

Refinando a precisão da detecção de conflitos: uma análise do CSDiff com abordagem focalizada

Felipe Araujo (CIn/UFPE)

Paulo Borba (CIn/UFPE)

Guilherme Cavalcanti (IFPE)

Conflitos de merge demandam esforço de resolução (mesmo com LLMs!)

base

```
1 def to_string(l: List[str]) -> str:
2     if len(l) == 0:
3         return ""
4     return "__".join(l)
```

left

```
1 def to_string(l: List[str]) -> str:
2     if l is null or len(l) == 0:
3         return ""
4     return "__".join(l)
```

right

```
1 def to_string(l: List[str]) -> str:
2     if len(l) == 0:
3         return self.D
4     return "__".join(l)
```

result

```
1 def to_string(l: List[str]) -> str:
2 <<<<<<< ./left.py
3     if l is null or len(l) == 0:
4         return ""
5     return "__".join(l)
6 =====
7     if len(l) == 0:
8         return self.D
9     return "__".join(l)
10 >>>>>>> ./right.py
```

**Ferramentas de merge avançadas
(estruturadas, semi-estruturadas,
textual baseada em separadores)
evitam parte desses falsos conflitos**

CSDiff (Custom Separators Diff)

- Merge essencialmente textual, com pré e pós processamento
- Utiliza separadores sintáticos pré-definidos, como "(", ":", e ")", além da quebra de linha, para determinar o comportamento do merge
- Simula comportamento do merge estruturado, sem custo de construção para cada linguagem
- 3 etapas

Pré-processamento: Adição de marcadores, antes e depois de separadores, em *base*, *left* e *right*

base

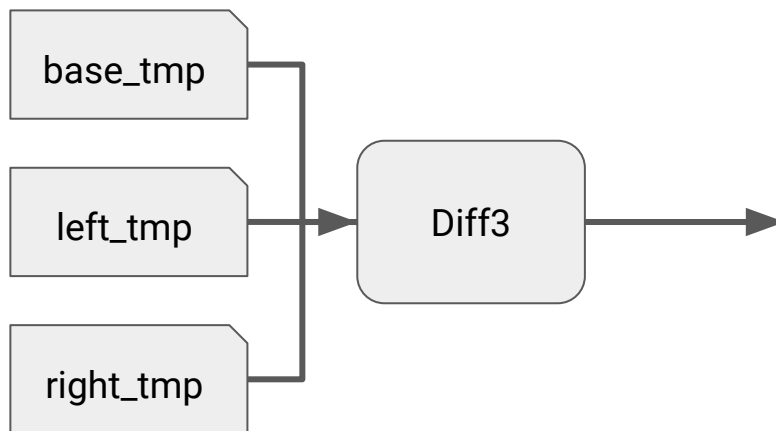
```
1 def to_string(l: List[str]) -> str:  
2     if len(l) == 0:  
3         return ""  
4     return "__".join(l)
```



base_tmp

```
1 def to_string  
2     $$$$$$$$(  
3     $$$$$$$l  
4     $$$$$$$:  
5     $$$$$$$ List[str]  
6     $$$$$$$)  
7     $$$$$$$ -> str  
8     $$$$$$$:  
9     $$$$$$$  
10        if len  
11     $$$$$$$$(  
12     $$$$$$$l  
13     $$$$$$$)  
14     $$$$$$$ == 0  
15     $$$$$$$:  
16     $$$$$$$  
17         return ""  
18     return "__".join  
19     $$$$$$$(  
20     $$$$$$$l  
21     $$$$$$$)  
22     $$$$$$$
```

Execução do Diff3



merged_tmp

```
def to_string
$$$$$$ (
$$$$$$ l
$$$$$$:
$$$$$$ List[str]
$$$$$$)
$$$$$$ -> str
$$$$$$:
$$$$$$
$$$$$$ if l is null or len
$$$$$$ (
$$$$$$ l
$$$$$$)
$$$$$$ == 0
$$$$$$:
$$$$$$
$$$$$$ return self.D
return "__".join
```

Pós-processamento: remoção dos marcadores

merged_tmp

```
def to_string
  $$$$$$(
  $$$$$$l
  $$$$$$:
  $$$$$$ List[str]
  $$$$$$)
  $$$$$$ -> str
  $$$$$$:
    return self.D
  return "__".join
```



merged

```
if l is null or len(l) == 0:
  return self.D
return "__".join(l)
```

Problema

- Apesar das vantagens, o CSDiff adiciona falsos positivos e falsos negativos em relação ao Diff3
- Falsos positivos do CSDiff são causados por problemas de alinhamento e pareamento das linhas durante a etapa de execução do Diff3 (devido aos marcadores \$\$\$\$\$)

Solução: SepMerge

Evita o uso dos marcadores, adiciona linhas

base_tmp (CSDiff)

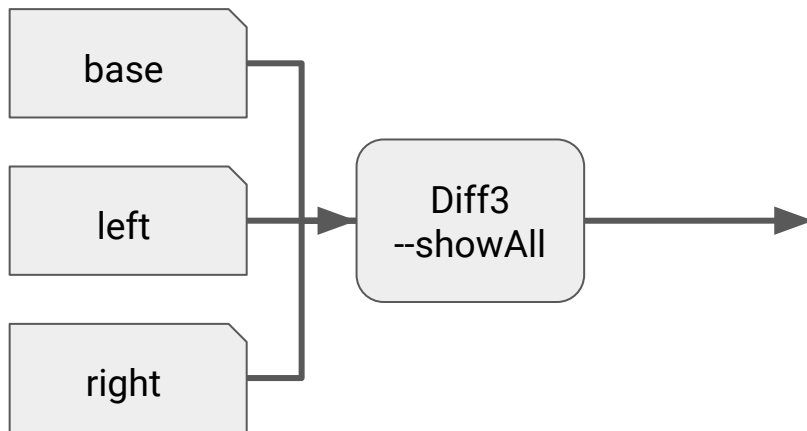
```
1 def to_string
2     $$$$$$(
3     $$$$$$l
4     $$$$$$:
5     $$$$$$ List[str]
6     $$$$$$)
7     $$$$$$ -> str
8     $$$$$$:
9     $$$$$$
10    |   if len
11    |   $$$$$$(
12    |   $$$$$$l
13    |   $$$$$$)
14    |   $$$$$$ == 0
15    |   $$$$$$:
16    |   $$$$$$
17    |   |   return ""
18    |   |   return "..".join
19    |   $$$$$$(
20    |   $$$$$$l
21    |   $$$$$$)
22    |   $$$$$$
23
```

base_tmp (SepMerge)

```
1 def to_string
2 (
3 l
4 :
5 List[str]
6 )
7 -> str
8 :
9 :
10 |   if len
11 |   (
12 |   l
13 |   )
14 |   == 0
15 |   :
16 |   :
17 |   |   return ""
18 |   |   return "..".join
19 |   (
20 |   l
21 |   )
22 |
23 |
```

**Execução focalizada, só
quando o diff3 reporta
conflito, e só no texto do
conflito**

Executa o Diff3 com opção --showAll



diff3_tmp

```
1 def to_string(l: List[str]) -> str:
2 <<<<<<< /left.py
3     if l is null or len(l) == 0:
4         return ""
5     return "__".join(l)
6 ||| ||| /base.py
7     if len(l) == 0:
8         return ""
9     return "..".join(l)
10 =====
11     if len(l) == 0:
12         return self.D
13     return "__".join(l)
14 >>>>>>> /right.py
```

Para cada conflito reportado, cria os arquivos left, right e base

diff3_tmp

```
1 def to_string(l: List[str]) -> str:
2 <<<<<<< /left.py
3     if l is null or len(l) == 0:
4         return ""
5     return "__".join(l)
6 ||||||| /base.py
7     if len(l) == 0:
8         return ""
9     return "..".join(l)
10 =====
11     if len(l) == 0:
12         return self.D
13     return "__".join(l)
14 >>>>>>> /right.py
```

base_tmp

```
1     if len(l) == 0:
2         return ""
3     return "..".join(l)
```

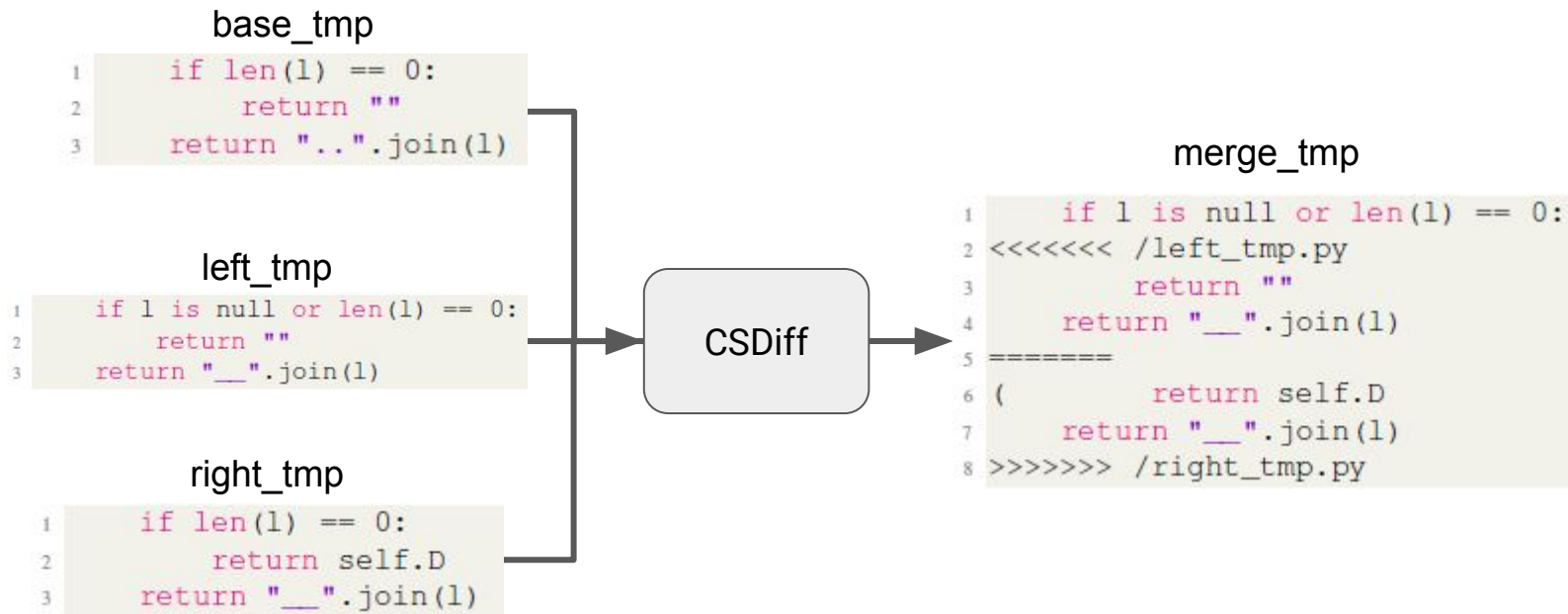
left_tmp

```
1     if l is null or len(l) == 0:
2         return ""
3     return "__".join(l)
```

right_tmp

```
1     if len(l) == 0:
2         return self.D
3     return "__".join(l)
```

Executa o CSDiff nos arquivos criados



Substituir o conflito no resultado do diff3 com o resultado do CSDiff

diff3_tmp

```
1 def to_string(l: List[str]) -> str:
2 <<<<<<< /left.py
3     if l is null or len(l) == 0:
4         return ""
5     return "__".join(l)
6 ||||| /base.py
7     if len(l) == 0:
8         return ""
9     return "..".join(l)
10 =====
11     if len(l) == 0:
12         return self.D
13     return "__".join(l)
14 >>>>>>> /right.py
```

merge_tmp

```
1     if l is null or len(l) == 0:
2 <<<<<<< /left_tmp.py
3         return ""
4     return "__".join(l)
5 =====
6 (         return self.D
7     return "__".join(l)
8 >>>>>>> /right_tmp.py
```

result

```
1 def to_string(l: List[str]) -> str:
2     if l is null or len(l) == 0:
3 <<<<<<< /left.py
4         return ""
5     return "__".join(l)
6 =====
7 (         return self.D
8     return "__".join(l)
9 >>>>>>> /right.py
```

Avaliação

Replay de cenários de merge

- Foi utilizado o MiningFramework usando as seguintes configurações:
 - Projetos: Certbot, Flask, IPython, Matplotlib, Requests, Salt, Scrapy, Sentry, Tensorflow, Tornado
 - Período: 2 anos (01/01/2021 até 01/01/2023)
 - Critério de seleção dos cenários: Cenários com modificações nos mesmos arquivos
- Ferramentas executadas e analisadas:
 - **CSDiff** usando os separadores: "():,"
 - **CSDiffI** usando os separadores: "():," e Indentação
 - **SepMerge** com os separadores: "():,"
 - **SepInMerge** usando os separadores: "():," e Indentação

PP1: A Abordagem Focalizada do SepMerge Reduz a Quantidade de Conflitos Reportados em Comparação ao Diff3 e ao CSDiff?

	SepInMerge	CSDiff1	SepMerge	CSDiff	Diff3
Conflitos	329	726	323	537	373
Arquivos com Conflitos	206	245	205	226	235
Cenários com Conflitos	75	77	75	77	84

PP2: A Abordagem Focalizada do SepMerge

Reduz a Quantidade de Cenários com Conflitos Reportados em Comparação ao Diff3 e ao CSDiff?

	SepInMerge	CSDiff1	SepMerge	CSDiff	Diff3
Conflitos	329	726	323	537	373
Arquivos com Conflitos	206	245	205	226	235
Cenários com Conflitos	75	77	75	77	84

PP3: A Abordagem Focalizada do SepMerge

Reduz a Quantidade de Arquivos com Erros de Identificação em Comparação com o Diff3 e o CSDiff?

Comparação entre SepInMerge e Diff3		
	SepInMerge	Diff3
aFP arquivos	0	21
aFN arquivos	8	0

Comparação entre SepMerge e Diff3		
	SepMerge	Diff3
aFP arquivos	0	22
aFN arquivos	8	0

Comparação entre CSDiff1 e Diff3		
	CSDiff1	Diff3
aFP arquivos	36	19
aFN arquivos	9	0

Comparação entre CSDiff e Diff3		
	CSDiff	Diff3
aFP arquivos	19	21
aFN arquivos	8	0

PP4: A Abordagem Focalizada do SepMerge

Reduz a Quantidade de Cenários com Erros de Identificação em Comparação com o Diff3 e o CSDiff?

Comparação entre SepInMerge e Diff3		
	SepInMerge	Diff3
aFP cenários	0	10
aFN cenários	2	0

Comparação entre SepMerge e Diff3		
	SepMerge	Diff3
aFP cenários	0	10
aFN cenários	2	0

Comparação entre CSDiff1 e Diff3		
	CSDiff1	Diff3
aFP cenários	1	8
aFN cenários	3	0

Comparação entre CSDiff e Diff3		
	CSDiff	Diff3
aFP cenários	1	9
aFN cenários	2	0

Conclusões

- O SepMerge traz benefícios em relação ao CSDiff (reduz FPs) e Diff3 (reduz FPs, mas adiciona FNs)
- O suporte a indentação como separador não traz benefícios significativos
- Integração com merge semiestruturado é uma abordagem híbrida promissora
- Baixo custo em relação a abordagens estruturadas e semi-estruturadas

Obrigado!

Refinando a precisão da detecção de conflitos: uma análise do CSDiff com abordagem focalizada

Felipe Araujo (CIn/UFPE)

Paulo Borba (CIn/UFPE)

Guilherme Cavalcanti (IFPE)



pauloborba.cin.ufpe.br

