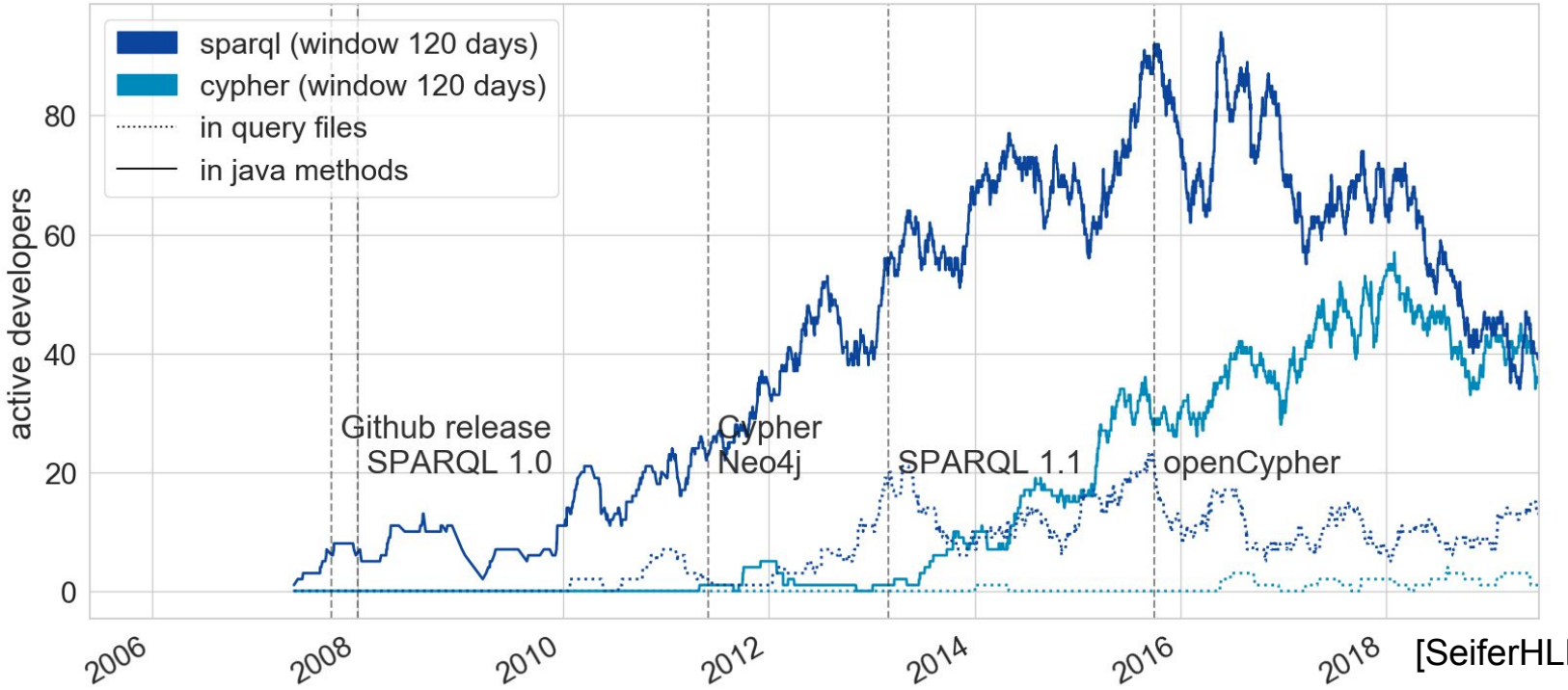
Incremental Map-Reduce on Repository History

Johannes Härtel and Ralf Lämmel
University of Koblenz-Landau, Germany
Software Languages Team

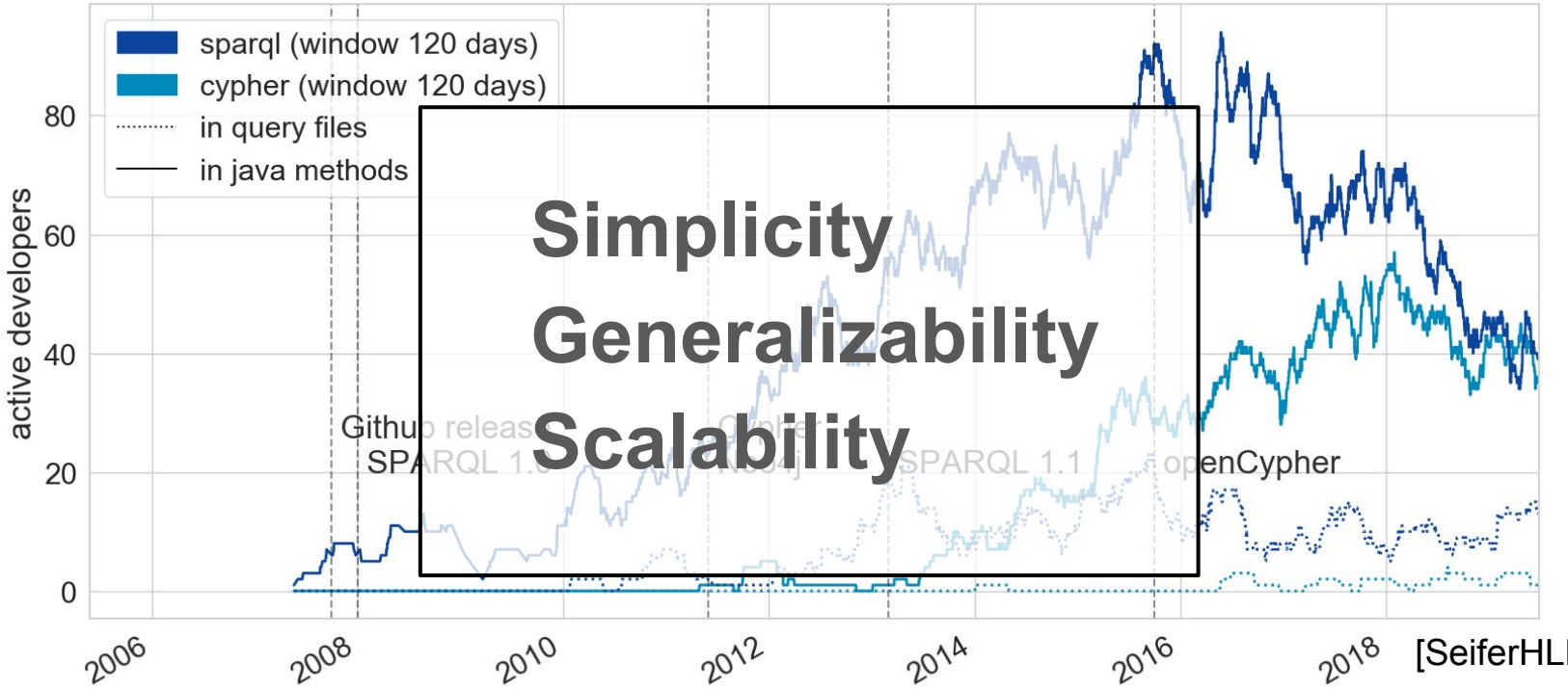
Challenges of Working with Repository History



[SeiferHLLS19]

7274 repositories, 1.373.000 commits where 22.000 touch sparql/cypher queries

Challenges of Working with Repository History



[SeiferHLLS19]

7274 repositories, 1.373.000 commits where 22.000 touch sparql/cypher queries

Demo

1. How to **represent** History?
2. How to **process** History?
3. How to **memoize** History?

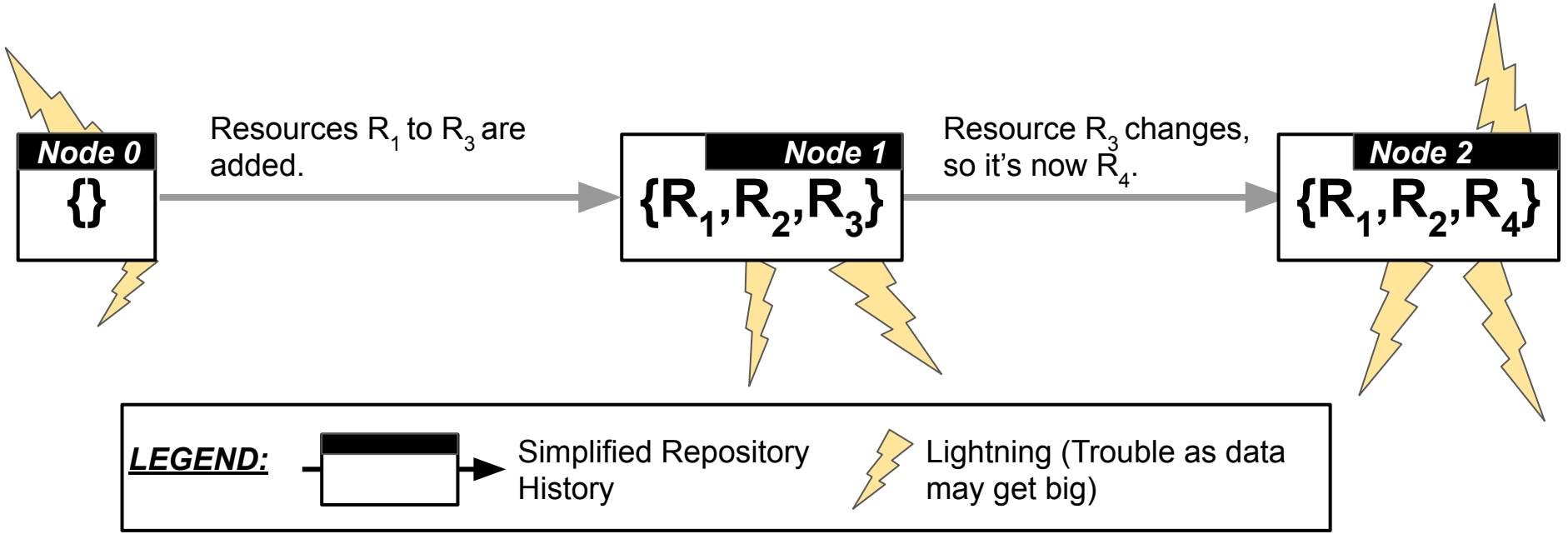
How to represent History?

Use Abelian Groups!

Johannes Härtel - University of Koblenz - johanneshaertel@uni-koblenz.de

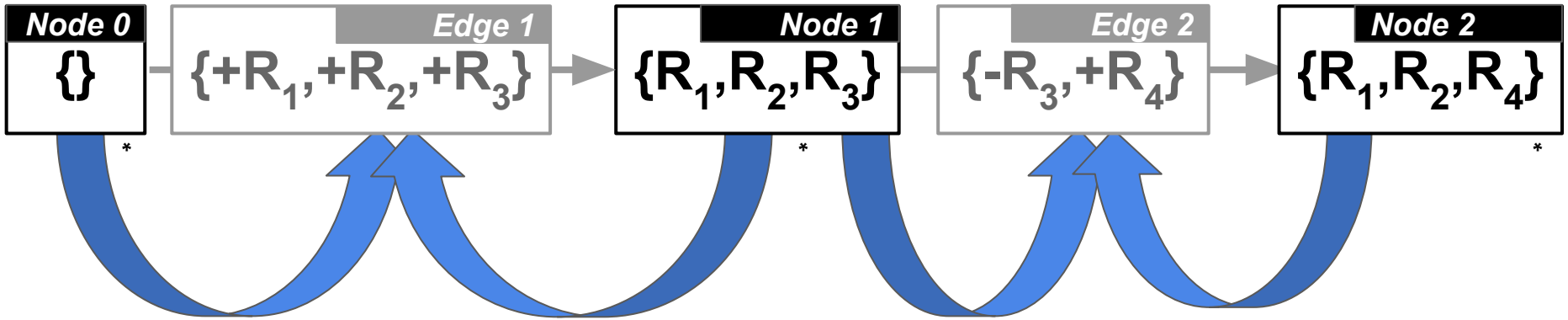
How to represent History?

We model repository history as the Nodes 0-2, storing bags (‘{‘ ‘}’) of immutable resources ($R_{1\text{-}4}$).



How to represent History?

We use Nodes 0-2 to compute the changes on Edges 1-2 in terms of bags with positive and negative elements ($\pm R_{1\text{-}4}$).

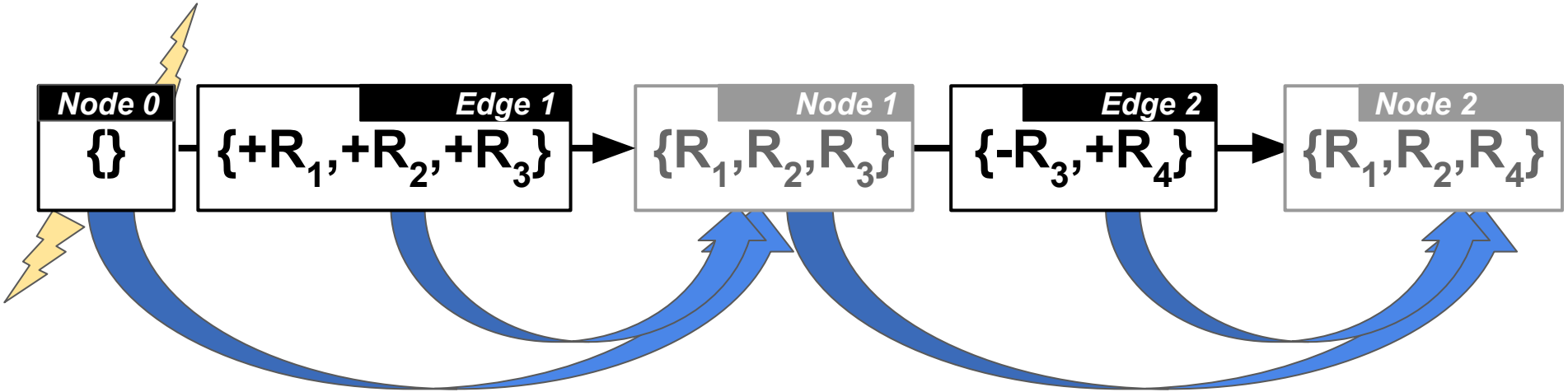


**** lightnings are avoided due to limitations in space on slide***

Johannes Härtel - University of Koblenz - johanneshaertel@uni-koblenz.de

How to represent History?

Data on Node 1 and Node 2 can be dropped, as we can use Node 0, Edge 1 and Edge 2 for reproducing all historic data.

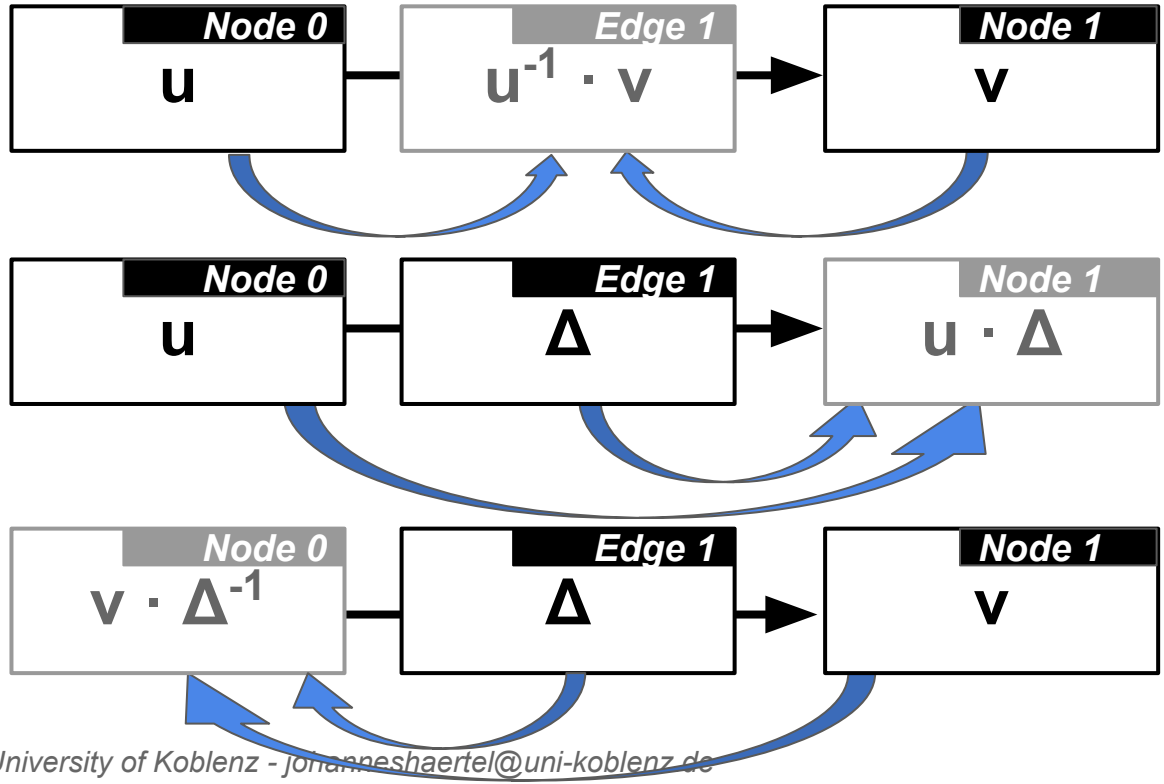


Johannes Härtel - University of Koblenz - johanneshaertel@uni-koblenz.de

How to represent History?
Use Abelian Groups!

Abelian groups (for u,v and Δ) consist of:

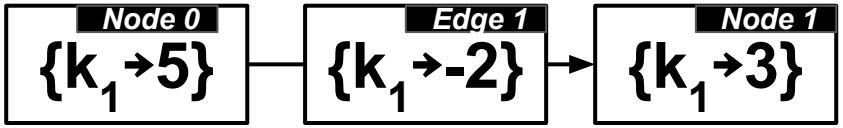
1. A group operator ‘ · ’ (commutative).
2. An inverse element x^{-1}.
3. An identity element z.



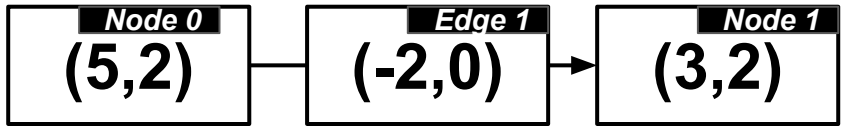
Some Abelian Groups

Integers (Int)
Group operator **+**.
Inverser element **-x**.
Identity element **0**.

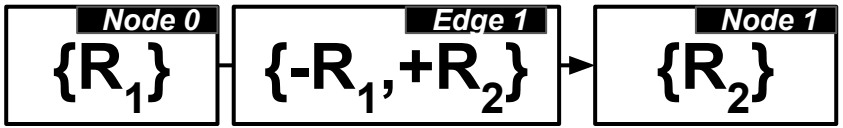
Maps (Map[K, V])
Group operator does an element-wise product, using the operator of a **nested group V**; K can hold arbitrary types . . .



Tuples ((A, B))
All operators delegates to **nested groups for A and B**.



Bags (Bag[K])
Alias for Map[K, Int].

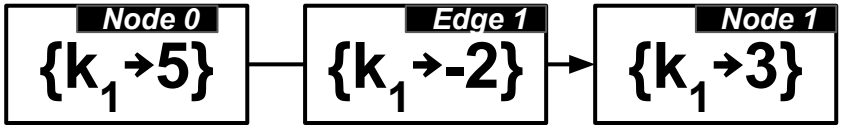


Johannes Härtel - University of Koblenz - johanneshaertel@uni-koblenz.de

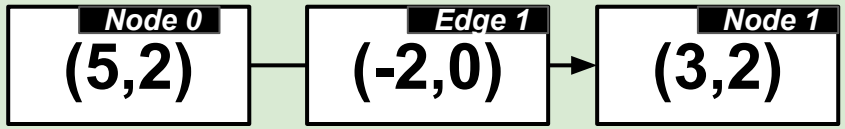
Some Abelian Groups

Integers (Int)
Group operator **+**.
Inverser element **-x**.
Identity element **0**.

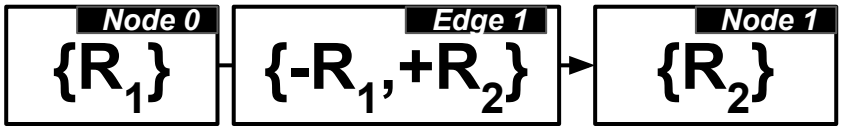
Maps (Map[K, V])
Group operator does an element-wise product, using the operator of a **nested group V**; K can hold arbitrary types . . .



Tuples ((A, B))
All operators delegates to **nested groups for A and B**.



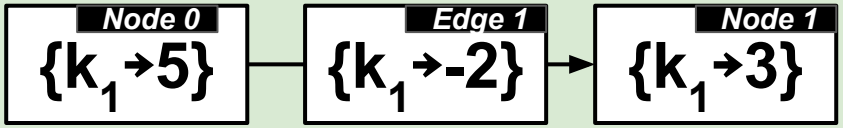
Bags (Bag[K])
Alias for Map[K, Int].



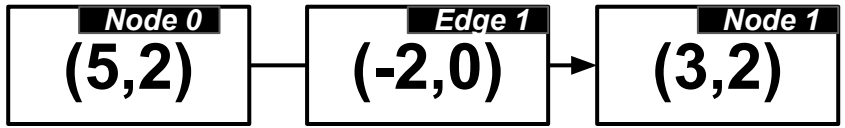
Some Abelian Groups

Integers (Int)
Group operator **+**.
Inverser element **-x**.
Identity element **0**.

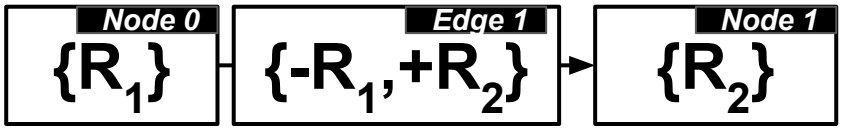
Maps (Map[K, V])
Group operator does an element-wise product, using the operator of a **nested group V**; K can hold arbitrary types . . .



Tuples ((A, B))
All operators delegates to **nested groups for A and B**.



Bags (Bag[K])
Alias for Map[K, Int].

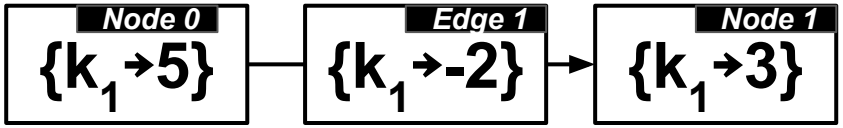


Johannes Härtel - University of Koblenz - johanneshaertel@uni-koblenz.de

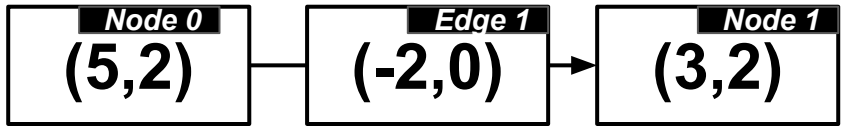
Some Abelian Groups

Integers (Int)
Group operator **+**.
Inverser element **-x**.
Identity element **0**.

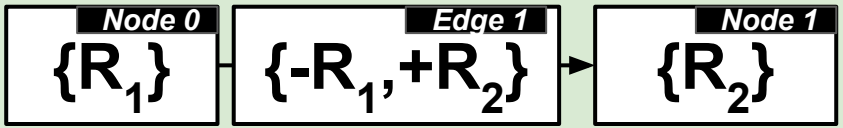
Maps (Map[K, V])
Group operator does an element-wise product, using the operator of a **nested group V**; K can hold arbitrary types . . .



Tuples ((A, B))
All operators delegates to **nested groups for A and B**.



Bags (Bag[K])
Alias for Map[K, Int].



Johannes Härtel - University of Koblenz - johanneshaertel@uni-koblenz.de

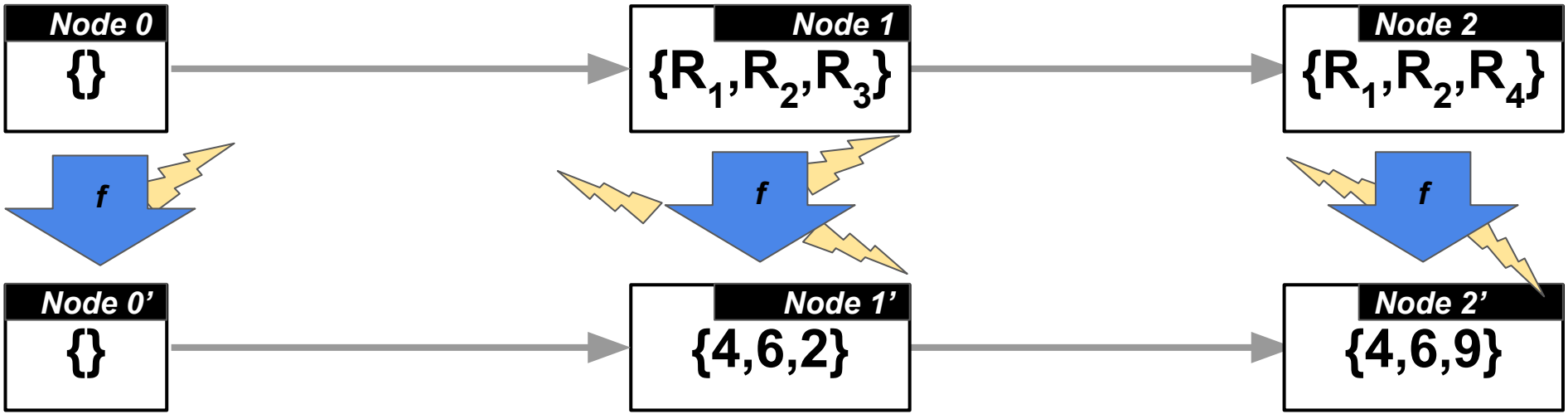
How to process History?

Use Group Homomorphisms!

Johannes Härtel - University of Koblenz - johanneshaertel@uni-koblenz.de

How to process History?

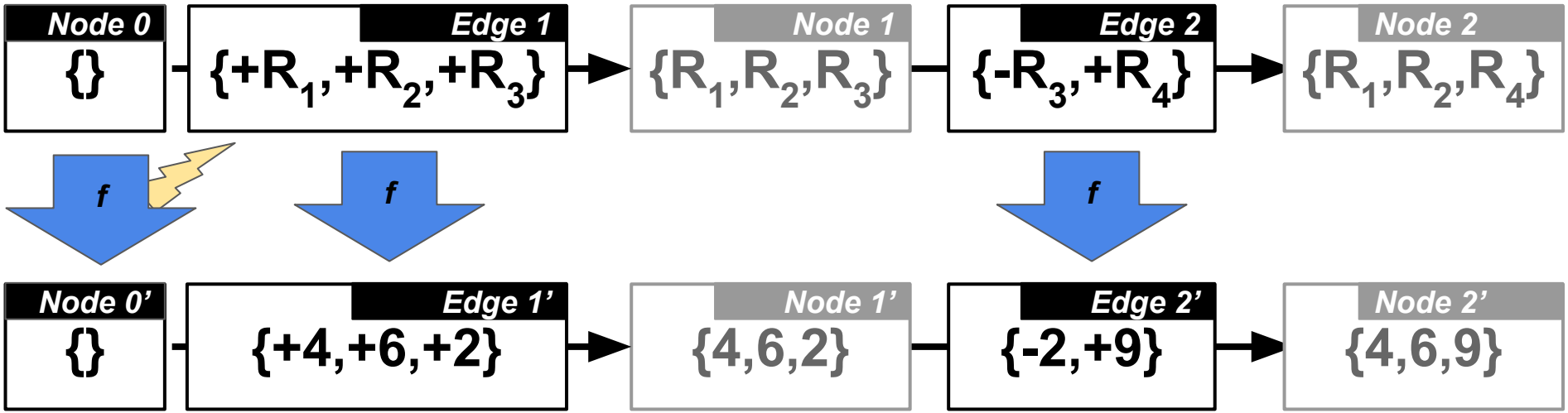
Extracting metrics can be done by applying a function f to the bags on Node 0-2.



(In this example metrics are f({R_1}) = 4, f({R_2}) = 6, f({R_3}) = 2, f({R_4}) = 9)

How to process History?

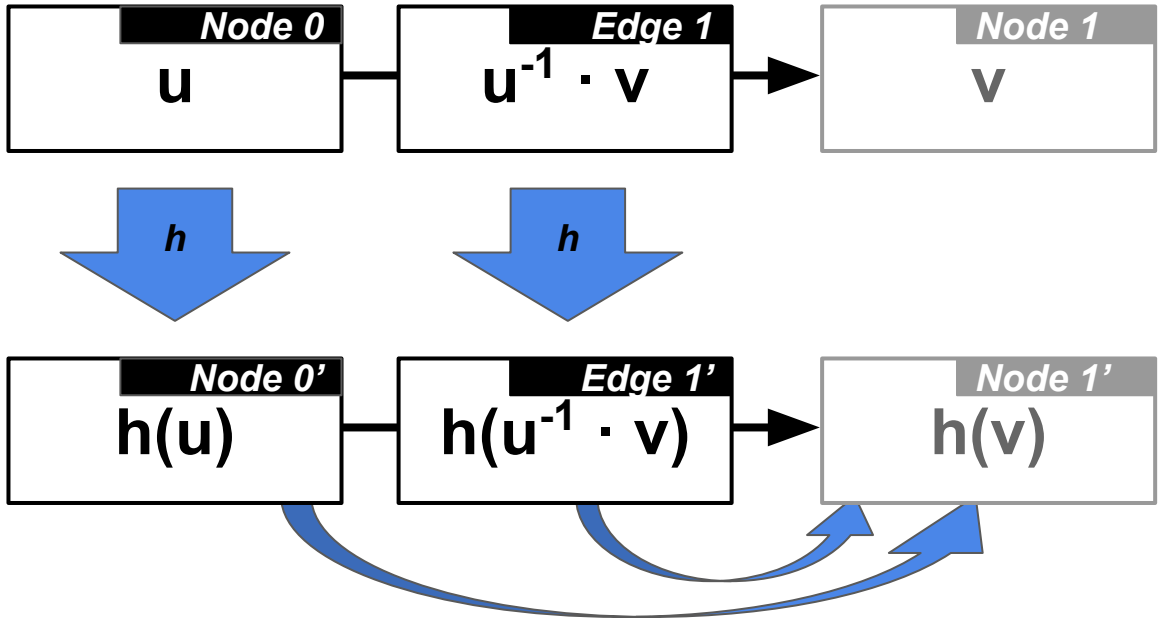
We can apply f to Node 0, Edge 1 and Edge 2 and get the same results;
Node 1' and Node 2' can be restored.



(In this example metrics are f({R_1}) = 4, f({R_2}) = 6, f({R_3}) = 2, f({R_4}) = 9)

How to process History?
Use Group Homomorphisms!

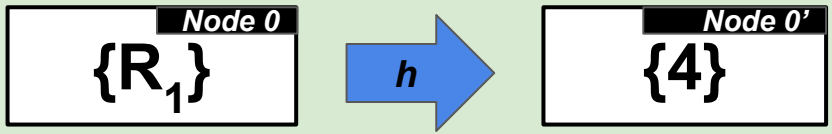
For a group homomorphism **h**,
$h(u^{-1} \cdot v) = h(u)^{-1} \cdot h(v)$ holds.



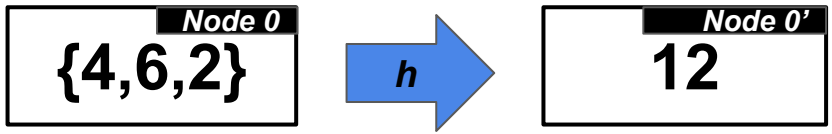
Johannes Härtel - University of Koblenz - johanneshaertel@uni-koblenz.de

Some Group Homomorphisms

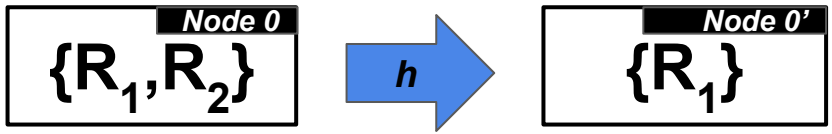
map (Bag[V1] → Bag[V2])
Maps the elements of a bag.



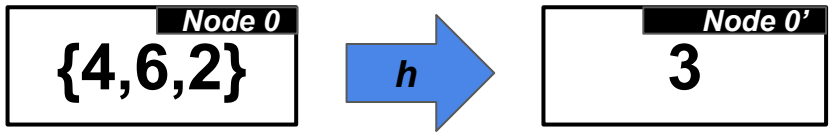
sum (Bag[Int] → Int)
Sums up the integer entries of a bag producing an integer.



filter (Bag[V1] → Bag[V2])
Filters the elements of a bag.



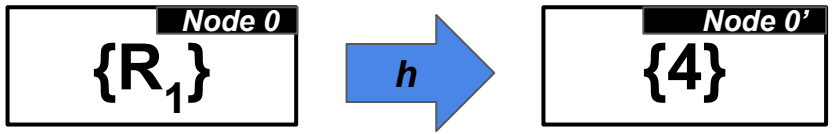
count (Bag[K] → Int)
Counts the elements of a bag producing an integer.



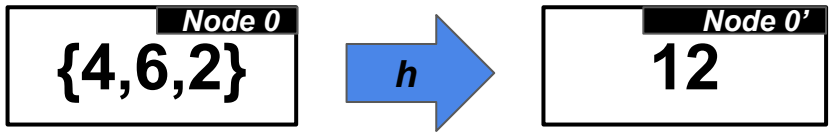
Johannes Härtel - University of Koblenz - johanneshaertel@uni-koblenz.de

Some Group Homomorphisms

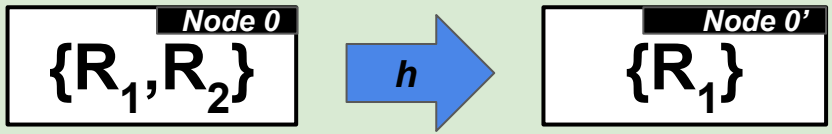
map (Bag[V1] → Bag[V2])
Maps the elements of a bag.



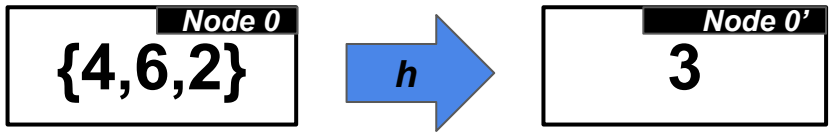
sum (Bag[Int] → Int)
Sums up the integer entries of a bag producing an integer.



filter (Bag[V1] → Bag[V2])
Filters the elements of a bag.



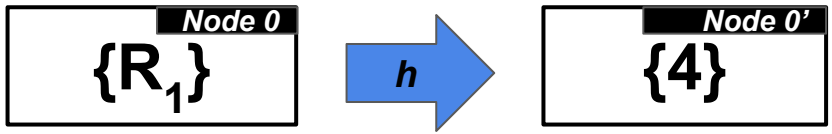
count (Bag[K] → Int)
Counts the elements of a bag producing an integer.



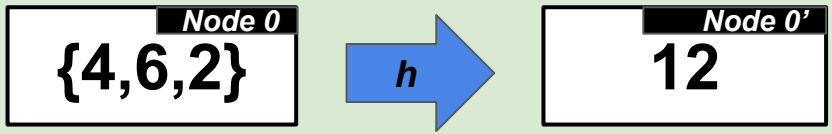
Johannes Härtel - University of Koblenz - johanneshaertel@uni-koblenz.de

Some Group Homomorphisms

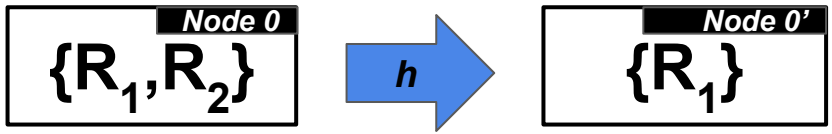
map (Bag[V1] → Bag[V2])
Maps the elements of a bag.



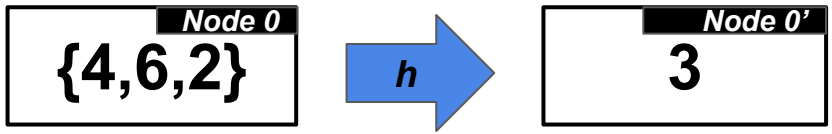
sum (Bag[Int] → Int)
Sums up the integer entries of a bag producing an integer.



filter (Bag[V1] → Bag[V2])
Filters the elements of a bag.



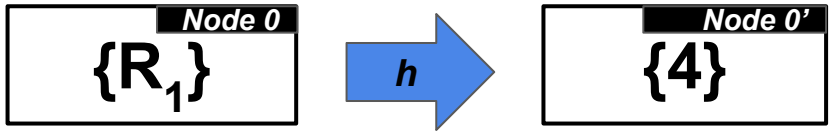
count (Bag[K] → Int)
Counts the elements of a bag producing an integer.



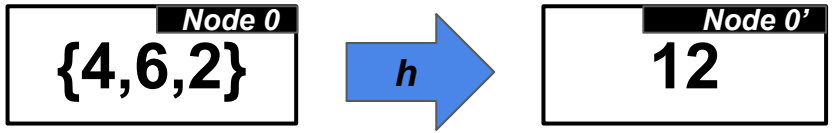
Johannes Härtel - University of Koblenz - johanneshaertel@uni-koblenz.de

Some Group Homomorphisms

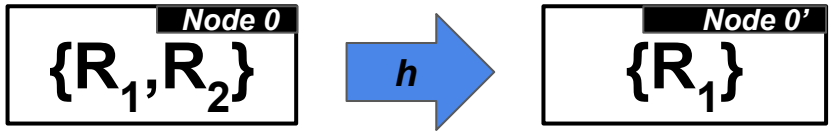
map (Bag[V1] → Bag[V2])
Maps the elements of a bag.



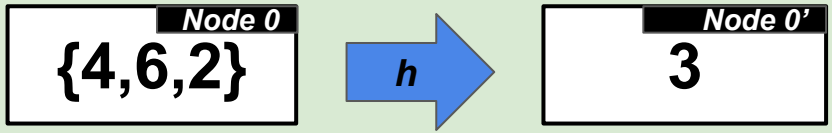
sum (Bag[Int] → Int)
Sums up the integer entries of a bag producing an integer.



filter (Bag[V1] → Bag[V2])
Filters the elements of a bag.



count (Bag[K] → Int)
Counts the elements of a bag producing an integer.



Johannes Härtel - University of Koblenz - johanneshaertel@uni-koblenz.de

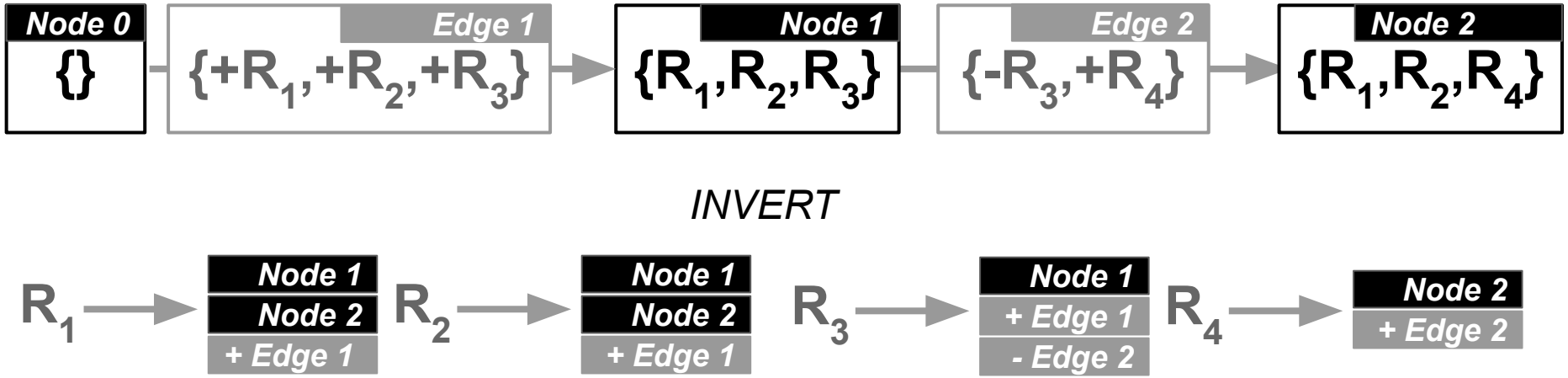
How to memoize History?

Invert it!

Johannes Härtel - University of Koblenz - johanneshaertel@uni-koblenz.de

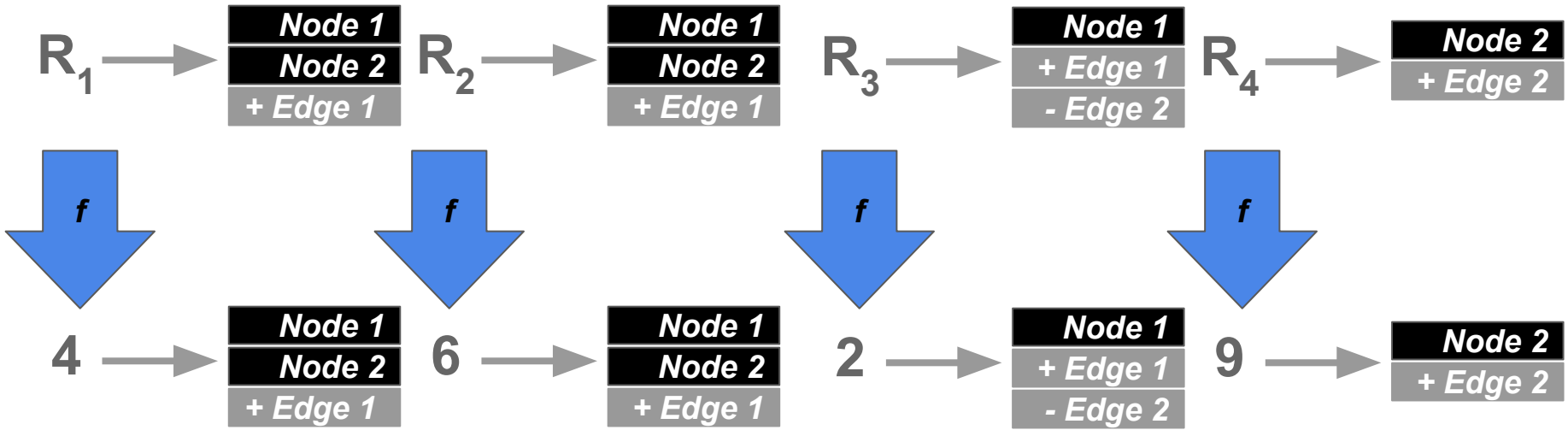
How to memoize History?

Instead of pointing from nodes/edges to the data, history can be inverted, pointing from data to the edges/nodes.



How to memoize History?

Functions can be applied without causing redundant computations.



(Metrics are f({R_1}) = 4, f({R_2}) = 6, f({R_3}) = 2, f({R_4}) = 9)

Evaluation

Johannes Härtel - University of Koblenz - johanneshaertel@uni-koblenz.de

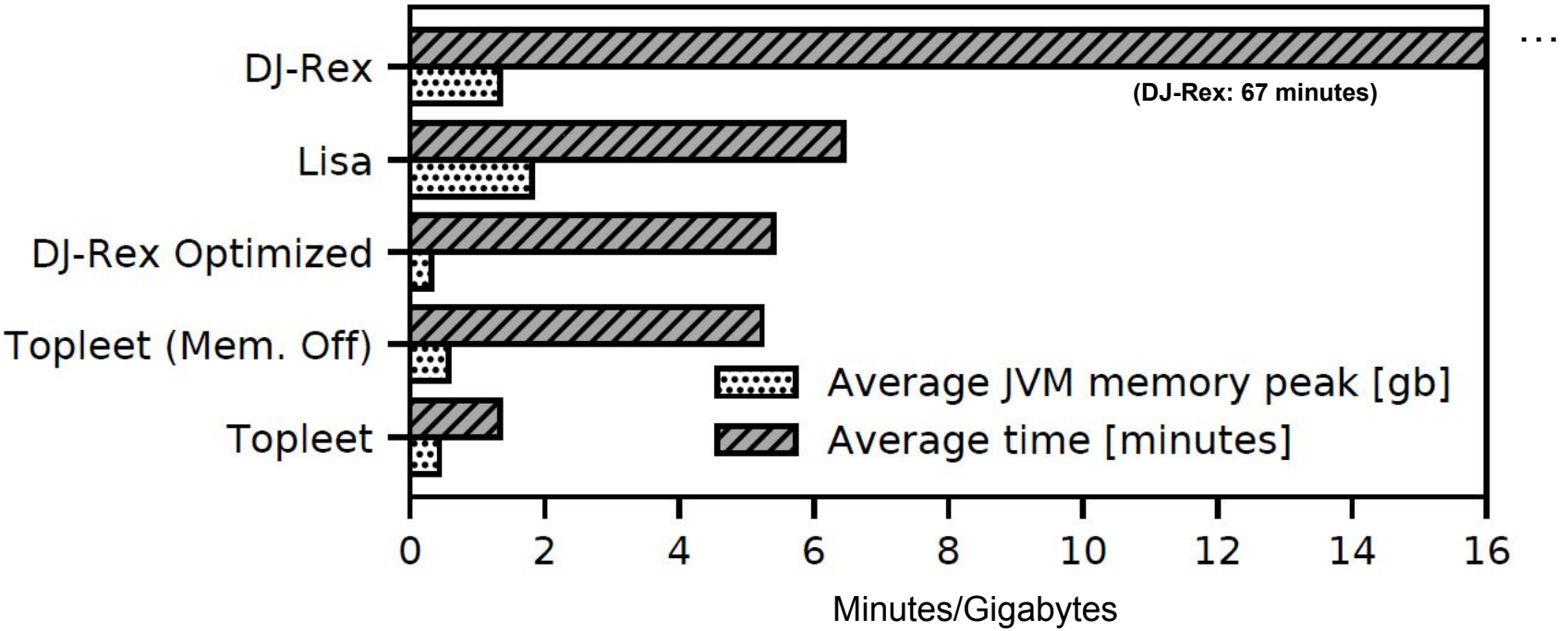
* Excluded from evaluation.

Evaluated Approaches

	Language/ Interface	History Model	Distribution	Parallelization	Compression/ Red. Reduction/ Memoization	Change Processing/ Incrementalization
DJ-Rex [ShangJAH09]	Map-Reduce		Generic	Generic	Manual Optimization	Manual Optimization
Boa* [DyerNRN13]	DSL		Generic	Generic	Specific to Java	Manual Optimization
LISA [AlexandruG15]	Signal/Collect	Linear		Specific to Resources	Specific to AST	Incremental Messaging
Topleet	Map-Reduce	Acyclic	Generic	Generic	Abelian groups	Group Homomorphisms & Inc. Indexing

Average Time and Memory Usage

Running cyclomatic complexity solutions on 98 repositories.



Johannes Härtel - University of Koblenz - johanneshaertel@uni-koblenz.de

Try Topleet

https://github.com/topleet/topleet

- **Extended map-reduce functionality** is provided, such as, join or cartesian.
- **Concise interface** defined by 5 core operators.
- **Interchangeable implementations**, e.g., incremental vs. non-incremental.
- **Differential testing** is used for checking correctness.
- **More application** cases are available online.
- **Future work** may include authorship attribution, refactoring detection and graph queries to the history.

Johannes Härtel - University of Koblenz - johanneshaertel@uni-koblenz.de

Fin

https://github.com/topleet/topleet

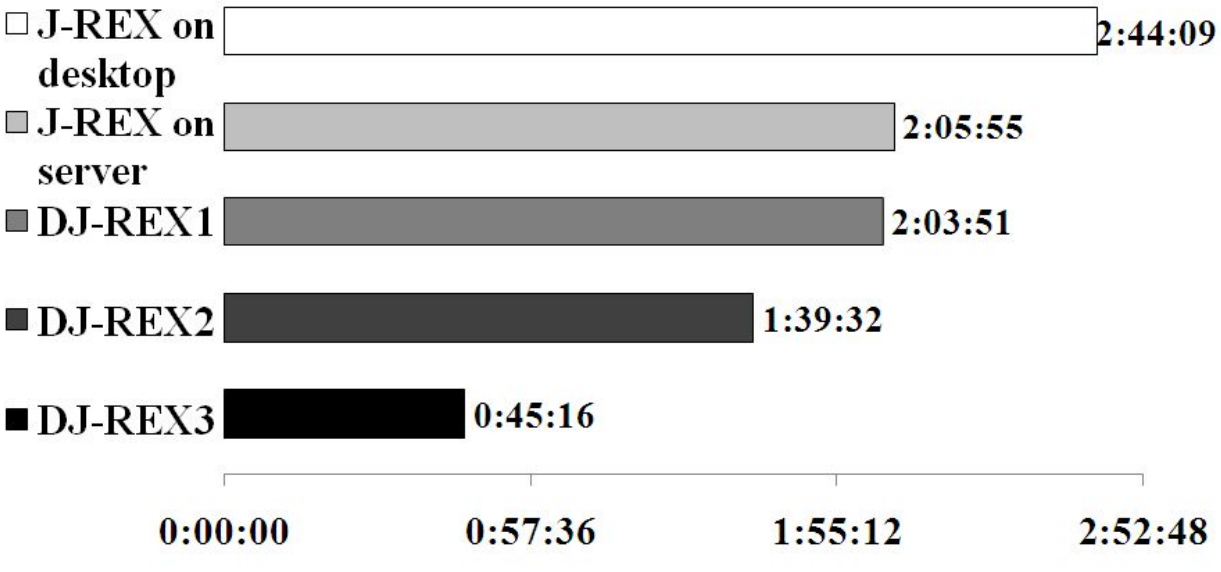
Johannes Härtel - University of Koblenz - johanneshaertel@uni-koblenz.de

References

- [ShangJAH09] Shang, Weiyi, Zhen Ming Jiang, Bram Adams, and Ahmed E. Hassan. "MapReduce as a general framework to support research in Mining Software Repositories (MSR)." In 2009 6th IEEE International Working Conference on Mining Software Repositories, pp. 21-30. IEEE, 2009.
- [DyerNRN13] Dyer, R., Nguyen, H. A., Rajan, H., & Nguyen, T. N. (2013, May). Boa: A language and infrastructure for analyzing ultra-large-scale software repositories. In 2013 35th International Conference on Software Engineering (ICSE) (pp. 422-431). IEEE.
- [StutzBC10] Stutz, Philip, Abraham Bernstein, and William Cohen. "Signal/collect: graph algorithms for the (semantic) web." In International Semantic Web Conference, pp. 764-780. Springer, Berlin, Heidelberg, 2010.
- [AlexandruG15] Alexandru, Carol V., and Harald C. Gall. "Rapid multi-purpose, multi-commit code analysis." In 2015 IEEE/ACM 37th IEEE International Conference on Software Engineering, vol. 2, pp. 635-638. IEEE, 2015.
- [SeiferHLLS19] Seifer, Philipp, Härtel, Johannes, Leinberger, Martin, Lämmel, Ralf and Staab, Steffen (2019) Empirical study on the usage of graph query languages in open source Java projects. 12th ACM SIGPLAN International Conference on Software Language Engineering (SLE 2019), Greece. 20 - 25 Oct 2019. 15 pp .

DJ-Rex Solution: Migration to Distributed Map-Reduce

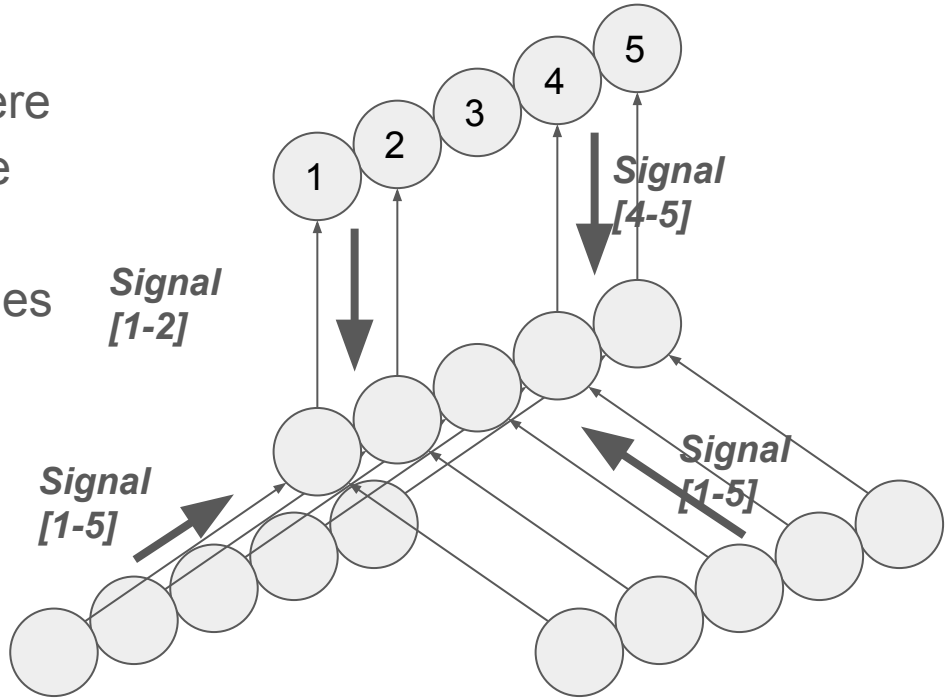
Successively adding Hardware leads to runtime improvements (DJ-REX1-3).



[ShangJAH09]

LISA Solution:
Reduction of Redundancies

LISA uses Signal/Collect [StutzBC10] to express queries, i.e., an actor system where asynchronous messages are sent until the system converges. LISA creates a multi-revision AST and just sends messages once for successive but equal version ranges.



Boa Solution: Domain Specific Languages

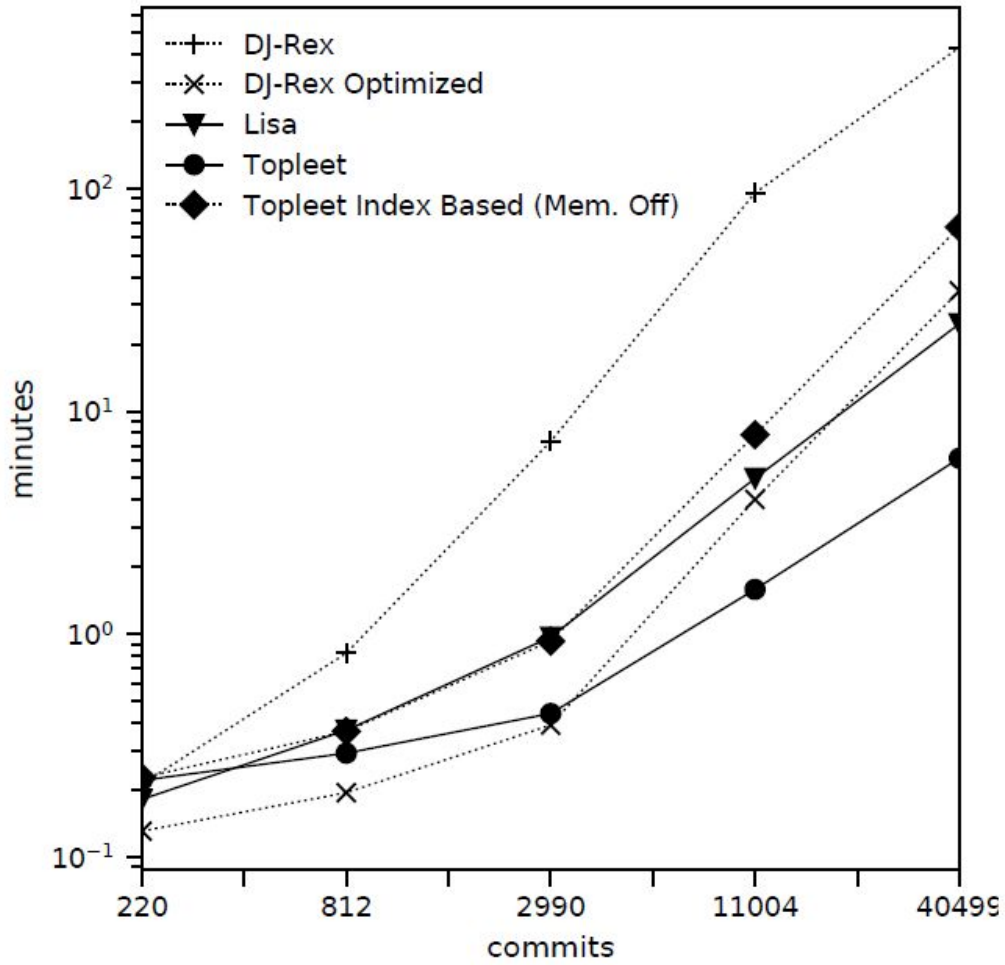
Analysis is written in a DSL and compiled to Map-Reduce. The right except answers: “How many fixing revisions added null checks?” (copy form the documentation). It contains handling of changed resources.

```
before node: ChangedFile -> {
    # if this is a fixing revision and
    # there was a previous version of the file
    if (isfixing && haskey(files, node.name)) {
        # count how many null checks were previously in the file
        count = 0;
        visit(getast(files[node.name]));
        last := count;

        # count how many null checks are currently in the file
        count = 0;
        visit(getast(node));

        # if there are more null checks, output
        if (count > last)
            AddedNullCheck << 1;
    }
    if (node.change == ChangeKind.DELETED)
        remove(files, node.name);
    else
        files[node.name] = node;
    stop;
```


[DyerNRN13]

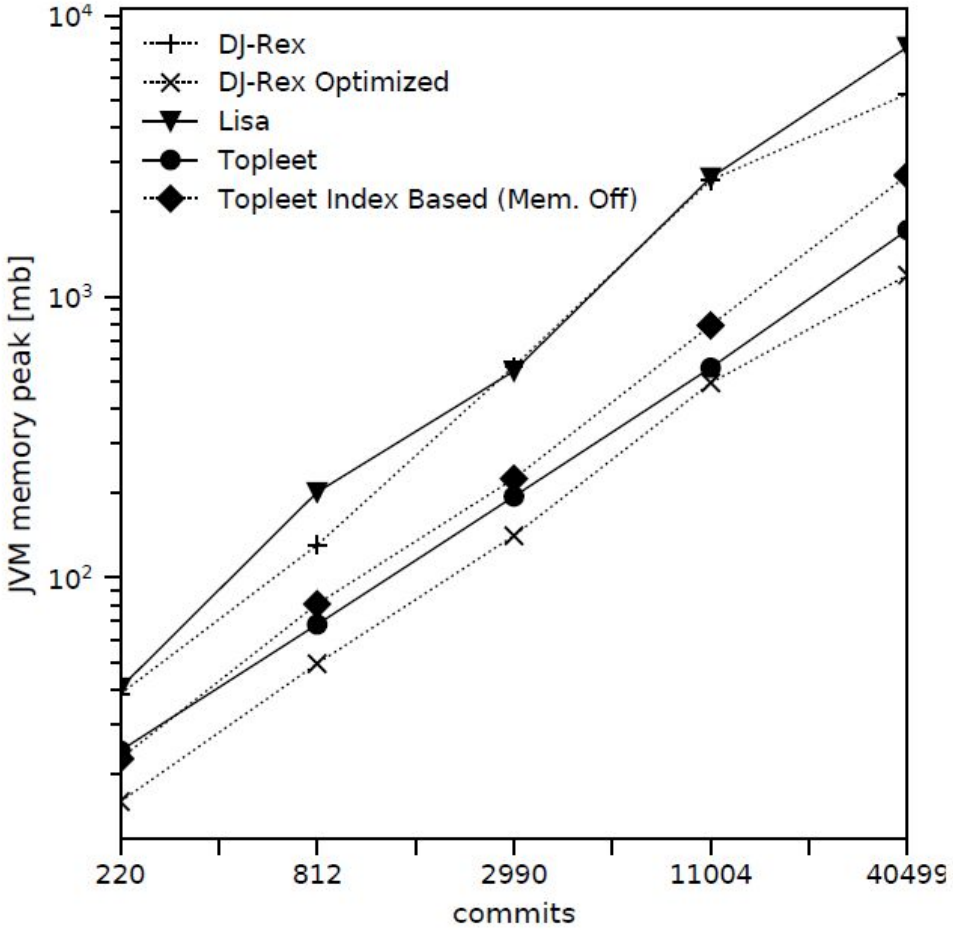
Time vs. Commits

Running the cyclomatic complexity solutions on 98 repositories grouped by commit count.



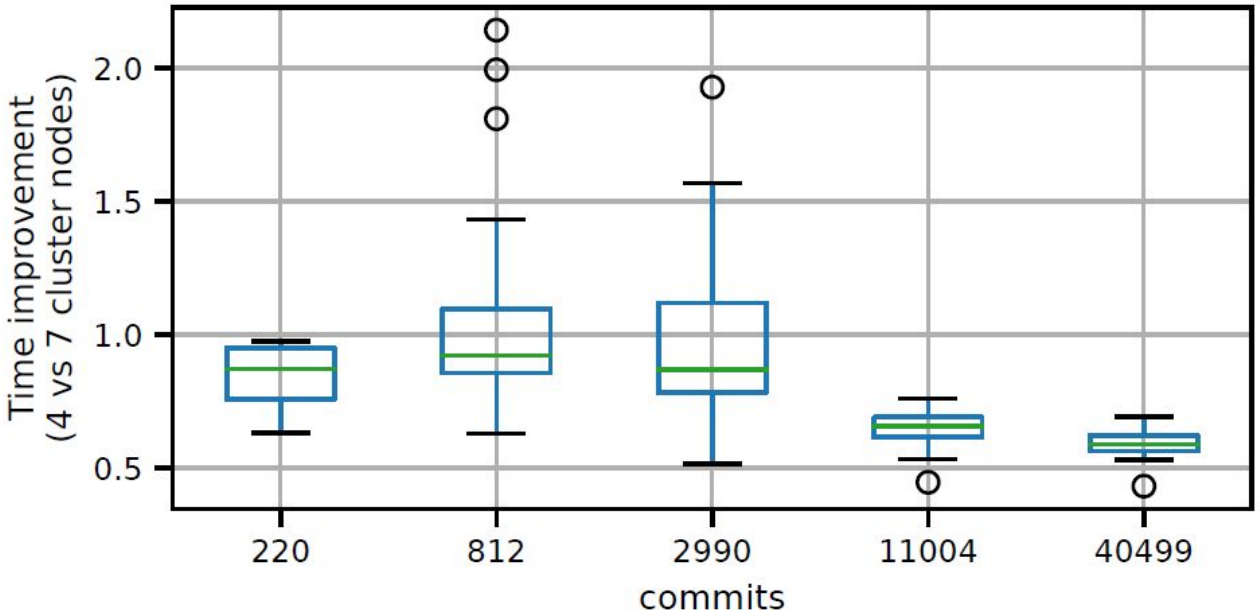
Memory vs. Commits

Running the cyclomatic complexity solutions on 98 repositories grouped by commit count.



Distribution vs. Commits

Running Topleet cyclomatic complexity in distribution on all 98 repositories grouped by commit count.



Johannes Härtel - University of Koblenz - johanneshaertel@uni-koblenz.de

Demo Backup

Johannes Härtel - University of Koblenz - johanneshaertel@uni-koblenz.de

Demo

```
9  object Sandbox {
10
11   // The foreign function computing MCC on a resource.
12   def computeMCC(resource: Resource): Int = {...}
70
71   // Library imports.
72   import org.topleet.libs.Natives._
73   import org.topleet.libs.Gits._
74
75   implicit val engine: Engine = IncrementalParallelEngine.create()
76
77   def main(args: Array[String]): Unit = {
78     |
79   }
80 }
81
```

Johannes Härtel - University of Koblenz - johanneshaertel@uni-koblenz.de

Demo

```
9  object Sandbox {
10
11   // The foreign function computing MCC on a resource.
12   def computeMCC(resource: Resource): Int = {...}
70
71   // Library imports.
72   import org.toplee
73   import org.topl
74
75   implicit val en
76
77   def main(args:
78     |
79   }
80 }
81
```

```
val parser = ASTParser.newParser(AST.JLS13)
parser.setSource(resource.read().toCharArray)
parser.setKind(ASTParser.K_COMPILATION_UNIT)

val cu = parser.createAST( monitor = null).asInstanceOf[CompilationUnit]

var result = 0
cu.accept(new ASTVisitor() {
  override def visit(node: WhileStatement): Boolean = {
    result = result + 1
    super.visit(node)
  }
...
```

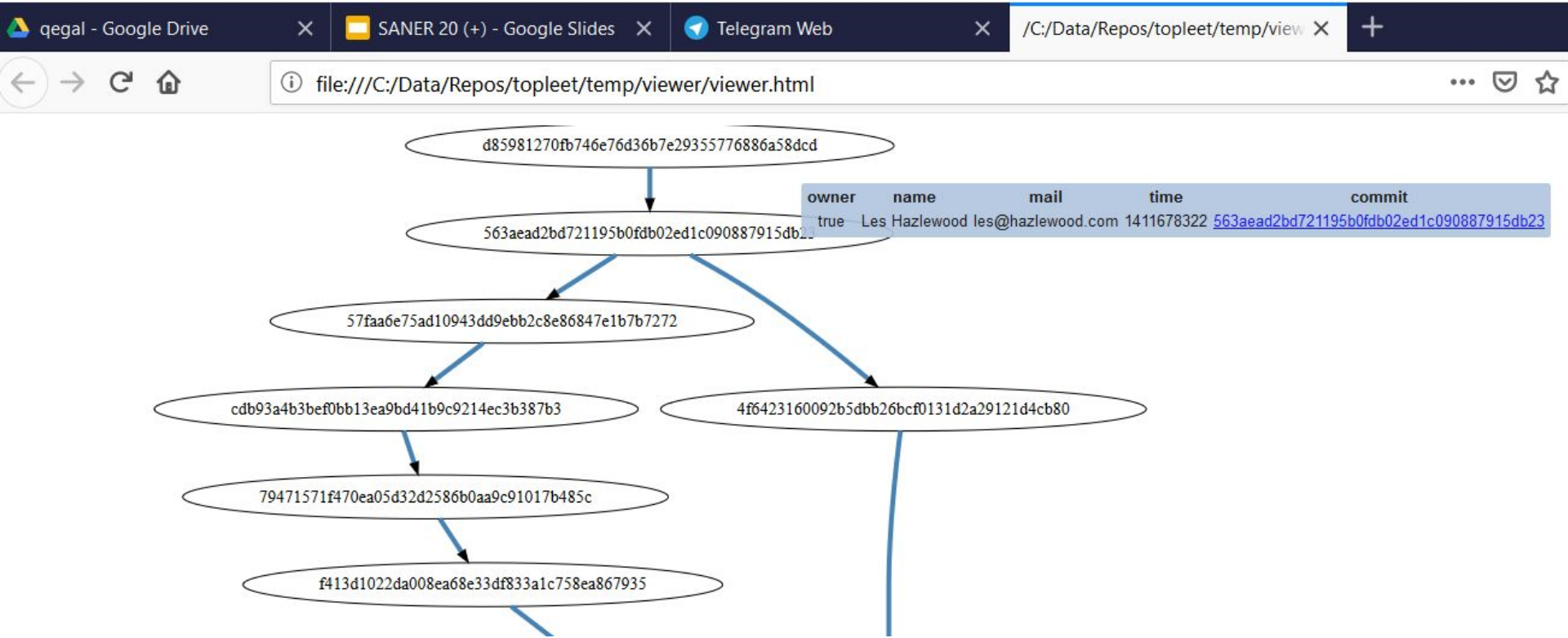
Demo

```
9  object Sandbox {
10
11   // The foreign function computing MCC on a resource.
12   def computeMCC(resource: Resource): Int = {...}
70
71   // Library imports.
72   import org.topleet.libs.Natives._
73   import org.topleet.libs.Gits._
74
75   implicit val engine: Engine = IncrementalParallelEngine.create()
76
77   def main(args: Array[String]): Unit = {
78     val shas = git( address = "jwtk/jjwt")
79   }
80 }
81
```

Demo

```
9  object Sandbox {
10
11   // The foreign function computing MCC on a resource.
12   def computeMCC(resource: Resource): Int = {...}
70
71   // Library imports.
72   import org.topleet.libs.Natives._
73   import org.topleet.libs.Gits._
74
75   implicit val engine: Engine = IncrementalParallelEngine.create()
76
77   def main(args: Array[String]): Unit = {
78     val shas = git( address = "jwtk/jjwt")
79
80     shas.show()
81   }
82 }
83
```

Demo


qegal - Google Drive
SANER 20 (+) - Google Slides
Telegram Web
/C:/Data/Repos/topleet/temp/viev
file:///C:/Data/Repos/topleet/temp/viewer/viewer.html
d85981270fb746e76d36b7e29355776886a58dcd
owner name mail time commit
true Les Hazlewood les@hazlewood.com 1411678322 563aead2bd721195b0fdb02ed1c090887915db23
563aead2bd721195b0fdb02ed1c090887915db23
57faa6e75ad10943dd9ebb2c8e86847e1b7b7272
cdb93a4b3bef0bb13ea9bd41b9c9214ec3b387b3
4f6423160092b5dbb26bcf0131d2a29121d4cb80
79471571f470ea05d32d2586b0aa9c91017b485c
f413d1022da008ea68e33df833a1c758ea867935

Demo

```
9  object Sandbox {
10
11   // The foreign function computing MCC on a resource.
12   def computeMCC(resource: Resource): Int = {...}
70
71   // Library imports.
72   import org.topleet.libs.Natives._
73   import org.topleet.libs.Gits._
74
75   implicit val engine: Engine = IncrementalParallelEngine.create()
76
77   def main(args: Array[String]): Unit = {
78     val shas = git( address = "jwtk/jjwt")
79   }
80 }
81
```

Demo

```
9  object Sandbox {
10
11   // The foreign function computing MCC on a resource.
12   def computeMCC(resource: Resource): Int = {...}
70
71   // Library imports.
72   import org.topleet.libs.Natives._
73   import org.topleet.libs.Gits._
74
75   implicit val engine: Engine = IncrementalParallelEngine.create()
76
77   def main(args: Array[String]): Unit = {
78     val shas: Leet[SHA, Single[SHA]] = git( address = "jwtk/jjwt")
79
80   }
81 }
82
```

Demo

```
9  object Sandbox {
10
11   // The foreign function computing MCC on a resource.
12   def computeMCC(resource: Resource): Int = {...}
70
71   // Library imports.
72   import org.topleet.libs.Natives._
73   import org.topleet.libs.Gits._
74
75   implicit val engine: Engine = IncrementalParallelEngine.create()
76
77   def main(args: Array[String]): Unit = {
78     val shas: Leet[SHA, Single[SHA]] = git( address = "jwtk/jjwt")
79
80     shas.res
81 m resources()        Leet[SHA, Natives.Bag[(Gits.Path, Resource)]]
82
83
84   Press Ctrl+. to choose the selected (or first) suggestion and insert a dot afterwards  Next Tip
```

Demo

```
9  object Sandbox {
10
11   // The foreign function computing MCC on a resource.
12   def computeMCC(resource: Resource): Int = {...}
70
71   // Library imports.
72   import org.topleet.libs.Natives._
73   import org.topleet.libs.Gits._
74
75   implicit val engine: Engine = IncrementalParallelEngine.create()
76
77   def main(args: Array[String]): Unit = {
78     val shas: Leet[SHA, Single[SHA]] = git( address = "jwtk/jjwt")
79
80     val resources: Leet[SHA, Bag[(Path, Resource)]] = shas.resources()
81
82   }
83 }
```

Demo

```
9  object Sandbox {
10
11   // The foreign function computing MCC on a resource.
12   def computeMCC(resource: Resource): Int = {...}
70
71   // Library imports.
72   import org.topleet.libs.Natives._
73   import org.topleet.libs.Gits._
74
75   implicit val engine: Engine = IncrementalParallelEngine.create()
76
77   def main(args: Array[String]): Unit = {
78     val shas: Leet[SHA, Single[SHA]] = git( address = "jwtk/jjwt")
79
80     val resources: Leet[SHA, Bag[(Path, Resource)]] = shas.resources()
81
82     val mcCabe: Leet[SHA, Bag[Int]] = resources
83       .filter { case (path, _) => path.endsWith(".java") }
84       .map { case (_, resource) => computeMCC(resource) }
85
86   }
87 }
88
```

Johannes Härtel - University of Koblenz - johanneshaertel@uni-koblenz.de

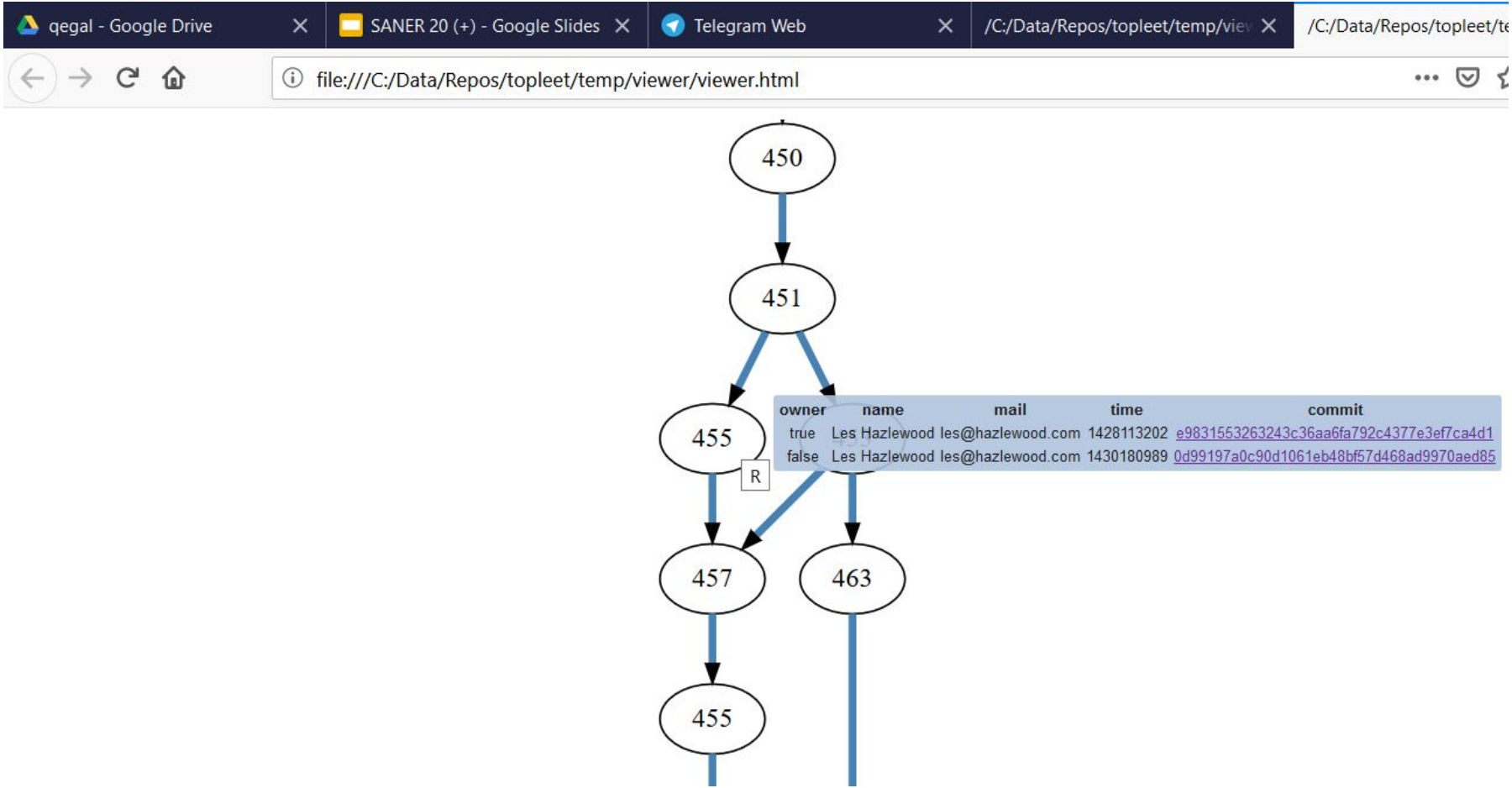
Demo

```
9  object Sandbox {
10
11   // The foreign function computing MCC on a resource.
12   def computeMCC(resource: Resource): Int = {...}
70
71   // Library imports.
72   import org.topleet.libs.Natives._
73   import org.topleet.libs.Gits._
74
75   implicit val engine: Engine = IncrementalParallelEngine.create()
76
77   def main(args: Array[String]): Unit = {
78     val shas: Leet[SHA, Single[SHA]] = git( address = "jwtk/jjwt")
79
80     val resources: Leet[SHA, Bag[(Path, Resource)]] = shas.resources()
81
82     val mcCabe: Leet[SHA, Bag[Int]] = resources
83       .filter { case (path, _) => path.endsWith(".java") }
84       .map { case (_, resource) => computeMCC(resource) }
85
86     val output: Leet[SHA, Single[Int]] = mcCabe.sum()
87
88   }
```

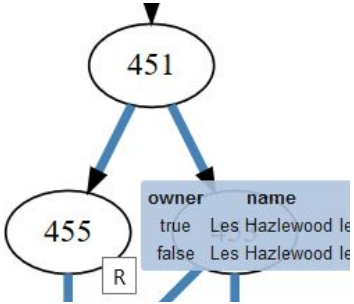
Demo

```
object Sandbox {

  // The foreign function computing MCC on a resource.
  def computeMCC(resource: Resource): Int = {...}

  // Library imports.
  import org.topleet.libs.Natives._
  import org.topleet.libs.Gits._

  implicit val engine: Engine = IncrementalParallelEngine.create()

  def main(args: Array[String]): Unit = {
    val shas: Leet[SHA, Single[SHA]] = git( address = "jwtk/jjwt")

    val resources: Leet[SHA, Bag[(Path, Resource)]] = shas.resources()

    val mcCabe: Leet[SHA, Bag[Int]] = resources
      .filter { case (path, _) => path.endsWith(".java") }
      .map { case (_, resource) => computeMCC(resource) }

    val output: Leet[SHA, Single[Int]] = mcCabe.sum()

    output.show()
  }
}
```

Demo


qegal - Google Drive
SANER 20 (+) - Google Slides
Telegram Web
/C:/Data/Repos/topleet/temp/viev
/C:/Data/Repos/topleet/te
file:///C:/Data/Repos/topleet/temp/viewer/viewer.html
450
451
455
R
457
463
455
owner name mail time commit
true Les Hazlewood les@hazlewood.com 1428113202 e9831553263243c36aa6fa792c4377e3ef7ca4d1
false Les Hazlewood les@hazlewood.com 1430180989 0d99197a0c90d1061eb48bf57d468ad9970aed85

Demo




```
24      -       return DatatypeConverter.printBase64Binary(data);
        27  +
        28  +       if (STANDARD_JVM) {
        29  +           return javax.xml.bind.DatatypeConverter.printBase64Binary(data);
        30  +       }
        31  +
        32  +       if (ANDROID) {
        33  +           int flags = android.util.Base64.NO_PADDING | android.util.Base64.NO_WRAP;
        34  +           return android.util.Base64.encodeToString(data, flags);
        35  +       }
        36  +
        37  +       throw new IllegalStateException("Unable to locate a Base64 codec for the current JVM");
25      38      }
26      39
27      40      @Override
28      41      public byte[] decode(String encoded) {
29      -       return DatatypeConverter.parseBase64Binary(encoded);
        42  +
        43  +       if (STANDARD_JVM) {
        44  +           return javax.xml.bind.DatatypeConverter.parseBase64Binary(encoded);
        45  +       }
        46  +
        47  +       if (ANDROID) {
        48  +           return android.util.Base64.decode(encoded, android.util.Base64.DEFAULT);
        49  +       }
        50  +
        51  +       throw new IllegalStateException("Unable to locate a Base64 codec for the current JVM");
30      52      }
31      53
32      54  }
```

Demo

Many Resources on every Commit ✓

```
object Sandbox {

  // The foreign function computing MCC on a resource.
  def computeMCC(resource: Resource): Int = {...}

  // Library imports.
  import org.topleet.libs.Natives._
  import org.topleet.libs.Gits._

  implicit val engine: Engine = IncrementalParallelEngine.create()

  def main(args: Array[String]): Unit = {
    val shas: Leet[SHA, Single[SHA]] = git( address = "jwtk/jjwt")

    val resources: Leet[SHA, Bag[(Path, Resource)]] = shas.resources()

    val mcCabe: Leet[SHA, Bag[Int]] = resources
      .filter { case (path, _) => path.endsWith(".java") }
      .map { case (_, resource) => computeMCC(resource) }

    val output: Leet[SHA, Single[Int]] = mcCabe.sum()

    output.show()
  }
}
```