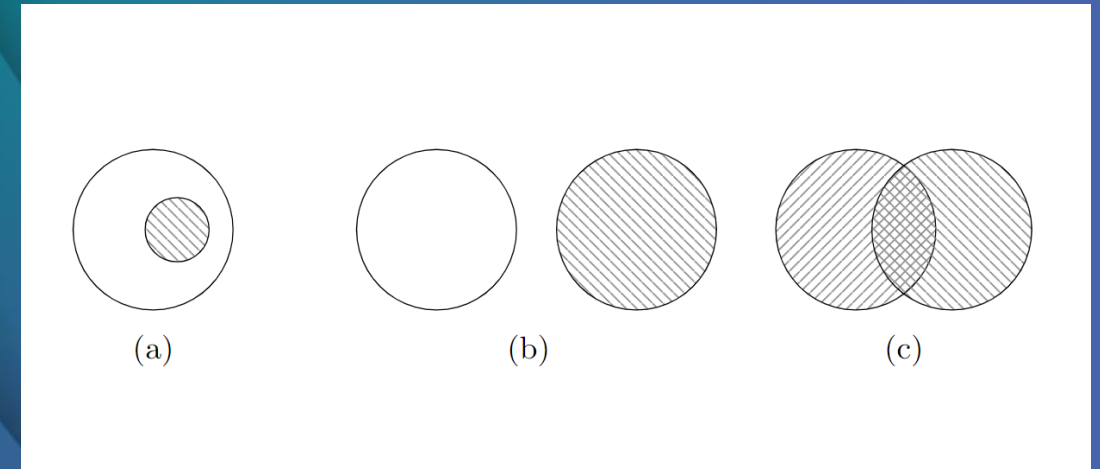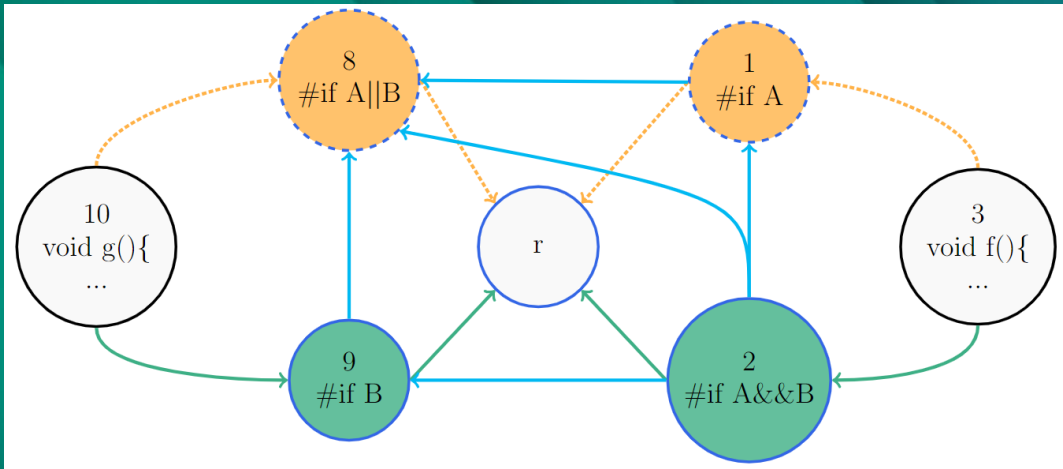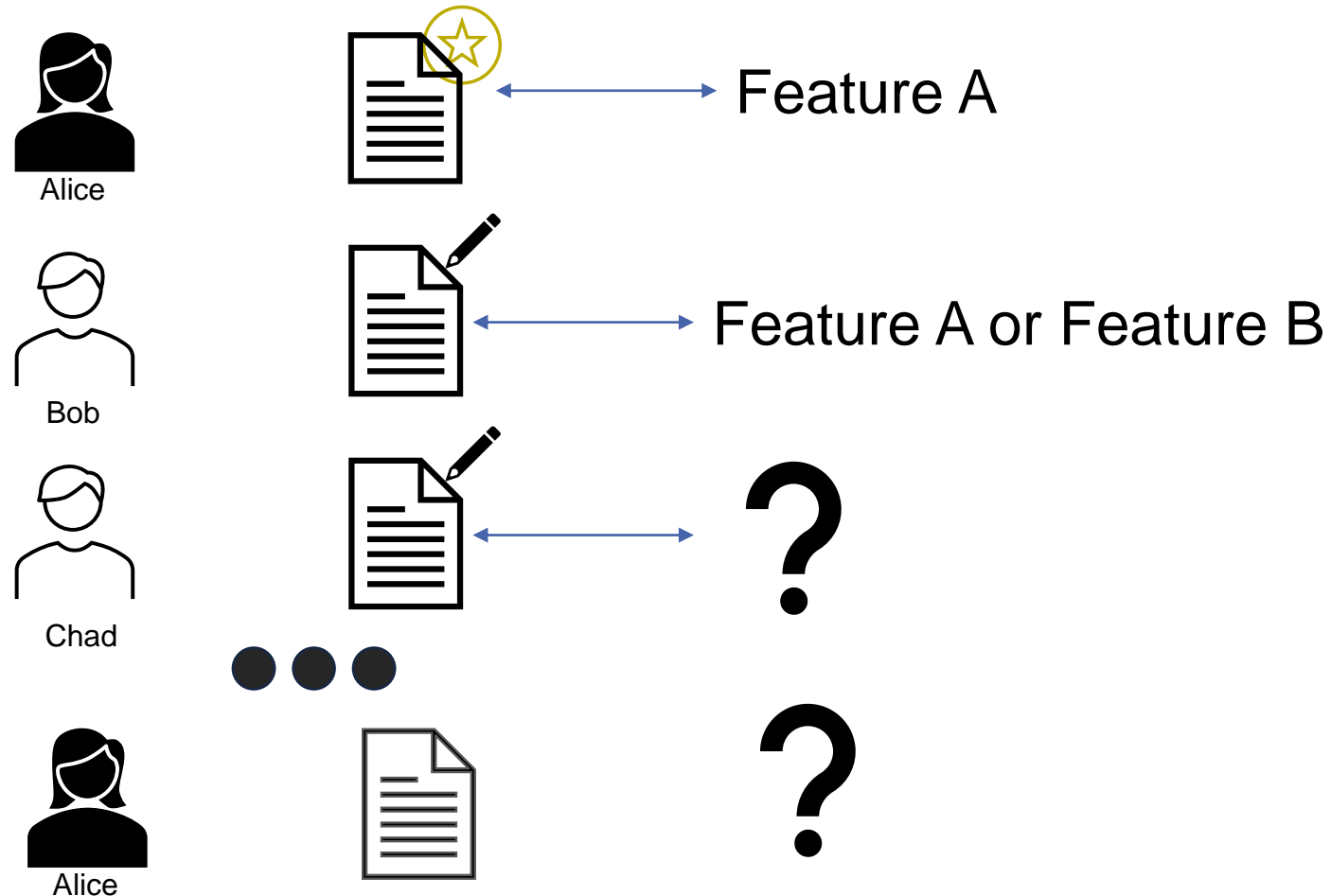# Explaining Edits to Variability Annotations in Evolving Software Product Lines

Lukas Güthing, Paul M. Bittner, Ina Schaefer, Thomas Thüm
VaMoS24

Motivation – Changing Variability Information

Alice — Feature A

Bob — Feature A or Feature B

Chad — ?

Alice — ?

Goal

- Use automated analyses
  - In CI/CD
  - For user-centered feedback

- To reduce
  - variability-induced bugs
  - variability misconfiguration

Variability Annotations

```
1  #ifdef A
2  void f(){
3     ...
4  }
5  #endif

6  #ifdef A || B
7  void g(){
8     ...
9  }
10 #endif
```
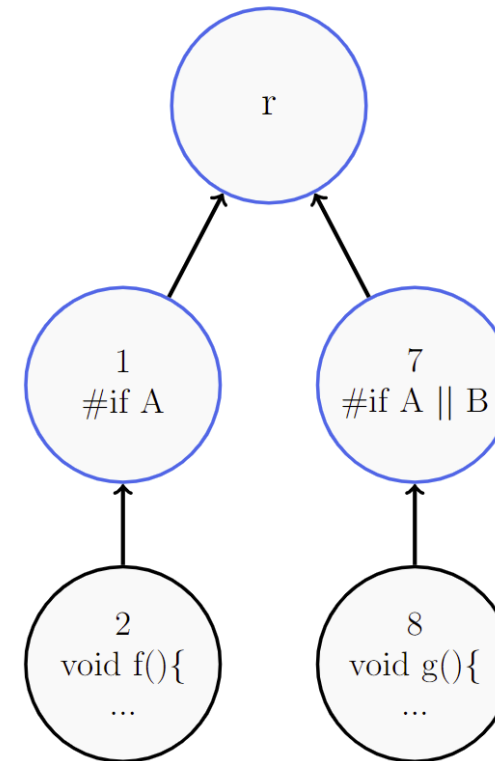
Variability Annotations – Variation Trees [Bittner22]


```
1  #ifdef A
2  void f(){
3      ...
4  }
5  #endif


6  #ifdef A || B
7  void g(){
8      ...
9  }
10 #endif
```
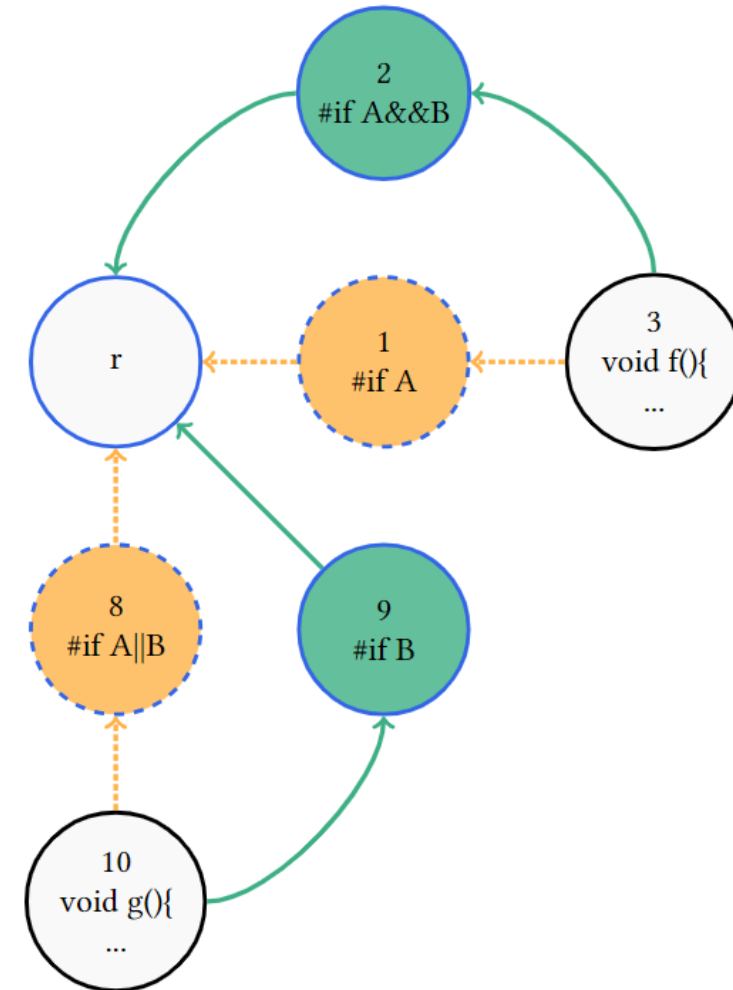
Edits to Variability Annotations

```
1  - #ifdef A
2  + #ifdef A && B
3  void f(){
4    ...
5  }
6  #endif


7  - #ifdef A || B
8  + #ifdef B
9  void g(){
10   ...
11 }
12 #endif
```
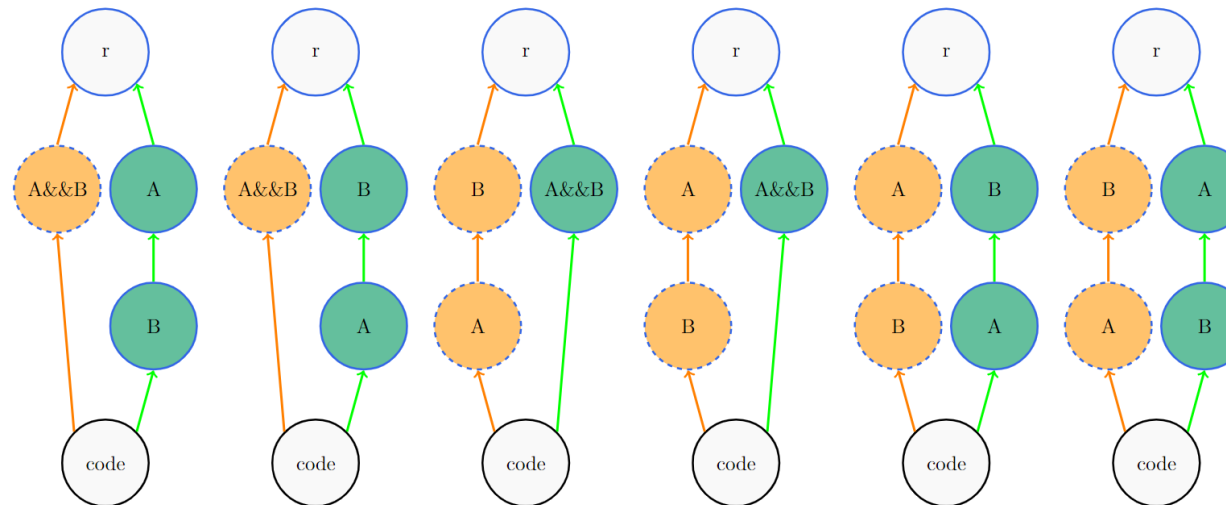
Edits – Variation Diffs [Bittner22]

February 8, 2024   VaMoS24 - Explaining Edits to Variability Annotations in Evolving Software Product Lines

Edit Classes [Bittner22]

$$Refactoring(c) := unchanged(c)$$
$$\wedge(\text{PC}_b(c) \models \text{PC}_a(c))$$
$$\wedge(\text{PC}_a(c) \models \text{PC}_b(c))$$
$$\wedge(path_b(c) \neq path_a(c))$$

```
-   #if A || (B && !A)
+   #if A || B
        c
    #endif
```

Edge-Typed Variation Diff


```
1  - #ifdef A
2  + #ifdef A && B
3  void f(){
4     ...
5  }
6  #endif


7  - #ifdef A || B
8  + #ifdef B
9  void g(){
10    ...
11 }
12 #endif
```
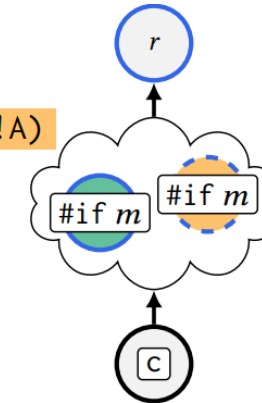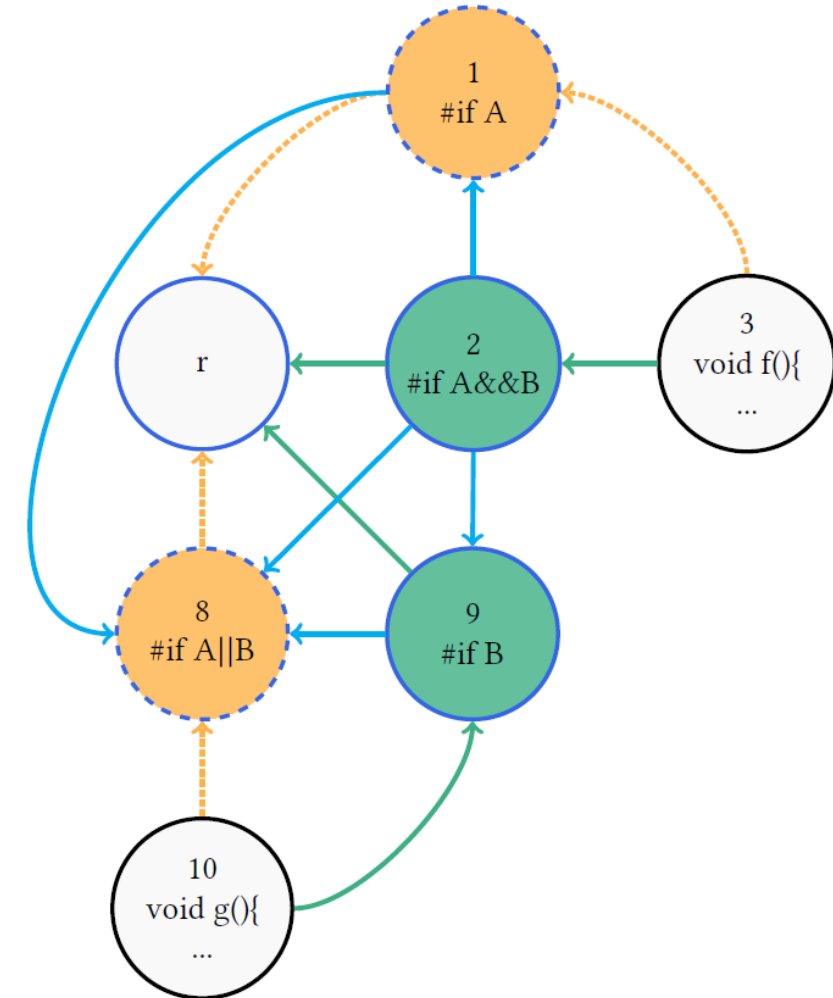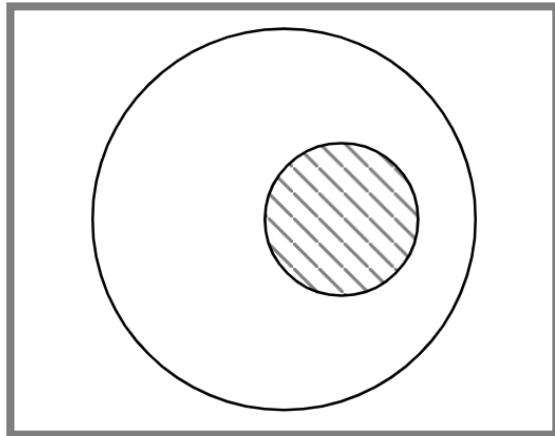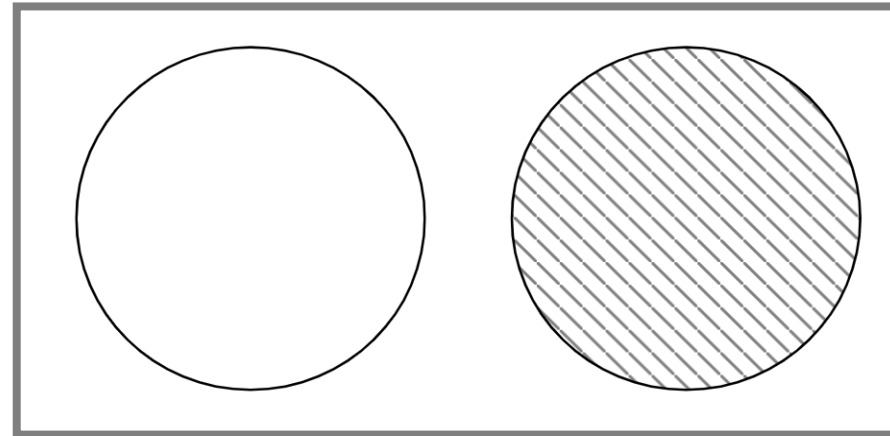
Feature Mapping Relations

Basic Set Relations:
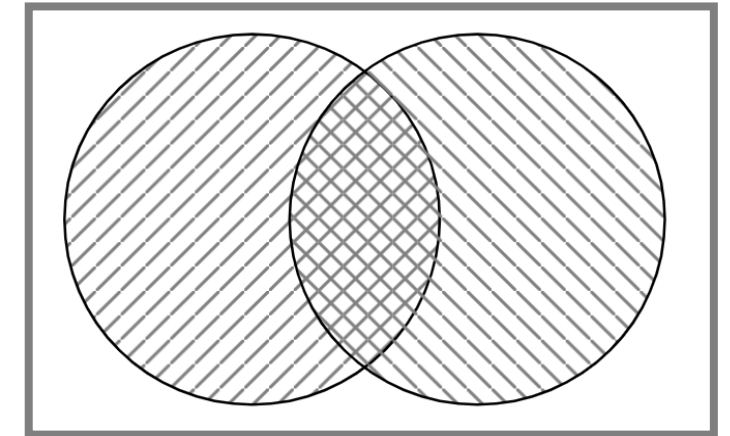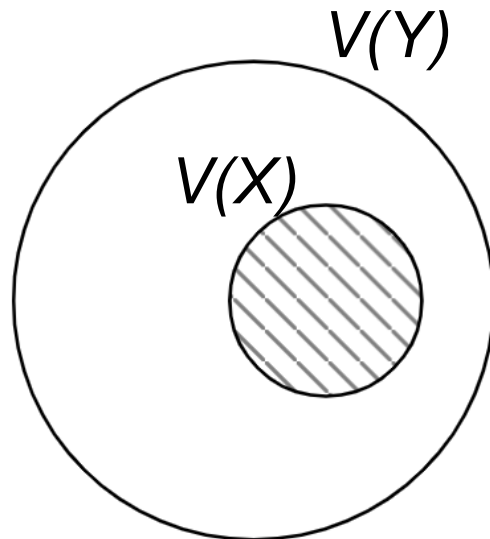

a) Subset   b) Disjoint   c) Intersected

Complete Set

The Subset Relation

V(Y)

V(X)

- Feature Mappings *X* and *Y* describe sets of variants
- *V(X)* is subset of *V(Y)*
  - $\rightarrow$ Code annotated with *X* also appears in every variant with *Y* present
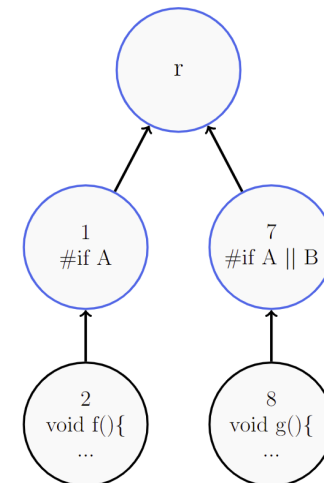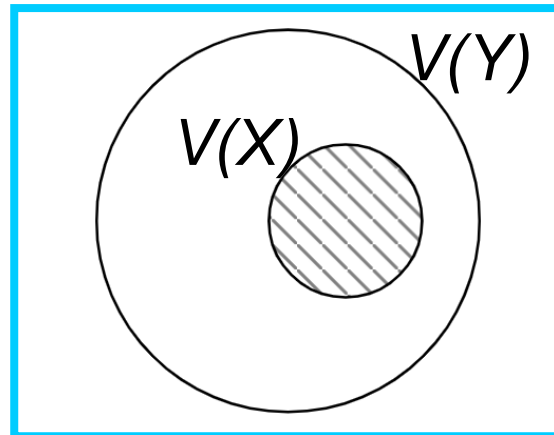- Example:

```
1  #ifdef A
2  void f(){
3    ...
4  }
5  #endif

6  #ifdef A || B
7  void g(){
8    ...
9  }
10 #endif
```
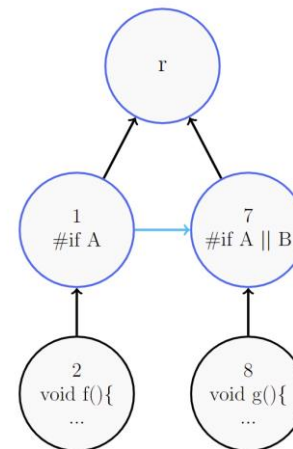
Implication Edges

■ Subset relation equal to SAT call:
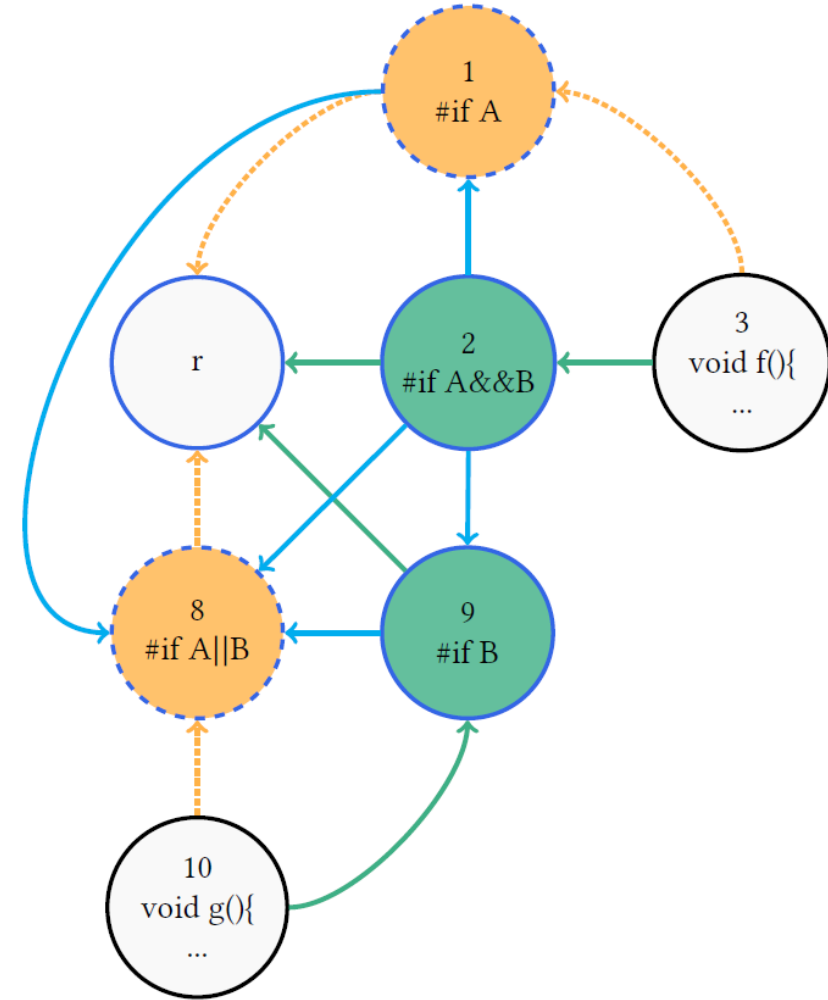   ■ $V(X) \subseteq V(Y)$

$$\leftrightarrow X \vDash Y$$
$$\leftrightarrow TAUT(X \rightarrow Y)$$
$$\leftrightarrow \neg SAT(X \wedge \neg Y)$$

■ $X = A$

■ $Y = A \vee B$

$\rightarrow V(X) \subseteq V(Y)$

Implication Edges
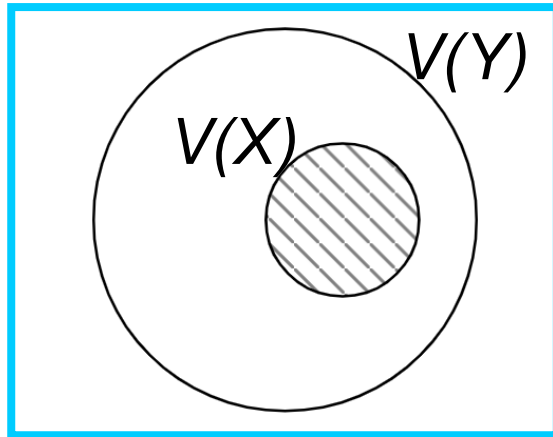
Alternative Edges

$$V(X) \cap V(Y) = \emptyset$$

$$\leftrightarrow X \vDash \neg Y$$
$$\leftrightarrow \neg SAT(X \wedge Y)$$
$$(\leftrightarrow Y \vDash \neg X)$$

V(X)　　　　　　　　　V(Y)

Research Questions

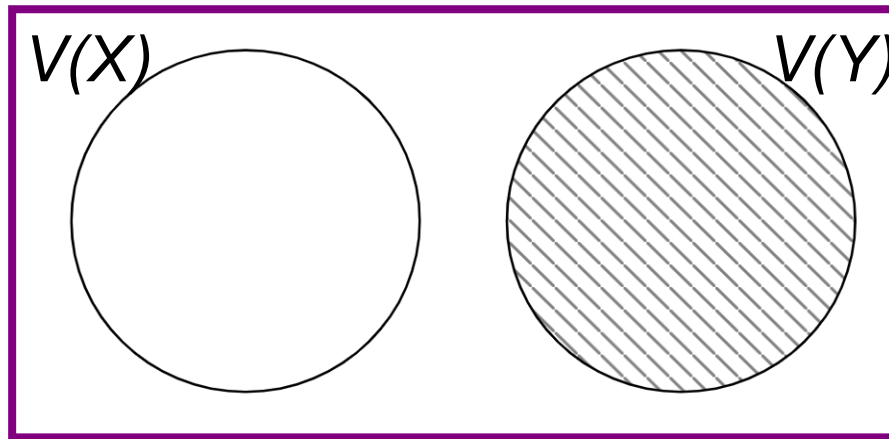- **RQ1: What is the runtime for identifying relations of feature annotations?**

- **RQ2: How many edge-typed variation diffs constructed from real software contain Implication or Alternative edges?**

- **RQ3: How often are edited annotations related in terms of implication or being alternative?**
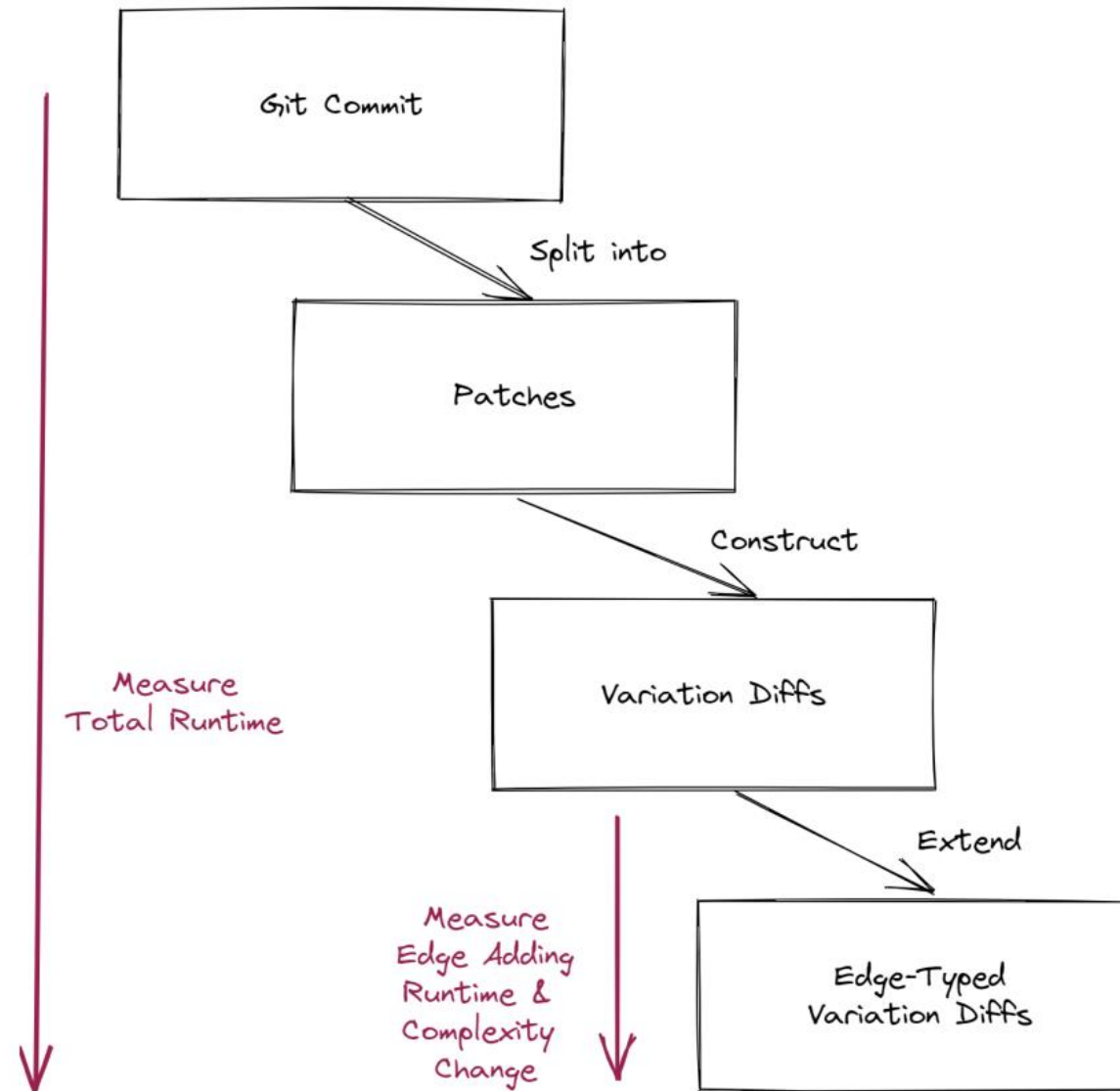
Implementation

- *DiffDetective:*
  - Variation Diff for
    - Every changed file in
    - Every commit of
    - Given Git repositories
- Extension:
  - Extends every VD to Edge-Typed VD
  - Record
    - Additional run time
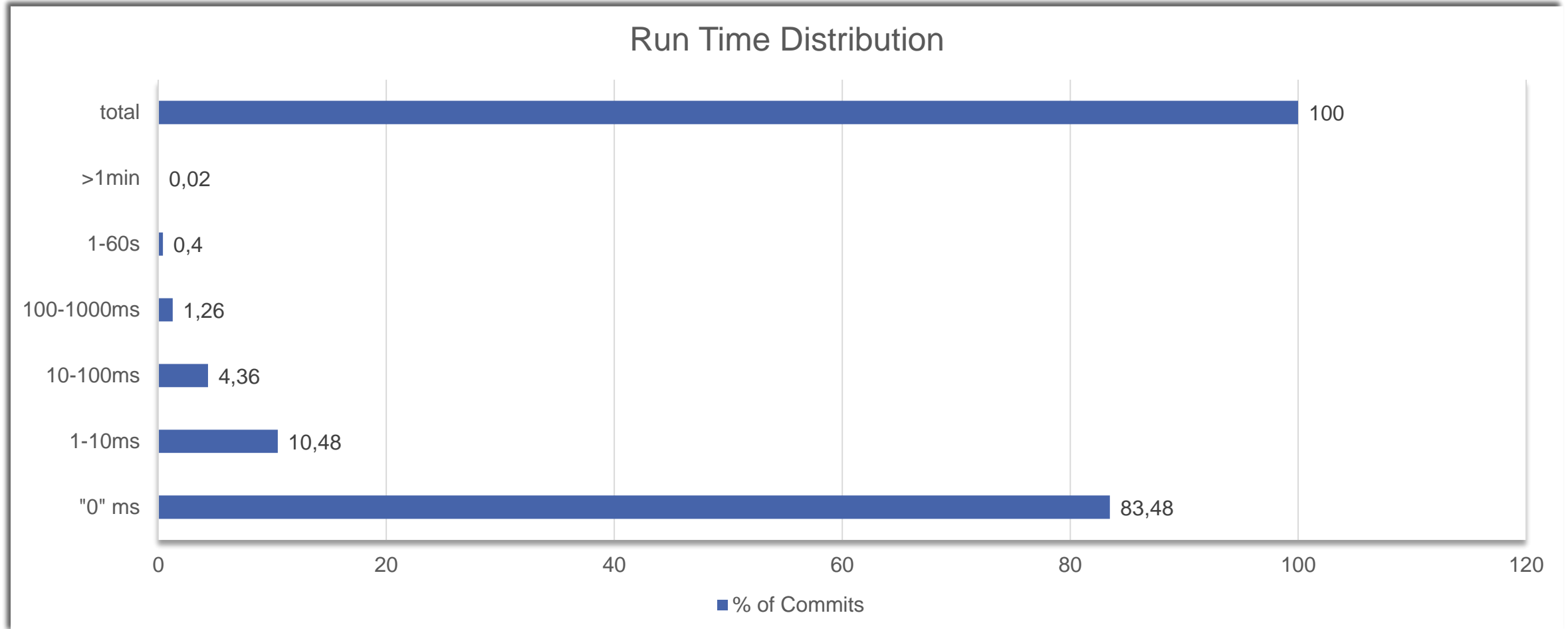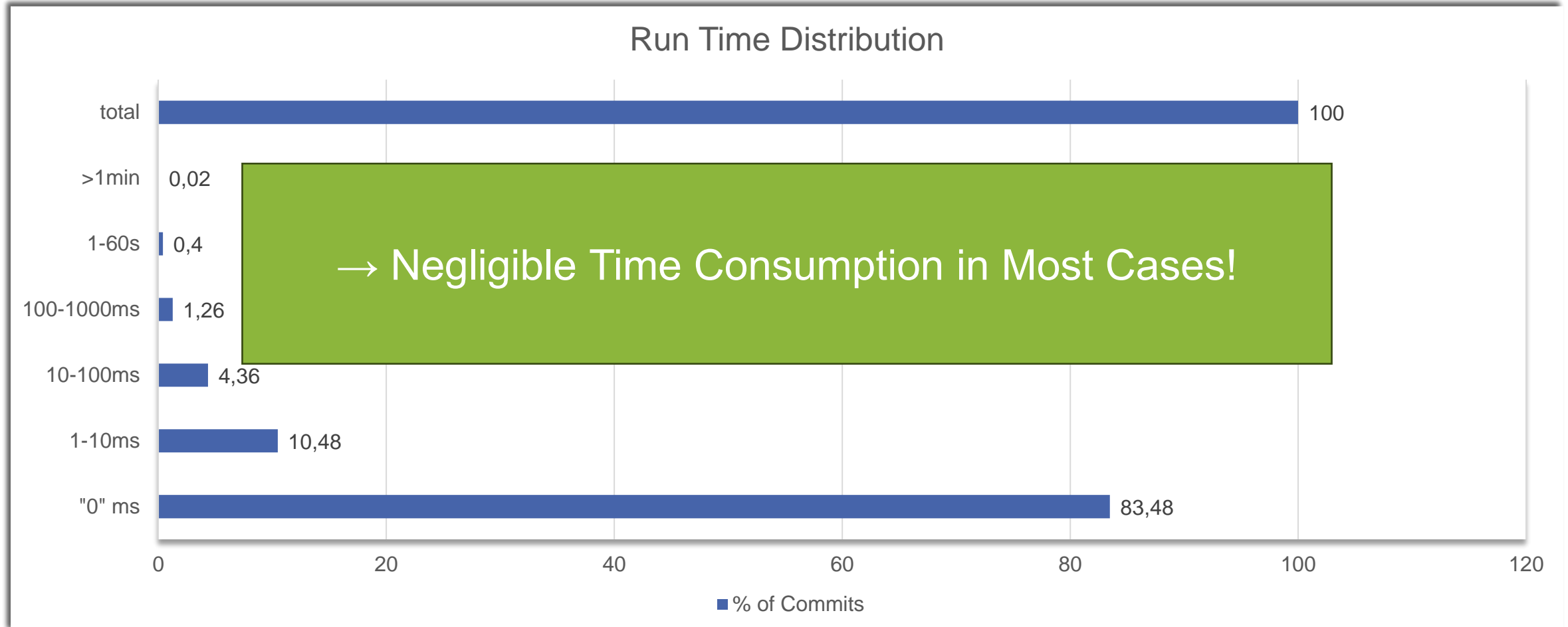   - Occurrence of Relationship Edges

Evaluation

Subject Systems

- 42 C preprocessor open-source systems
- Real-world systems
- Examples:
  - Busybox
  - Libssh
  - Freebsd
  - …
- Diverse domains

- 862,969 commits
- 3,117,797 patches

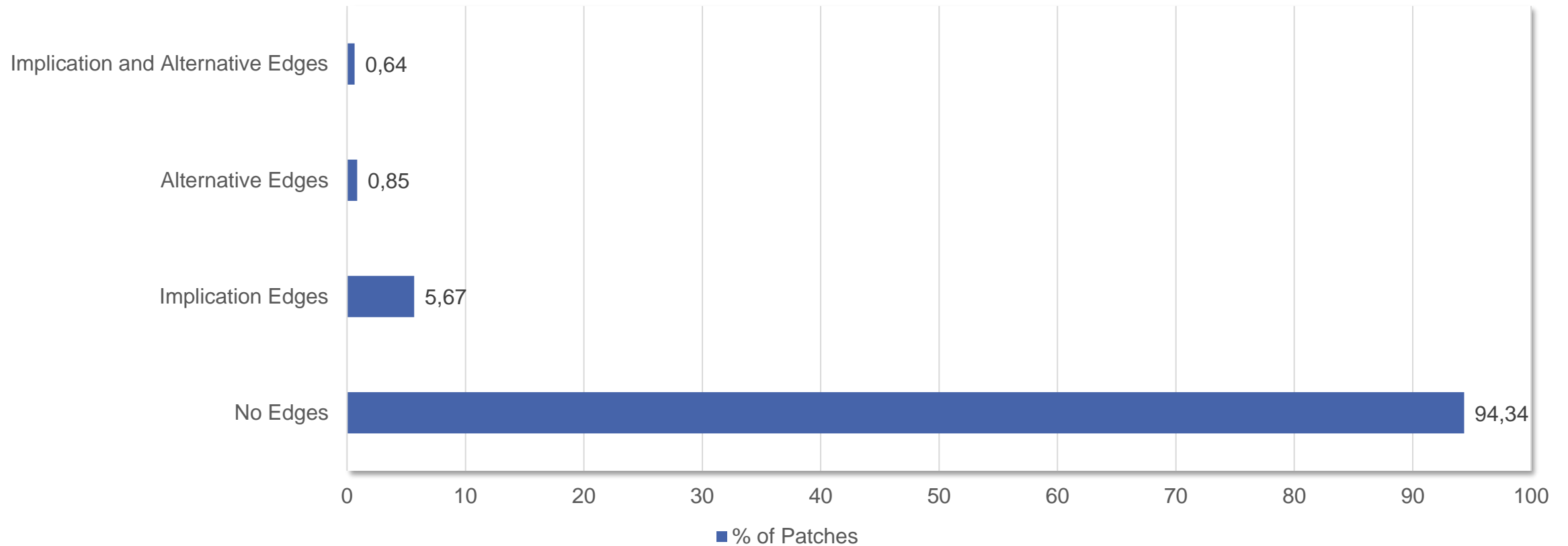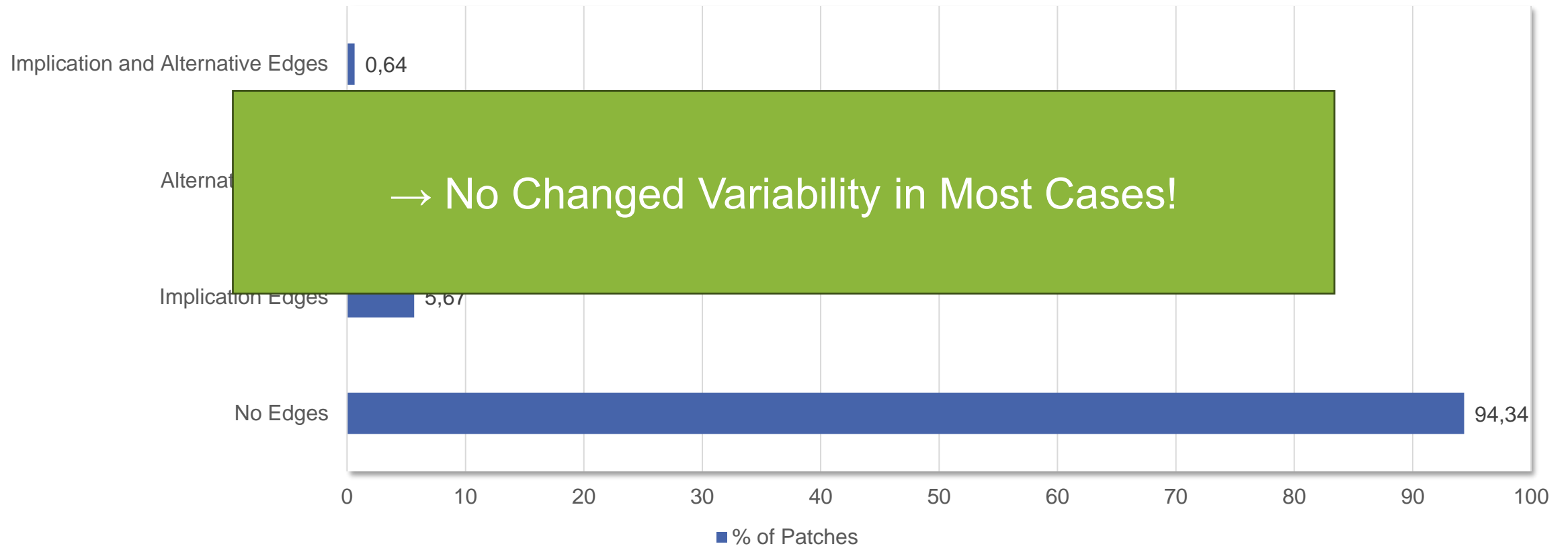| apache-httpd	web server	y	y	https://github.com/apache/httpd
berkeley-db-libdb	database system	y	y	https://github.com/berkeleydb/libdb
busybox	embedded systems	y	y	https://git.busybox.net/busybox
cherokee-webserver	web server	y	y	https://github.com/cherokee/webserver
clamav	antivirus program	y	y	https://github.com/Cisco-Talos/clamav
dia	diagramming software	y	y	https://github.com/GNOME/dia
emacs	text editor	y	y	https://github.com/emacs-mirror/emacs

Evaluation – RQ1

Run Time Distribution

Category	% of Commits
total	100
>1min	0,02
1-60s	0,4
100-1000ms	1,26
10-100ms	4,36
1-10ms	10,48
"0" ms	83,48

■ % of Commits

Evaluation – RQ1

Run Time Distribution

Category	% of Commits
total	100
>1min	0,02
1-60s	0,4
100-1000ms	1,26
10-100ms	4,36
1-10ms	10,48
"0" ms	83,48

→ Negligible Time Consumption in Most Cases!

Evaluation – RQ2

Relationship Edge Distributions

Category	% of Patches
Implication and Alternative Edges	0,64
Alternative Edges	0,85
Implication Edges	5,67
No Edges	94,34

■ % of Patches

Evaluation – RQ2

Relationship Edge Distributions

→ No Changed Variability in Most Cases!

- Implication and Alternative Edges: 0,64
- Implication Edges: 5,67
- No Edges: 94,34

■ % of Patches

Evaluation – RQ3

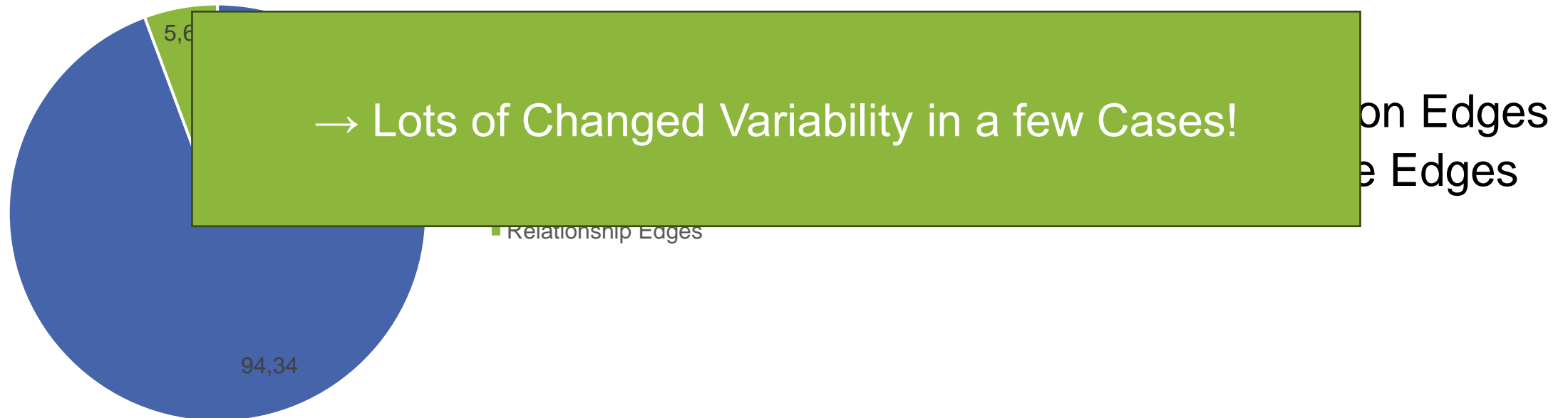**Edge-Typed Variation Diffs**

- No Relationship Edges
- Relationship Edges

5,66

94,34

Changed ETVDs:
- Avg. 354.24 Implication Edges
- Avg. 76.31 Alternative Edges
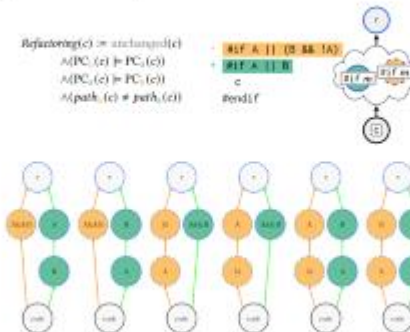
Evaluation – RQ3

Edge-Typed Variation Diffs

→ Lots of Changed Variability in a few Cases!

5,6

94,34

Relationship Edges

...on Edges

...e Edges

Thank You for Your Attention

Edge-Typed Variation Diff

Edge-Typed Variation Diff $(V, \textcolor{red}{E}, r, \tau, l, \Delta, \textcolor{red}{\pi})$

$\textcolor{red}{E}$: Extended Set of Edges

$\textcolor{red}{\pi}$: Edge Label Function

Edge-Typed Variation Diff - Correctness

$$
\begin{array}{ccc}
\text{Edge-Typed} & \xrightarrow{\;\;project_{ET}\;\;} & \text{Edge-Typed} \\
\text{Variation Diff} & & \text{Variation Tree} \\
\big\downarrow reduce_{VD} & & \big\downarrow reduce_{VT} \\
\text{Variation Diff} & \xrightarrow{\;\;project\;\;} & \text{Variation Tree}
\end{array}
$$
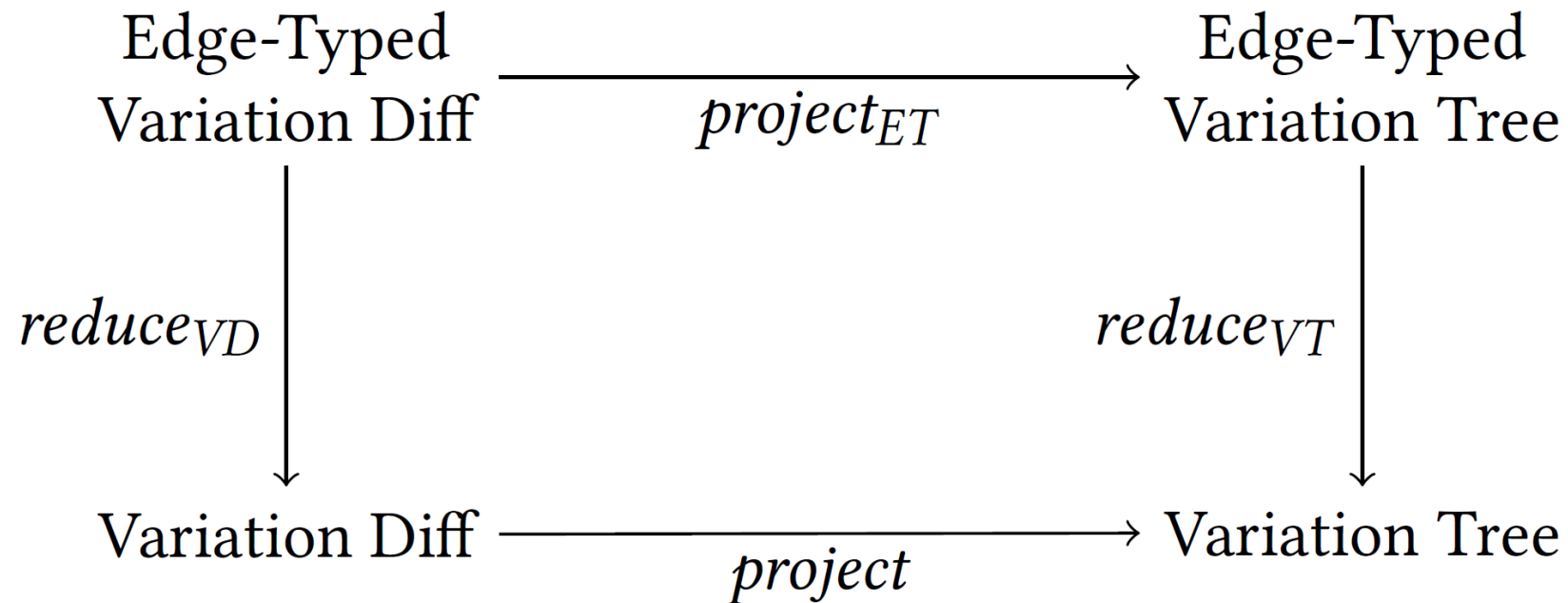
Evaluation

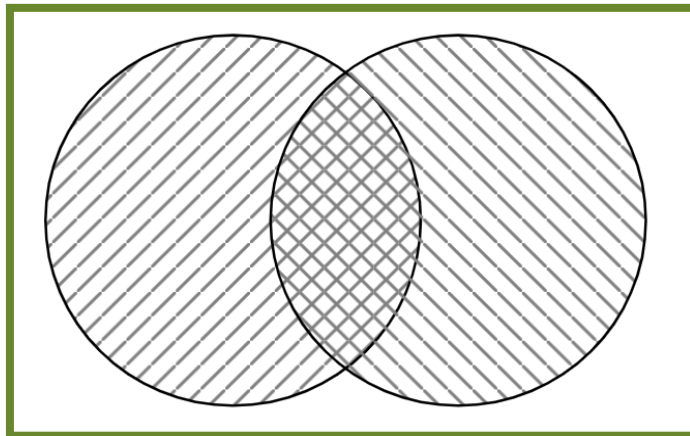|	<0 ms		[1 ms,10 ms]		(10 ms, 100 ms]		(100 ms, 1 s]		(1 s, 1 min]		>1 min		Σ
	#	%	#	%	#	%	#	%	#	%	#	%	#
Commits	720,439	83.48	90,423	10.48	37,587	4.36	10,845	1.26	3,460	0.40	215	0.02	862,969
Patches	2,821,529	90.50	199,471	6.40	73,531	2.36	18,334	0.59	4,750	0.15	182	0.01	3,117,797

Table 1: Share of patches and commits with runtimes in the specified time intervals

No relation edges		Implication		Alternative		Implication and Alternative		Σ
#	%	#	%	#	%	#	%	#
2,941,302	94.34	176,495	5.67	26,358	0.85	19,887	0.64	3,117,797

Table 2: Number of patches with and without *vset* relation edges.

Independent Edges

- $V(X) \cap V(Y) \neq \emptyset \land V(X)\backslash V(Y) \neq \emptyset \land V(Y)\backslash V(X) \neq \emptyset$

$$\leftrightarrow SAT(X \land Y)$$
$$\land SAT(\neg X \land Y)$$
$$\land SAT(X \land \neg Y)$$