Evaluation of Version Control Merge Tools

Benedikt Schesch
ETH Zürich

Ryan Featherman
Microsoft

Kenneth J. Yang
University of Washington

Ben R. Roberts
University of Washington

Michael D. Ernst
University of Washington

Merging Algorithms make mistakes

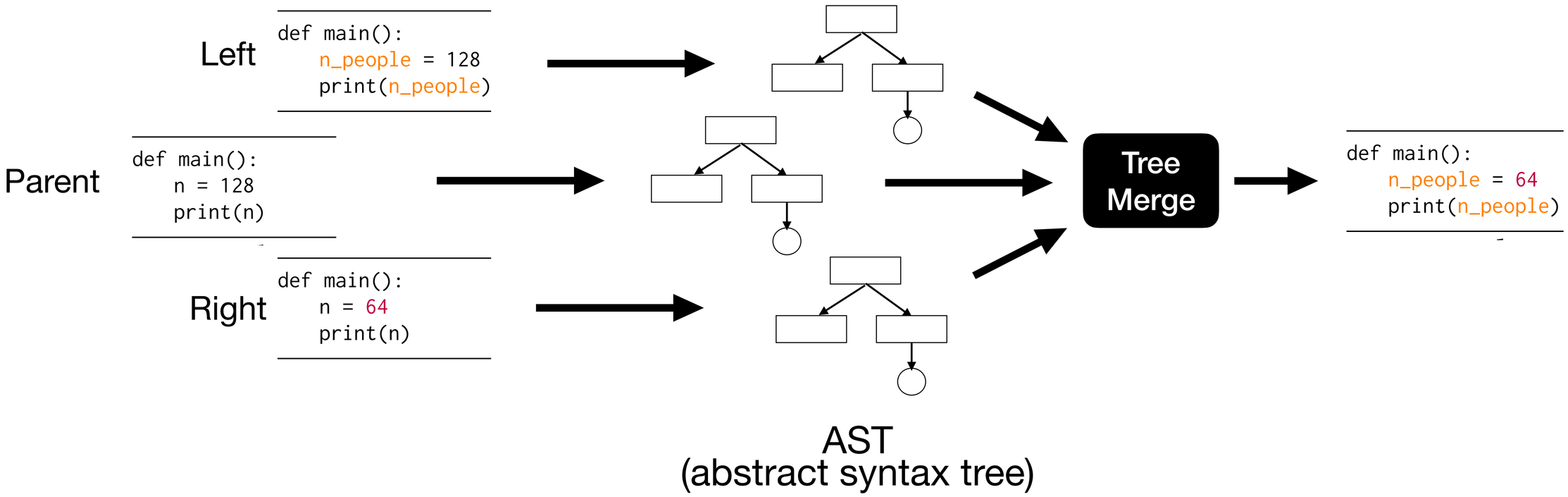
Too conservative algorithms

Incorrect Merges

Git Merge outputs a conflict

AST merge can **solve** this

AST Merging (Spork, IntelliMerge)



Is AST Merging any good?

No one knows!

Problems with previous evaluations:

1. Few tool comparisons
2. Cost of incorrect merges
3. Unrepresentative merges
   - Cherry-picked examples
   - Main branch only

Problem 1: Which merge tools?

Compared 16 merge tools

Do complex merging algorithms outperform simple heuristics?

- Imports: Resolve Java *import* conflict
  - Spork and IntelliMerge do this
- Version-Numbers: Resolve version numbers conflicts
- I+Vn: Combines Imports and Version-Numbers

Problem 2: Incorrect merges

Old Methodology

	Conflict	Merged	
		Correctly	Incorrectly
Git Merge	51%	46%	3%
Spork [1]	35%	54%	11%
IntelliMerge [2]	26%	24%	50%

Worst

Which is best depends on #conflicts and #incorrect

Dataset: >6k merges with good test coverage

[1] Larsén et al. (2023). Spork: Structured merge for Java with formatting preservation. IEEE Trans. Softw. Eng., 49(01), 64–83.

[2] Shen et al. (2019). IntelliMerge: A refactoring-aware software merging technique. In OOPSLA 2019 (pp. 170:1–170:28). Athens, Greece.

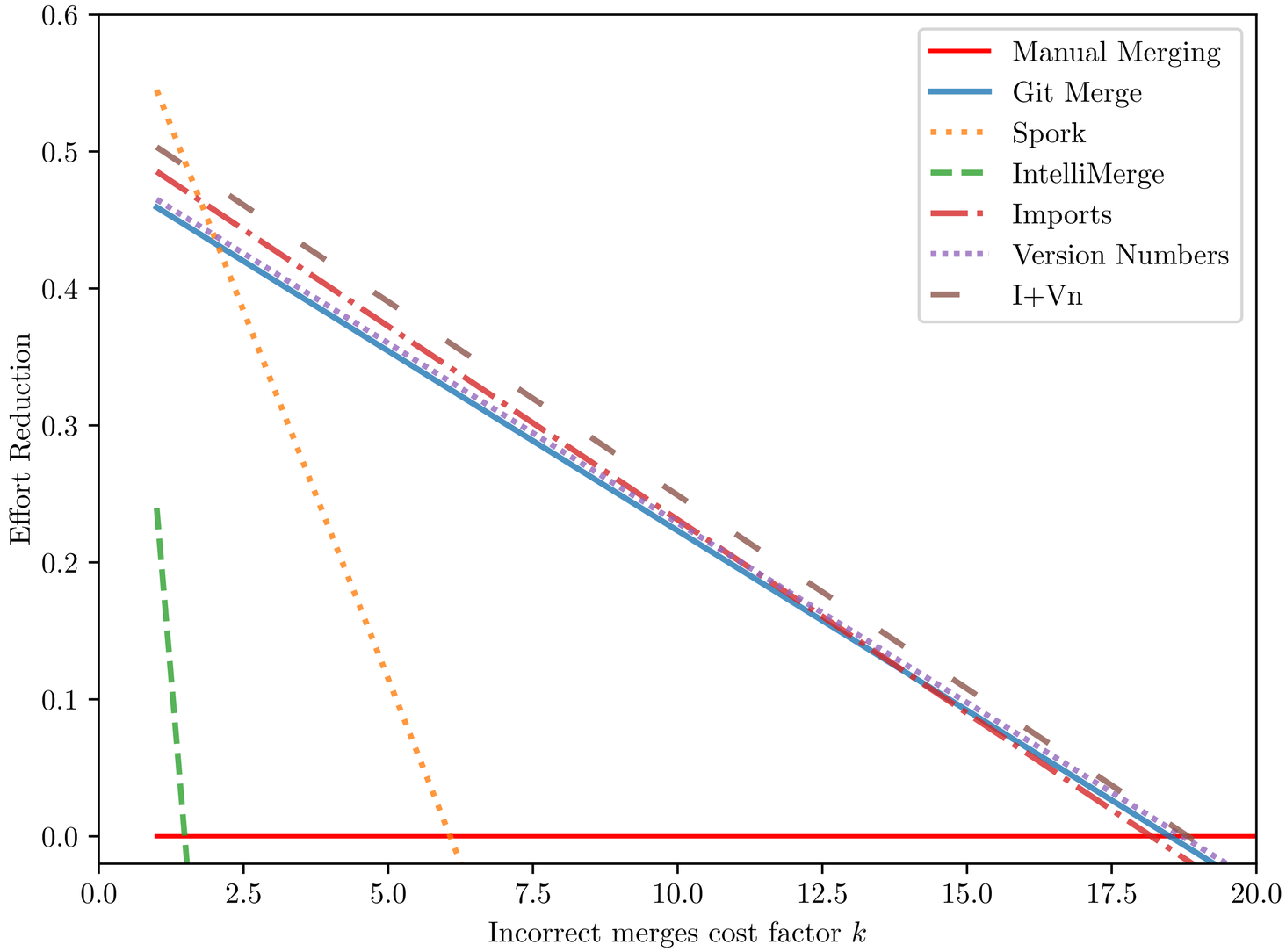
Which merge algorithm is the best?

$$k = \frac{\text{Cost of incorrect merge}}{\text{Cost of unhandled merge}}$$

$$Cost(T) = numUnhandled(T) + numIncorrect(T) \times k$$

$$EffortReduction(T) = \frac{Cost(Manual) - Cost(T)}{Cost(Manual)}$$

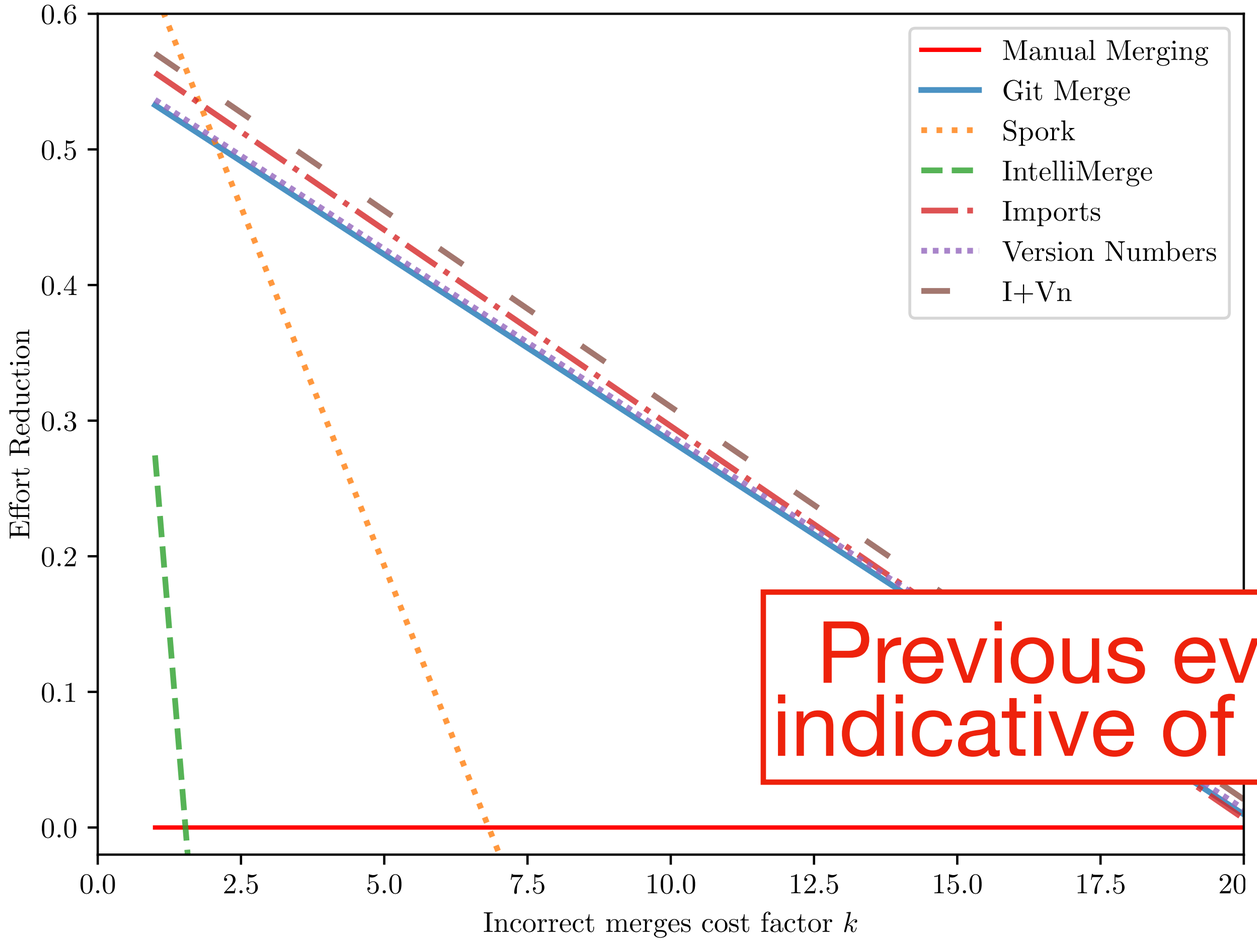
Effort Reduction Results

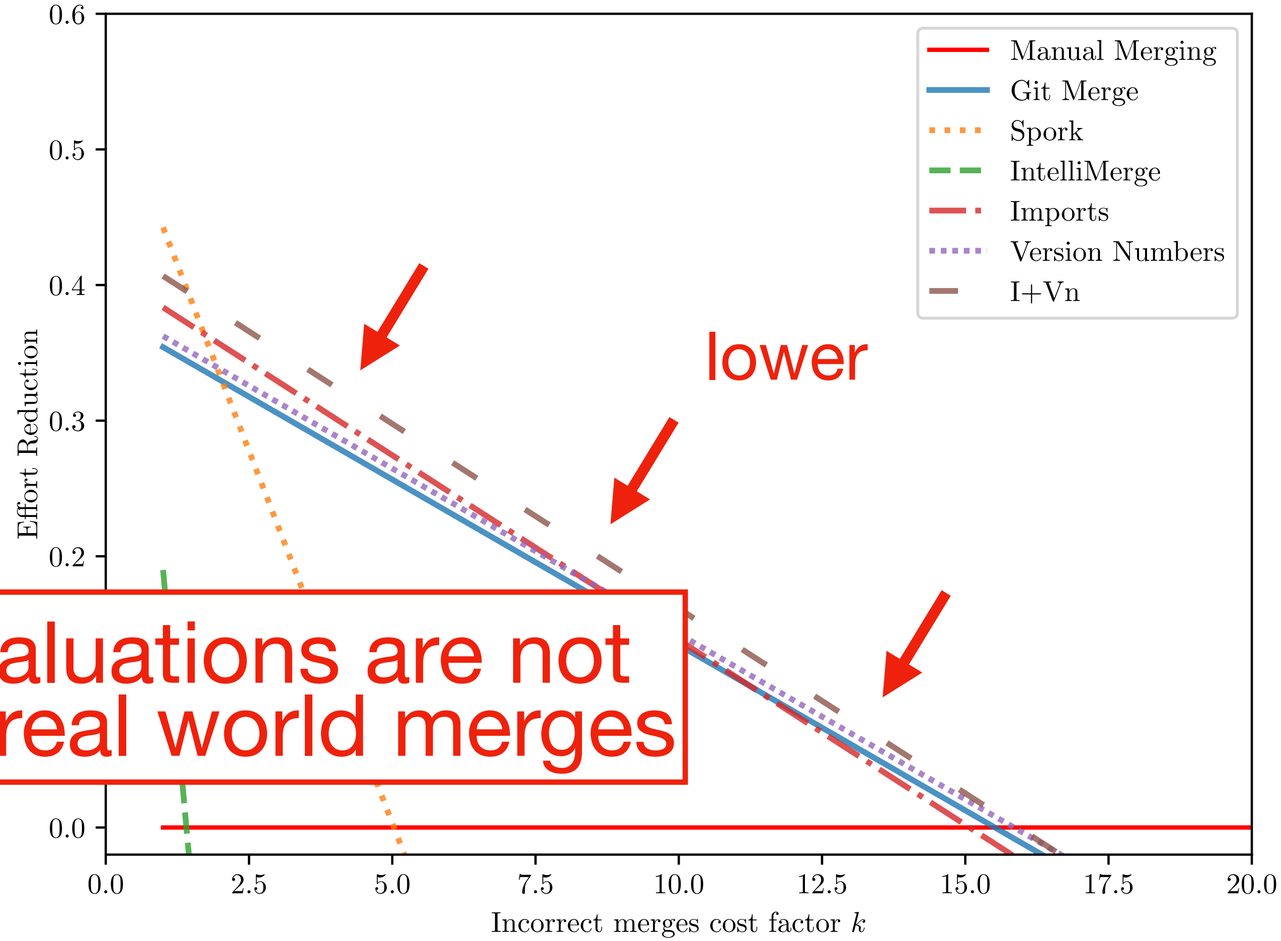


Problem 3: Features branches are harder

Main branches

Feature branches
(not in Git history)





Previous evaluations are not indicative of real world merges

Contributions

1. New methodology corrects 3 problems with previous merge evaluations:
   - Many merge tools
   - Cost of incorrect merges (changes the ranking)
   - Feature branch merges

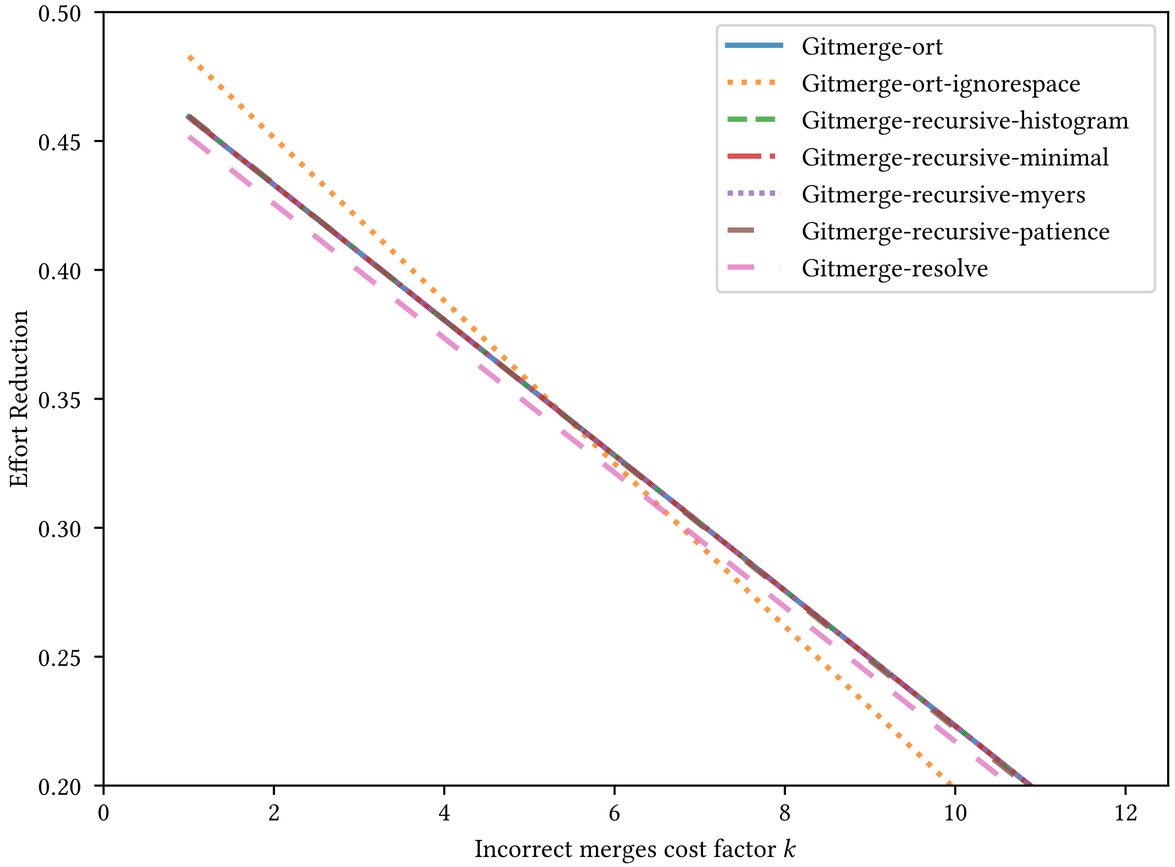
2. Created new tools that outperform all others.

3. Simple tools outperform complex ones.
   The research community should reward effectiveness.

4. Our merge tools, experimental scripts, and data are public.

Git Evaluation Results

Tool	Merges					
	Correct		Unhandled		Incorrect	
	#	%	#	%	#	%
Gitmerge-ort	2748	46%	3078	51%	157	3%
Gitmerge-ort-ignorespace	2889	48%	2905	49%	189	3%
Gitmerge-recursive-histogram	2748	46%	3078	51%	157	3%
Gitmerge-recursive-minimal	2748	46%	3078	51%	157	3%
Gitmerge-recursive-myers	2748	46%	3078	51%	157	3%
Gitmerge-recursive-patience	2751	46%	3074	51%	158	3%
Gitmerge-resolve	2703	45%	3124	52%	156	3%



Merge tools we did not evaluate

- Many other tools only have only a GUI interface, unsuitable for automated testing
- DeepMerge and MergeBERT are not publicly available making it impossible to evaluate
- JDime discards comments, file headers and arbitrarily reorders methods and fields. Often requires more than 15 minutes for a merge.

General notes on merge tools

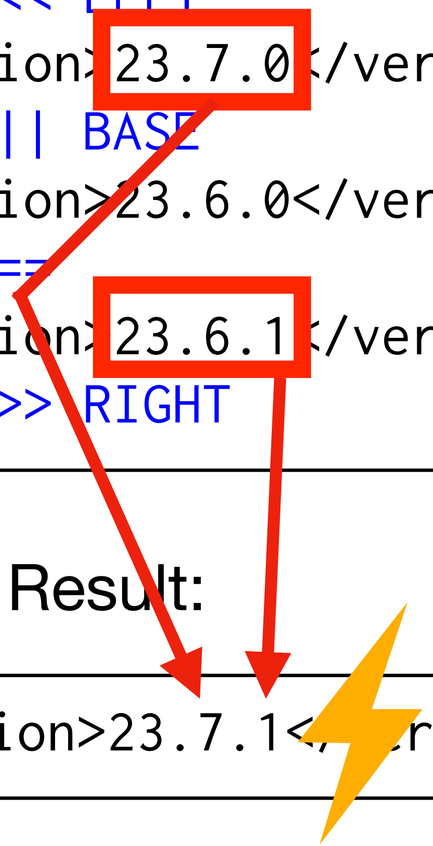
- Sporks implementation is buggy (in discussion with maintainers to fix it), underlying algorithm might still be competitive
- Intellimerge compared itself to git but only on a few examples where git failed and refactoring was the main problem. Implementation is also buggy and non deterministic.
- Many other tools only have only a GUI interface, unsuitable for automated testing
- DeepMerge and MergeBERT are not publicly available making it impossible to evaluate

Character Merge Case Analysis

```
<<<<<<< LEFT
<version>23.7.0</version>
||||||| BASE
<version>23.6.0</version>
=======
<version>23.6.1</version>
>>>>>>> RIGHT
```

Merge Result:

```
<version>23.7.1</version>
```



Adjacent Case Analysis

```
<<<<<<< LEFT
String comments = SourcesHelper.readerToString(reader);
CompilationUnit cu = new JavaParser().setSource(comments).parse();
||||||| BASE
String comments = SourcesHelper.readerToString(reader);
CompilationUnit cu = new InstanceJavaParser(comments).parse();
=======
String comments = readerToString(reader);
CompilationUnit cu = new InstanceJavaParser(comments).parse();
>>>>>>> RIGHT
```

Adjacent Case Analysis

```
<<<<<<< LEFT
synchronized (cacheMap) {
    List<DNSEntry> entryList = cacheMap.get(dnsEntry.getKey());
    if (entryList != null) {
        entryList.remove(dnsEntry);
||||||| BASE
List<DNSEntry> entryList = this.get(dnsEntry.getKey());
if (entryList != null) {
    synchronized (entryList) {
        entryList.remove(dnsEntry);
=======
List<DNSEntry> entryList = this.get(dnsEntry.getKey());
if (entryList != null) {
    synchronized (entryList) {
        result = entryList.remove(dnsEntry);
>>>>>>> RIGHT
    }
}
/* Remove from DNS cache when no records remain with this key */
if (result && entryList.isEmpty()) {
    this.remove(dnsEntry.getKey());
```

Merge Result:

```
synchronized (cacheMap) {
    List<DNSEntry> entryList = cacheMap.get(dnsEntry.getKey());
    if (entryList != null) {
        result = entryList.remove(dnsEntry);
    }
}
/* Remove from DNS cache when no records remain with this key */
if (result && entryList.isEmpty()) {
    this.remove(dnsEntry.getKey());
}
```

Github

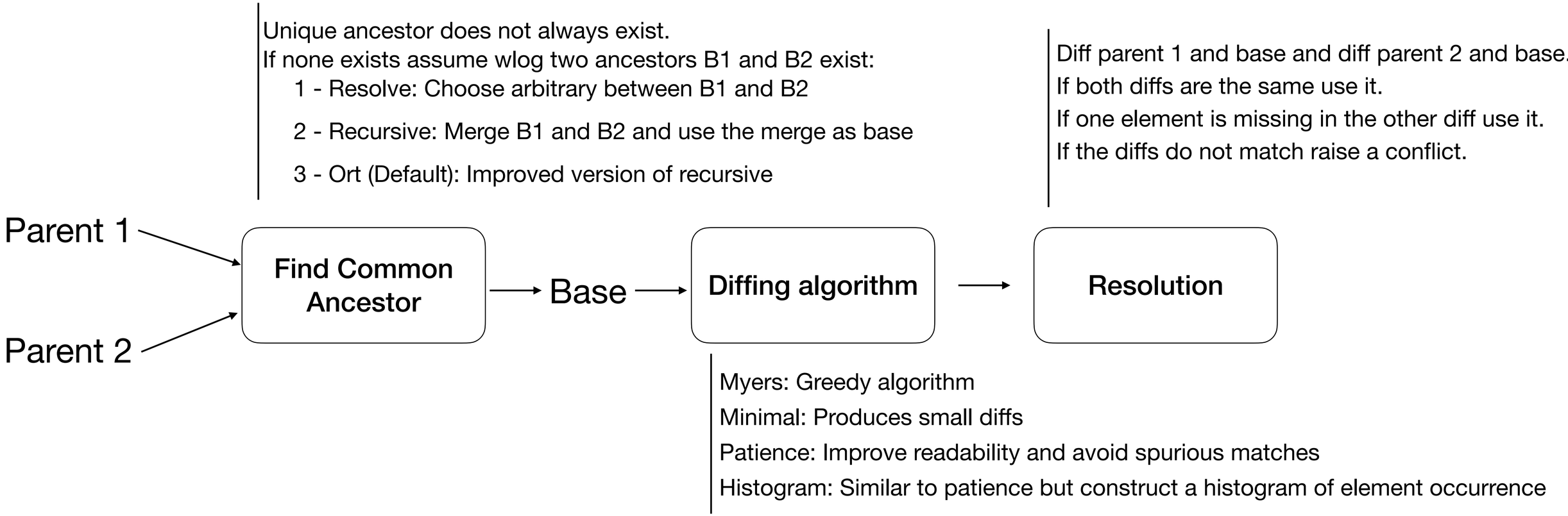
Testing infrastructure:

https://github.com/benedikt-schesch/AST-Merging-Evaluation

Merging Tool:

https://github.com/plume-lib/merging

Git Merging Algorithm



- Adjacent: Merge two adjacent lines as opposed to Git
- Hires merge uses exactly the same idea as git but operates at a character level instead of a line level

Motivation

Old Methodology

	Conflict	Merged
Git Merge	51%	49%
Spork [1]	35%	65%
IntelliMerge [2]	26%	74%

Best

Our Methodology

	Conflict	Merged	
		Correctly	Incorrectly
Git Merge	51%	46%	3%
Spork	35%	54%	11%
IntelliMerge	26%	24%	50%

Worst

[1] Larsén et al. (2023). Spork: Structured merge for Java with formatting preservation. IEEE Trans. Softw. Eng., 49(01), 64–83.

[2] Shen et al. (2019). IntelliMerge: A refactoring-aware software merging technique. In OOPSLA 2019 (pp. 170:1–170:28). Athens, Greece.