Translating Code Comments to Procedure Specifications

Arianna Blasi[♦♥] · Alberto Goffi[♦] · Konstantin Kuznetsov[♣]
Alessandra Gorla[♥] · Michael D. Ernst[♠] · Mauro Pezzè[♦] · Sergio Delgado[♥]

[♦]USI Università della Svizzera italiana, Switzerland

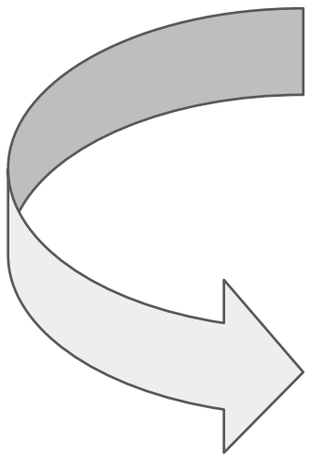
[♥]IMDEA Software Institute, Spain

[♣]Saarland University/CISPA, Germany

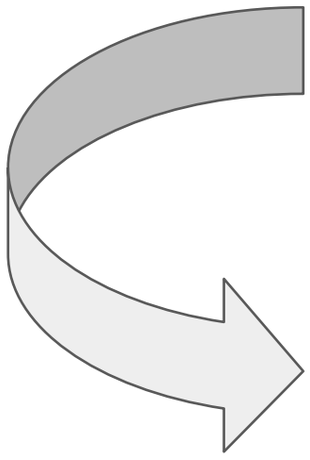
[♠]University of Washington Seattle, WA, USA

```
public static boolean removeAll(List myList){…}
```

```
public static boolean removeAll(List myList){…}
```

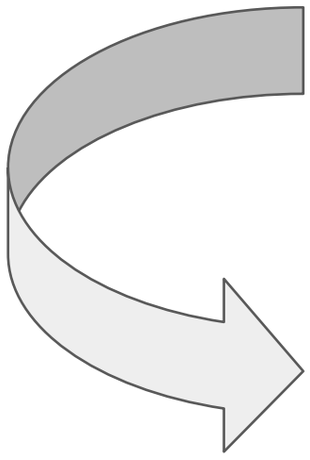


```java
public static boolean removeAll(List myList){…}
```



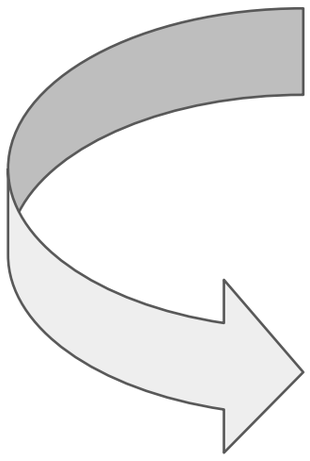
```java
@Test
public void testRemoveAll(){



}
```

```java
public static boolean removeAll(List myList){…}
```



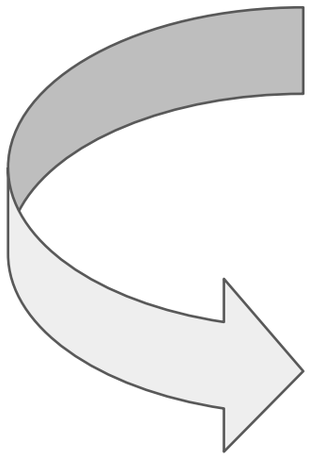
```java
@Test
public void testRemoveAll(){
    List<Integer> list = Arrays.asList(1, 2, 3);


}
```

```java
public static boolean removeAll(List myList){…}
```



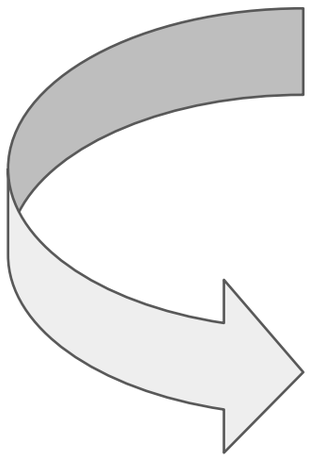
```java
@Test
public void testRemoveAll(){
    List<Integer> list = Arrays.asList(1, 2, 3);
    boolean result = removeAll(list);

}
```

```java
public static boolean removeAll(List myList){…}
```



```java
@Test
public void testRemoveAll(){
    List<Integer> list = Arrays.asList(1, 2, 3);
    boolean result = removeAll(list);
    assert ?
}
```

```
public static boolean removeAll(List myList){…}
```



```
@Test
public void testRemoveAll(){
    List<Integer> list = Arrays.asList(1, 2, 3);
    boolean result = removeAll(list);
    assert ?
}
```

EVOSUITE

Randoop

Automatic unit test generation for Java

EVOSUITE

Randoop

Automatic unit test generation for Java

Where is the oracle?

EVOSUITE

Randoop

Automatic unit test generation for Java

Where is the oracle?

Where is the specification?

Why not generating executable specifications automatically?

Why not generating executable specifications automatically?

Let’s rely on Javadoc comments!

Our solution

```
/**
 * @return true if the list is empty
 */
public static boolean removeAll(List myList)
{…}
```

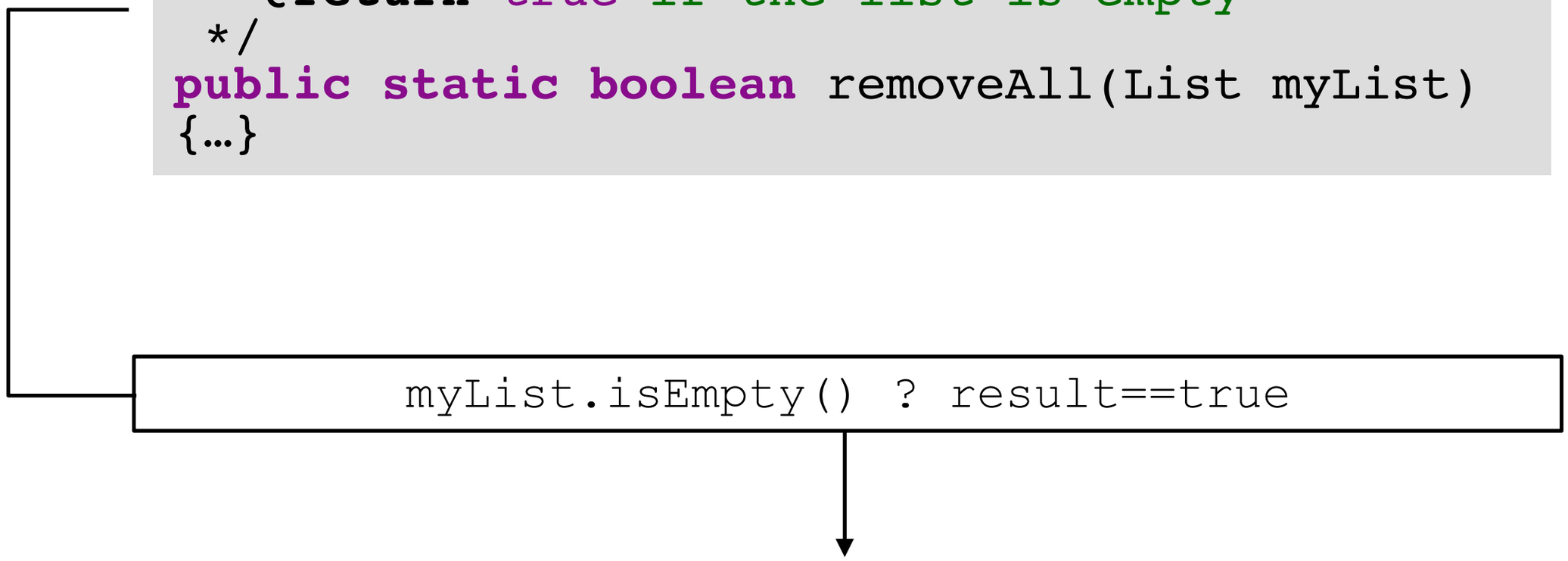
Our solution

```
/**
 * @return true if the list is empty
 */
public static boolean removeAll(List myList)
{…}
```

```
myList.isEmpty() ? result==true
```

Our solution

```
/**
 * @return true if the list is empty
 */
public static boolean removeAll(List myList)
{…}
```



Our solution

```
/**
 * @return true if the list is empty
 */
public static boolean removeAll(List myList)
{…}
```

```
myList.isEmpty() ? result==true
```

```
@Test
public void testRemoveAll(){
    List<Integer> list = Arrays.asList(1, 2, 3);
    boolean result = removeAll(list);

}
```

Our solution

```
/**
 * @return true if the list is empty
 */
public static boolean removeAll(List myList)
{…}
```

```
myList.isEmpty() ? result==true
```

```
@Test
public void testRemoveAll(){
    List<Integer> list = Arrays.asList(1, 2, 3);
    boolean result = removeAll(list);
    if(list.isEmpty()) assert result==true;
}
```

Our solution

```
/**
 * @return true if the list is empty
 */
public static boolean removeAll(List myList)
{…}
```

myList.isEmpty() ? result==true

```
@Test
public void testRemoveAll(){
    List<Integer> list = Arrays.asList(1, 2, 3);
    boolean result = removeAll(list);
    if(list.isEmpty()) assert result==true;
}
```

The idea is not new...

The idea is not new...

@tComment

Tan et al. ICST 2012

The idea is not new...

@tComment

Tan et al. ICST 2012

Toradocu

Goffi et al. ISSTA 2016

The idea is not new...

@tComment

Tan et al. ICST 2012

Toradocu

Goffi et al. ISSTA 2016

ALICS

Pandita et al. ICSE 2012

The idea is not new...

@tComment

Tan et al. ICST 2012

Toradocu

Goffi et al. ISSTA 2016

ALICS

Pandita et al. ICSE 2012

Preconditions	Normal Postconditions	Exceptional Postconditions

The idea is not new...

@tComment

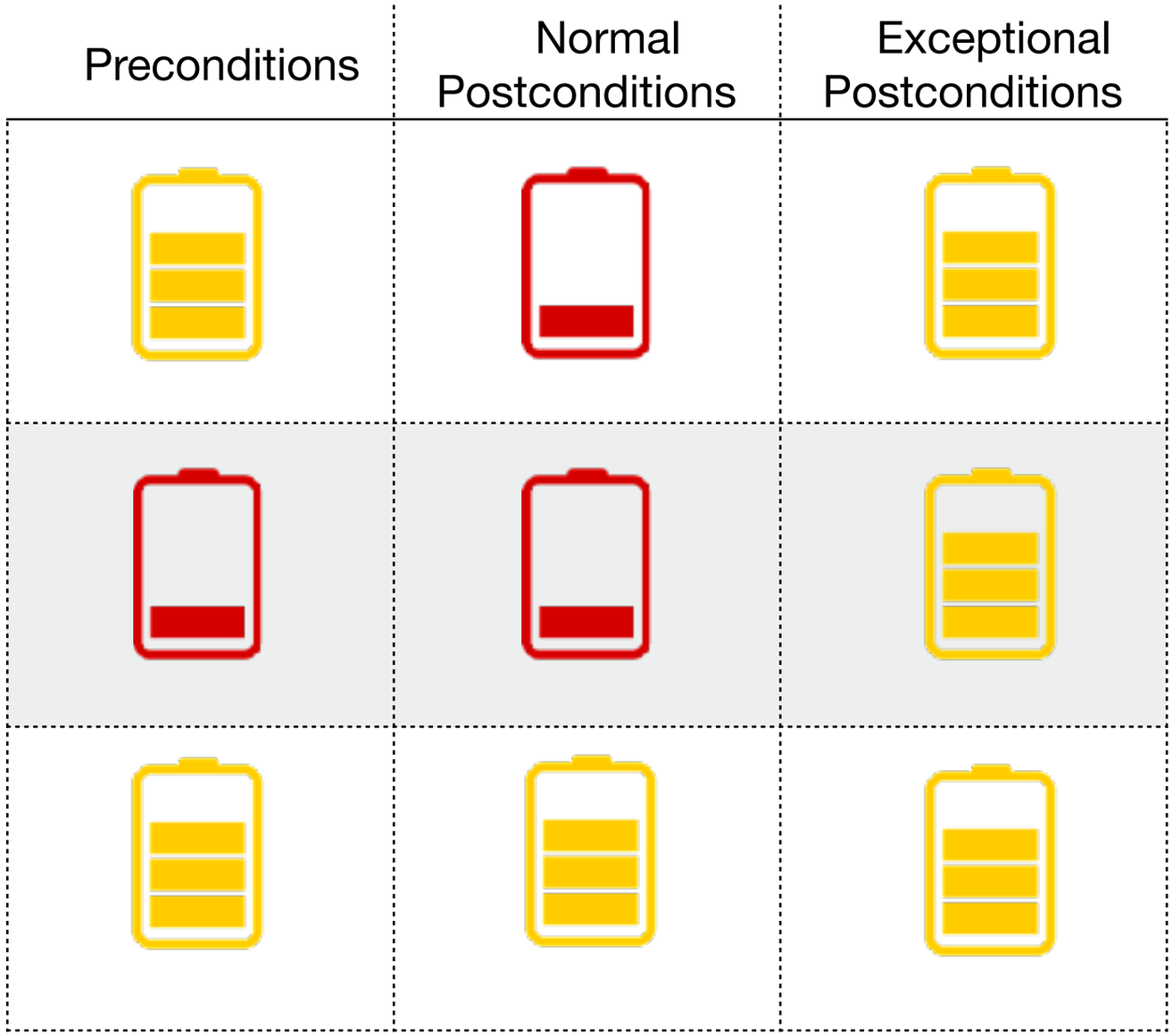
Tan et al. ICST 2012

Toradocu

Goffi et al. ISSTA 2016

ALICS

Pandita et al. ICSE 2012



The idea is not new...

@tComment

Tan et al. ICST 2012

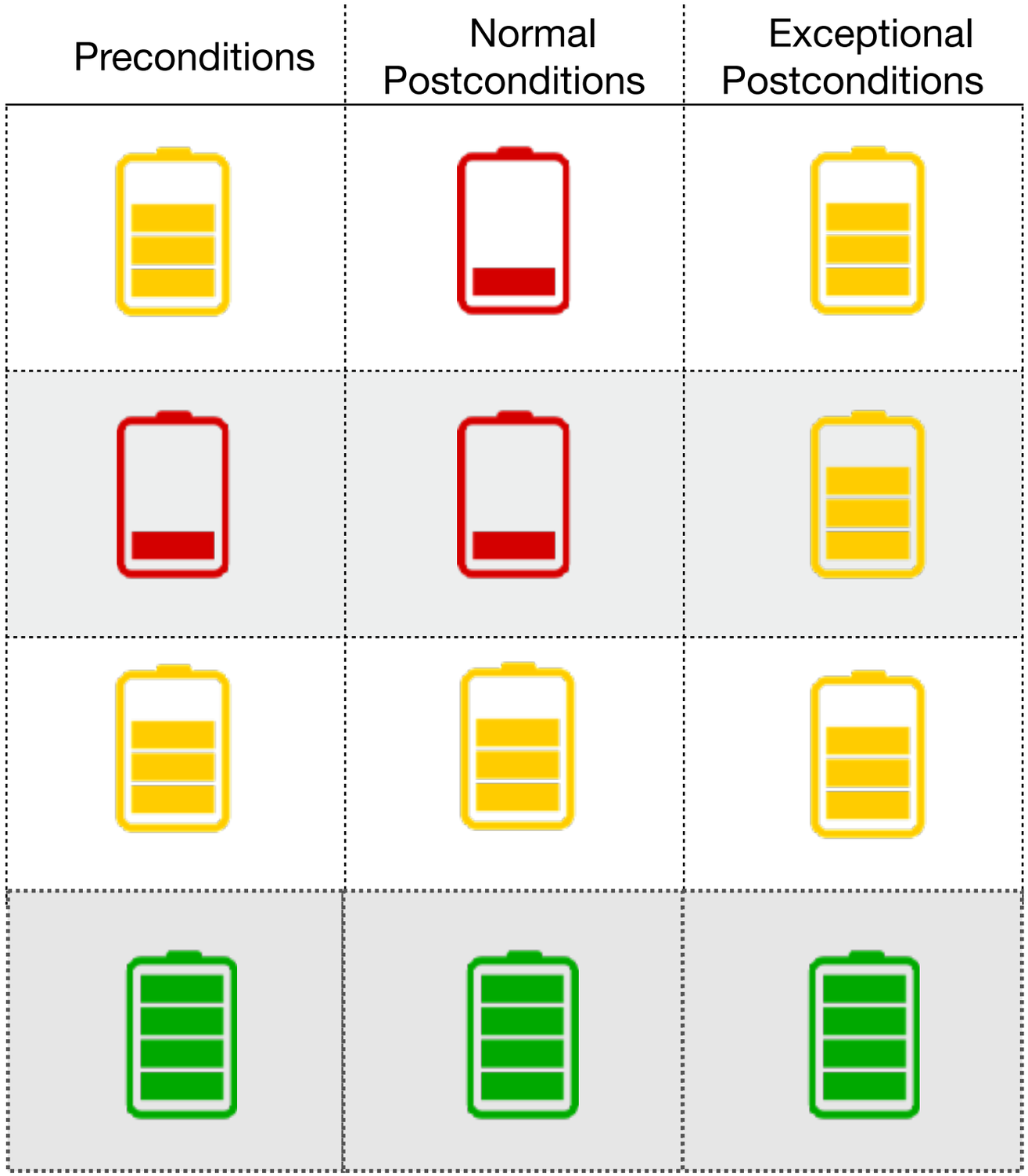
Toradocu

Goffi et al. ISSTA 2016

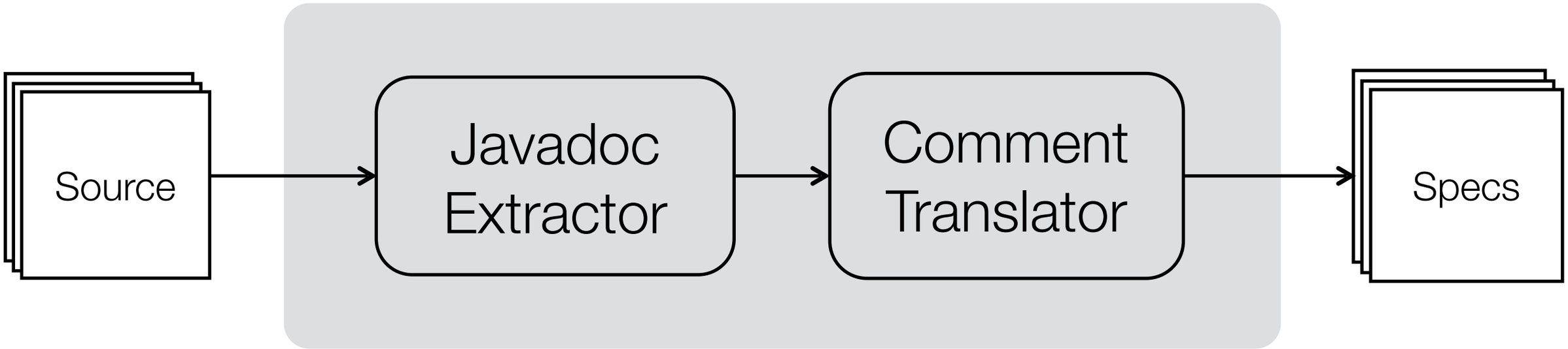
ALICS

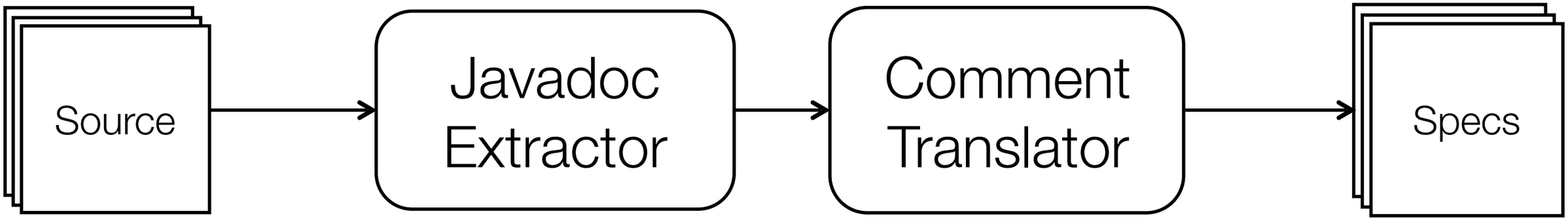
Pandita et al. ICSE 2012

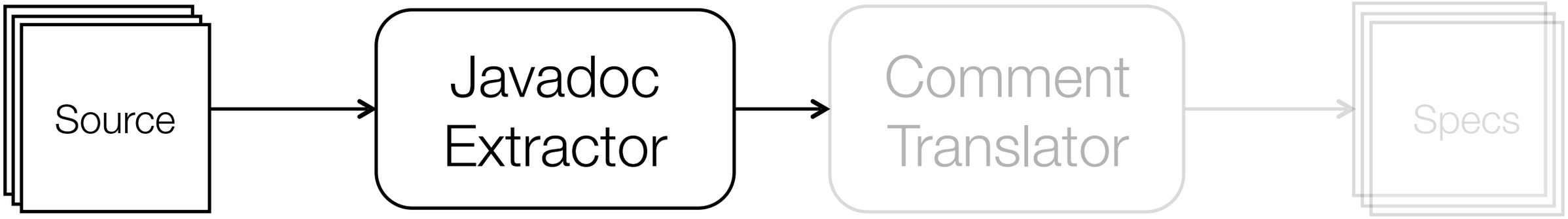
Jdoctor

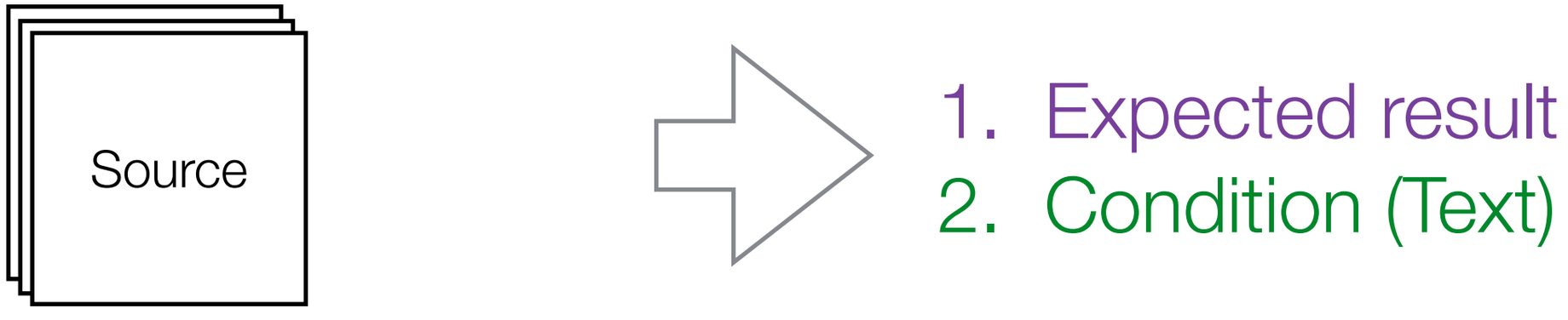


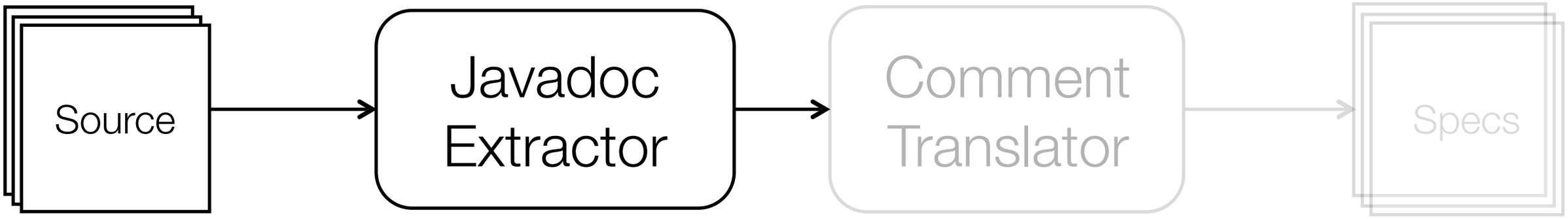
Jdoctor

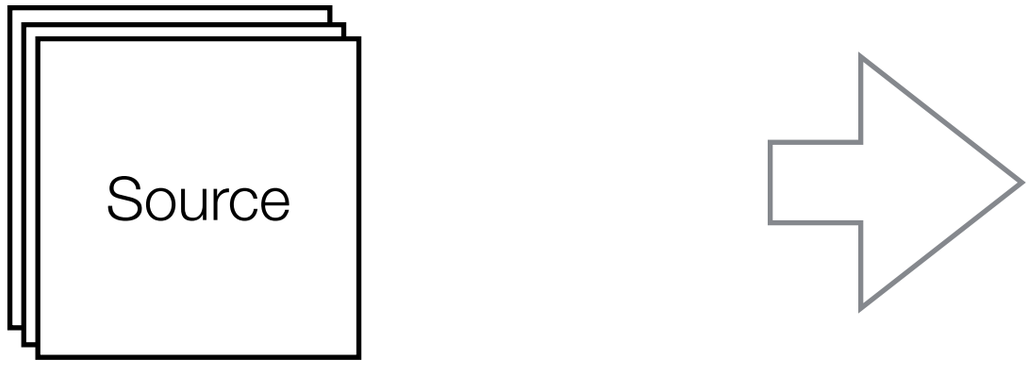



Source
Javadoc
Extractor
Comment
Translator
Specs


Source
Javadoc Extractor
Comment Translator
Specs

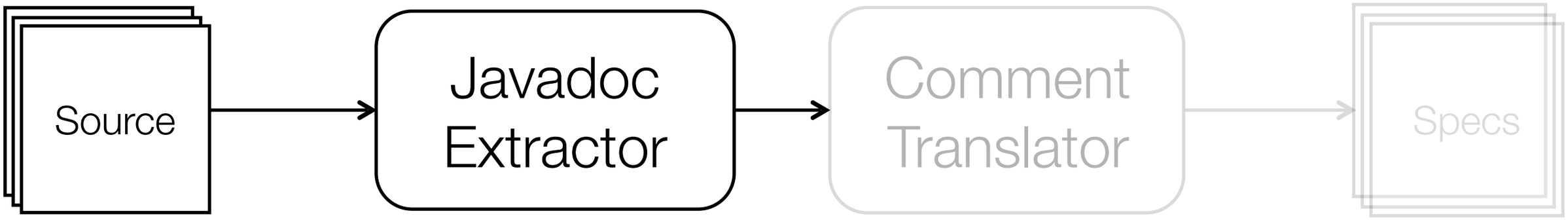

Source
1. Expected result
2. Condition (Text)





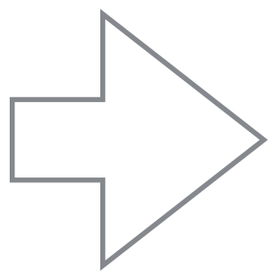

```
/**
 * @return true
 * if the list is empty
 */
public boolean removeAll(List
list){…}
```

1. true
2. “if the list is empty”

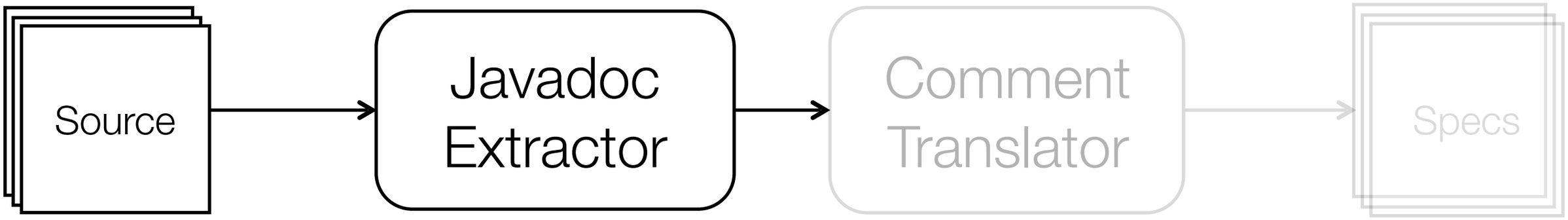


1. Expected result
2. Condition (Text)

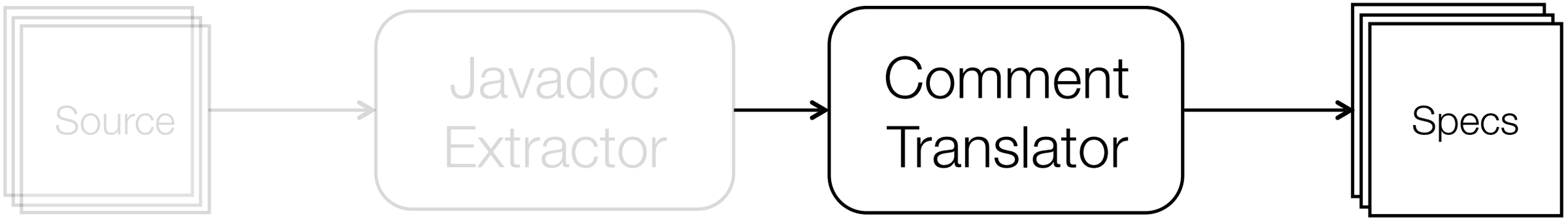
```
/**
 * @return true
 * if the list is empty
 */
public boolean removeAll(List
list){…}
```



1. true
2. “if the list is empty”

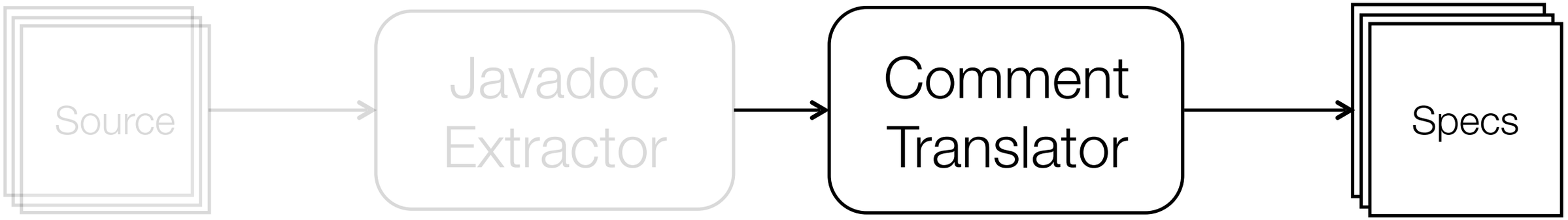


1. Expected result
2. Condition (Text)

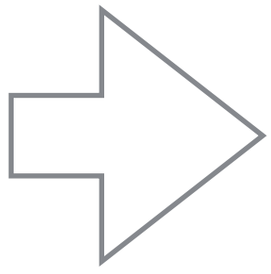


1. Expected result
2. Condition (Text)

1. Expected result
2. **Condition (Code)**

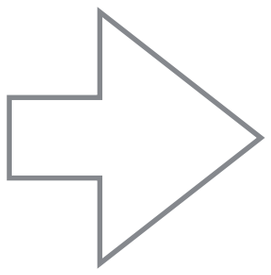


1. Expected result
2. Condition (Text)

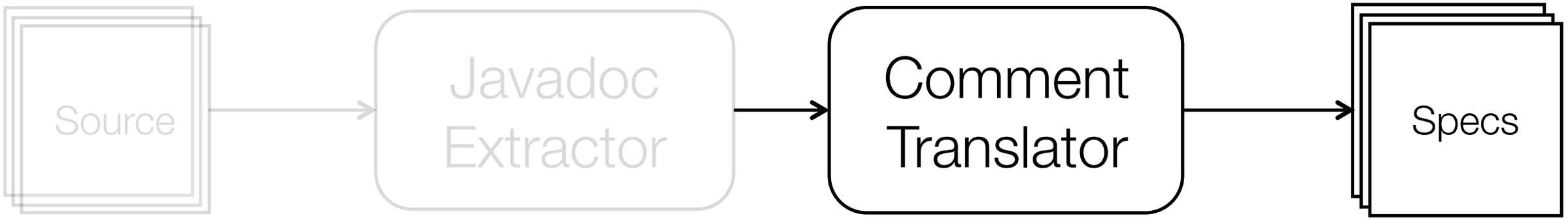


1. Expected result
2. **Condition (Code)**

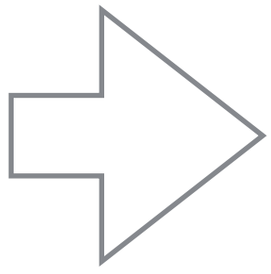
```
1. true
2. “if the list is empty”
```



```
1. true
2. myList.isEmpty()
```

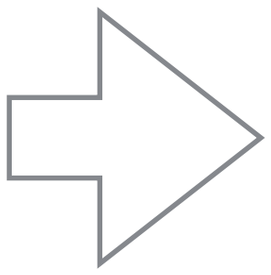


1. Expected result
2. Condition (Text)



1. Expected result
2. **Condition (Code)**

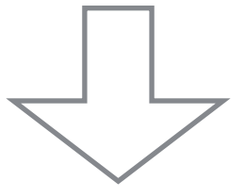
1. true
2. “if the list is empty”



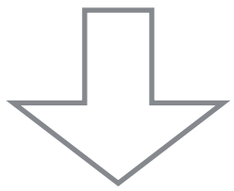
1. true
2. myList.isEmpty()

```
myList.isEmpty() ? result==true
```

Natural Language Specification

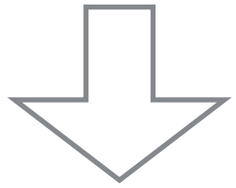


Comment Translator

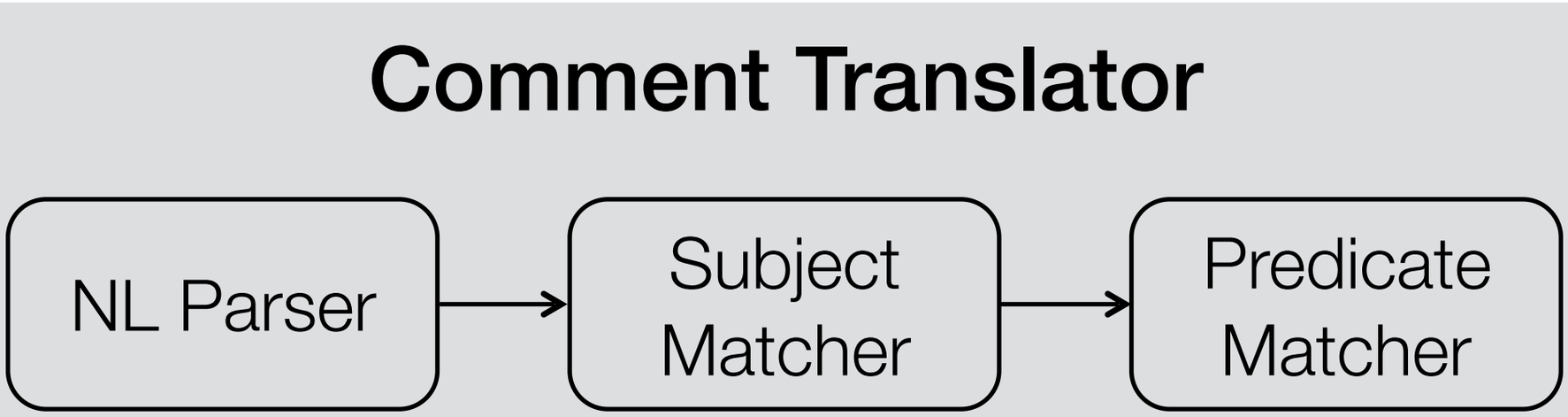


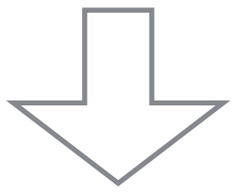
Java Specification

Natural Language Specification



Comment Translator

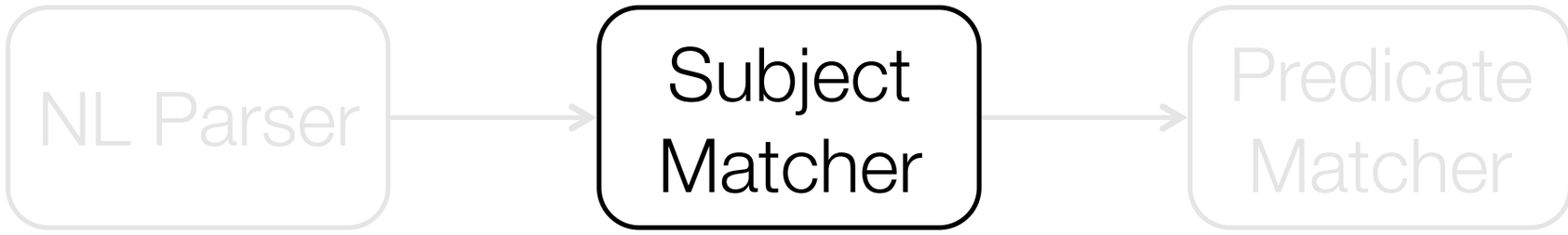




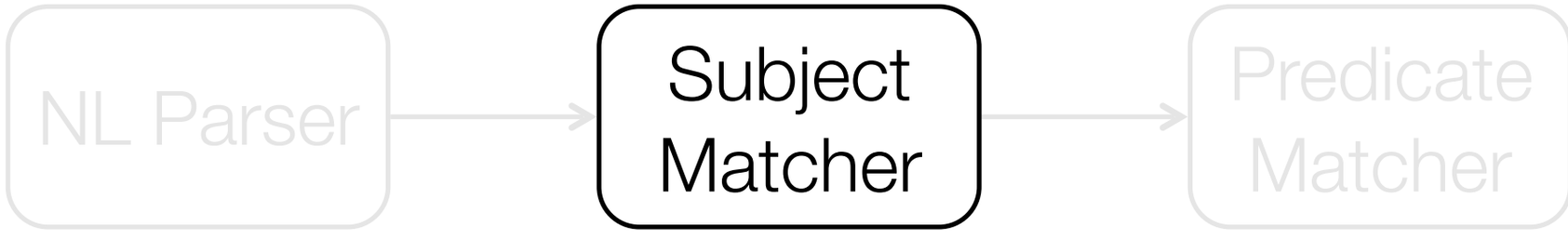
Java Specification


```
“if the list is empty”
```

Subject	Predicate
“list”	“is empty”



Subject: `“list”`



Subject: `“list”`

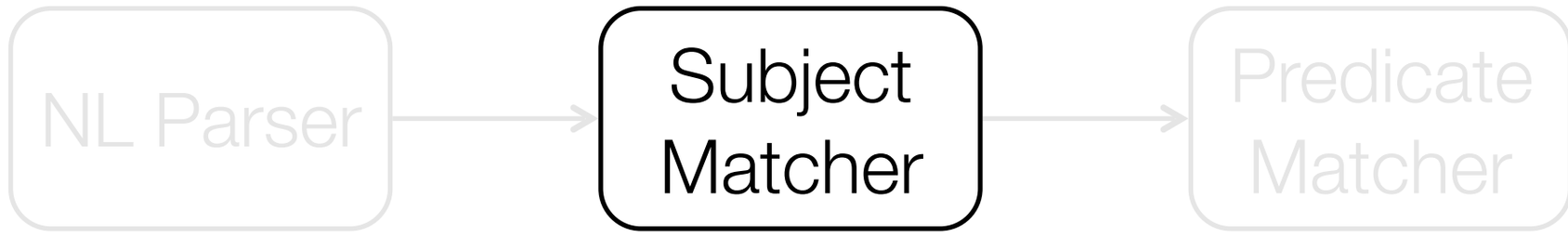
Candidates

Formal Parameters

Class Name

Methods

Fields

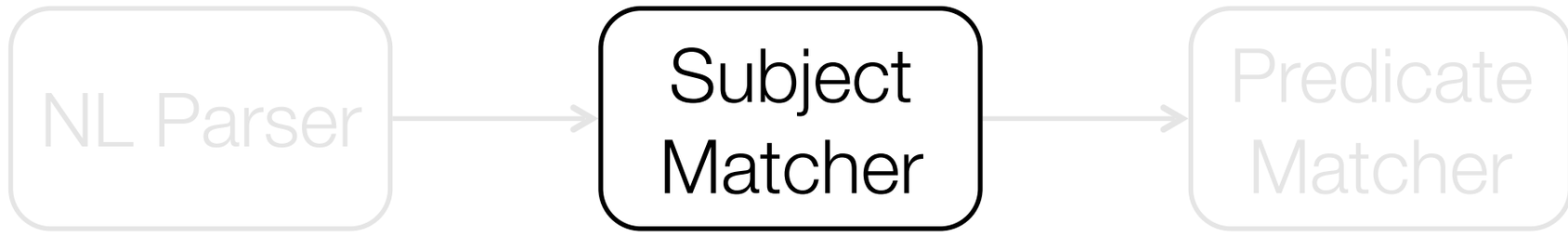


Subject: `"list"`

Candidates

Formal Parameters
Class Name
Methods
Fields

Candidate	Distance
`myList`	2
`isEmpty`	7
`iterator`	8
`size`	4
…	

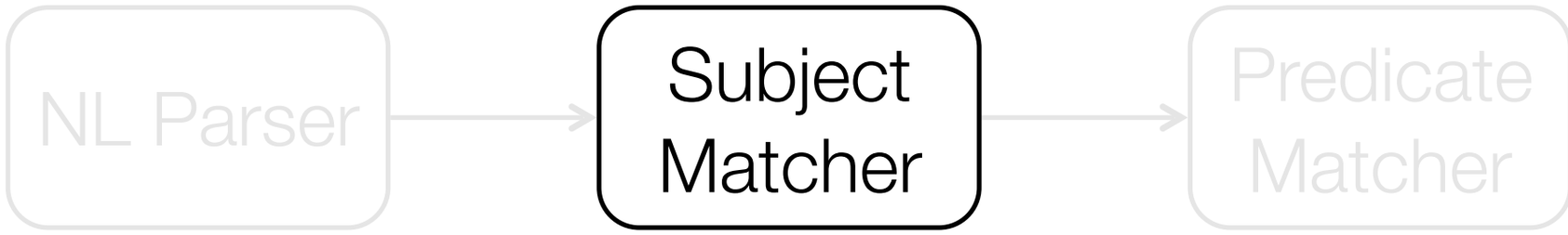


Subject: `"list"`

Candidates

Formal Parameters
Class Name
Methods
Fields

Candidate	Distance
`myList`	**2**
`isEmpty`	7
`iterator`	8
`size`	4
…	



Subject: `"list"` => **`myList`**

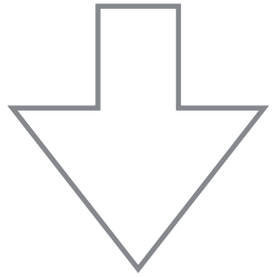
Candidates

Formal Parameters
Class Name
Methods
Fields

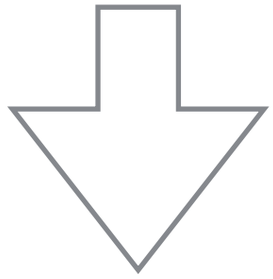
Candidate	Distance
`myList`	**2**
`isEmpty`	7
`iterator`	8
`size`	4
...	

Subject	Predicate
`"list"`	`"is empty"`

Subject	Predicate
`"list"`	`"is empty"`

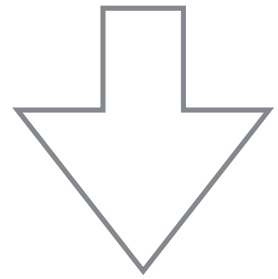


Subject	Predicate
`"list"`	`"is empty"`



1: Exact pattern match?

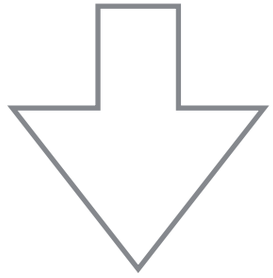
Subject	Predicate
`"list"`	`"is empty"`



is null ⟶ `== null`

1: Exact pattern match?

Subject	Predicate
`"list"`	`"is empty"`



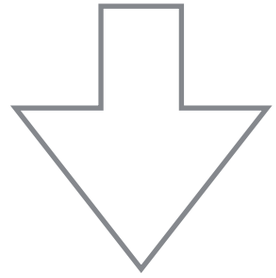
1: Exact pattern match?

is null ⟶ `== null`

is negative ⟶ `< 0`

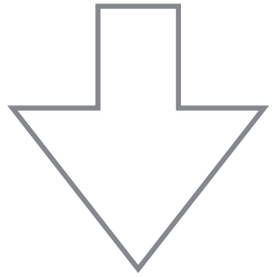
Subject	Predicate
`"list"`	`"is empty"`

1: Exact pattern match?



is null	→	`== null`
is negative	→	`< 0`
is positive	→	`> 0`

Subject	Predicate
`"list"`	`"is empty"`

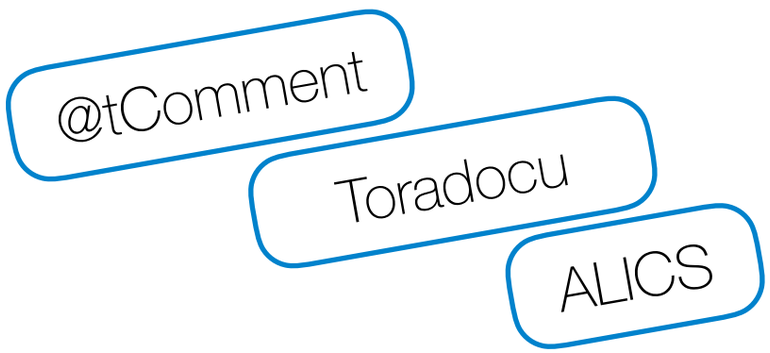


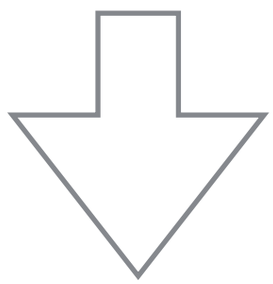
1: Exact pattern match?

is null	→	`== null`
is negative	→	`< 0`
is positive	→	`> 0`
...		

Subject	Predicate
`"list"`	`"is empty"`

1: Exact pattern match?

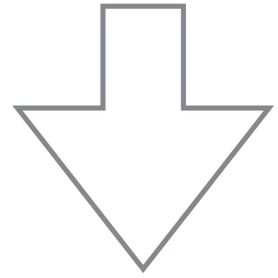




is null	→	`== null`
is negative	→	`< 0`
is positive	→	`> 0`
...		

Subject	Predicate
`"list"`	`"is empty"`

"is empty" not found...



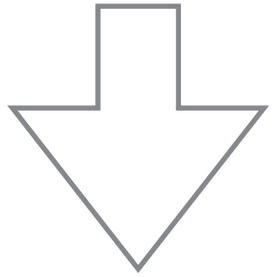
is null → `== null`

is negative → `< 0`

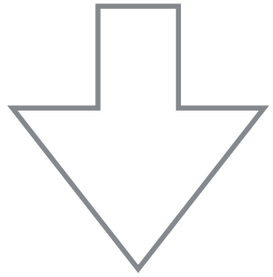
is positive → `> 0`

...

Subject	Predicate
`"list"`	`"is empty"`



Subject	Predicate
`"list"`	`"is empty"`



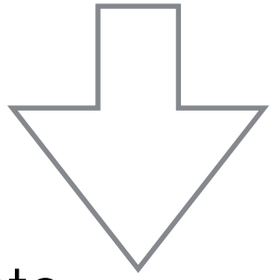
2: Syntax match?

Toradocu

Subject	Predicate
`"list"`	`"is empty"`

2: Syntax match?

Toradocu

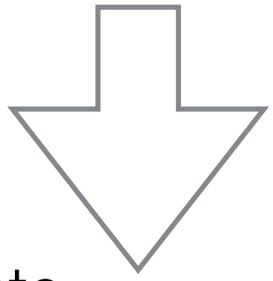


Candidate	Distance
`myList`	9
`isEmpty`	1
`iterator`	6
`size`	7
...	

Subject	Predicate
`"list"`	`"is empty"`

2: Syntax match?

Toradocu



Candidate	Distance
`myList`	9
`isEmpty`	**1**
`iterator`	6
`size`	7
...	

Pattern and syntax match are not enough

Pattern and syntax match are not enough

Return true if the list **is empty**

Pattern and syntax match are not enough

Return true if the list **is empty**

Candidate	Distance
`clear`	7
`isEmpty`	**1**
`iterator`	6
`size`	7
...	

Pattern and syntax match are not enough

Return true if the list **is empty**

contains no element

Pattern and syntax match are not enough

Return true if the list **is empty**

contains no element

has no content

Pattern and syntax match are not enough

Return true if the list **is empty**

contains no element

has no content

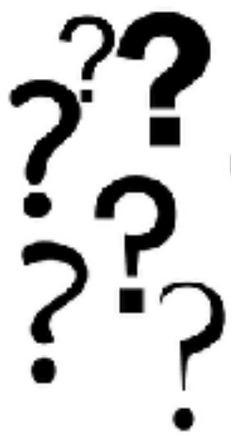
Candidate	Distance
`myList`	
`isEmpty`	
`iterator`	
`size`	
…	

Pattern and syntax match are not enough

Return true if the list **is empty**

contains no element

has no content

Candidate	Distance
`myList`	
`isEmpty`	
`iterator`	
`size`	
…	

Pattern and syntax match are not enough

Return true if the list **is empty**

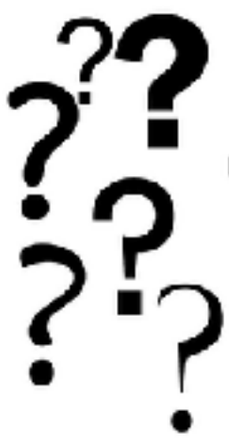
contains no element

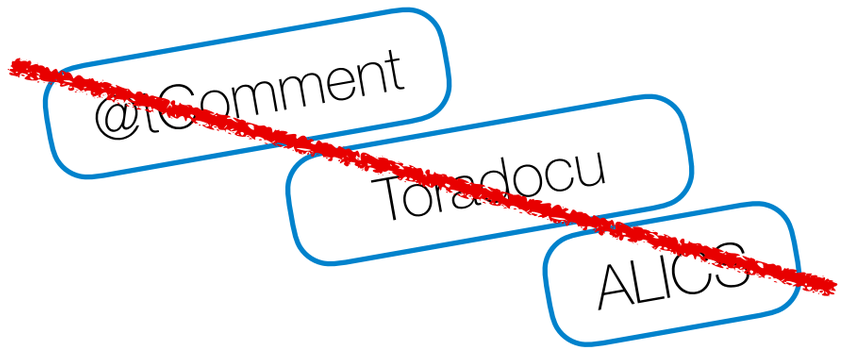
has no content

Candidate

```
myList
isEmpty
iterator
size
...
```

Distance





Analyzing complex comments

```
/**
* Checks if there is a credible threatening unit to
* this unit within a range of moves.
*
* @param moves list of moves
* @return True if a threat was found.
*/
public boolean isInDanger(List<Object> moves)
```

Analyzing complex comments

```
/**
* Checks if there is a credible threatening unit to
* this unit within a range of moves.
*
* @param moves list of moves
* @return True if a threat was found.
*/
public boolean isInDanger(List<Object> moves)
```

Analyzing complex comments

```
/**
* Checks if there is a credible threatening unit to
* this unit within a range of moves.
*
* @param moves list of moves
* @return True if a threat was found.
*/
public boolean isInDanger(List<Object> moves)
```

searchForDanger(*List<Object>*)

PRINCETON UNIVERSITY

WordNet

A Lexical Database for English

threat was found

search for **danger**

PRINCETON UNIVERSITY

WordNet

A Lexical Database for English

Verb

- S: (v) find, happen, chance, bump, encounter (come upon, as if by accident; meet with)
- S: (v) detect, observe, find, discover, notice (discover or determine the existence, presence, or fact of)
- S: (v) find, regain (come upon after searching; find the location of something that was missed or lost)
- S: (v) determine, find, find out, ascertain (establish after a calculation, investigation, experiment, survey, or study)
- S: (v) find, feel (come to believe on the basis of emotion, intuitions, or indefinite grounds)
- S: (v) witness, find, see (perceive or be contemporaneous with)
- S: (v) line up, get hold, come up, find (get something or somebody for a specific purpose)
- S: (v) discover, find (make a discovery, make a new finding)
- S: (v) discover, find (make a discovery)
- S: (v) find (obtain through effort or management)

Verb

- S: (v) find, happen, chance, bump, encounter (come upon, as if by accident; meet with)
- S: (v) detect, observe, find, discover, notice (discover or determine the existence, presence, or fact of)
- S: (v) find, regain (come upon after searching; find the location of something that was missed or lost)
- S: (v) determine, find, find out, ascertain (establish after a calculation,

"*search*" is **not** displayed as a result

- S: (v) witness, find, see (perceive or be contemporaneous with)
- S: (v) line up, get hold, come up, find (get something or somebody for a specific purpose)
- S: (v) discover, find (make a discovery, make a new finding)
- S: (v) discover, find (make a discovery)
- S: (v) find (obtain through effort or management)

Verb

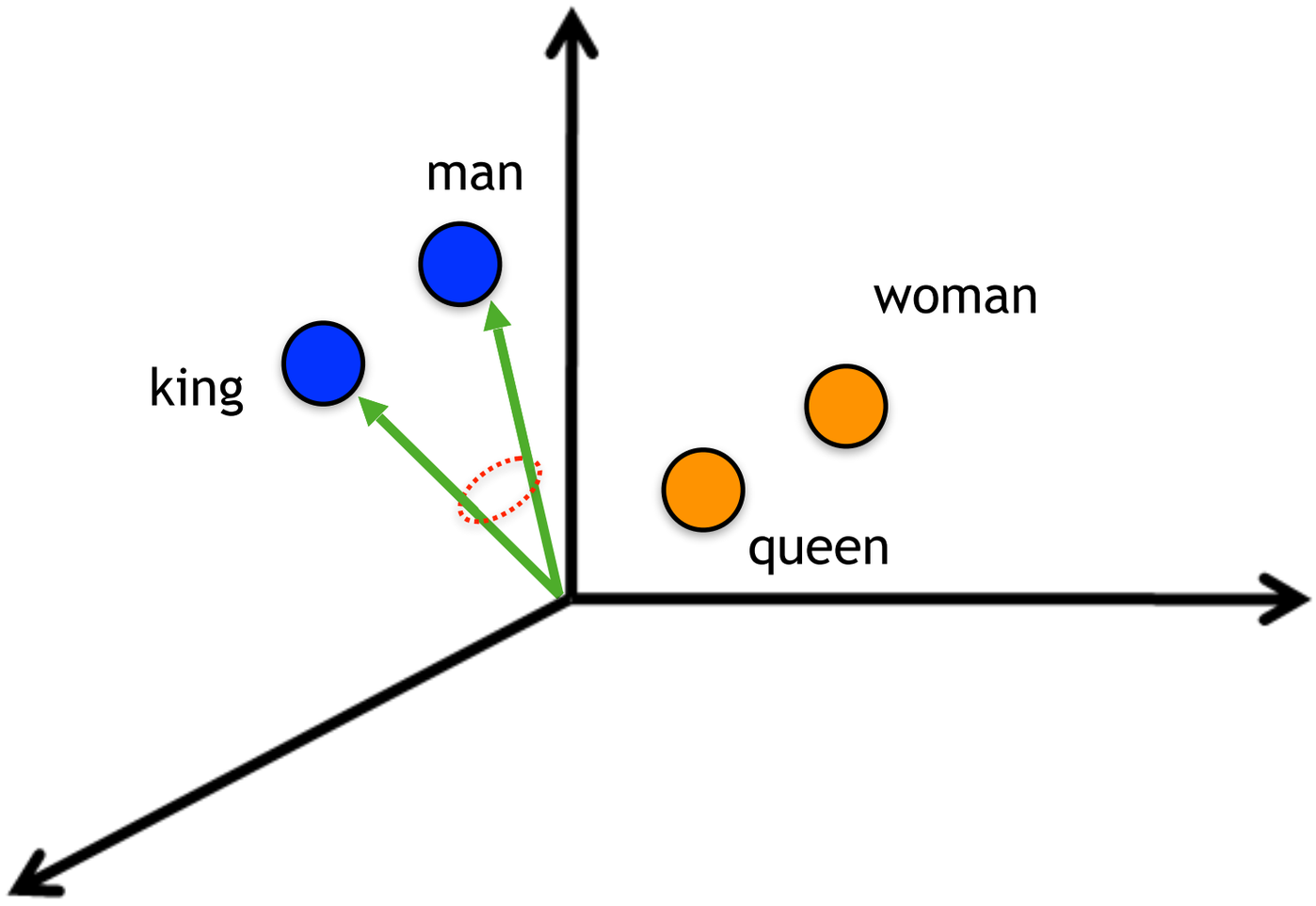
- S: (v) find, happen, chance, bump, encounter (come upon, as if by accident; meet with)
- S: (v) detect, observe, find, discover, notice (discover or determine the existence, presence, or fact of)
- S: (v) find, regain (come upon after searching; find the location of something that was missed or lost)
- S: (v) determine, find, find out, ascertain (establish after a calculation,

We need more than semantic **equivalence**

- S: (v) witness, find, see (perceive or be contemporaneous with)
- S: (v) line up, get hold, come up, find (get something or somebody for a specific purpose)
- S: (v) discover, find (make a discovery, make a new finding)
- S: (v) discover, find (make a discovery)
- S: (v) find (obtain through effort or management)

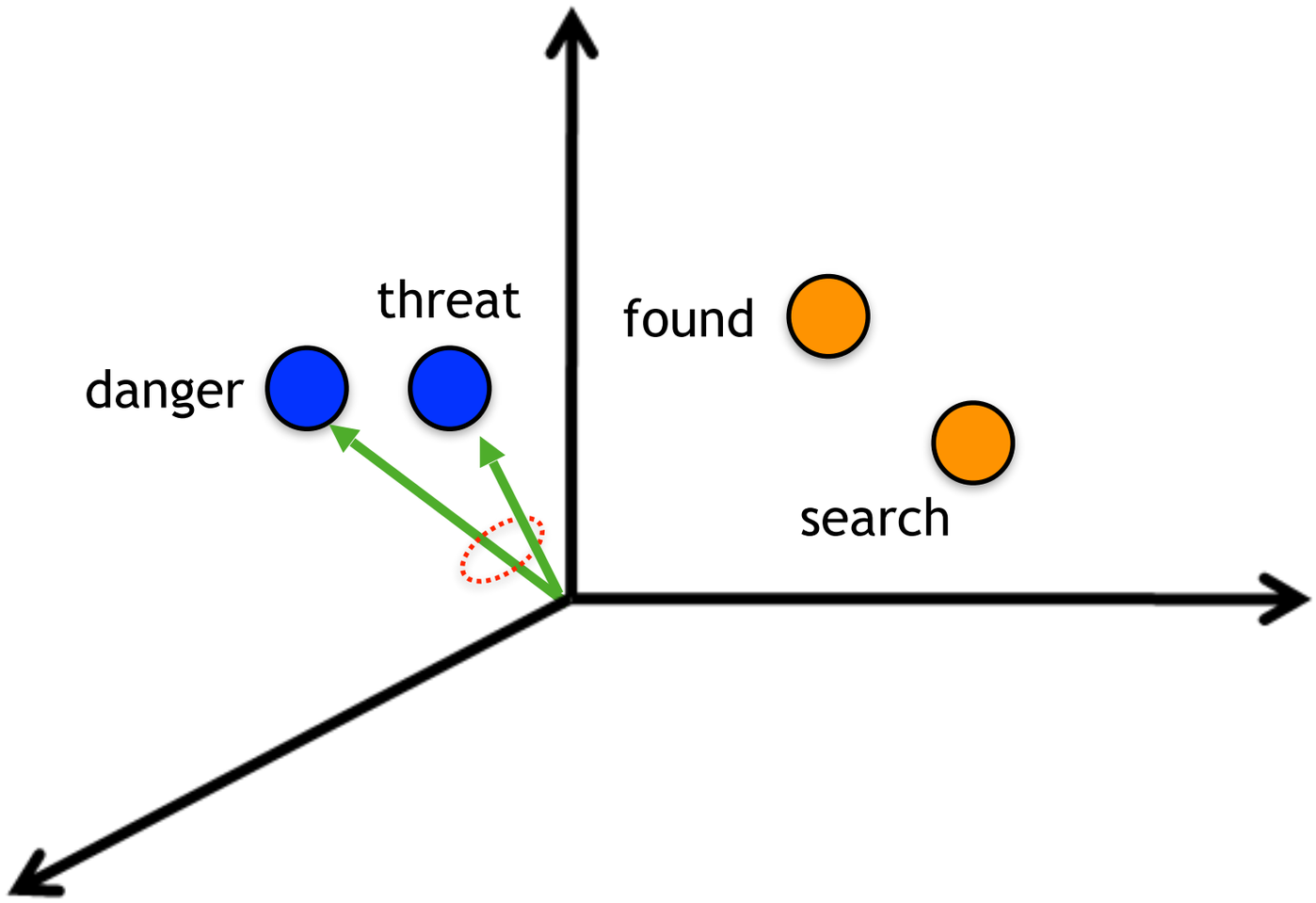
Word embedding

Word embedded in vector space: GloVe, Word2Vec



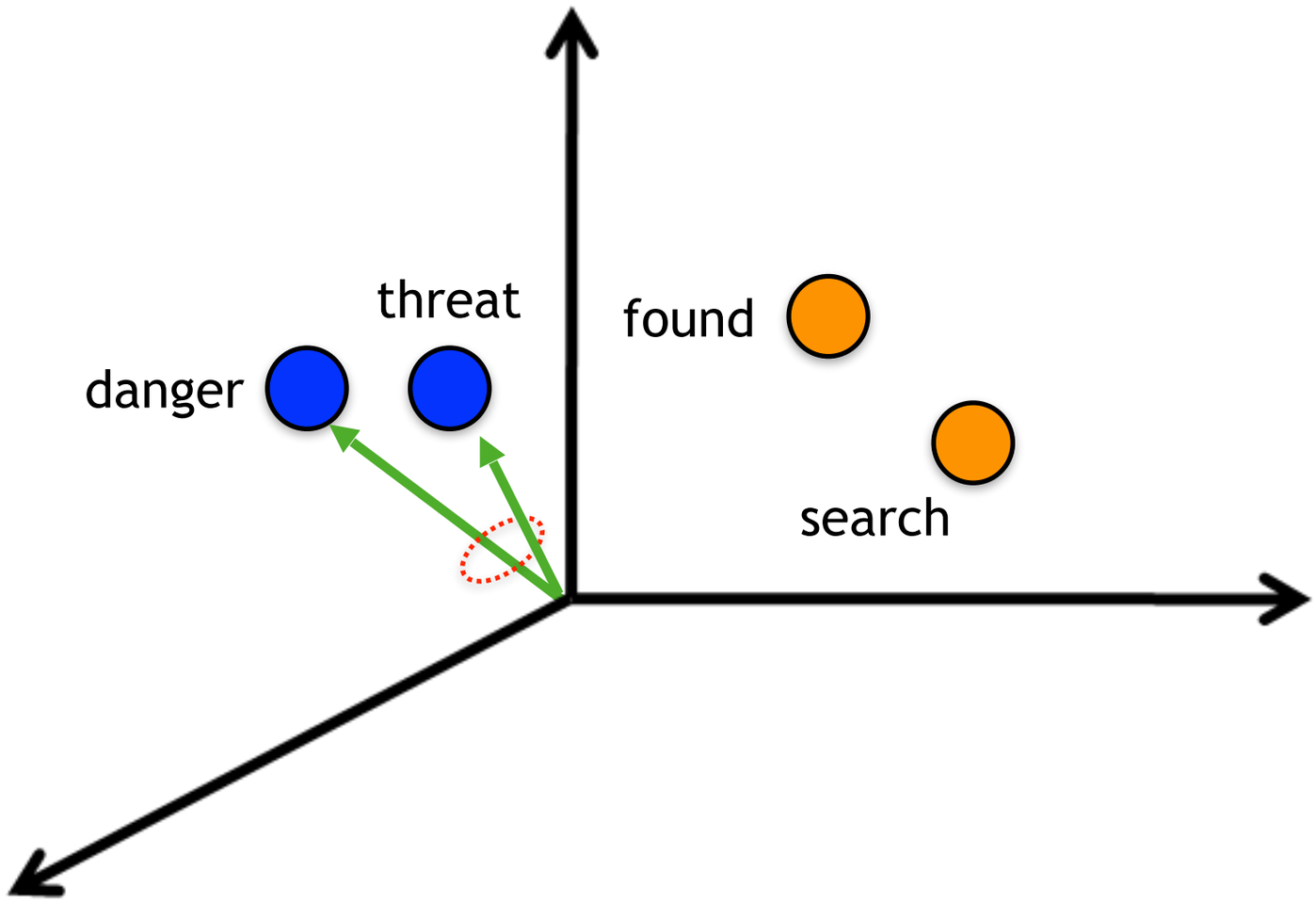
Word embedding

Word embedded in vector space: GloVe, Word2Vec



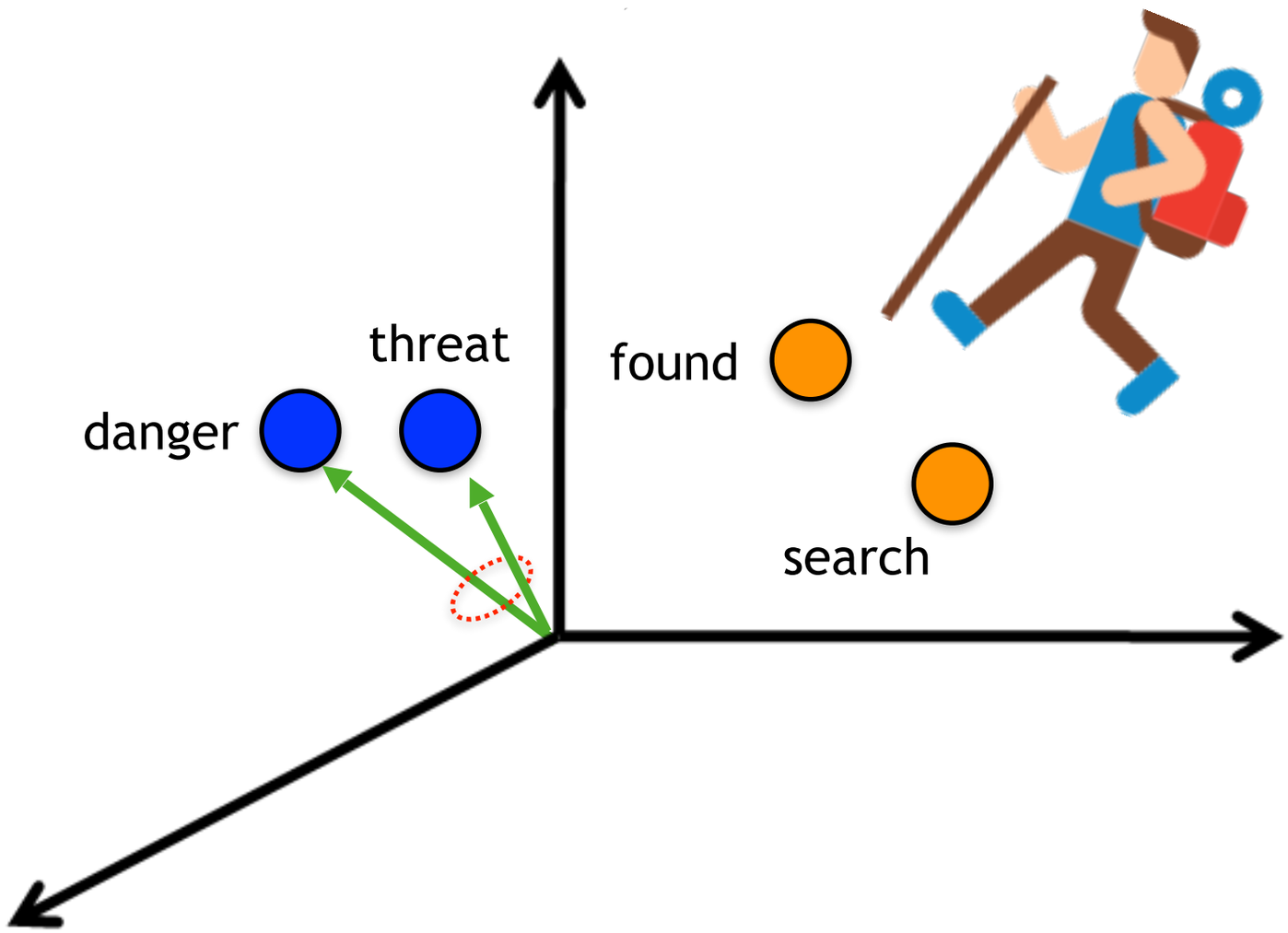
Word embedding

Word mover's distance (Matt J. Kusner et al., 2015)



Word embedding

Word mover's distance (Matt J. Kusner et al., 2015)

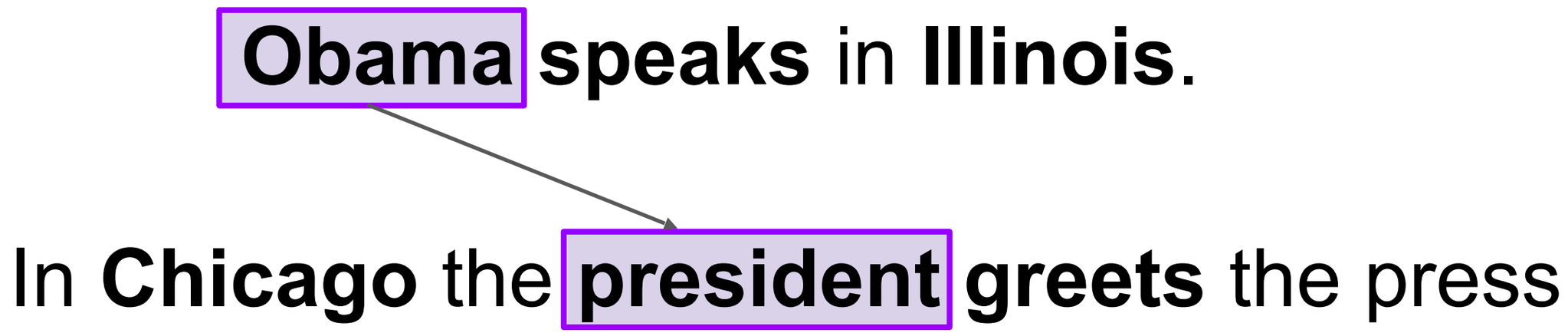


Word Mover's distance

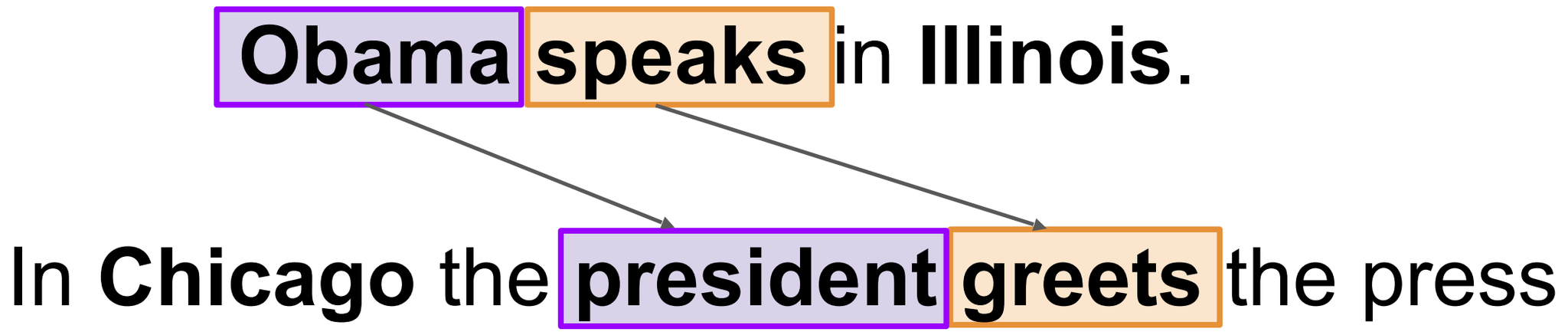
Obama speaks in **Illinois**.

In **Chicago** the **president greets** the press

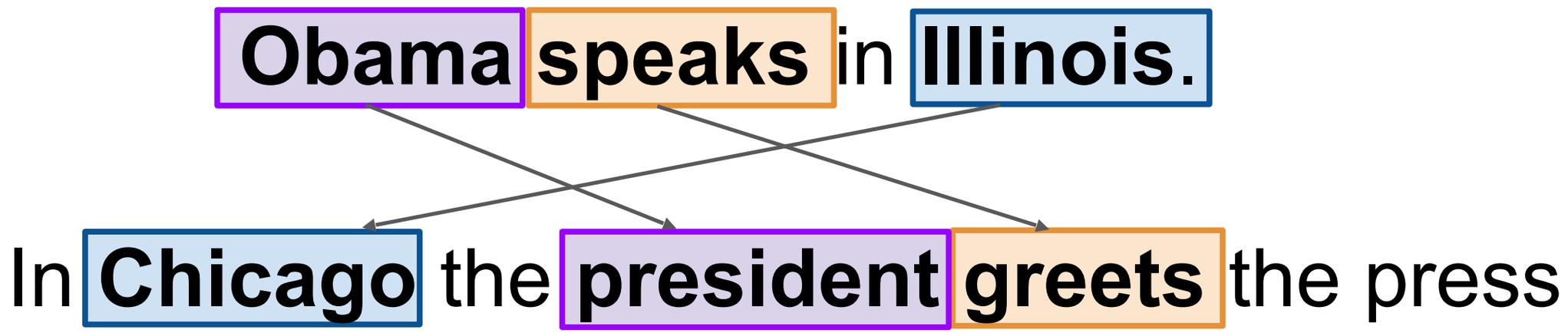
Word Mover's distance



Word Mover's distance



Word Mover's distance



Word Mover's distance

Threat was **found**

Search For Danger

Word Mover's distance

Threat was **found**

Search For Danger

Word Mover's distance

Threat was **found**

Search For Danger

How accurate is Jdoctor?

Experimental setup

6 Popular open-source Java systems

Experimental setup

563 Analyzed Java methods

Experimental setup

829 Manually-written Java conditions

92

%

Precision

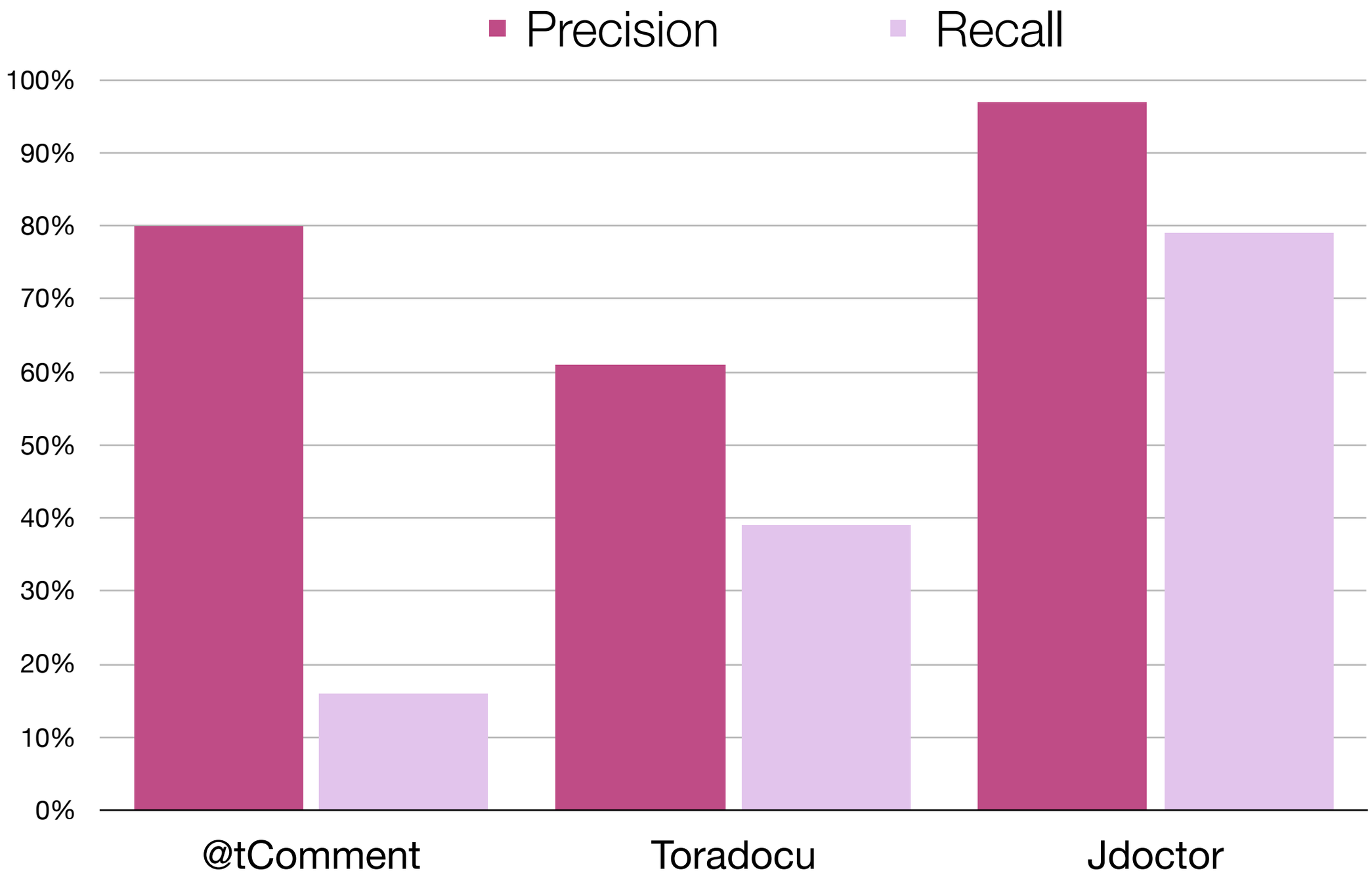
92%

Precision

83%

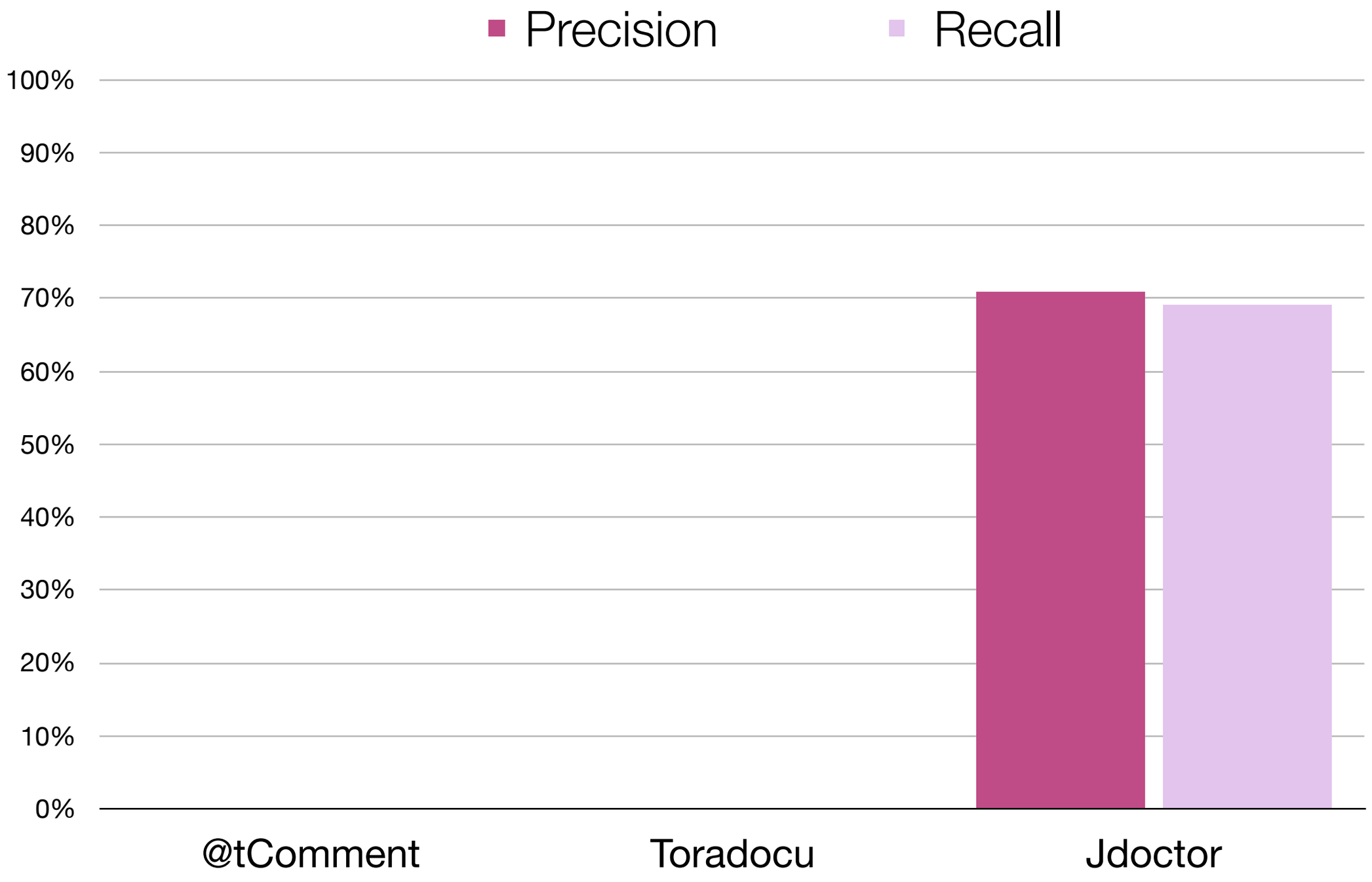
Recall

Exceptional Postconditions

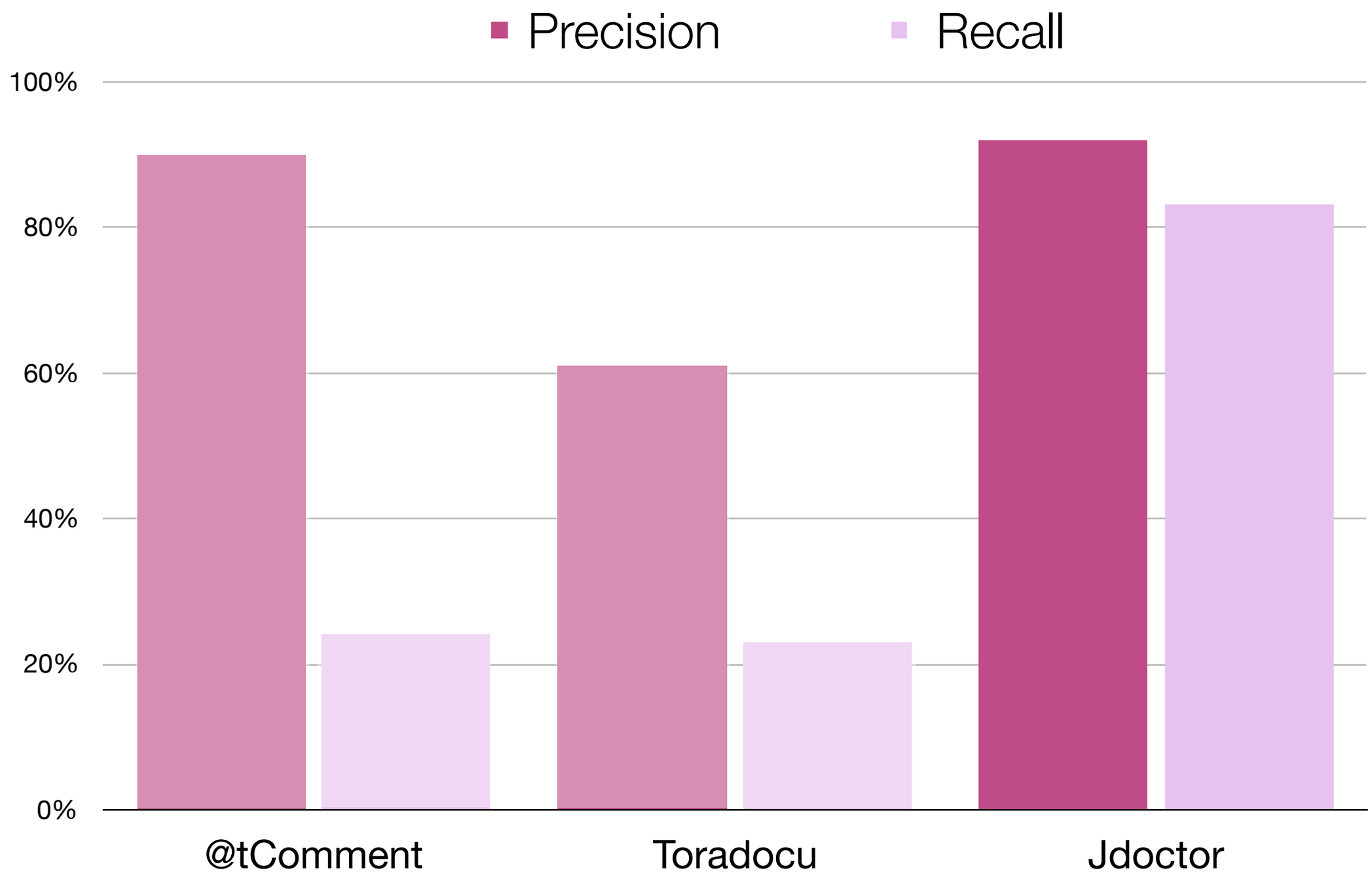


Preconditions

Normal Postconditions



Overall accuracy



How can you use Jdoctor?

Automatic Test Case Generator

+

Jdoctor

Randoop

Automatic unit test generation for Java

+

Jdoctor

Randoop+Jdoctor reclassification

passing

passing

failing

invalid

Randoop+Jdoctor reclassification

passing

passing

failing

invalid

Randoop+Jdoctor reclassification

+ Jdoctor

passing

passing

failing

invalid

Randoop+Jdoctor reclassification

+ Jdoctor

passing → invalid

passing

failing

invalid

Randoop+Jdoctor reclassification

+ Jdoctor

passing → invalid **= invalid**

passing

failing

invalid

Randoop+Jdoctor reclassification

+ Jdoctor

passing → invalid **= invalid**

passing → failing

failing

invalid

Randoop+Jdoctor reclassification

+ Jdoctor

passing	→	invalid	**= invalid**
passing	→	failing	**= missed alarm**
failing			
invalid			

Randoop+Jdoctor reclassification

+ Jdoctor

passing	→	invalid	**= invalid**
passing	→	failing	**= missed alarm**
failing	→	passing	
invalid			

Randoop+Jdoctor reclassification

+ Jdoctor

passing	→	invalid	**= invalid**
passing	→	failing	**= missed alarm**
failing	→	passing	**= false alarm**
invalid			

Randoop+Jdoctor reclassification

+ Jdoctor

passing	→	invalid	**= invalid**
passing	→	failing	**= missed alarm**
failing	→	passing	**= false alarm**
invalid	→	passing	

Randoop+Jdoctor reclassification

+ Jdoctor

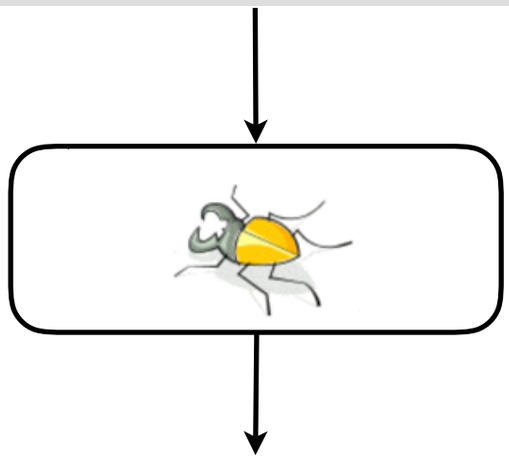
passing	→	invalid	**= invalid**
passing	→	failing	**= missed alarm**
failing	→	passing	**= false alarm**
invalid	→	passing	**= new test**

Missed alarm

```
public static boolean removeAll(List myList){…}
```

Missed alarm

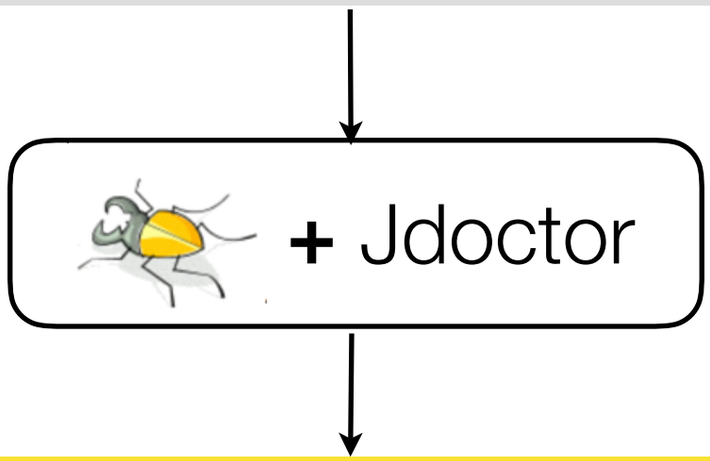
```
public static boolean removeAll(List myList){…}
```



```
@Test
public void testRemoveAll(){
     List<Integer> list = Arrays.asList(1, 2, 3);
     boolean result = removeAll(list);

}
```

Missed alarm

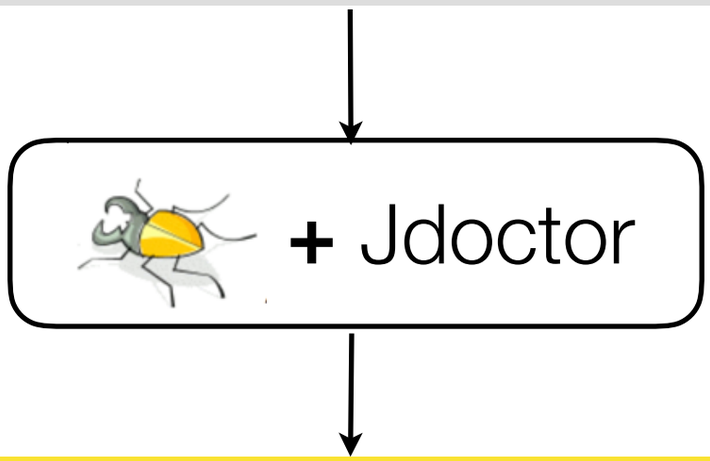
```
/**
 * @return true if the list is empty
 */
public static boolean removeAll(List myList){…}
```




```
@Test
public void testRemoveAll(){
     List<Integer> list = Arrays.asList(1, 2, 3);
     boolean result = removeAll(list);

}
```

Missed alarm

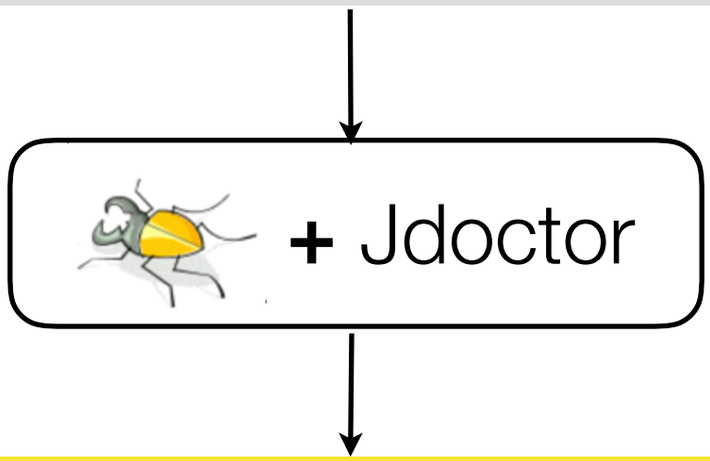
```
/**
 * @return true if the list is empty
 */
public static boolean removeAll(List myList){…}
```




```
@Test
public void testRemoveAll(){
     List<Integer> list = Arrays.asList(1, 2, 3);
     boolean result = removeAll(list);
     if(list.isEmpty()) assert result==true;
}
```

Missed alarm

```
/**
 * @return true if the list is empty
 */
public static boolean removeAll(List myList){…}
```



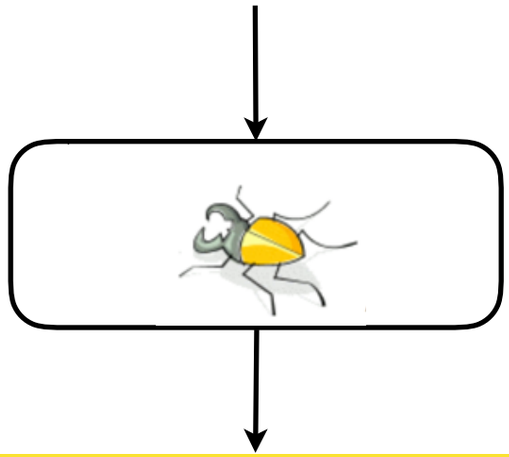

```
@Test
public void testRemoveAll(){
      List<Integer> list = Arrays.asList(1, 2, 3);
      boolean result = removeAll(list);
      if(list.isEmpty()) assert result==true;
}
```

Invalid test

```
public static float max(float[] floatArray){…}
```

Invalid test

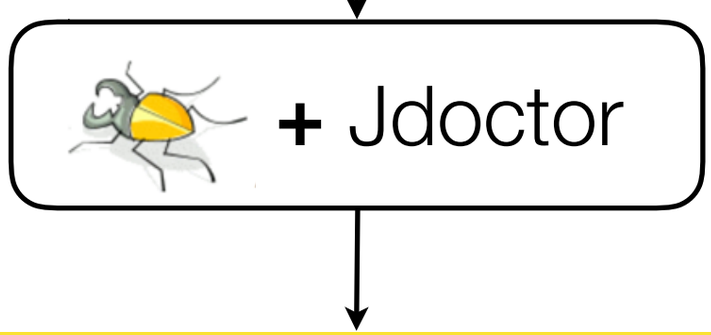
```
public static float max(float[] floatArray){…}
```



```
@Test
public void testRemoveAll(){
        float[] floatArray0 = null;
        float float1 = max(floatArray0);
}
```

Invalid test

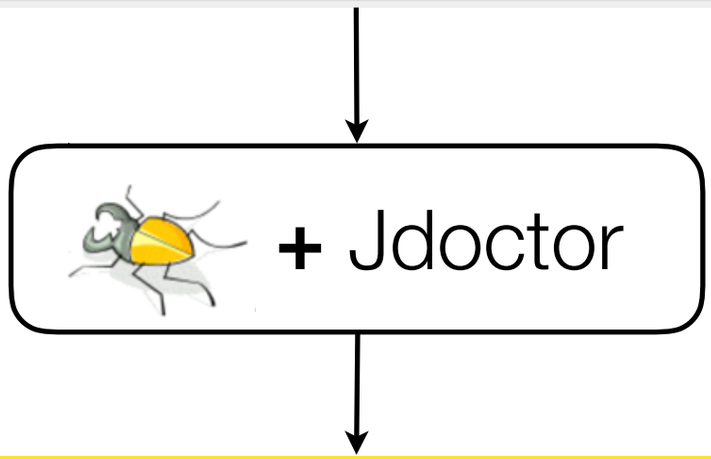
```
/**
 * @param floatArray array of floats, not null
 */
public static float max(float[] floatArray){…}
```




```
@Test
public void testRemoveAll(){
        float[] floatArray0 = null;
        float float1 = max(floatArray0);
}
```

Invalid test

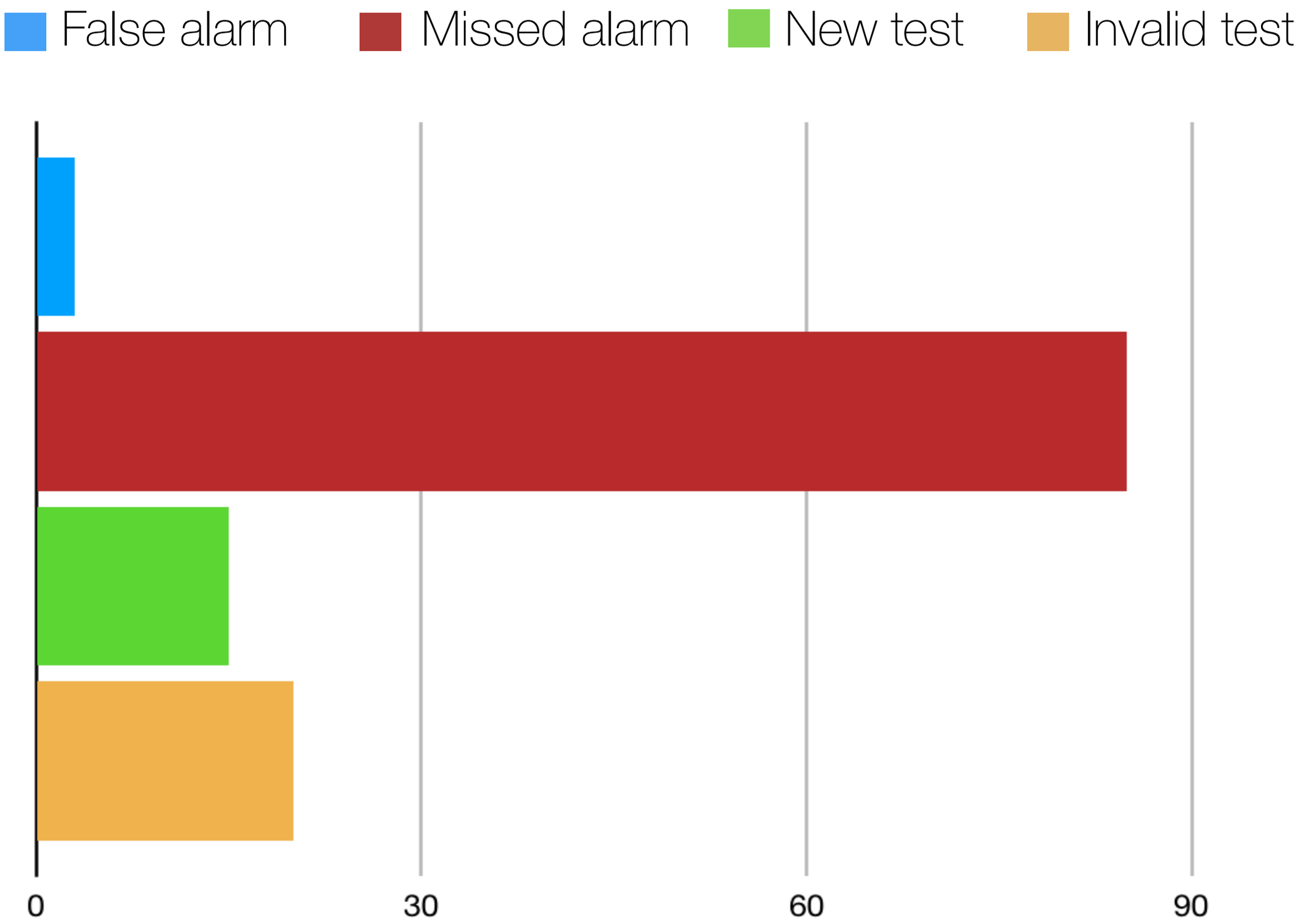
```java
/**
 * @param floatArray array of floats, not null
 */
public static float max(float[] floatArray){…}
```



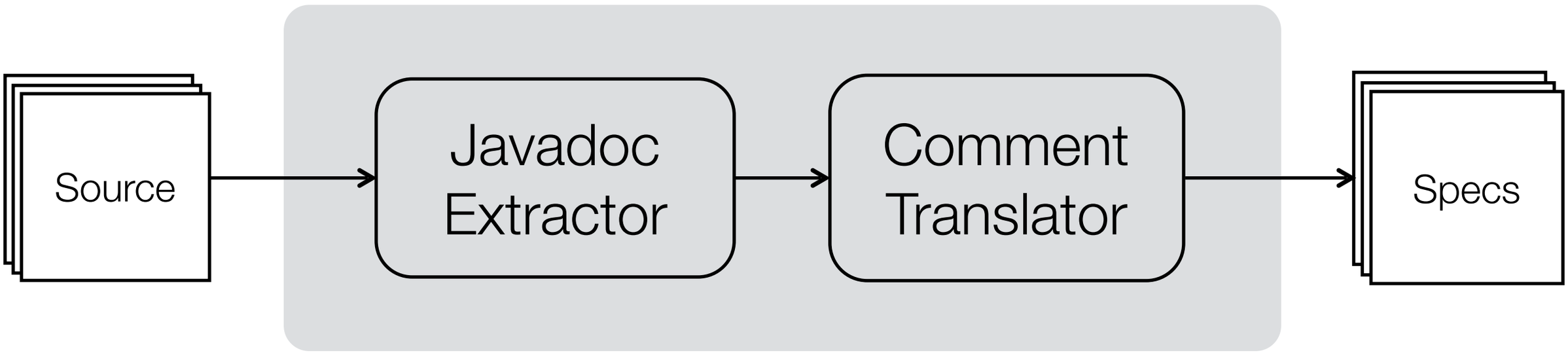

```java
@Test
public void testRemoveAll(){
        float[] floatArray0 = null;
        float float1 = max(floatArray0);
}
```

INVALID

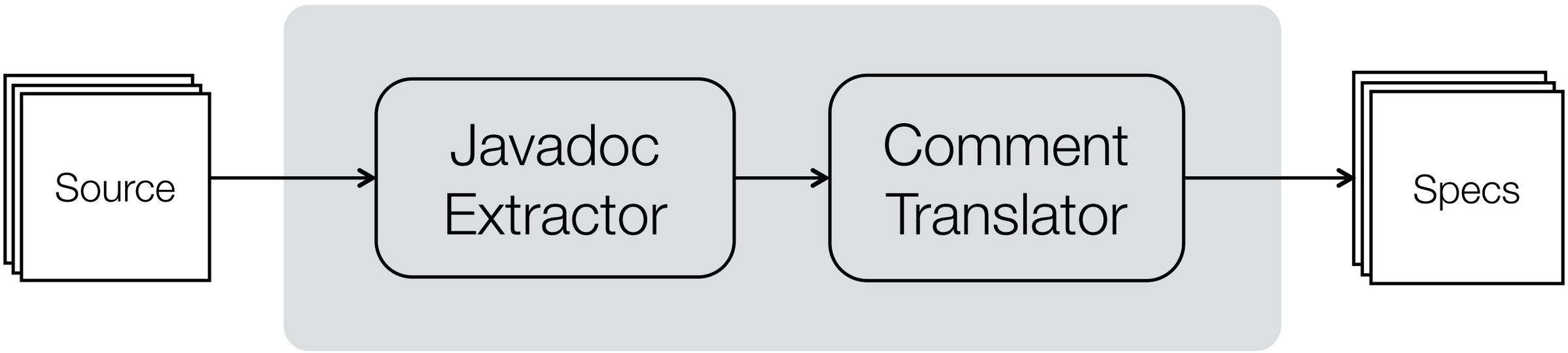
Reclassification



Jdoctor

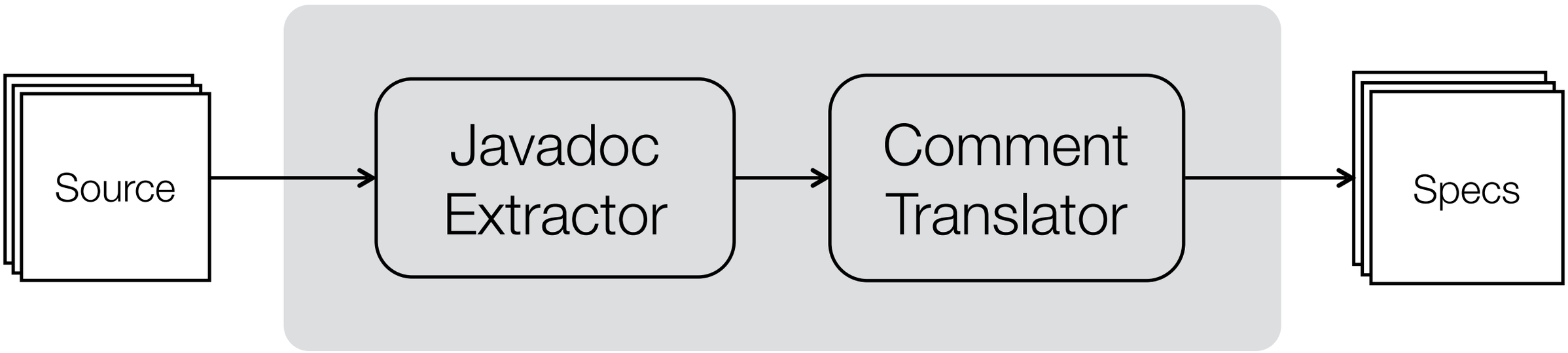


Jdoctor



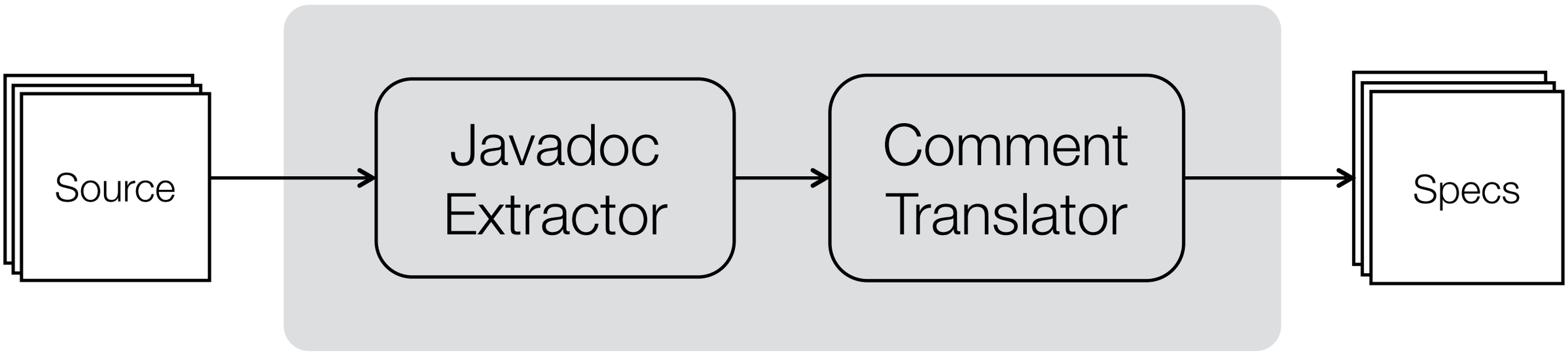
- ✓ Preconditions

Jdoctor



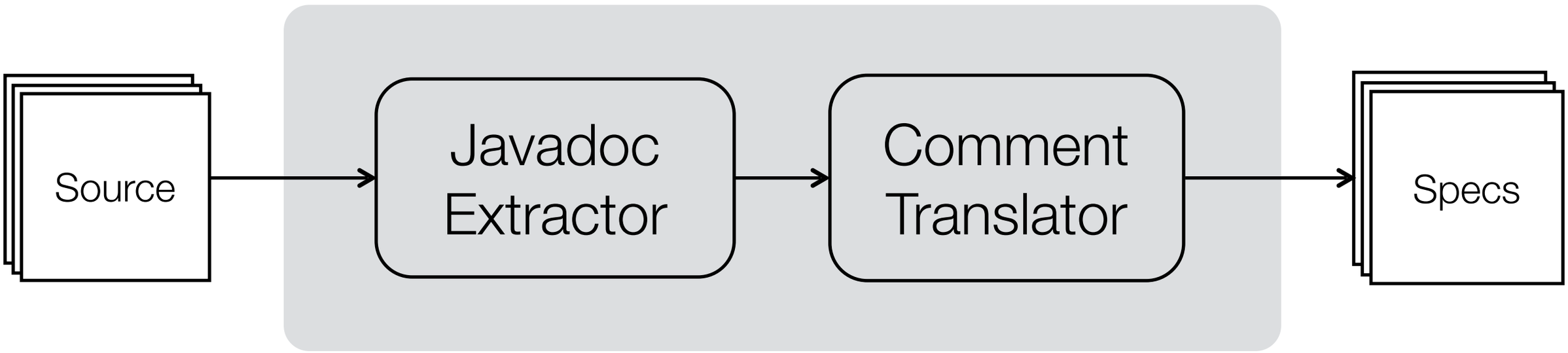
- ✓ Preconditions
- ✓ Normal Postconditions

Jdoctor



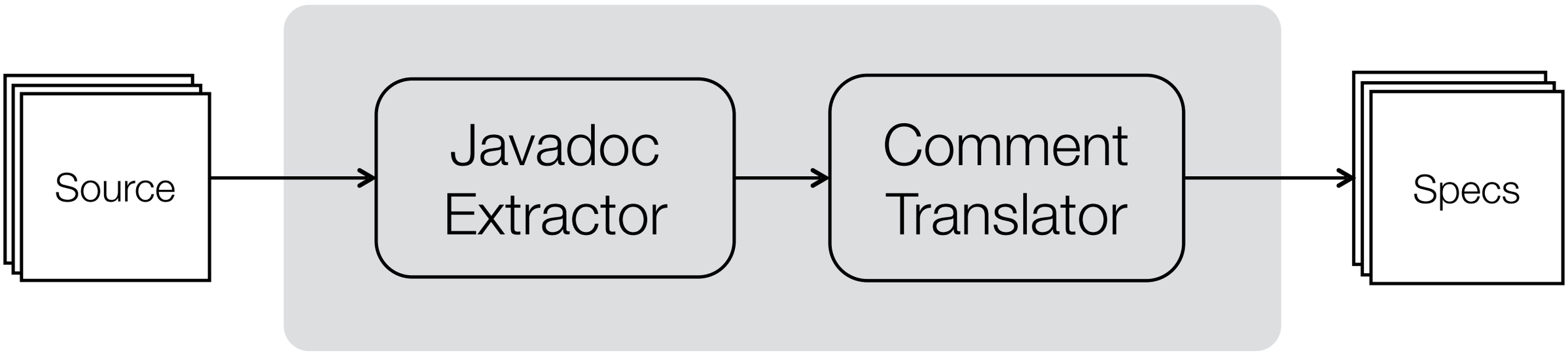
- ✓ Preconditions
- ✓ Normal Postconditions
- ✓ Exceptional Postconditions

Jdoctor



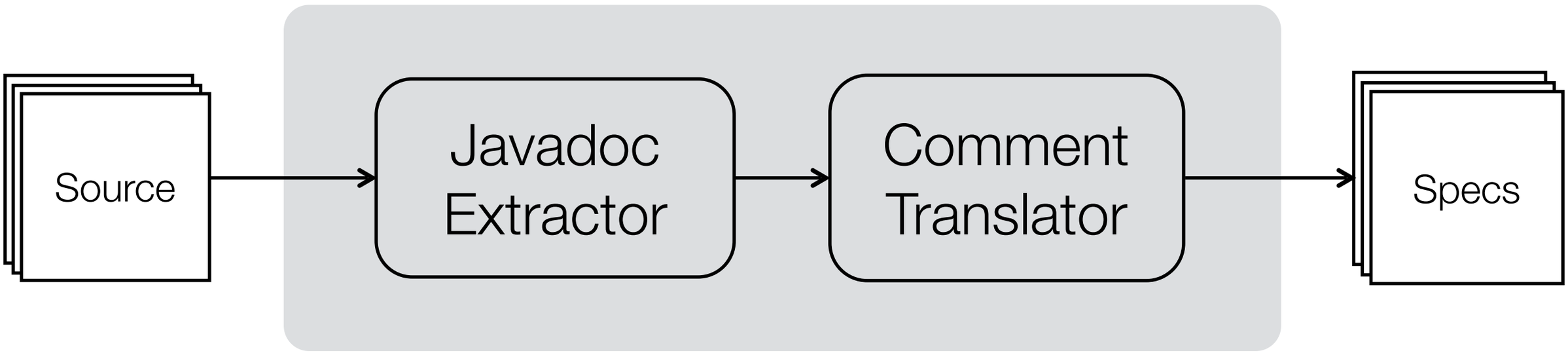
- ✓ Preconditions
- ✓ Normal Postconditions
- ✓ Exceptional Postconditions
- ✓ Pattern Match

Jdoctor



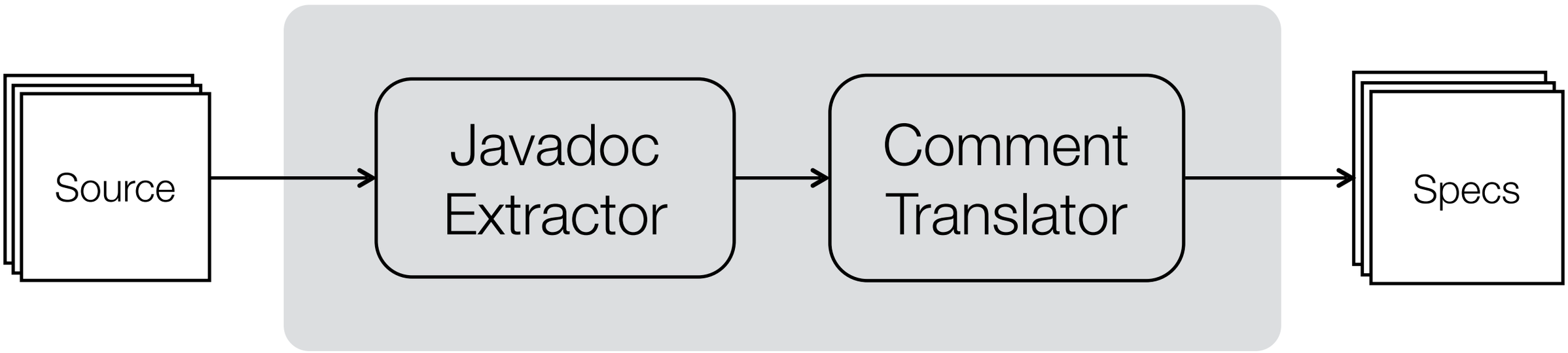
- ✓ Preconditions
- ✓ Normal Postconditions
- ✓ Exceptional Postconditions
- ✓ Pattern Match
- ✓ Syntax Match

Jdoctor



- ✓ Preconditions
- ✓ Normal Postconditions
- ✓ Exceptional Postconditions
- ✓ Pattern Match
- ✓ Syntax Match
- ✓ Semantic Match

Jdoctor



- ✓ Preconditions
- ✓ Normal Postconditions
- ✓ Exceptional Postconditions
- ✓ Pattern Match
- ✓ Syntax Match
- ✓ Semantic Match

https://github.com/albertogoffi/toradocu