
Algorithm 1: Checking compatibility

```
1  $L \leftarrow \{l_a, l_b, l_c, \dots\}$ 
   /* Set of licenses */
2  $C \leftarrow \{C_{l_a}, C_{l_b}, \dots\}$ 
   /* The way to reblicense derivative works,  $C_{li}=1$  means permissive
      license, 2 means file-level weakly restricted license License,
      3 means library-level weakly restricted license, and 4 means
      restricted license */
3  $P \leftarrow \{p_{Interaction}, p_{Retain}, p_{Enhance}, p_{Modification}, p_{Acceptance}, p_{Patent\_term}\}$ 
   /* The set of compatibility determination conditions */
4  $SP_{li} \leftarrow \{p | p \in P, li \in L\}$ 
   /* The open source license  $li$  contains the set of decision
      conditions in  $P$  */
5  $FV_{li} \leftarrow \{lx | lx \in L\}$ 
   /* Set of compatible subsequent versions of license  $li$  */
6  $SL_{li} \leftarrow \{lm | lm \in L\}$ 
   /* Set of secondary licenses of license  $li$  */
Input:  $l_a, l_b \in L$ 
Output:  $SecondaryCompatibility_{a \rightarrow b}, CombinativeCompatibility_{a \rightarrow b}$ 
7 if  $C_{l_a} == 1 \&\& SP_{l_a} \subseteq SP_{l_b}$  then
8   |  $SecondaryCompatibility_{a \rightarrow b} \leftarrow 1$ 
9 if  $C_{l_a}, C_{l_b} \in \{2, 3, 4\} \&\& (l_b \in FV_{l_a} || l_b \in SL_{l_a})$  then
10  |  $SecondaryCompatibility_{a \rightarrow b} \leftarrow 1$ 
11 if  $C_{l_a} \in \{1, 2, 3\} \&\& C_{l_b} \in \{1, 2\}$  then
12  |  $CombinativeCompatibility_{a \rightarrow b} \leftarrow 1$ 
13 return  $SecondaryCompatibility_{a \rightarrow b}, CombinativeCompatibility_{a \rightarrow b}$ 
```
