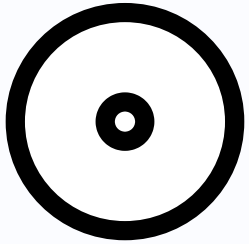
Детекција нарушавања двоструког лиценцирања

Милош Цветановић
Захарије Радивојевић
Саша Стојановић

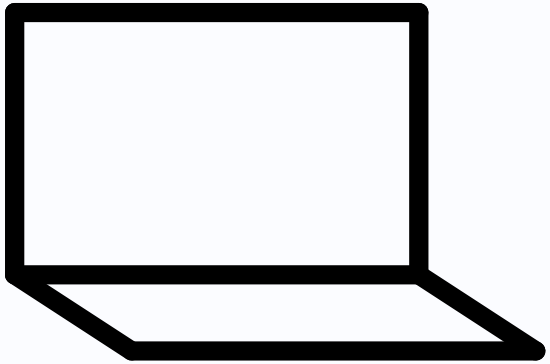
Детекција нарушавања двоструког лиценцирања

- Вишеструка употребљивост софтверског кода
  - Софтверске библиотеке
  - Софтвер “отвореног кода”
  - Двоструко лиценцирање
- Неовлашћена употреба софтверског кода
  - Финансијски губици
- Потреба за откривањем неовлашћене употребе софтверског кода

Дохватање кода

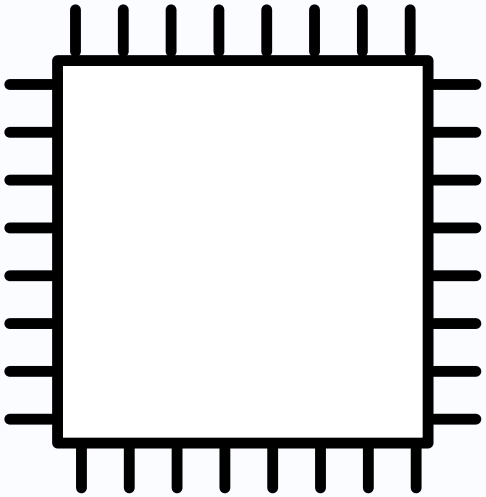


Source code



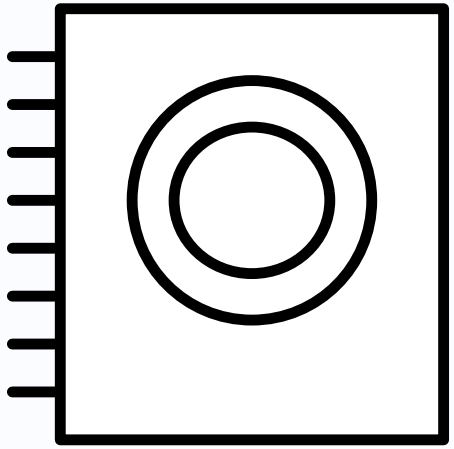
Device

Дохватање кода

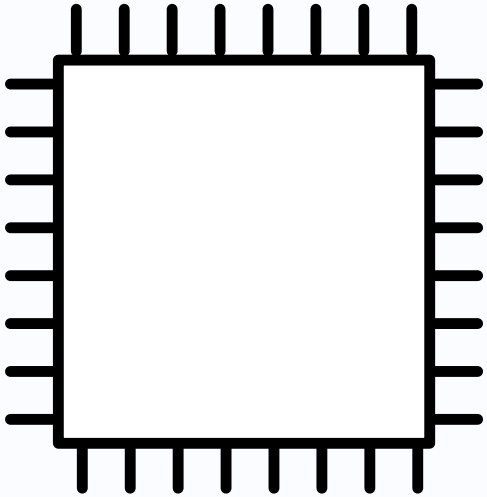


Live chip

Дохватање кода

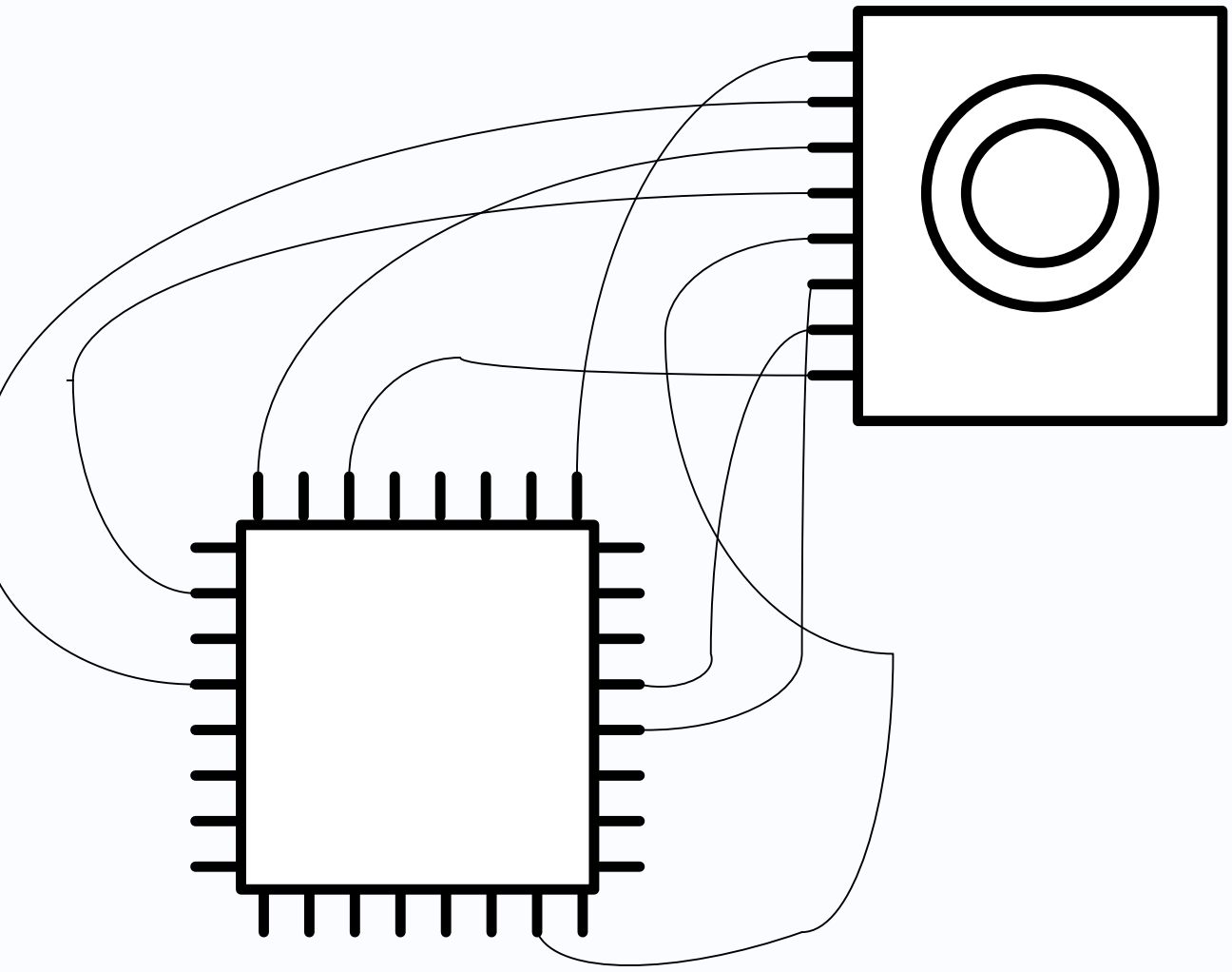


Connection device

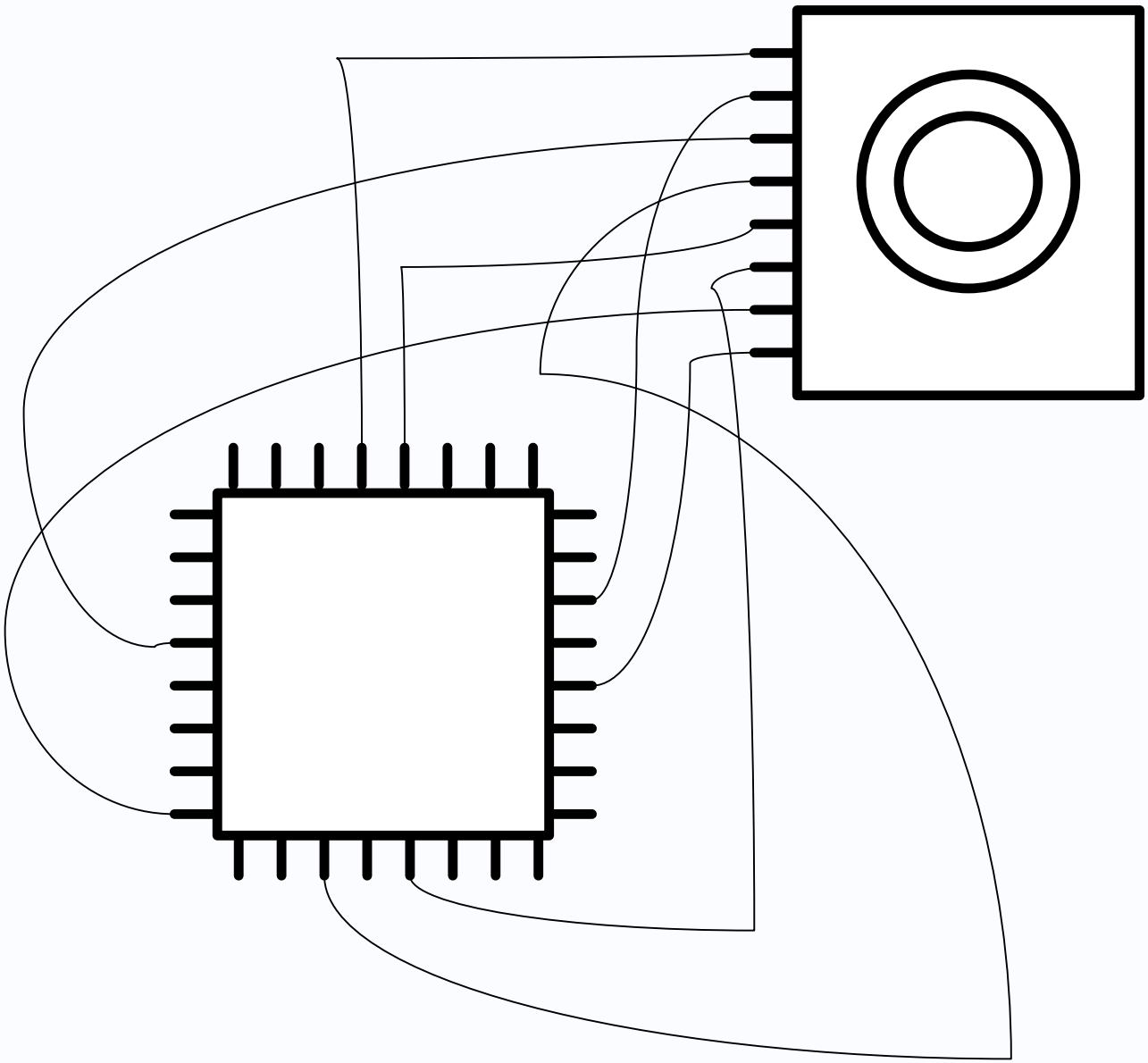


Live chip

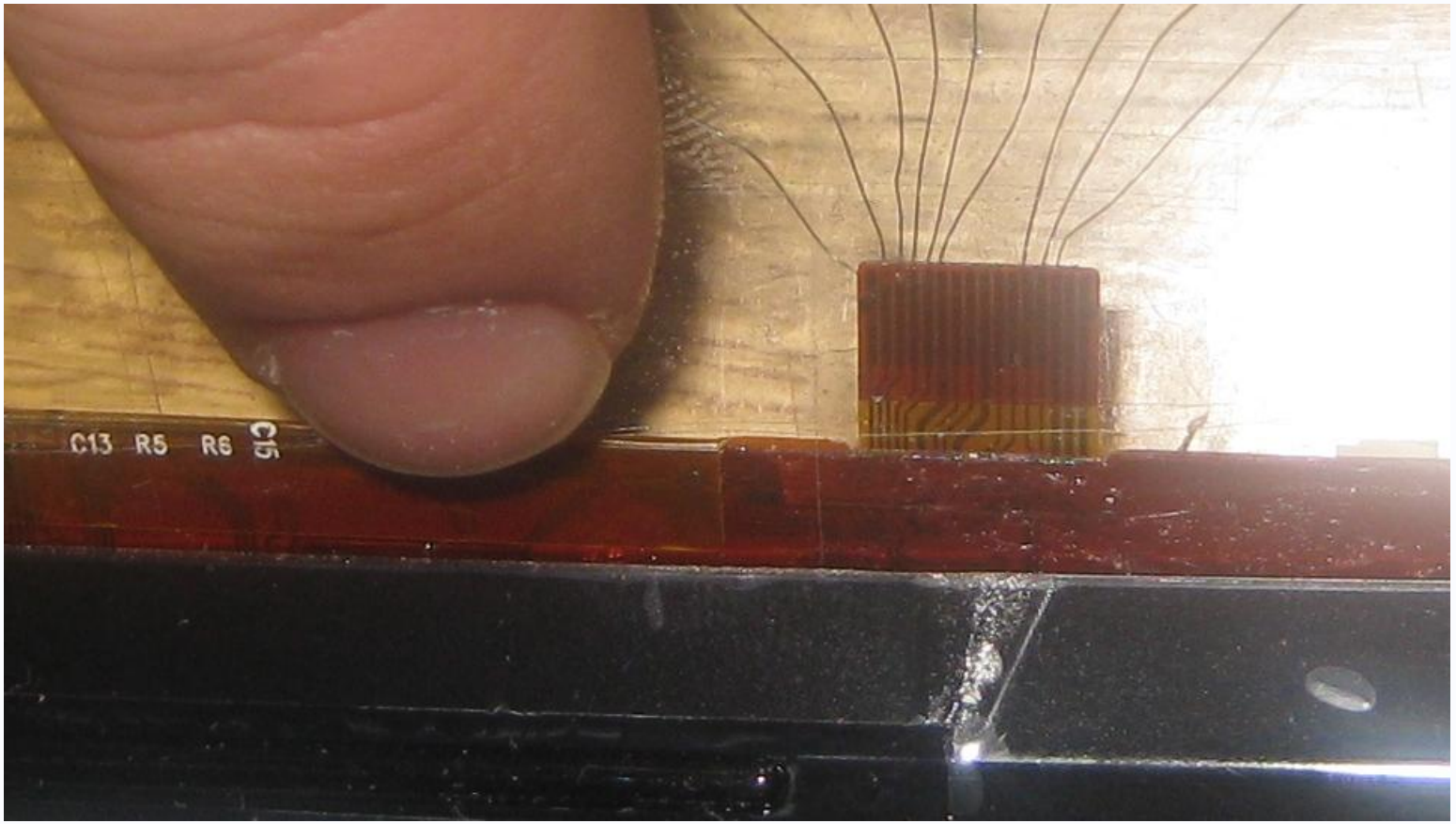
Дохватање кода



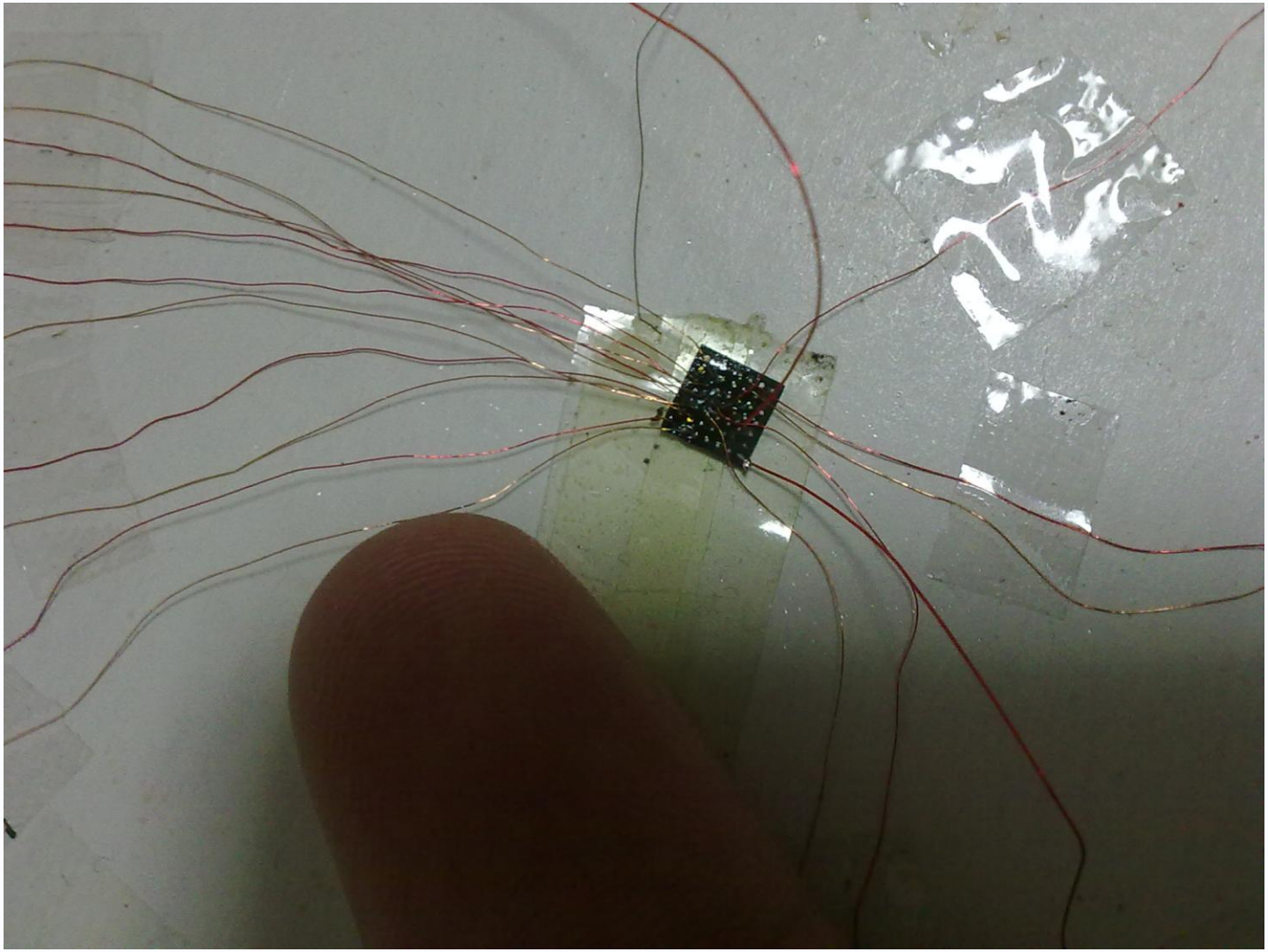
Дохватање кода



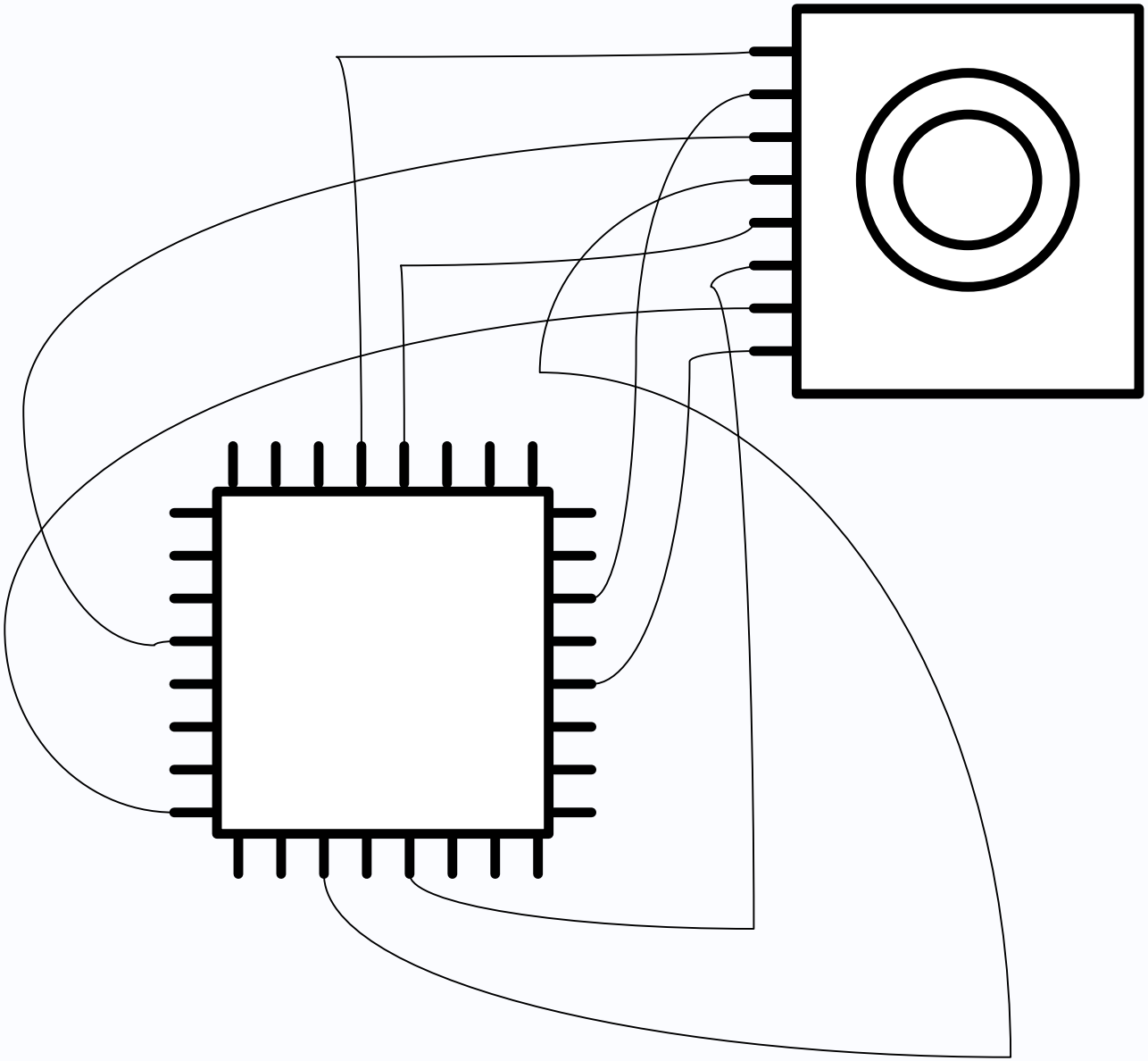
Дохватање кода



Дохватање кода



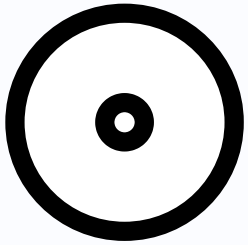
Дохватање кода



01001011011010010

Binary code!

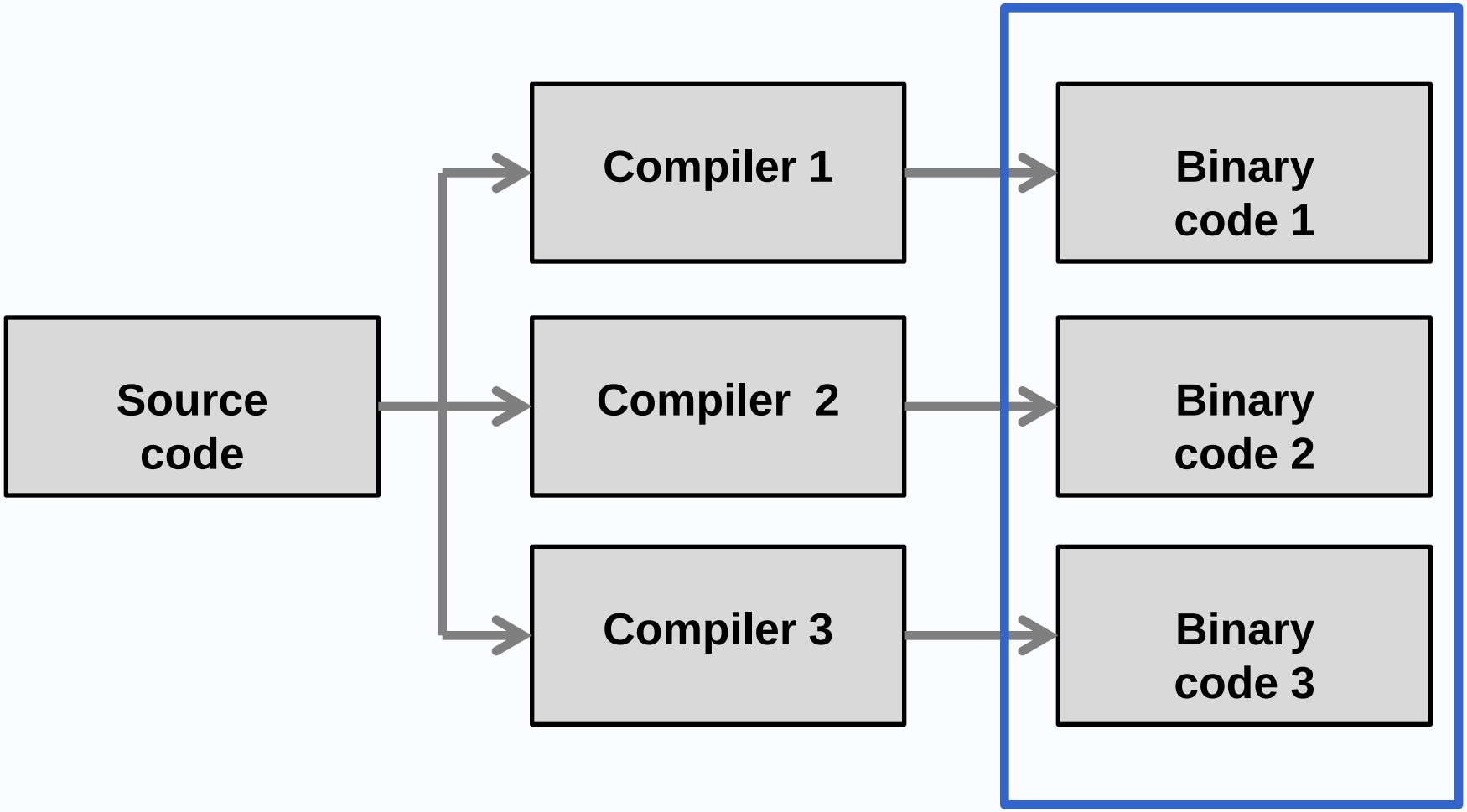
Одређивање сличности кода



01001011011010001O

What compiler?

Одређивање сличности кода



Одређивање сличности кода

- Како одредити да ли одређени бинарни код потиче од одређеног изворног кода?

Декомпозиција проблема

- Код у који је уграђена библиотека доступан само у бинарном облику
- Процес у којем је библиотека преведена није познат:
  - Употребљени преводилац
  - Употребљене опције
  - Окружење у којем је библиотека преведена (остатак изворног кода који користи библиотеку)
- Код уграђен у уређаје – нема додатних информација

Поређење процедура

- Поређење изворног и бинарног кода:
  - Превести и изворни код у бинарни,
    потом применити неку технику
    за поређење бинарног кода
  - Вратити бинарни код у изворни,
    потом применити неку технику
    за поређење изворног кода
  - Трансформисати оба облика кода у неки трећи облик
- Могући резултати поређења:
  - Процењена мера сличности процедура
  - Процењена једнакост процедура

Постојећи приступи

- Приступи који претежно раде на нивоу изворног кода:
  - Приступи за откривање софтверских клонова
  - Приступи за откривање плагијаризма
- Приступи који претежно раде на нивоу бинарног кода:
  - Приступи за откривање нарушавања лиценцних права
  - Приступи за откривање рањивости софтвера
  - Приступи за откривање злонамерног кода
  - Приступи за компакцију кода

Софтверски клонови

- Понављање делова кода у потенцијално измењеном облику
  - Лоша програмерска пракса (одржавање софтвера)
  - Намерна употребе туђег кода (плагијаризам)
- Четири типа клонова:
  - Тип 1
    - Потенцијално измењено форматирање
  - Тип 2
    - Преименовани идентификатори
  - Тип 3
    - Додата, измењена или обрисана инструкција
  - Тип 4
    - Семантички клонови

Приступи за откривање софтверских клонова и плагијаризма

- Према нивоу на којем раде обраду деле се на:
  - Приступи базирани на тексту
  - Приступи базирани на токенима
  - Приступи базирани на софтверским метрикама
  - Приступи базирани на апстрактним синтаксним стаблима
  - Приступи базирани на графовима зависности у програму
  - Хибридни приступи

Приступи за откривање софтверских клонова и плагијаризма

	Текст	Токени	Метрике	AST	PDG
Примери алата	Duploc SimCad	Dup CCFinder	CLAN Davey	Cpdetector Deckard	GPlag Duplix
Ниво језика	Сви	Претежно виши	Претежно виши	Виши	Виши
Зависност од језика	Нема	Висока	Висока	Висока	Висока
Проширивост	Ништа или лексер	Лексер	Лексер или парсер	Парсер	Парсер
Типови клонова	1, 3	1, 2	1, 2, 3, 4	1, 2, 3	1, 2, 3, 4
Комплексност рачунања	Средња	Мала	Средња	Средња	Висока

Приступи за откривање нарушености лиценцних права

- BAT – Binary Analysis Tool
  - Упаривање стринг литерала са скупом познатих
  - Компримовање поређених кодова
  - Рачунање бинарних разлика
- Постојећи алат имплементира само прву технику
- За потребе евалуације имплементирана је друга техника

Приступи за откривање рањивости софтера

- Поређење се користи да се открију разлике нове верзије у односу на претходну верзију истог софтвера
- Верзије су најчешће преведене на исти начин
  - Разлике настају услед измена које уносе програмери
- За потребе евалуације имплементиран је приступ који је предложио Dullien
  - Упаривање процедура које се по селекторима упарују 1-1 (број блокова, број грана у CFG, број позиваних потпрограма)
  - Сужавање скупова из којих се процедуре упарују
  - Процедуре се даље упарују користећи граф позива

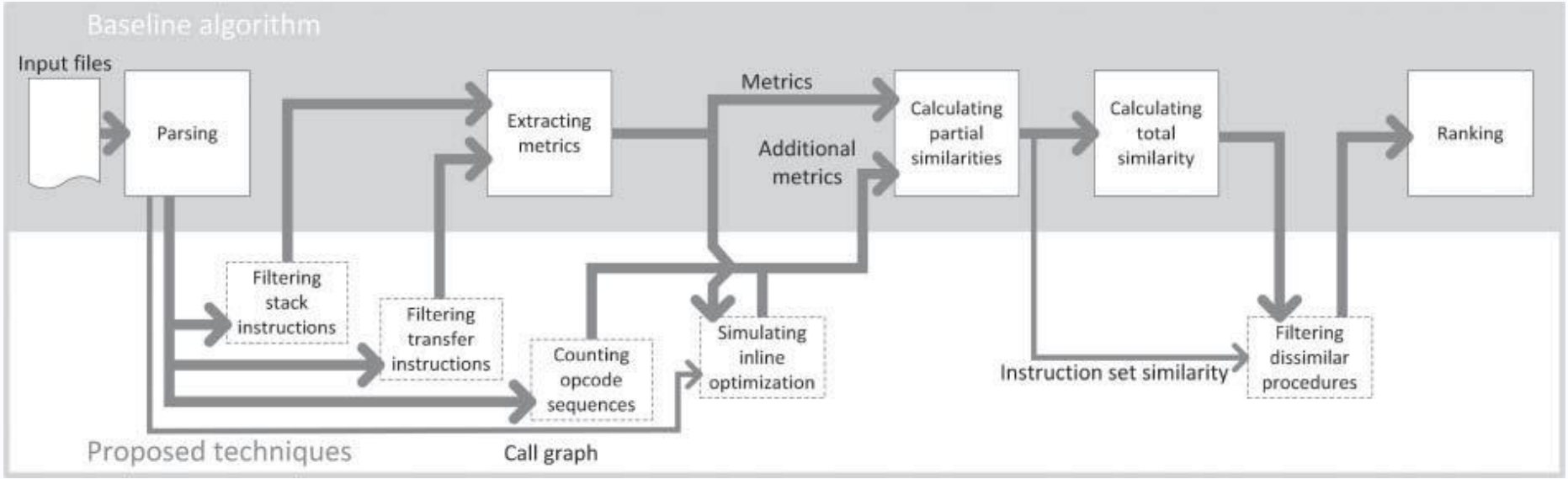
Приступи за откривање злонамерног кода

- На основу понашања познатог злонамерног кода препознати да је неки други код злонамеран
- За потребе евалуације имплементиран је приступ за поређење секвенци инструкција
  - За сваку процедуру се броје појављивања сваке од секвенци инструкција одређене дужине
  - Вектори броја појављивања појединих секвенци се пореде користећи косинусну сличност

Опис експеримента

- Библиотека која се тражи
  доступна у облику изворног кода
- Код који користи библиотеку
  доступан само у бинарном облику
  - Уграђени уређаји –> ARM архитектура
  - Бинарни код прочитан из меморије уређаја
    –> не постоје додатне информације
- Циљ је убрзати откривање коришћења
  библиотеке
  - Открити део кода који потиче
    од изворног кода тражене процедуре
  - Тражена процедура се за потребе тестова
    увек појављује у посматраном бинарном коду

Технике за побољшање резултата



Метрике

Опис	Акроним	Тип вредности	Тип мере	Тип тока
Број инструкција	AIN	S	A	-
Број скокова	ABN	S	A	C
Број позива	ACN	S	A	C
Број петљи	APN	S	A	C
Број аритметичких инструкција	AAN	S	A	D
Број логичких инструкција	ALN	S	A	D
Број инструкција за пренос података	ATN	S	A	D

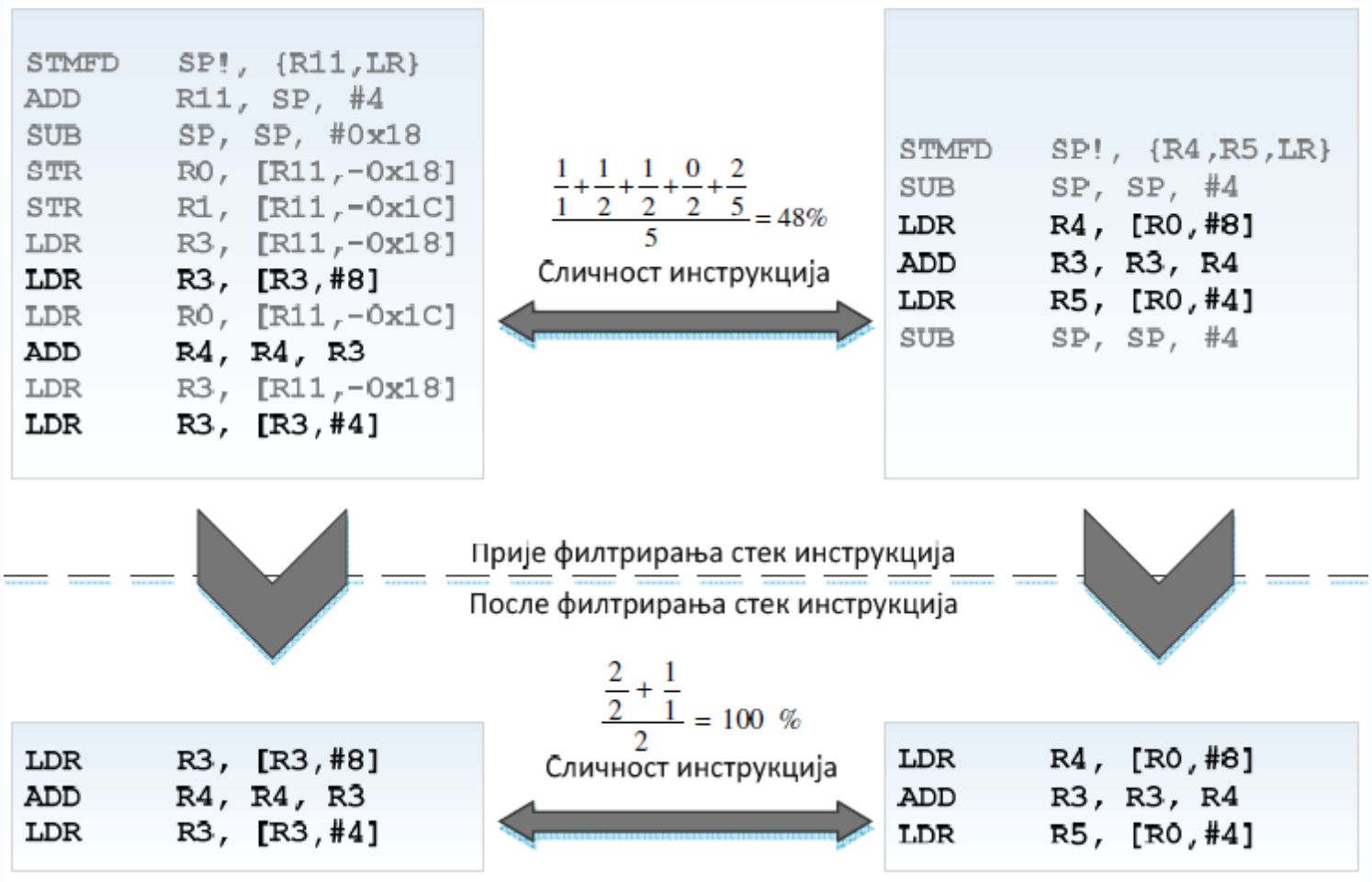
Метрике

Опис	Акроним	Тип вредности	Тип мере	Тип тока
Фреквенција скокова	ABF	S	N	C
Фреквенција позива	ACF	S	N	C
Фреквенција петљи	APF	S	N	C
Фреквенција аритметичких инструкција	AAF	S	N	D
Фреквенција логичких инструкција	ALF	S	N	D
Фреквенција инструкција за пренос података	ATF	S	N	D

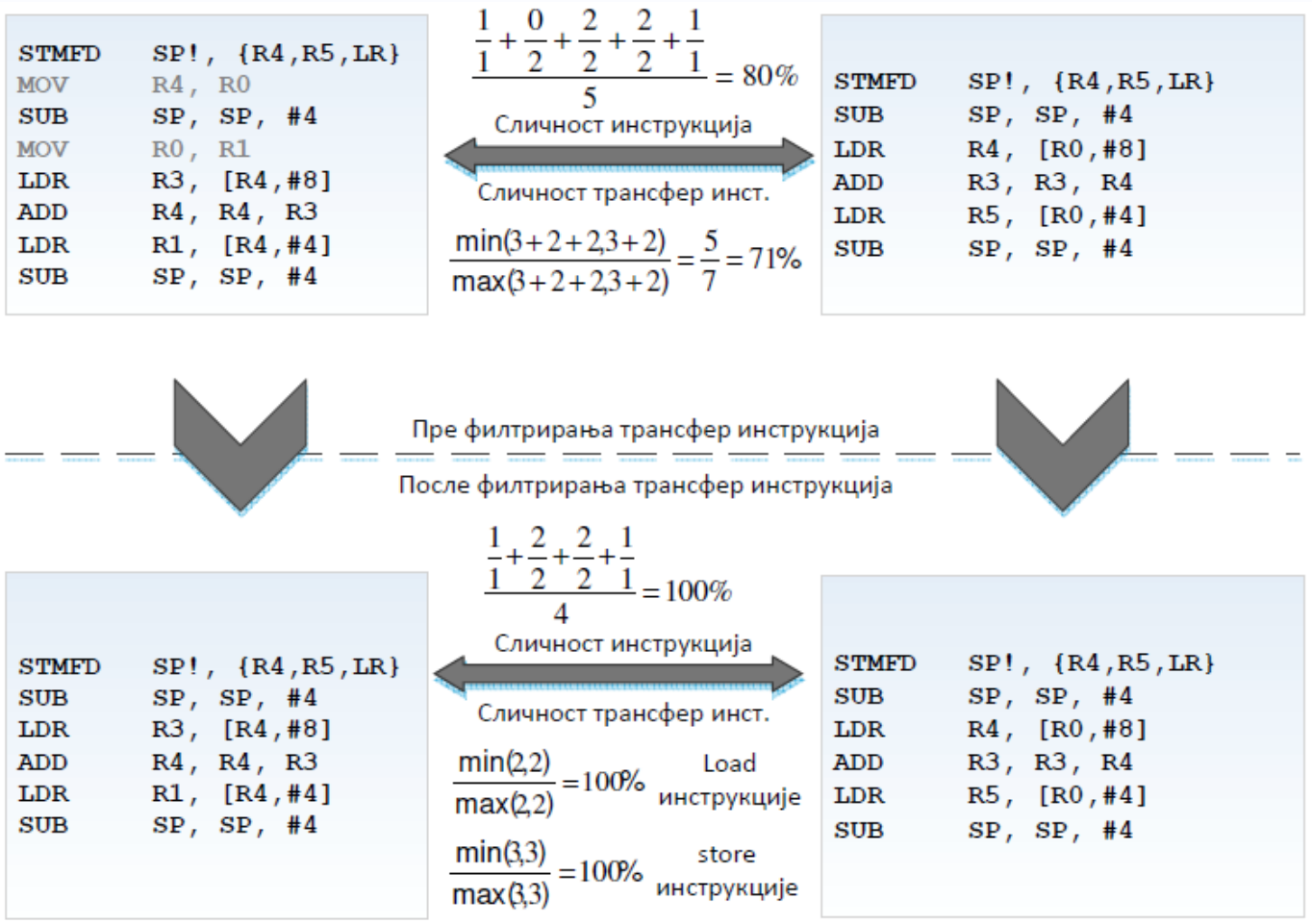
Метрике

Опис	Акроним	Тип вредности	Тип мере	Тип тока
Бројеви појављивања за сваку инструкцију засебно	EIN	V	A	-
Фреквенције појављивања за сваку инструкцију засебно	EIF	V	N	-
Бројеви доскока за сваку одредишну адресу засебно	EBN	V	A	C
Фреквенције доскока за сваку одредишну адресу засебно	EBF	V	N	C
Бројеви позива за сваку позивану процедуру засебно	ECN	V	A	C
Фреквенције позива за сваку позивану процедуру засебно	ECF	V	N	C

Филтрирање инструкција за рад са стеком

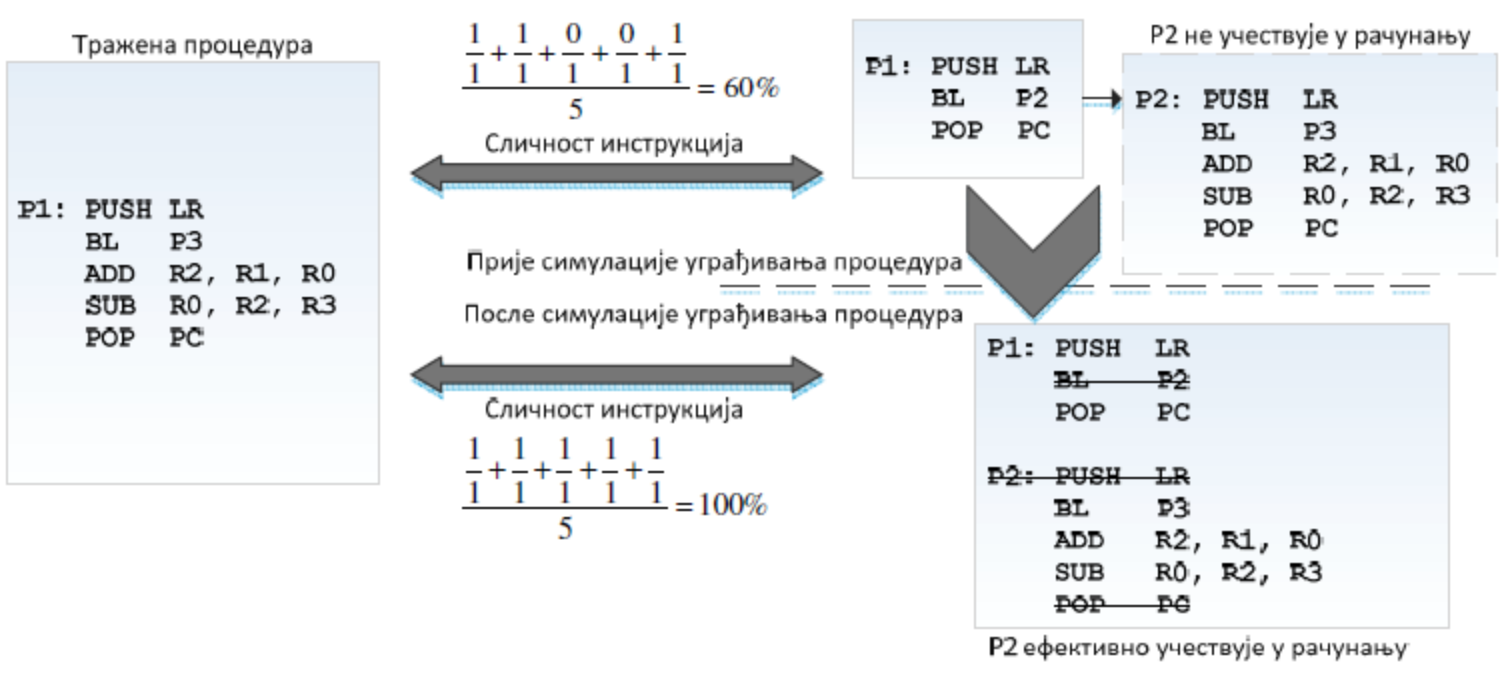


Филтрирање инструкција за пренос података

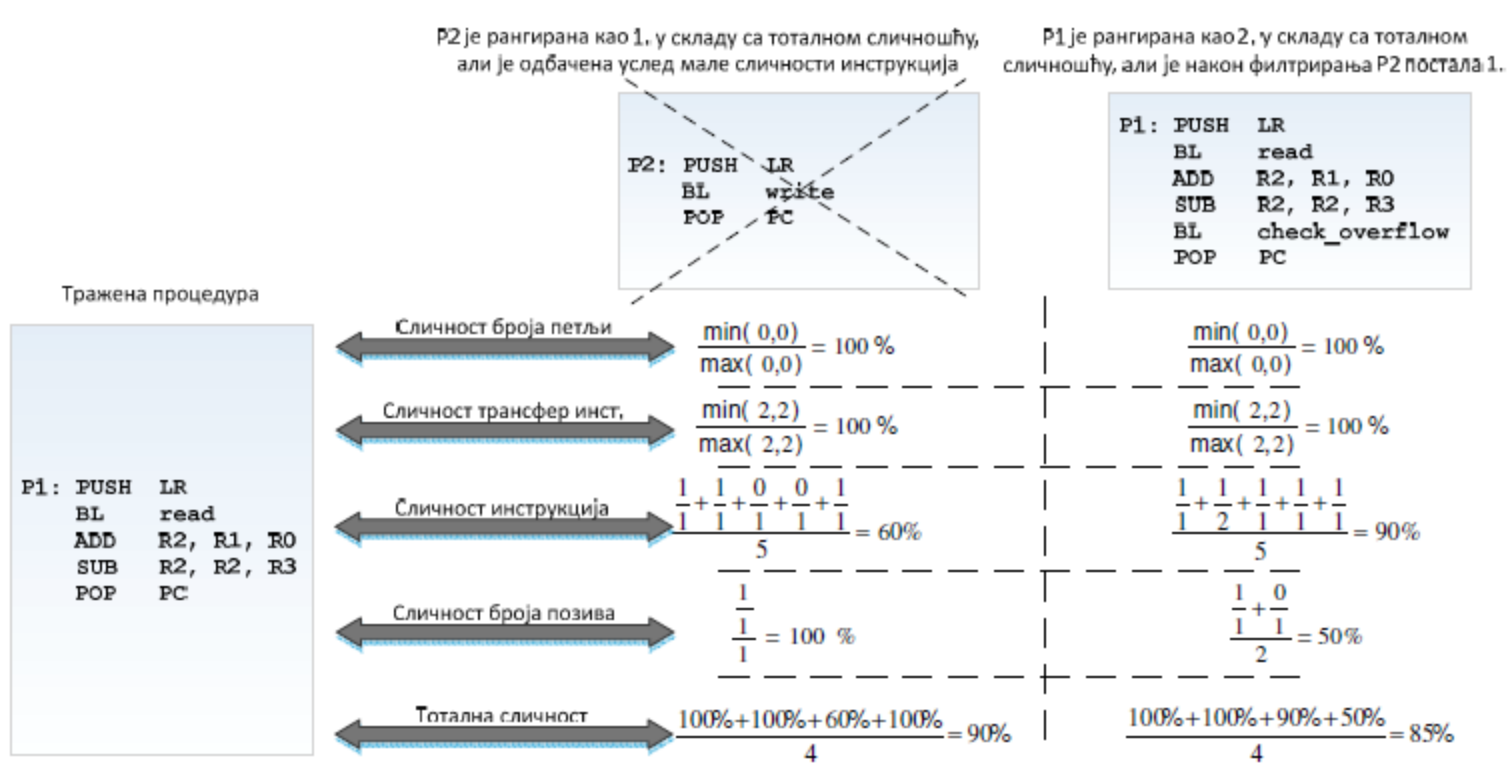


Сличност секвенци операционих кодова

Симулација уграђивања процедура на месту позива



Филтрирање значајно различитих процедура



Окружење за евалуацију

- STAMP Benchmark
  (дељене процедуре су посматране као библиотека)
  - Bayes
  - Genome
  - Intruder
  - Ssca2
  - Vacation
- BusyBox (networking део)
  - MatrixSSL
- Преводиоци (оптимизације O0, O3, Os, Of)
  - Keil
  - IAR
  - CodeSourcery
  - CrossWorks
  - SysProg

Мерење резултата

- Одзив
- Прецизност
- F мера

$$F_\beta = \left(1+\beta^2\right)\frac{preciznost \cdot odziv}{\beta^2 \cdot preciznost + odziv}$$

  - F1 мера – прецизност и одзив имају исти значај
  - F2 мера – одзив има већи значај

Рангирање и n посматраних процедура

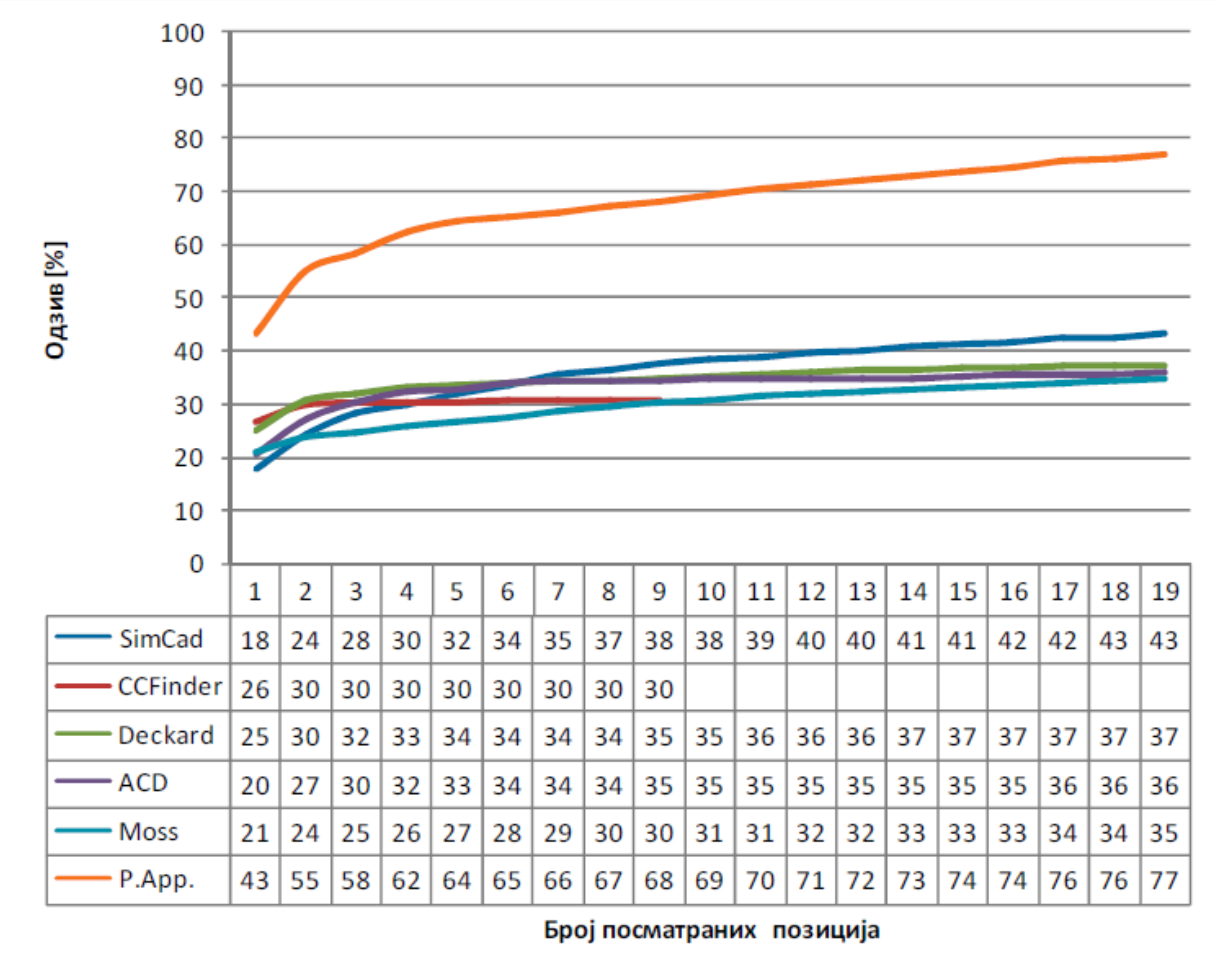
- Посматра се првих n процедура
- Тражена процедура је међу првих n

- Тражена процедура није међу првих n

Постојећи алати коришћени у евалуацији

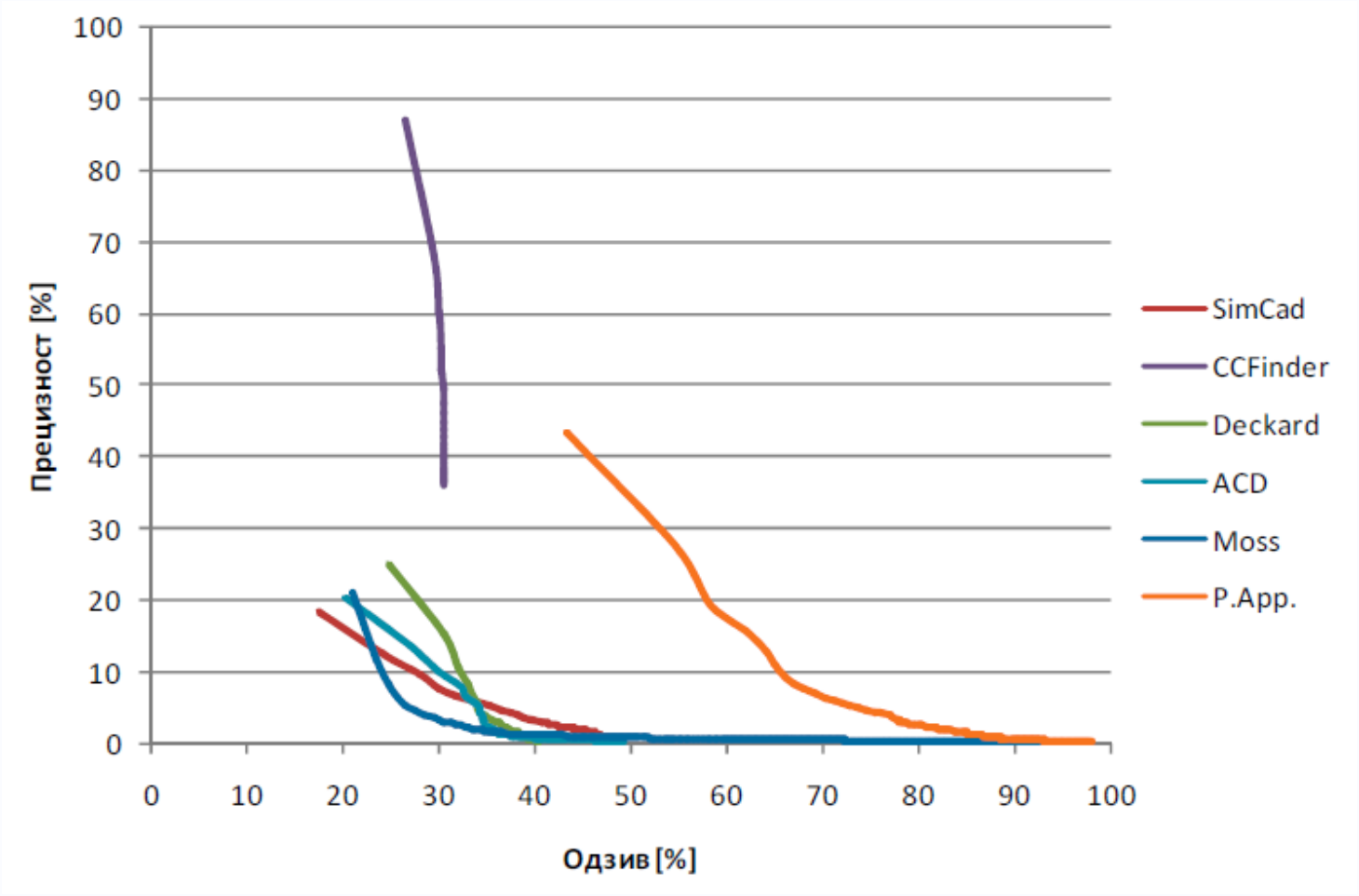
	SimCad	CCFinder	Deckard	ACD	Moss
Подржани језици	C, C#, Java, Py	C/C++, C#, Cobol, Java, VB, Text	C, Java, Php	C/C++	C/C++, C#, Cobol, Java, VB, MIPS, Text…
Језик у експерименту	C	C	C	C	ASM
Ниво поређења	блок, процедура	датотека	датотека	датотека	датотека
Техника откривања клонова	базиран на тексту	базиран на токенима	базиран на AST	базиран на тексту (ASM добијен од C)	базиран на тексту
Типови клонова	1, 2 и 3	1, 2 и 3	1, 2 и 3	1, 2 и 3	1, 2 и 3

Одзив у зависности од броја посматраних позиција(BusyBox)

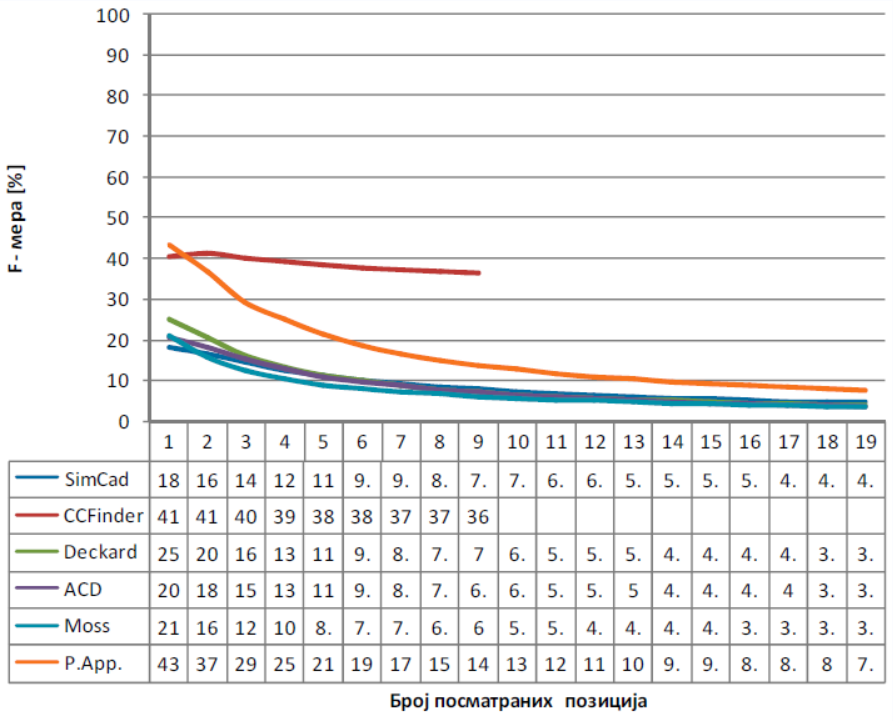


	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
SimCad	18	24	28	30	32	34	35	37	38	38	39	40	40	41	41	42	42	43	43
CCFinder	26	30	30	30	30	30	30	30	30										
Deckard	25	30	32	33	34	34	34	34	35	35	36	36	36	37	37	37	37	37	37
ACD	20	27	30	32	33	34	34	34	35	35	35	35	35	35	35	35	36	36	36
Moss	21	24	25	26	27	28	29	30	30	31	31	32	32	33	33	33	34	34	35
P.App.	43	55	58	62	64	65	66	67	68	69	70	71	72	73	74	74	76	76	77

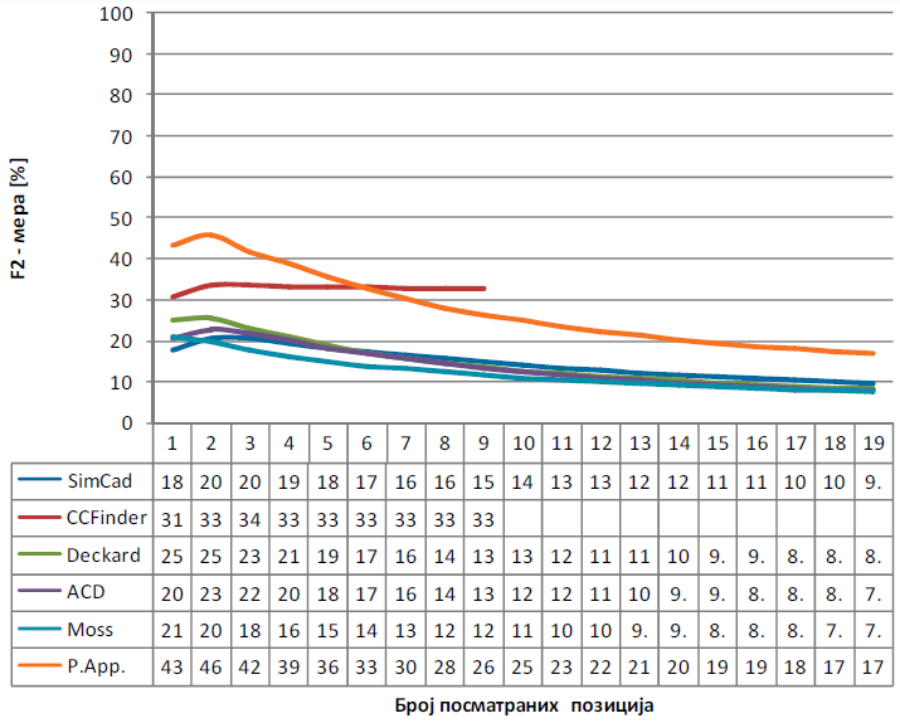
Прецизност у зависности од одзива (BusyBox)



Ф и Ф2 мере у зависности од броја посматраних позиција(BusyBox)



	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
SimCad	18	16	14	12	11	9.	9.	8.	7.	7.	6.	6.	5.	5.	5.	5.	4.	4.	4.
CCFinder	41	41	40	39	38	38	37	37	36										
Deckard	25	20	16	13	11	9.	8.	7.	7	6.	5.	5.	5.	4.	4.	4.	4.	3.	3.
ACD	20	18	15	13	11	9.	8.	7.	6.	6.	5.	5.	5	4.	4.	4.	4	3.	3.
Moss	21	16	12	10	8.	7.	7.	6.	6	5.	5.	4.	4.	4.	4.	3.	3.	3.	3.
P.App.	43	37	29	25	21	19	17	15	14	13	12	11	10	9.	9.	8.	8.	8	7.



	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
SimCad	18	20	20	19	18	17	16	16	15	14	13	13	12	12	11	11	10	10	9.
CCFinder	31	33	34	33	33	33	33	33	33										
Deckard	25	25	23	21	19	17	16	14	13	13	12	11	11	10	9.	9.	8.	8.	8.
ACD	20	23	22	20	18	17	16	14	13	12	12	11	10	9.	9.	8.	8.	8.	7.
Moss	21	20	18	16	15	14	13	12	12	11	10	10	9.	9.	8.	8.	8.	7.	7.
P.App.	43	46	42	39	36	33	30	28	26	25	23	22	21	20	19	19	18	17	17

Закључак

- Предложен нови приступ
  - 19 метрика
  - 6 трансформатора
  - 7 формула
  - Испробане 3 технике машинског учења
  - 5 техника за повећање одзива
  - Поновна имплементација одабраних приступа
  - Поређење са постојећим алатима и приступима
- Постигнут највећи одзив, и то:
  - 53% у случају тестова формираних од STAMP
  - 56% у случају тестова формираних од BusyBox
- Постигнута трећа највећа прецизност
  - CCFinder и приступ који је предложио Dulien постижу већу прецизност, али са знатно мањим одзивом
- Постигнута највећа вредност F мере, и то када се посматра само прва позиција
- Постигнута највећа вредност F_2 мере, и то када се посматрају прве две позиције.

Детекција нарушавања двоструког лиценцирања

Милош Цветановић
Захарије Радивојевић
Саша Стојановић

Хвала на пажњи!