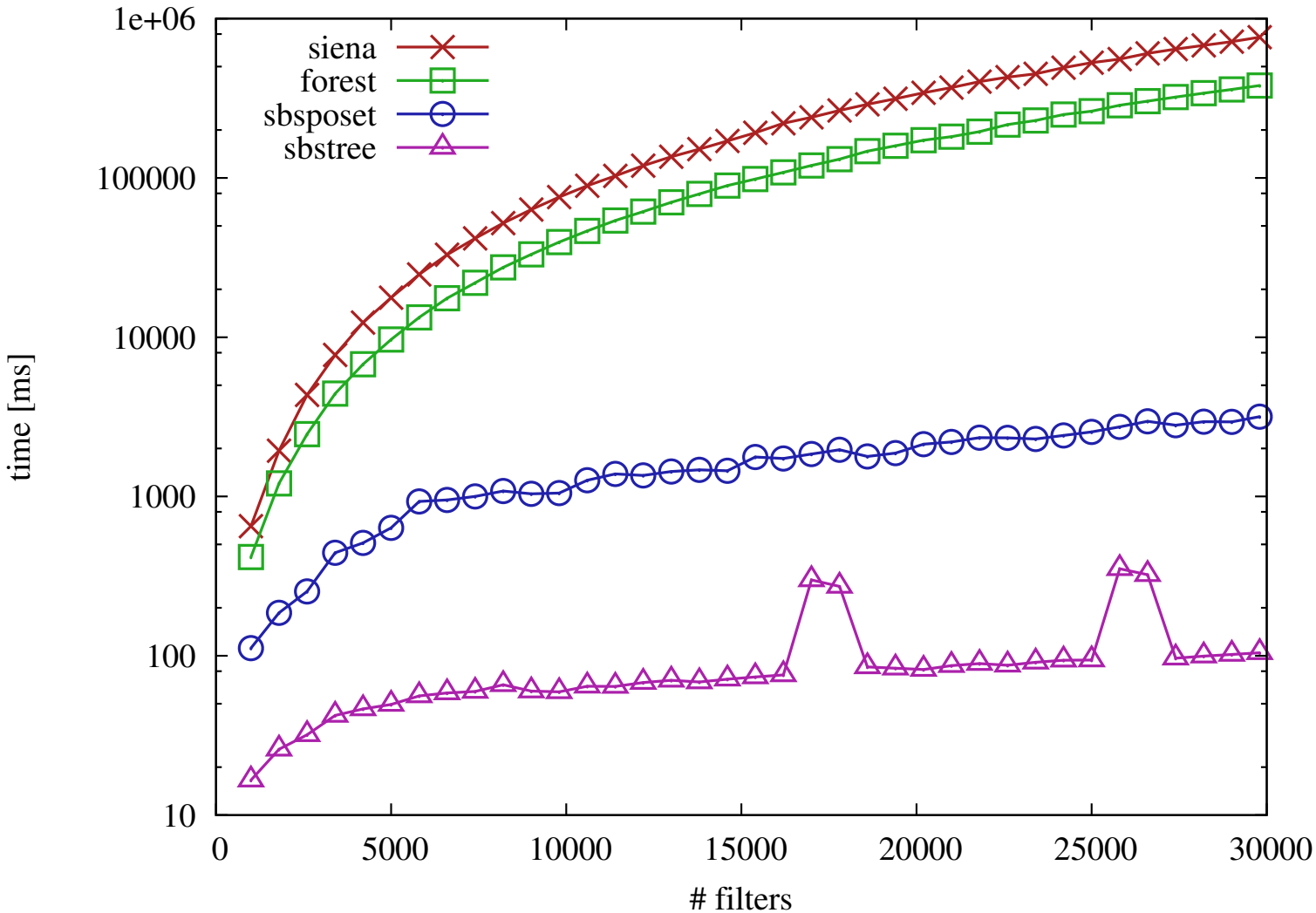
Evaluation Environment

- SIENA 1.5.4
- Apache Mina
- Stochastic Simulation in Java (SSJ) library
- BloomFilter
  - `http://wwwse.inf.tu-dresden.de/xsiena`
  - Based on code by: Hongbin Liu and Arash Partow

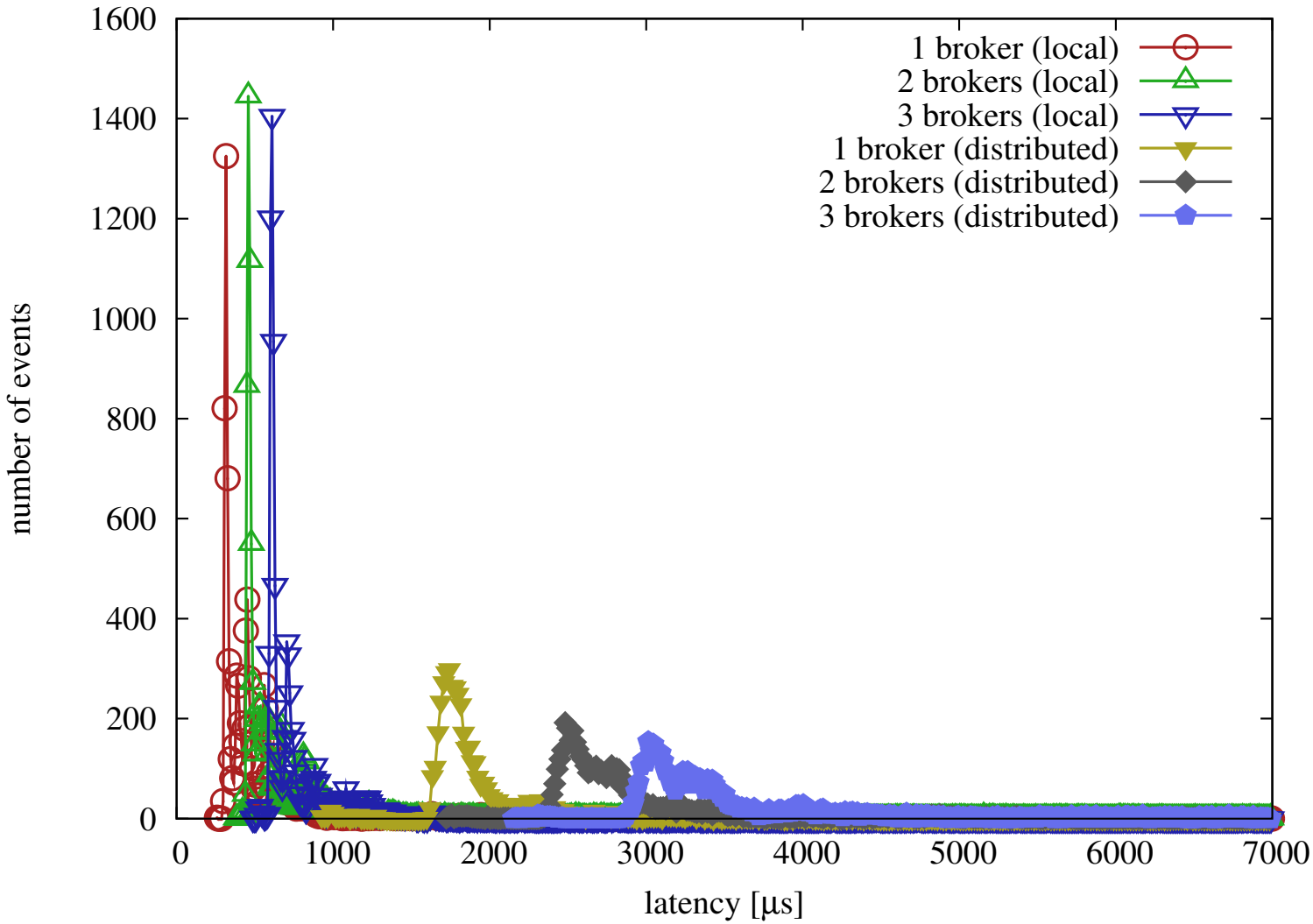
Event forwarding



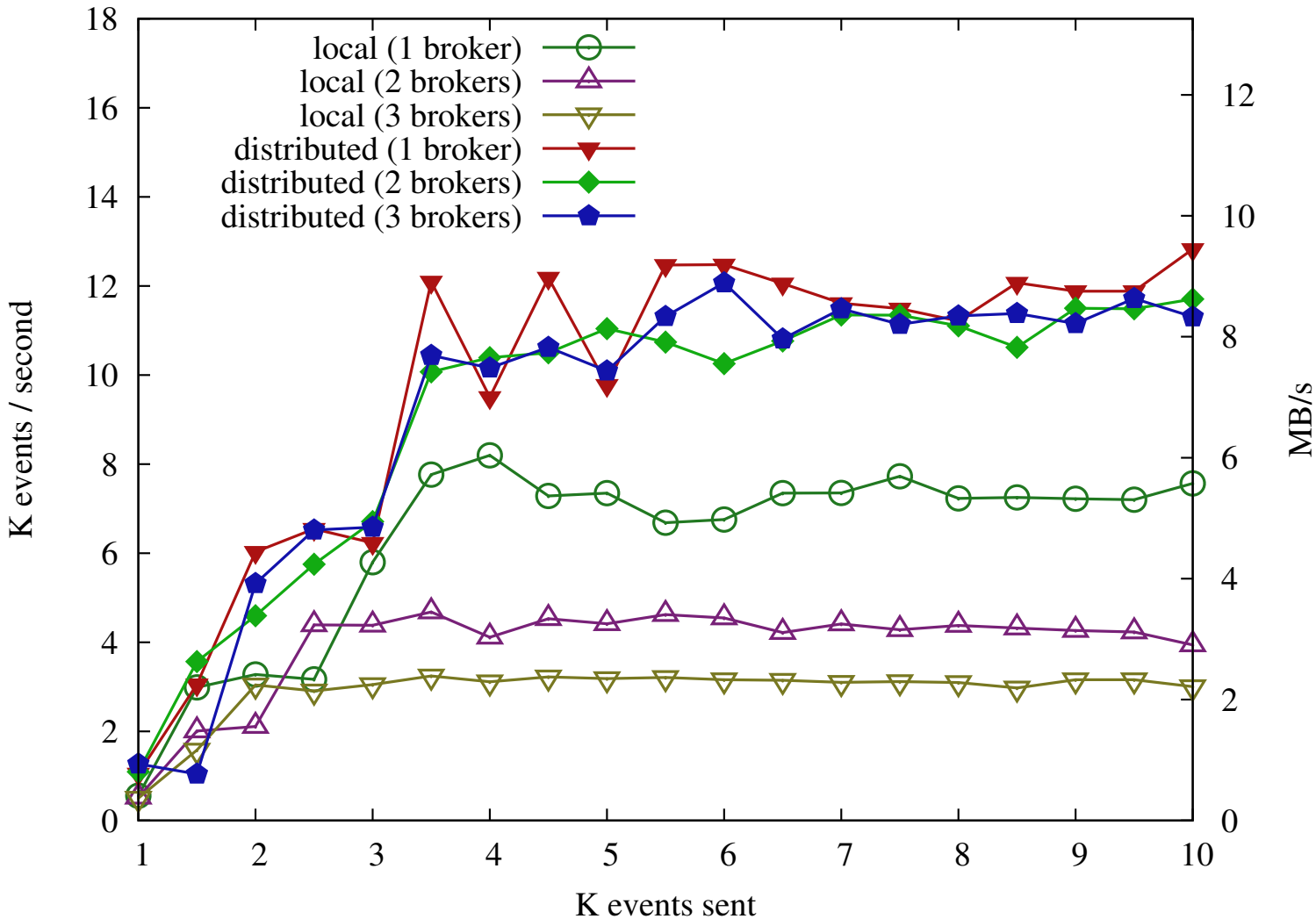
Subscription routing



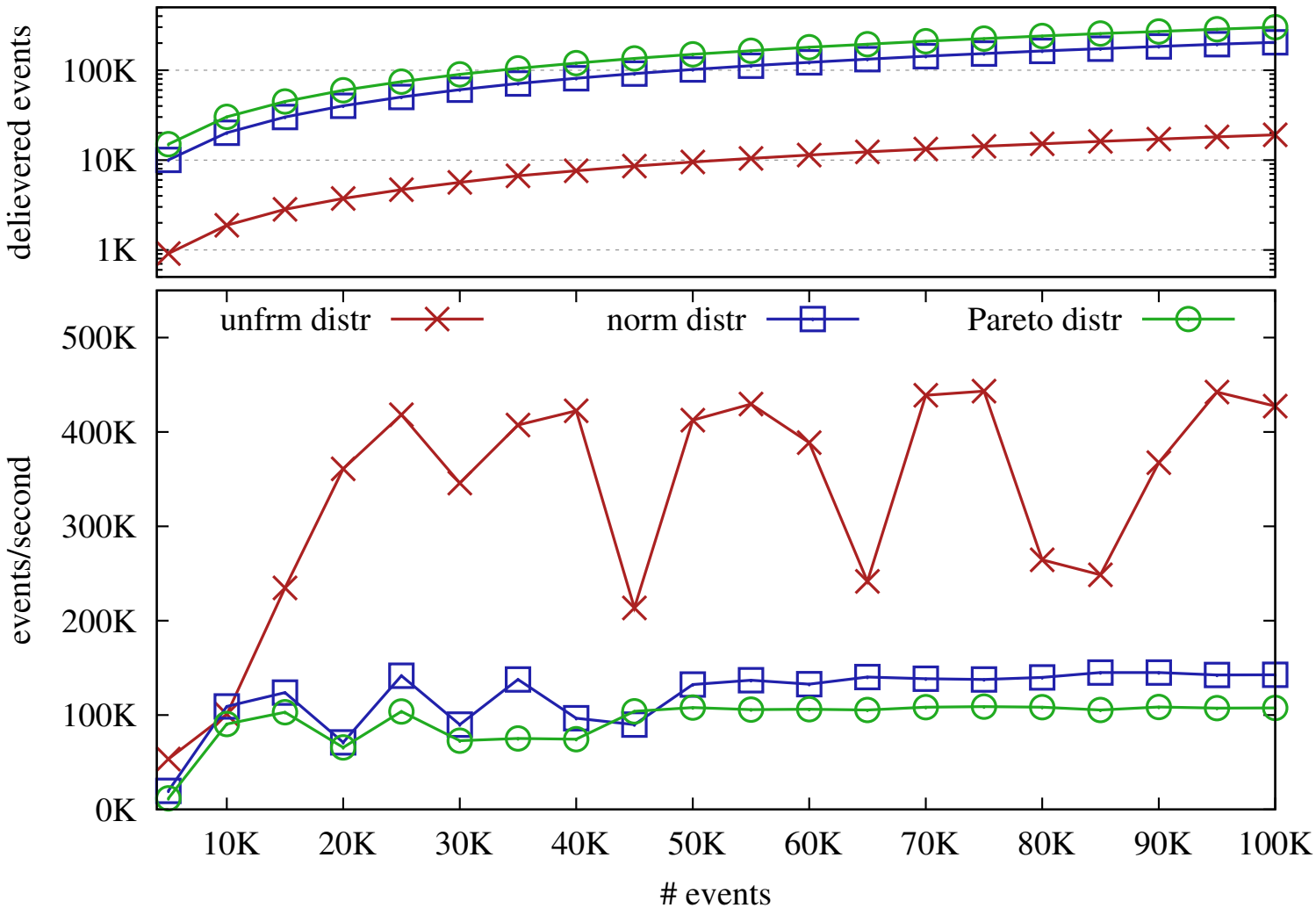
TCP/IP Latency



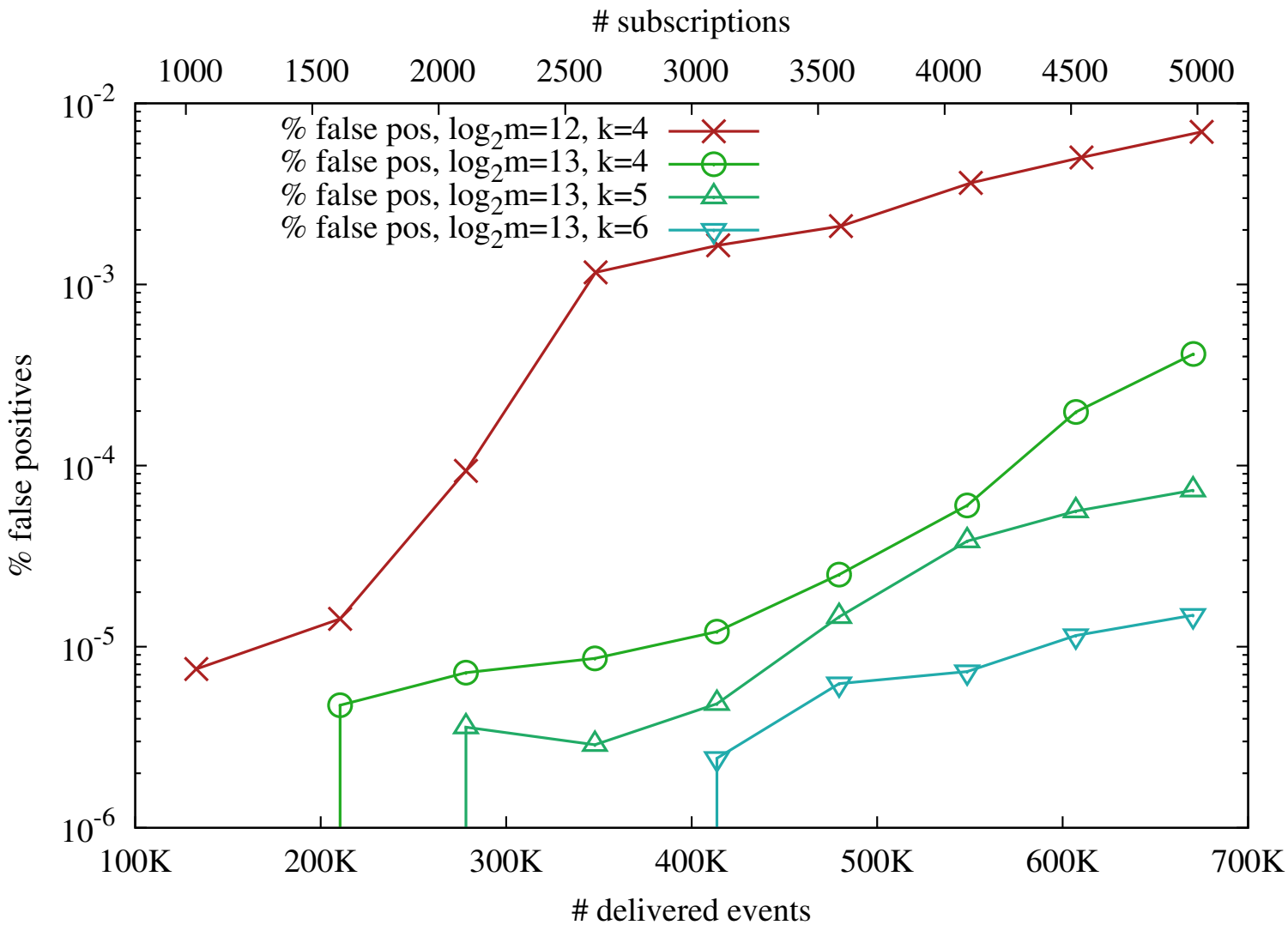
TCP/IP Throughput



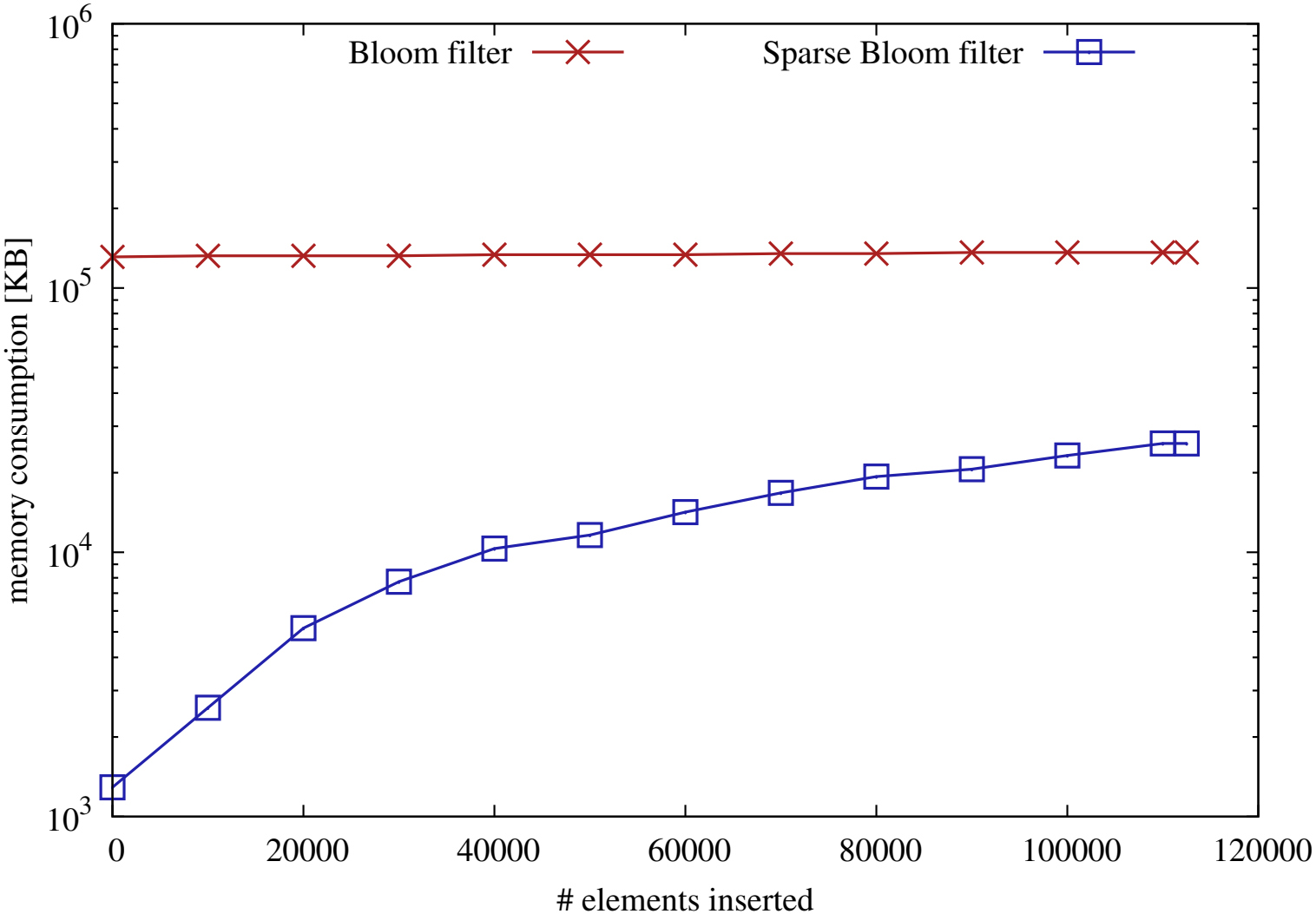
Local Throughput



False Positives



Bloom Filter vs Sparse Bloom Filter



Summary

- Low latency, high throughput
- Event forwarding based on Bloom filters
- No restrictions on the subscription language
- Decoupled management of predicates and their conjunctions

Thank You!

http://wwwse.inf.tu-dresden.de/xsiena/

References

Burton H. Bloom.

Space/time trade-offs in hash coding with allowable errors.
Communications of the ACM, 13(7):422–426, 1970.

Antonio Carzaniga, David S. Rosenblum, and Alexander L. Wolf.

Design and evaluation of a wide-area event notification service.
ACM Trans. Comput. Syst., 19(3):332–383, 2001.

Patrick Th. Eugster, Pascal A. Felber, Rachid Guerraoui, and Anne-Marie Kermarrec.

The many faces of publish/subscribe.
ACM Comput. Surv., 35(2):114–131, 2003.

Zbigniew Jerzak and Christof Fetzer.

Prefix forwarding for publish/subscribe.
In *DEBS '07: Proceedings of the 2007 Inaugural International Conference on Distributed Event-Based Systems*, pages 238–249, Toronto, Ontario, Canada, June 2007. ACM Press.

Sasu Tarkoma and Jaakko Kangasharju.

Optimizing content-based routers: posets and forests.
Distributed Computing, 19(1):62–77, September 2006.