

基于负关联规则的软件缺陷挖掘

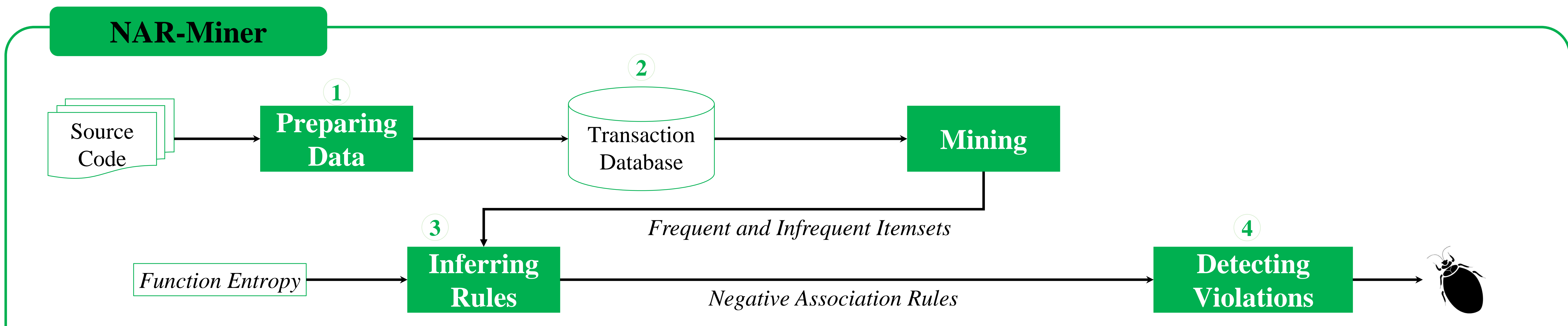
NAR-Miner: Discovering Negative Association Rules from Code for Bug Detection

蔡彦 (yancai@ios.ac.cn) · 中国科学院软件研究所, 计算机科学国家重点实验室

合作者: 边攀、梁彬、石文昌、黄建军, 中国人民大学

软件Bug是普遍存在的, 且很多Bug难以检测。一种方式是通过挖掘程序中元素之间的关系, 并以某个可信度建立元素之间的依赖规则。若程序中出现违反该规则的元素时, 则对应的程序片段有可能是一个Bug。这种方法已经被实践证明是行之有效的。然而, 已有的基于关联规则挖掘的Bug检测方法, 只考虑了程序元素之间的正关联关系, 即当某个元素A出现时, 另一个元素B也应该出现。实际中, 除了正关联规则外, 还存在着程序元素之间的负关联规则, 即当某个程序元素A出现时, 另一个元素B不应该出现。

我们提出了首个基于负关联规则的程序Bug挖掘方法。不同于正关联规则, 任何不相关的两个元素通常会行成负关联规则, 从而导致了负关联规则爆炸问题。我们通过引入信息熵及一种排序算法, 使得所挖掘得到的负关联规则是有意义的。并将之用于Linux内核等开源程序, 从中找出了29个疑似Bug, 其中23个已经被确认为真实Bug。而基于正关联规则的挖掘方法则仅仅检测到了5个被确认的缺陷。



NAR-Miner: focus on mining negative association rules ($A \Rightarrow \neg B$) to detect bugs that contain unexpected program elements. For $A \Rightarrow \neg B$, A and B are frequent itemsets, while $A \cup B$ is an infrequent itemset. Negative rules with low confidence (e.g., lower than min_conf) are filtered out!

$$confidence(A \Rightarrow \neg B) = 1 - \frac{support(A \cup B)}{support(A)}$$

Solve Rule Explosion

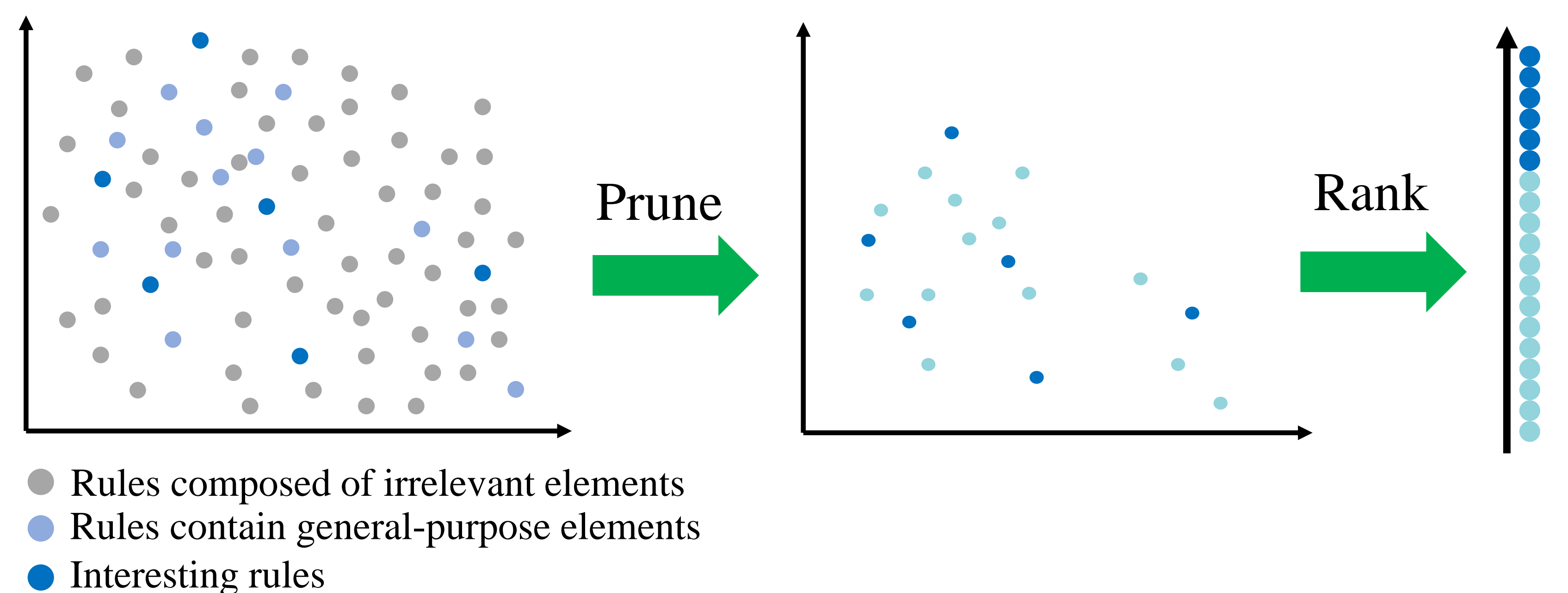
Many mined negative association rules are **uninteresting**, i.e., they do not embody real programming logics. Example: we got > 180,000 negative association rules and about 310,000 violations initially.

- **Pruning:** Introducing **semantic constraints** during mining to reduce negative association rules consist of irrelevant elements

$$interestingness(R) = \frac{confidence(R)}{\sum H(g_i)}$$

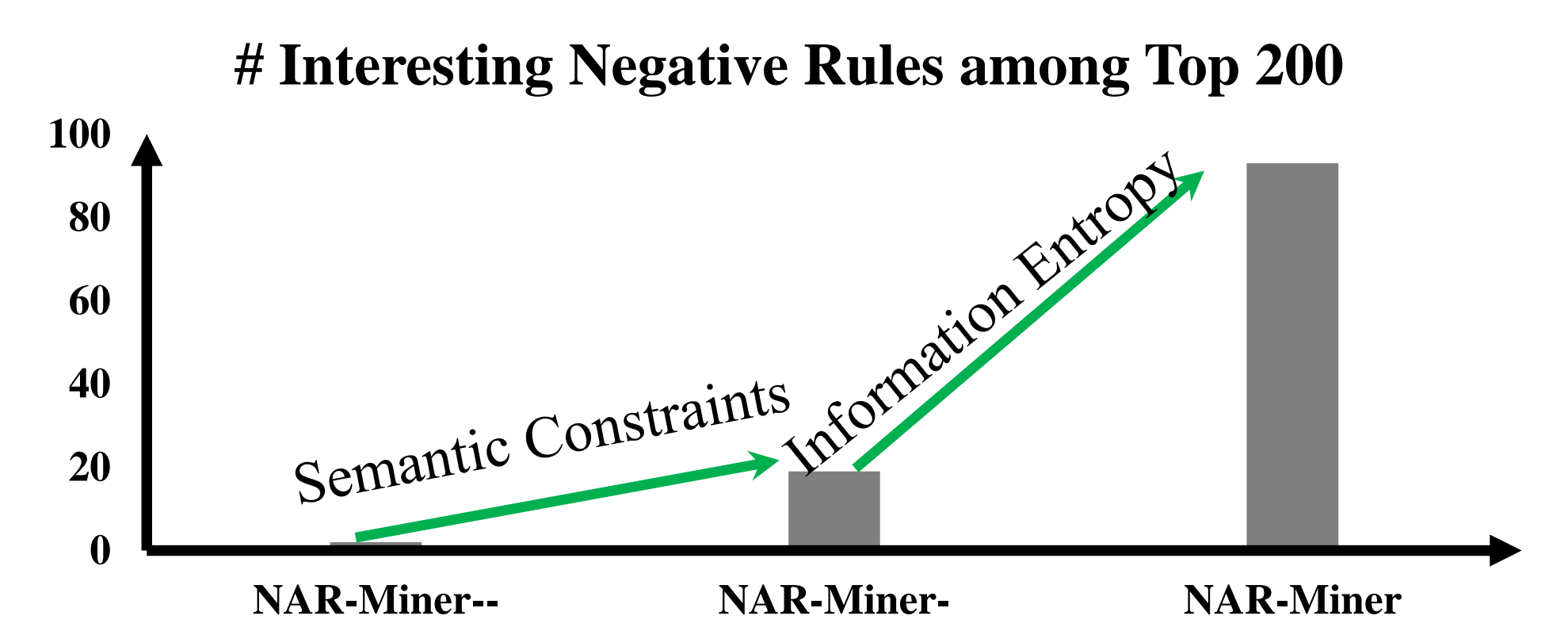
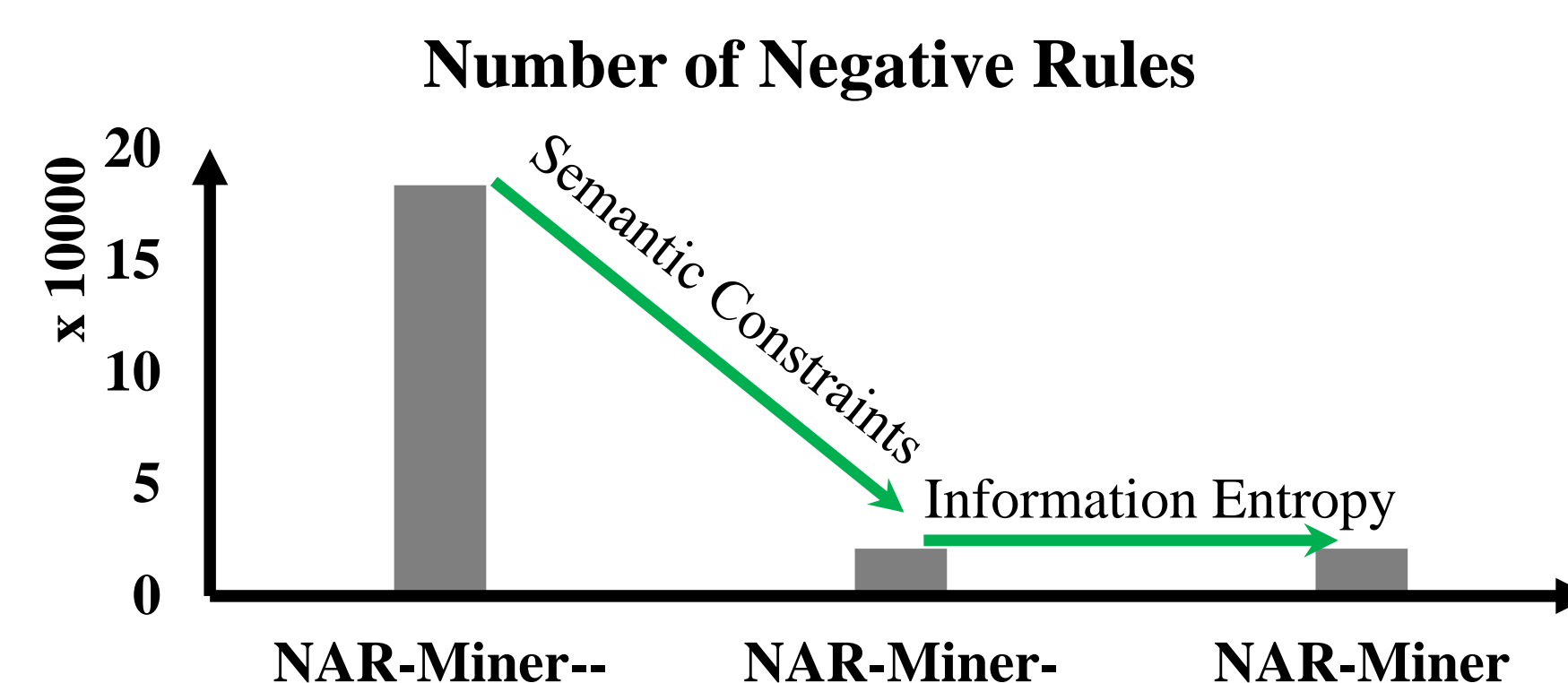
- **Ranking:** Introducing **information entropy** (信息熵) to measure the generality of program elements and further measure the interestingness of rules. Rank rules according to their interestingness

$$H(g) = -\frac{1}{\lg(N)} \sum_{i=1}^N p_i * \log_2(p_i)$$



Experiment

Target Project	Suspicious Bugs	Confirmed Bugs
Linux-v4.12-rc6	23	17
PostgreSQL-v10.3	2	2
OpenSSL-v1.1.1	2	2
FFmpeg-v4.3.2	2	2
Total	29	23



- Detected 29 suspicious bug; 23 confirmed (at time of paper publications)

- Introducing semantic constraints reduces a large number of negative rules (**about 90%**)

- Improve the accuracy of mined rules (**1% ↑ 9.5%**)
- Improve the accuracy of top ranked rules (**9.5% ↑ 46.5%**)