

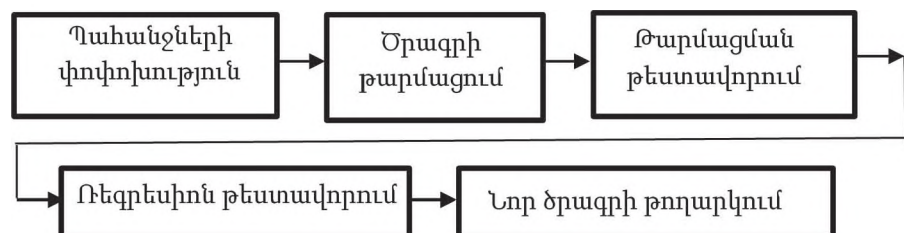
Հ.Հ. ՀԱԿՈԲՅԱՆ, Ա.Գ. ՀԱՐՈՒԹՅՈՒՆՅԱՆ

ԾՐԱԳՐԱՑԻՆ ՀԱՍՏԱԿԱՐԳԵՐԻ ՌԵԳՐԵՍԻՈՆ ԹԵՍՏԱՎՈՐՄԱՆ
ԱՐԴՅՈՒՆԱՎԵՏՈՒԹՅԱՆ ԲԱՐՁՐԱՑՈՒՄԸ ԲԼՈՒՄԻ ՖԻԼՏՐԻ ԿԻՐԱՌՄԱՄԲ

Ներկայացված է ծրագրային միջոցների ֆունկցիոնալության փոփոխության պայմաններում ռեգրեսիոն թեստերի ընտրության մեխանիզմ, որը հիմնված է հեշավորման ալգորիթմի և Բլումի ֆիլտրի կիրառման վրա: Առաջարկվող մեխանիզմը թույլ է տալիս բարձրացնել մեծածավալ տեղեկության մեջ տվյալների որոնման արագագործությունը՝ առանց իջեցնելու սխալների հայտնաբերման հավանականությունը: Դա կատարվում է ամբողջական թեստերի փաթեթից թեստերի այնպիսի ենթախմբի ընտրությամբ, որը պատասխանատու լինի կատարված փոփոխության համար:

Առանցքային բառեր. ծրագրային ապահովման սպասարկում, ռեգրեսիոն թեստավորում, ռեգրեսիոն թեստերի ընտրություն, հեշավորում, Բլումի ֆիլտր:

Ներածություն: Ավտոմատացման համակարգերի զարգացմանը զուգահեռ՝ տեղի են ունենում ծրագրային սպասարկման ֆունկցիոնալության հաճախակի փոփոխություններ, որոնք հանգեցնում են դրանց թեստավորման ծավալների և անհրաժեշտ մեքենայական ռեսուրսների ավելացման: Այս պայմաններում, ծրագրային ապահովման սխալների հայտնաբերման նպատակով, կիրառվում է ռեգրեսիոն թեստավորում [1]: Ծրագրային միջոցի սպասարկման փուլում գործողությունների հաջորդականությունը և դրանում ռեգրեսիոն թեստավորման տեղը ցուցադրված են նկ. 1-ում: Թեստավորման ավարտից հետո կատարվում է ծրագրային միջոցի նոր տարբերակի թողարկումը, որը համակարգի զարգացմանը զուգահեռ՝ բազմակի անգամ կարող է անցնել բերված սպասարկման գործընթացով:



Նկ. 1. Ծրագրային սպասարկման գործընթացը

Ներկայումս մշակված են ռեգրեսիոն թեստերի ընտրության (ՌԹԸ) տարբեր մեխանիզմներ, որոնք կարելի է դասակարգել չորս խմբի [2 - 4]՝

- ծրագրի ամբողջական թեստավորում,
- ծրագրի մոդուլների պատահական ընտրողական թեստավորում,
- թեստերի «ձեռքով» ընտրություն,
- ծրագրի մոդուլների նպատակային ընտրողական թեստավորում:

Ժամանակակից ավտոմատացման համակարգերի դինամիկ զարգացման պայմաններում բերված մեխանիզմներից առավել արդյունավետ են ծրագրի մոդուլների նպատակային ընտրողական թեստավորման վրա հիմնված ՌԹԸ մեխանիզմները [4]: Դրանք հնարավորություն են տալիս թեստերի ամբողջական փաթեթից ընտրել թեստերի այնպիսի ենթախումբ, որ ամբողջական և ընտրված ենթախմբի թեստերի աշխատանքից հետո ստացվի նույն արդյունքը: Այդպիսի մոտեցման արդյունավետությունը կախված է թեստերի ընտրության այնպիսի մեթոդների կիրառումից, որոնք կնպաստեն թեստավորման ժամանակի նվազեցմանը, միաժամանակ ապահովելով անհրաժեշտ ճշտություն:

Ներկայումս, ռեգրեսիոն թեստավորման ծավալների անընդհատ աճի պայմաններում, դրանց համար անհրաժեշտ ռեսուրսները կարող են հասնել ծրագրային սպասարկման գումարային ռեսուրսների գրեթե կեսին [1, 2]: Այս պայմաններում ռեգրեսիոն թեստավորման համար անհրաժեշտ ռեսուրսների նվազեցման խնդիրը դառնում է շատ կարևոր և արդիական:

Այս նպատակով ստորև ներկայացված է ծրագրային համակարգերի ռեգրեսիոն թեստավորման արդյունավետության բարձրացման մեխանիզմ, որը հիմնված է ՌԹԸ հեշավորման և Բլումի ֆիլտրի կիրառման վրա:

Խնդրի դրվածքը և մեթոդի մշակումը: Ժամանակակից ավտոմատացման համակարգերի թեստավորման արդյունավետության բարձրացման խնդիրը հանգում է ռեգրեսիոն թեստավորման այնպիսի մեխանիզմների մշակմանը, որոնք թույլ կտան կառավարել թեստավորման սխալների հայտնաբերման հավանականության և պահանջվող հաշվողական ռեսուրսների միջև փոխկապվածությունը: Այս նպատակով առաջարկված է Բլումի ֆիլտրի ներդրմամբ հեշավորման վրա հիմնված ռեգրեսիոն թեստավորման մեթոդ: Հեշավորման օգտագործումը թույլ է տալիս օգտվել ծրագրի կոդի պահպանման և վերլուծության պարզ կառուցվածքներից, իսկ Բլումի ֆիլտրի ներդրումը բարձրացնում է տեղեկության որոնման արագությունը: ՌԹԸ ալգորիթմներում հաճախ պահանջների և թեստ-դեպքերի միջև կապը հաստատելու նպատակով օգտագործվում է հետազոտության մատրից, որում ներկայացվում է տարբեր թեստ-դեպքերի կարևորության աստիճանը: Սակայն այս մոտեցման կիրառումը մեծածավալ տվյալների մշակման դեպքում հանգեցնում է տվյալների որոնման դանդաղագործության, քանի որ թեստերի ավե-

լացմանը զուգընթաց՝ հետագծելիության մատրիցում ավելանում են նոր սյուներ, որտեղ գրված են 0-ներ, ինչը նշանակում է, որ տվյալ թեստ-դեպքը պատասխանատու չէ համապատասխան տողում գրված պահանջի համար: Այսպիսով, հավելյալ տվյալների պահպանման հետևանքով այս եղանակի կիրառումը մեծածավալ տվյալների դեպքում արդյունավետ չէ (ադ.):

Այս խնդրից խուսափելու նպատակով մշակվել է տվյալների պահպանման նոր կառուցվածք, որը ներկայացված է ծրագրերի կարգաբերման JSON ձևաչափով: Առաջարկվող կառուցվածքում նոր պահանջները և դրանց հեշ արժեքները պահվում են համապատասխան Բլումի ֆիլտրերում, իսկ թեստ-դեպքը ավելացվում է JSON ձևաչափում միայն տվյալ ֆունկցիայի տողում: Դա թույլ է տալիս կրճատել հետագծելիության մատրիցում պահվող հավելյալ 0-ների քանակը, չարունակելով հավելյալ տեղեկություն մնացած պահանջների մասին: Այս կերպ տեղի է ունենում հիշողության արդյունավետ օգտագործում:

Աղյուսակ

Հետագծելիության մատրից

<i>Պահանջներ</i>	<i>Թեստ-դեպքեր</i>			
	<i>Test 1</i>	<i>Test 2</i>	<i>...</i>	<i>Test N</i>
createRow	3	0	...	0
createColumn	0	4	...	0
...	...	...	...	...
createTop	0	0	...	1

Բլումի ֆիլտրի հիման վրա կառուցված ալգորիթմների հիմնական առավելությունը նրանց մեծ արագագործությունն է [5]: Սակայն այս մեթոդն ունի նաև թերություն, ինչը պայմանավորված է նրա հավանականային բնույթով, որի պտճառով, հնարավոր է, հանդիպեն «կեղծ դրական» արդյունքներ: «Կեղծ դրական» արդյունքը այն դեպքն է, երբ ստացվում է պատասխան, թե տվյալ տարրը գոյություն ունի, սակայն իրականում այն գոյություն չունի տվյալների ցուցակում: Նշենք նաև, որ այն բացառում է «կեղծ բացասական» դեպքերը: «Կեղծ բացասական» արդյունքը այն դեպքն է, երբ ստացվում է պատասխան, թե տվյալ տարրը գոյություն չունի, սակայն իրականում այն գոյություն ունի տվյալների ցուցակում:

Բլումի ֆիլտրում հնարավոր է կառավարել «կեղծ դրական» արդյունքների հավանականությունը՝ փոփոխելով բիթերի զանգվածի չափերը և հեշ ֆունկցիաների քանակը, ինչի շնորհիվ էլ այն լայն կիրառություն է գտել մեծածավալ տվյալների որոնման խնդիրներում:

Ստորև բերված են Բլումի ֆիլտրի մի քանի հատկություններ, որոնք կարևոր են ՌԹԸ մեխանիզմներում դրա կիրառության տեսանկյունից.

1. Ի տարբերություն հեշ աղյուսակի ֆիքսված երկարության, Բլումի ֆիլտրը կարող է պահել ցուցակ, որը բաղկացած է կամայական քանակությամբ տարրերից:

2. Բլումի ֆիլտրը երբեք չի գեներացնում «կեղծ բացասական» արդյունքը, այսինքն, եթե տարրը գոյություն չունի ցուցակում ապա ալգորիթմը չի ցույց տա որ այն գոյություն ունի:

3. Տարրերի հեռացումը հնարավոր չէ, քանի որ մեկ տարրի հեռացումը ենթադրում է բիթերի հեռացում, ինչը կարող է ազդել մնացած տարրերի արդյունքների վրա:

Բերված հատկություններից ամենահատկանշականն այն է, որ Բլումի ֆիլտրը երբեք չի վերադարձնում «կեղծ բացասական» արժեք, ինչն էլ հանդիսանում է այս ալգորիթմի հիմնական դրական կողմը: Բլումի ալգորիթմի կարևոր հատկությունն է նաև ճկունությունը, այսինքն՝ «կեղծ դրական» արդյունքների հանդես գալու հավանականության կառավարման հնարավորությունը: «Կեղծ դրական» արդյունքները նվազեցնելու համար կարելի է փոփոխել բիթերի զանգվածի չափերը, ինչի շնորհիվ հնարավոր կլինի խուսափել բազմակի կրկնվող բիթերից: Հնարավոր է նաև ավելացնել հեշ ֆունկցիաների քանակը, որոնց միջոցով էլ հաշվվում են զանգվածի ինդեքսները, ինչը նույնպես կնպաստի «կեղծ դրական» արդյունքների նվազեցմանը:

Տվյալների պահպանման առաջարկվող կառուցվածքը, որը ներկայացված է JSON ձևաչափով, հետևյալն է.

```
{
  <file path1>: {
    "def_filter": <bloom filter for defs>,
    "hash_filter": <bloom filter for hashes>,
    "defs": {
      <function name1>: <test path1>,
      <function name2>: <test path2>
      ....
      <function nameN>: <test pathN>
    }
  },
  ...
  <file path N>: {
    "def_filter": <bloom filter for defs>,
    "hash_filter": <bloom filter for hashes>,
    "defs": {
      <function name1>: <test path1>,
```

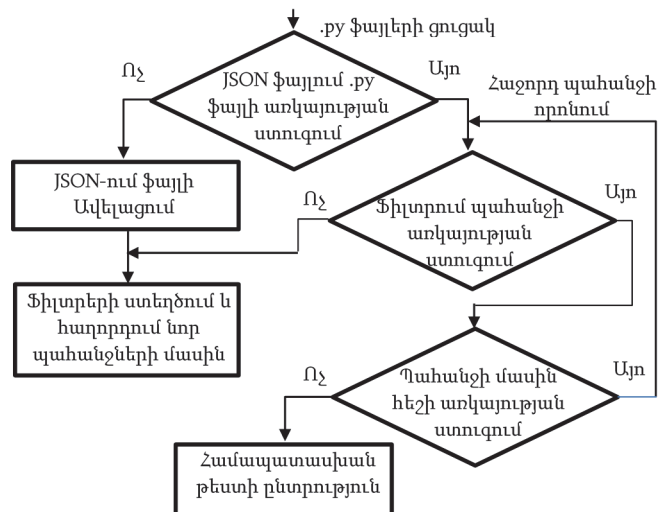
```

    <function name2>: <test path2>
    ....
    <function nameN>: <test pathN>
  }

}

```

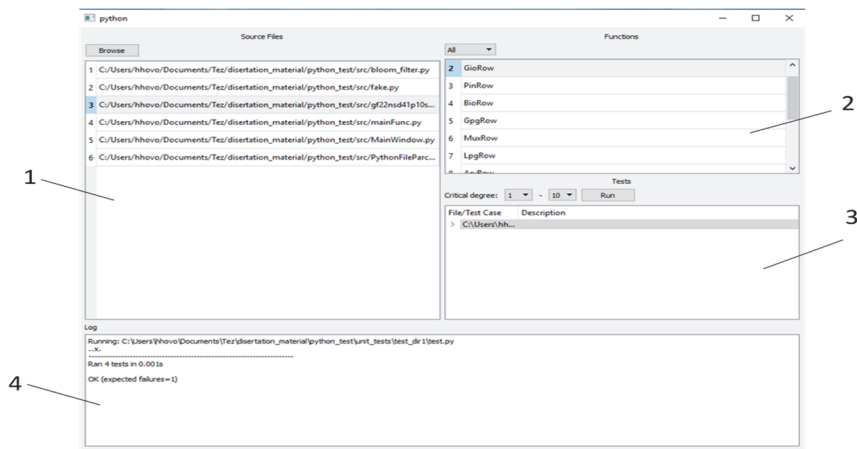
Վերը բերված ձևաչափում <file path1>, ..., <file path N>-ը ծրագրային համակարգի .py ընդլայնումով ֆայլերն են, <bloom filter for defs>, <bloom filter for hashes> պահանջների և հեշ արժեքների պահպանման և որոնման համար ստեղծված Բլումի ֆիլտրերն են, <function name1>, ... , <function name N>-ը՝ տվյալ ֆայլում պահանջների անվանումները՝ համապատասխան թեստերով: Տվյալների պահպանման JSON ֆայլի կիրառմամբ մոդուլային ռեգրեսիոն թեստերի ընտրության առաջարկվող ընթացակարգի բլոկ-սխեման պատկերված է նկ. 2-ում:



Նկ. 2. ՌԹՀ առաջարկվող ընթացակարգի բլոկ-սխեման

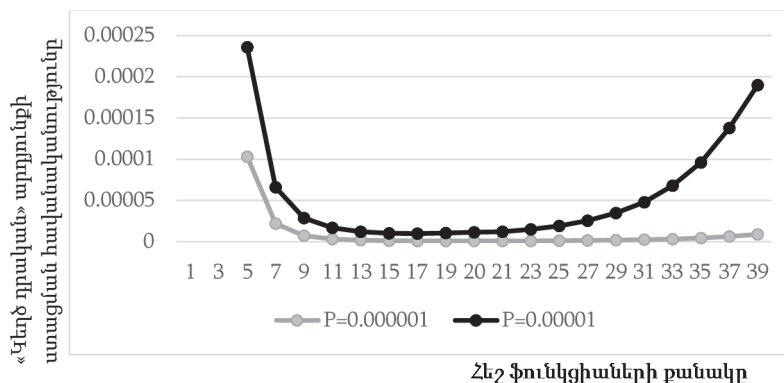
Ստացված արդյունքները: Վերը ներկայացված ընթացակարգի հիման վրա մշակվել և իրականացվել է մոդուլային ռեգրեսիոն թեստերի ընտրության ծրագրային միջոց, որի գլխավոր պատուհանը ներկայացված է նկ. 3-ում:

Ծրագրային գործիքի ինտերֆեյսը բաղկացած է 4 հիմնական պատուհաններից: 1-ին պատուհանում ներկայացվում է ծրագրային համակարգը նկարագրող ֆայլերի ցուցակը: 2-րդ պատուհանում ներկայացվում է պահանջների ցուցակը, որոնք նկարագրված են ընտրված ֆայլում: 3-րդ պատուհանում ցուցադրվում են պահանջին համախառնախանոթ թեստ-դեպքերը, իսկ 4-րդ պատուհանում՝ թեստերի աշխատանքից ստացված արդյունքը:



Նկ. 3. Թեստերի ընտրության ծրագրային միջոցի ինտերֆեյսը

Հետագոտվել է «կեղծ դրական» արդյունքի ստացման հավանականության կախվածությունը հեշ ֆունկցիաների քանակից, որի արդյունքները ներկայացված են նկ. 4-ում բերված գրաֆիկների տեսքով:



Նկ. 4. «Կեղծ դրական» արդյունքի ստացման հավանականության կախվածությունը հեշ ֆունկցիաների քանակից

Երբ «կեղծ դրական» արդյունքի ստացման հավանականությունը $P=10^{-6}$, հեշ ֆունկցիաների անհրաժեշտ քանակը ստացվել է $k=20$, իսկ երբ $P=10^{-5}$ հեշ ֆունկցիաների անհրաժեշտ քանակը նվազում է մինչև $k=17$:

Եզրակացություն: Առաջարկվել է ՌԹՀ մեխանիզմ, որը հիմնված է հեշավորման ալգորիթմի վրա: Մշակված մեխանիզմում ներդրվել է Բլումի ֆիլտրի վրա հիմնված որոնման համակարգ: Այս մեխանիզմի կիրառմամբ, ծրագրային համակարգի նոր տարբերակի թողարկումից առաջ, կարելի է ընտրել ռեգրեսիոն թեստերի այնպիսի ենթախումբ, որի աշխատանքի և ամբողջական ռեգրեսիոն թեստավորման աշխատանքի արդյունքները կլինեն նույնը:

ԳՐԱԿԱՆՈՒԹՅԱՆ ՑԱՆԿ

1. **Leung H. and White L.** Insights into regression test-ing // Proceedings of the Conference on Software Maintenance.- 1989. - P. 60–69.
2. **Kapfhammer G.** The Computer Science Handbook, chapter on Software testing. - 2nd edition.- CRC Press, Boca Raton, FL, 2004.
3. **Rothermel G. and Harrold M.** Selecting tests and identifying test coverage requirements for modified software // Proceedings of the International Symposium on Software Testing and Analysis.- August, 1994. – P. 169–184.
4. **Leung H. and White L.** A cost model to compare re-gression test strategies // Proceedings of the Conference on Software Maintenance.-1991. – P. 201–208.
5. <https://www.geeksforgeeks.org/bloom-filters-introduction-and-python-implementation/>.

Հայաստանի ազգային պոլիտեխնիկական համալսարան: Նյութը ներկայացվել է խմբագրություն 07.02.2020.

Օ.Գ. ԱԿՈՅԱՆ, Ա.Գ. ԱՐՄԵՆՅԱՆ

ПОВЫШЕНИЕ ЭФФЕКТИВНОСТИ РЕГРЕССИОННОГО ТЕСТИРОВАНИЯ ПРОГРАММНЫХ СИСТЕМ С ИСПОЛЬЗОВАНИЕМ ФИЛЬТРА БЛУМА

Представлен механизм выбора регрессионных тестов в условиях изменения функциональности программного обеспечения на основе алгоритма хэширования и фильтра Блума. Предлагаемый механизм позволяет повысить быстродействие поиска данных в большом объеме информации, не снижая вероятность обнаружения ошибок. Это осуществляется путем выбора подгруппы тестов из полного набора тестов, ответственных за внесенные изменения.

Ключевые слова: обслуживание программного обеспечения, регрессионное тестирование, выбор регрессионных тестов, хэширование, фильтр Блума.

Ո.Ո. ՈՒԿՅԱՆ, Ա.Գ. ՀԱՐՄԵՆՅԱՆ

IMPROVING THE EFFICIENCY OF REGRESSION TESTING FOR SOFTWARE SYSTEMS BY USING THE BLOOM FILTER

A regression test selection mechanism based on the hashing algorithm and Bloom filter in case of software changes is introduced. The proposed mechanism allows to improve the searching process in a large amount of data without decreasing the probability of error detection. It is carried out by selecting the test subset among the whole set of tests responsible for the changes made.

Keywords: software system maintenance, regression testing, regression test selection, hashing, Bloom filter.