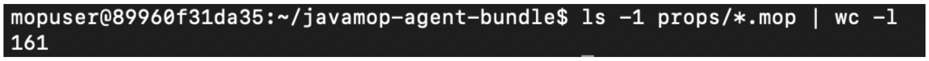
3.2.3

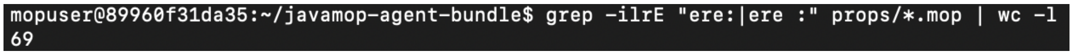
(a) 161



```
mopuser@89960f31da35:~/javamop-agent-bundle$ ls -1 props/*.mop | wc -l
161
```

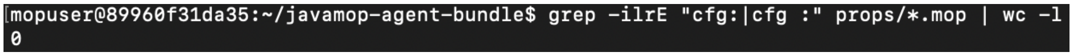
(b)

ERE Answer:
69.



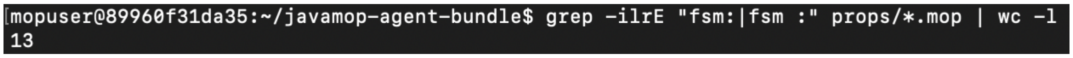
```
mopuser@89960f31da35:~/javamop-agent-bundle$ grep -ilrE "ere:|ere :" props/*.mop | wc -l
69
```

CFG Answer:
0.



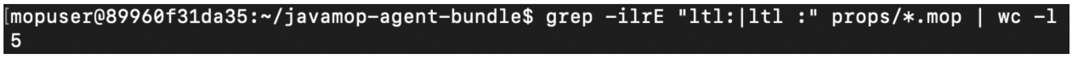
```
[mopuser@89960f31da35:~/javamop-agent-bundle$ grep -ilrE "cfg:|cfg :" props/*.mop | wc -l
0
```

FSM Answer:
13.



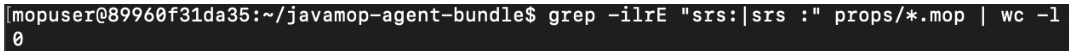
```
[mopuser@89960f31da35:~/javamop-agent-bundle$ grep -ilrE "fsm:|fsm :" props/*.mop | wc -l
13
```

LTL Answer:
5.



```
[mopuser@89960f31da35:~/javamop-agent-bundle$ grep -ilrE "ltl:|ltl :" props/*.mop | wc -l
5
```

SRS Answer:
0.



```
[mopuser@89960f31da35:~/javamop-agent-bundle$ grep -ilrE "srs:|srs :" props/*.mop | wc -l
0
```

(c)

Sum = 87. So, sum does not equal to .mop count in 3a.

74 that are not expressed in the formalisms:

```
props/ArrayDeque_NonNull.mop
props/Arrays_Comparable.mop
props/Arrays_DeepHashCode.mop
props/Arrays_MutuallyComparable.mop
```

```
props/Authenticator_OverrideGetPasswordAuthentication.mop
props/Byte_BadDecodeArg.mop
props/Byte_BadParsingArgs.mop
props/CharSequence_NotInMap.mop
props/CharSequence_NotInSet.mop
props/CharSequence_UndefinedHashCode.mop
props/Character_ValidateChar.mop
props/ClassLoader_UnsafeClassDefinition.mop
props/Closeable_MeaninglessClose.mop
props/Collection_HashCode.mop
props/Collections_Comparable.mop
props/Collections_CopySize.mop
props/Collections_ImplementComparable.mop
props/Collections_UnnecessaryNewSetFromMap.mop
props/Comparable_CompareToNull.mop
props/Comparable_CompareToNullException.mop
props/ContentHandler_GetContent.mop
props/DatagramPacket_Length.mop
props/DatagramPacket_SetLength.mop
props/DatagramSocket_Port.mop
props/DatagramSocket_SoTimeout.mop
props/DatagramSocket_TrafficClass.mop
props/Dictionary_NullKeyOrValue.mop
props/EnumMap_NonNull.mop
props/EnumSet_NonNull.mop
props/Enum_NoExtraWhiteSpace.mop
props/File_LengthOnDirectory.mop
props/HttpCookie_Domain.mop
props/HttpCookie_Name.mop
props/IDN_ToAscii.mop
props/InetAddress_IsReachable.mop
props/InetSocketAddress_Port.mop
props/InputStream_MarkReset.mop
props/Integer_BadDecodeArg.mop
props/InvalidPropertiesFormatException_NonSerializable.mop
props/Long_BadDecodeArg.mop
props/Long_BadParsingArgs.mop
props/Map_ItselfAsKey.mop
props/Map_ItselfAsValue.mop
props/MulticastSocket_TTL.mop
props/NetPermission_Actions.mop
props/NetPermission_Name.mop
props/Object_MonitorOwner.mop
props/Object_NoClone.mop
props/PriorityQueue_NonComparable.mop
props/PriorityQueue_NonNull.mop
props/Random_OverrideNext.mop
props/Reader_MarkReset.mop
props/Runnable_OverrideRun.mop
```

```
props/RuntimePermission_NullAction.mop
props/RuntimePermission_PermName.mop
props/Serializable_NoArgConstructor.mop
props/ServerSocket_Backlog.mop
props/ServerSocket_Port.mop
props/ServerSocket_Timeout.mop
props/Set_ItselfAsElement.mop
props/Short_BadDecodeArg.mop
props/Short_BadParsingArgs.mop
props/SocketPermission_Actions.mop
props/Socket_Timeout.mop
props/Socket_TrafficClass.mop
props/SortedSet_Comparable.mop
props/String_UseStringBuilder.mop
props/System_NullArrayCopy.mop
props/TreeMap_Comparable.mop
props/TreeSet_Comparable.mop
props/URLConnection_OverrideGetPermission.mop
props/URLDecoder_DecodeUTF8.mop
props/URLEncoder_EncodeUTF8.mop
props/Vector_InsertIndex.mop
```

Taking look at first 5:

props/ArrayDeque_NonNull.mop:

It impelements single event - insertnull and logs errorrs if null is inserted. Plain java is used in event definition.

props/Arrays_Comparable.mop:

It defines invalid_sort event and iterates over array to check comparability and logs violations. There are java loops and catches classcastexception without labeled formalism.

props/Arrays_DeepHashCode.mop:

It implements a private cycle(…) method to detect recursive references and an invalid_deephadhcode event. Deep-hash logic is in custom java recursion without formal label.

props/Arrays_MutuallyComparable.mop:

It sets up invalid_sort with nested java loops and logs to inspect pair of element with comparator.

props/Authenticator_OverrideGetPasswordAuthentication.mop:

This checks if subclasses of autheticator override getpasswordauthenticator() using reflection. Custom event staticinit plus java checks.

3.4.4

Without JavaMOP:

```
real	0m42.844s
user	0m52.674s
sys	0m1.387s
```

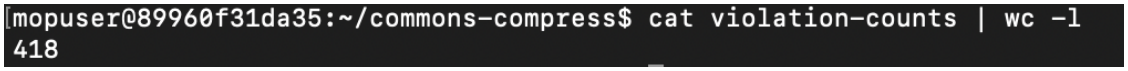
```
With JavaMOP:
```

```
real	3m48.567s
user	4m30.736s
sys	0m2.639s
```

```
Overhead: 5.335
```

3.4.5

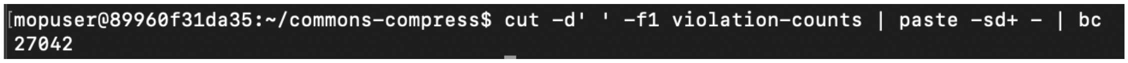
```
How many unique violations are there? 418.
```



```
[mopuser@89960f31da35:~/commons-compress$ cat violation-counts | wc -l
418
```

3.4.6

```
How many times did violations occur?  27042.
```



```
[mopuser@89960f31da35:~/commons-compress$ cut -d' ' -f1 violation-counts | paste -sd+ - | bc
27042
```

3.5.4

1 Specification Collection_UnsafeIterator has been violated on line org.eclipse.jetty.util.thread.QueuedThreadPool.doStop(QueuedThreadPool.java:128). Documentation for this property can be found at https://github.com/SoftEngResearch/tracemop/tree/master/scripts/props/Collection_UnsafeIterator.mop

3.6.1

JavaMOP sees a loop over a collection that can be modified by other threads and that is why it triggers the warning. JavaMOP flags any iteration of collection that might be changed concurrently without explicit synchronization or concurrency checks.

3.6.2

I ran QueuedThreadPoolTest#testMaxStopTime with JavaMOP. Then I checked violation-counts which was pointing at line 128. I inspected line 128 in /home/mopuser/jetty.project/jetty-util/src/main/java/org/eclipse/jetty/util/thread/QueuedThreadPool.java. _threads is a concurrent linkedqueue<Thread> and noticed that iteration is flagged for possible concurrent modification.

3.6.3

This is false alarm, because concurrentlinkedqueue is safe for concurrent iteration. JavaMOP's rule triggers as it sees a loop iteration without explicit synchronization.

3.6.4

I do not think this is a bug so it is okay to ignore this. We can add synchronization but I think it is unnecessary.

3.7.1

At first it was hard to see if concurrentlinkedqueue iteration was truly risky or a false alarm. I had to check if other threads could modify _threads during doStop().

3.7.2

I think it took me about half an hour. Manually inspecting remains a challenge as RV flags a lot of issues and identifying real bugs might take a lot of time.

3.7.3

Tool could show why spec believes it's unsafe, this might help identifying false alarms quickly.

Grouping violations by pattern can make manual checks easier in large codebase as many violation may have the same cause.

It would be helpful if tool could suggest minimal changes for violations

As the bug I inspected in previous task was false alarm, I think it would be helpful if tool could make users mark violations as safe to silence the warning.