Модел. и анализ информ. систем. Т. **21**, № **6** (2014) 7–17

УДК 004.052.42+004.4’23

Использование случайной выборки моделей для решения задачи интерполяции Крейга в рамках ограниченной проверки моделей

Ахин М.Х., Колтон С.Л., Ицыксон В.М.

Санкт-Петербургский политехнический университет
195251, Россия, г. Санкт-Петербург, Политехническая ул., 29

e-mail: {akhin,kolton}@kspt.icc.spbstu.ru, vlad@icc.spbstu.ru

получена 30 июня 2014

Ключевые слова: ограниченная проверка моделей, статический анализ программ, интерполяция Крейга, аппроксимация функций, SMT

Одной из самых сложных проблем при статическом анализе программ является анализ вызовов функций, также известный как межпроцедурный анализ. Классическим способом решения этой проблемы является подстановка тел функций в места вызовов, однако при этом значительно возрастает вычислительная сложность анализа из-за увеличения размера модели программы. Для решения этой проблемы можно использовать различные алгоритмы аппроксимации функций, которые заменяют полное тело функции на ее упрощенное описание, тем самым снижая сложность анализа. В контексте ограниченной проверки моделей в последнее время начала активно использоваться интерполяция Крейга, однако ее использование возможно только для пары несовместных логических формул.

В данной статье предлагается подход к аппроксимации функций, основанный на интерполяции Крейга, который лишен данного ограничения за счет усиления интерполяции при помощи случайной выборки моделей. При помощи поиска интересных взаимозависимостей между входными и выходными аргументами функции, случайная выборка моделей позволяет усилить совместные формулы до несовместных, тем самым делая возможным использование интерполяции Крейга. Результаты проведенных предварительных экспериментов подтверждают применимость данного подхода; в дальнейшем планируется провести более подробные исследования его характеристик на реальных примерах.

Введение

В последние годы одним из наиболее перспективных методов обеспечения качества программного обеспечения (ПО) стал метод ограниченной проверки моделей [2]. В его основе лежат два ключевых компонента: полная раскрутка программы для

того, чтобы ее можно было представить в виде формулы в рамках определенной теории (Satisfiability Modulo Theories, SMT), и решатели, которые могут достаточно эффективно определять выполнимость или невыполнимость подобных формул (SMT-решатели). К сожалению, как и большинство других методов анализа ПО, ограниченная проверка моделей плохо справляется со вложенными и рекурсивными вызовами функций, так как они приводят к значительному росту или даже взрыву пространства возможных состояний программы. Для того, чтобы справиться с этой проблемой, обычно используют различные аппроксимации — упрощенные представления отдельных частей анализируемой программы, которые содержат только информацию, относящуюся к проверяемым свойствам безопасности, и отбрасывают все остальное.

Одним из способов аппроксимации является интерполяция Крейга [6]. Для заданной формулы вида $B \to Q$ процедура интерполяции находит интерполянт I, который обобщает B относительно Q. Существенным ограничением данного подхода является то, что он может применяться только если $B \to Q$ всегда выполняется, то есть если $B \wedge \neg Q$ невыполнимо. В контексте ограниченной проверки моделей, однако, часто необходимо решить задачу аппроксимации как раз для случая, когда $B \to Q$ может выполняться не всегда.

В данной статье предлагается подход к интерполяции Крейга, расширяющий ее для случая, когда интересующее нас свойство выполняется не всегда. Он основан на идее случайной выборки точек из входного пространства состояний функции и усилении ими $B \to Q$ так, чтобы данная формула всегда выполнялась. После этого итоговая усиленная формула может быть либо обобщена до контракта функции, либо использоваться при построении аппроксимации функции. Результаты предварительных экспериментов показывают, что подход может использоваться для определенного набора интересных практических случаев аппроксимации.

Статья построена следующим образом. В разделе 1 кратко рассматриваются метод ограниченной проверки моделей и интерполяция Крейга. Основная идея предлагаемого подхода представлена в разделе 2. Экспериментальные исследования и их результаты анализируются в разделе 3. Раздел 4 посвящен обзору предметной области и планам на дальнейшие исследования.

1. Основы

В данном разделе рассматривается метод ограниченной проверки моделей, интерполяция Крейга и ее применение в рамках ограниченной проверки моделей. Кроме того, кратко описываются проблемы, возникающие при применении интерполяции Крейга к задаче аппроксимации функций «в лоб».

1.1. Ограниченная проверка моделей

Проверка моделей является одним из наиболее широко используемых методов проверки корректности ПО, однако при его применении часто возникает проблема взрыва пространства состояний. С учетом современного развития вычислительных ресурсов, полная проверка всего пространства состояний для любой реальной про-

граммы является невозможной. Одним из подходов, пытающихся справиться с этой проблемой, является ограниченная проверка моделей [2].

Идея, положенная в основу ограниченной проверки моделей, чрезвычайно проста: проверяются не все возможные пути выполнения программы, а пути, ограниченные определенной длиной k. Для этого сперва раскручиваются все циклы и выполняется полная подстановка всех функций в местах вызовов. После этого полученная раскрученная программа представляется в виде формулы в логике первого порядка, которая комбинируется с интересующими нас свойствами безопасности и затем проверяется при помощи SAT- или SMT-решателя. В случае если формула невыполнима (UNSAT), программа корректна относительно заданного свойства безопасности, в противном случае (формула выполнима, SAT) — нет.

Как было показано выше, межпроцедурный анализ в рамках ограниченной проверки моделей выполняется при помощи полной подстановки функций, что приводит к значительному увеличению размера анализируемых логических формул, то есть увеличению сложности анализа. Для борьбы с этим возможно использовать аппроксимации вместо тел функций при выполнении подстановки. Под аппроксимацией функции обычно понимается переаппроксимация[1] ее тела, выраженная в терминах ее аргументов и возвращаемого значения; благодаря тому, что это переаппроксимация, ее использование не влияет на полноту ограниченной проверки моделей, то есть подстановка аппроксимации функции в место ее вызова не приведет к пропуску ошибки в ПО. Однако аппроксимации могут повлиять на точность анализа и привести к возникновению ложных обнаружений — ошибок, отсутствующих в оригинальной программе. Основным преимуществом использования аппроксимаций функций является то, что они значительно сокращают размер проверяемой логической формулы, так как содержат только фрагменты, относящиеся к проверяемому свойству безопасности, и являются более компактными.

1.2. Интерполяция Крейга

Одним из способов извлечения аппроксимаций функций в рамках ограниченной проверки моделей является интерполяция Крейга [6], впервые использованная для этой цели в [13]. Вкратце рассмотрим предложенный в [13] подход.

Интерполянтом Крейга для несовместной пары формул (B, Q) в логике первого порядка является формула I такая, что:

- $B \to I$;
- (I, Q) также является несовместной парой формул;
- I содержит только неинтерпретируемые символы, общие для B и Q.

Было доказано, что интерполянт Крейга можно построить для любой несовместной пары формул (B, Q). Различные существующие SMT-решатели поддерживают интерполяцию Крейга и могут генерировать интерполянты из доказательства невыполнимости пары формул [12, 4, 3].

По построению, I является переаппроксимацией B, то есть если принять за B полное тело функции, а за Q — интересующее нас свойство, то I будет являться

[1] Аппроксимация сверху.

искомой аппроксимацией[2] B относительно Q. Например, при проверке программы на разыменования нулевых указателей, интерес представляет, может ли функция вернуть нулевой указатель в качестве возвращаемого значения[3], и формула Q выглядит как $\backslash result \neq 0$.

1.3. Свойства безопасности

Основным ограничением интерполяции Крейга является то, что пара формул (B, Q) должна быть несовместной для того, чтобы существовал интерполянт I. В контексте ограниченной проверки моделей это означает, что аппроксимация функции может быть построена, только если интересующее нас свойство безопасности всегда выполняется для заданной функции. Понятно, что для большинства реальных программ данный факт не будет выполняться. Рассмотрим пример 1, для которого невозможно построить интерполянт и, следовательно, невозможно получить аппроксимацию функции при выполнении анализа разыменований нулевых указателей.

Листинг 1.

```
int* unsafe_index(int* arr, unsigned size, unsigned index) {
    // @assume arr != \nullptr
    if (index >= size) {
        return NULL;
    } else {
        return arr + index;
    }
}
```

Простым подходом является использование вместо аппроксимации полного тела функции, как предлагается в [13], что является вариантом частичной подстановки функций в места вызовов. В данной статье предлагается альтернативный подход, целью которого является генерация аппроксимации функций даже в случае, если пара формул является совместной.

2. Случайная выборка моделей

Пусть нас интересует построение аппроксимации для формулы B относительно свойства Q. Для этого сперва при помощи SMT-решателя проверяется, можно ли найти удовлетворяющее присваивание для $B \wedge \neg Q$; в случае если этого сделать невозможно (UNSAT), то задача может быть решена при помощи классической интерполяции Крейга. В противном случае (SAT) интерполяция Крейга не может использоваться из-за нарушения необходимого условия существования интерполянта.

Предлагаемый подход в общем виде представлен на рисунке 1 и основывается на идее усиления формулы $B \wedge \neg Q$ набором дополнительных условий M относительно аргументов функции так, чтобы $B \wedge \neg Q \wedge M$ стала нетривиально невыполнимой, то есть чтобы к ней можно было применить интерполяцию Крейга. Поиск M использует следующий принцип: если $B \wedge Q$ является выполнимой, то ее удовлетворяющее присваивание (модель) SA делает $B \wedge \neg Q \wedge SA$ невыполнимой.

[2]Для того, чтобы быть корректной аппроксимацией, формула I должна быть выражена относительно аргументов функции и ее возвращаемого значения.

[3]Без потери общности можно пренебречь нулевыми указателями, приходящими через глобальные переменные и аргументы функции.

Рис. 1. Алгоритм случайной выборки моделей

if $B \wedge \neg Q$ невыполнима **then**
  Применяем обычную интерполяцию Крейга к $B \wedge \neg Q$
else
  FA = false
  repeat
    SA_i = SAT-модель для $B \wedge Q$ ▷ с учетом всех предыдущих моделей
    $FA_i = \{A \in SA_i | A$ включает аргумент функции$\}$
    if $B \wedge \neg Q \wedge FA_i$ выполнима **then**
      continue
    end if
    $FA = FA \vee FA_i$
  until FA состоит из достаточного количества моделей
  Применяем обычную интерполяцию Крейга к $B \wedge \neg Q \wedge FA$
end if

Очевидно, что использование полного удовлетворяющего присваивания не имеет никакого смысла, так как оно не позволяет интерполяции эффективно обобщить интересные факты об особенностях поведения функции. В рассматриваемом случае B — это формула, представляющая тело функции, Q — интересующее свойство безопасности, и решается задача построения аппроксимации функции относительно ее окружения.

Окружение функции может влиять на ее поведение только через соответствующие аргументы[4], поэтому возможно сократить SA и оставить только удовлетворяющие присваивания аргументов функции FA. Данное ослабление может сделать $B \wedge \neg Q \wedge FA$ выполнимой[5], поэтому требуется дополнительный вызов SMT-решателя для того, чтобы проверить достаточность FA для наших задач. В случае, если $B \wedge \neg Q \wedge FA$ является невыполнимой, алгоритм может продолжать свою работу, в противном случае данная модель отбрасывается и в дальнейшем не рассматривается.

Для того, чтобы лучше исследовать пространство возможных значений аргументов функции, собираются N удовлетворяющих присваиваний аргументов FA_i. В итоге получается невыполнимая формула $B \wedge \neg Q \wedge \vee_i FA_i$, которая может быть проинтерполирована двумя различными способами.

В случае, если необходимо получить аппроксимацию функции, можно выполнить интерполяцию ее тела B, в результате чего получается аппроксимация S, которая обеспечивает невыполнимость формулы $S \wedge \neg Q \wedge \vee_i FA_i$. Так как функция может быть вызвана с любыми возможными значениями аргументов, то ограничения FA_i необходимо отбросить, после чего проверить, сохраняется ли невыполнимость формулы $S \wedge \neg Q$. Если это так, то S представляет собой корректную аппроксимацию функции, которая может использоваться вместо B. Если же $S \wedge \neg Q$ является совместной, то предложенный подход не смог извлечь подходящую аппроксимацию,

[4]Без потери общности можно пренебречь влиянием глобальных переменных.

[5]Некоторые из удаленных дизъюнктов могут находиться в UNSAT-ядре, и их удаление приведет к удовлетворению оригинальной формулы.

Листинг 2.

```
int* safe_index(int* arr, unsigned size, unsigned index) {
    // @assume arr != \nullptr
    if (index >= size) {
        return arr;
    } else {
        return arr + index;
    }
}
```

и можно либо попробовать снова с другой начальной формулой SA, либо применить альтернативный способ аппроксимации.

При необходимости извлечения контракта функции, можно выполнить интерполяцию от $\vee_i FA_i$ до контракта C. В случае, если интерполирующая процедура смогла успешно обобщить ограничения на аргументы функции, формула C является интересным контрактом оригинальной функции. В противном случае формула C будет тривиально равна $\vee_i FA_i$, и можно, аналогично извлечению аппроксимации, либо попробовать снова, либо воспользоваться другим алгоритмом извлечения контрактов.

3. Экспериментальные исследования

В данном разделе описываются экспериментальные исследования предложенного подхода, проведенные над его реализацией в рамках прототипа Borealis [1]. В данной реализации использовались следующие два SMT-солвера:

- MathSAT 5 для генерации интерполянтов Крейга;
- Z3 для выполнения алгоритма выборки моделей.

3.1. Пример работы подхода

Рассмотрим пример функции, приведенной на листинге 2. В случае поиска разыменований нулевых указателей нас интересует свойство, что функция не может вернуть нулевой указатель. Это свойство может быть представлено следующим образом: $\backslash result \neq 0$. Данное свойство будет выполняться в случае, если `arr` — ненулевой указатель, что задается при помощи специальной аннотации `@assume arr != \nullptr`). В этом случае можно построить аппроксимацию функции, напрямую используя интерполяцию Крейга, так как формула $B \wedge Q$ невыполнима. Полученный интерполянт (аппроксимация) совпадает с интересующим свойством $\backslash result \neq 0$, и его размер в 20 раз меньше, чем оригинальное представление исходной функции.

Теперь рассмотрим пример функции `unsafe_index` (см. листинг 3), которая отличается от функции `safe_index` тем, что может вернуть нулевой указатель. В этом случае мы не можем напрямую применить интерполяцию Крейга для построения аппроксимации функции. Для решения этой проблемы применим предложенный алгоритм выборки моделей.

При помощи алгоритма случайной выборки моделей по 32 точкам была построена следующая аппроксимация данной функции: $\neg((index < size) \wedge (\backslash result = 0))$,

Листинг 3.

```
int* unsafe_index(int* arr, unsigned size, unsigned index) {
    // @assume arr != \nullptr
    if (index >= size) {
        return NULL;
    } else {
        return arr + index;
    }
}
```

что соответствует: $(index < size) \rightarrow (\backslash result \neq 0)$. Видно, что данная формула содержит только ту часть функции, которая позволяет верифицировать интересующее нас свойство, и в 8 раз меньше, чем оригинальное представление исходной функции.

3.2. Результаты экспериментов

Для проведения экспериментального исследования был построен специальный набор тестовых примеров с различными ограничениями, такими как наличие нескольких функций, в том числе пригодных для аппроксимации, работа с указателями, и так далее. Эти ограничения не позволили в полной мере применить существующие тестовые наборы, используемые для проверки средств статического анализа. Выбранный для экспериментов тестовый набор основывается на популярном наборе NECLA [8], из которого было отобрано 6 примеров. Кроме того, было дополнительно разработано еще 6 тестовых примеров.

Результаты проведенных экспериментов приведены в таблице 1. Были рассмотрены три стратегии реализации межпроцедурного анализа:

- подход на основе случайной выборки моделей;
- режим без подстановки функций;
- режим с полной подстановкой функций.

В таблице сравниваются времена работы анализатора, а также количество обнаруженных дефектов[6]. В примерах № 1–6 аппроксимация функции строится напрямую с использованием интерполяции Крейга, в примерах № 7–12 для построения аппроксимаций применяется предложенный алгоритм случайной выборки моделей.

Стратегия полной подстановки функций позволяет добиться максимальной точности анализа за счет потери производительности. Стратегия без подстановки функций, напротив, обеспечивает высокую производительность анализа при более низкой точности. Из таблицы 1 видно, что при использовании аппроксимаций удается сохранить точность анализа по сравнению со стратегией с полной подстановкой функций и одновременно повысить его производительность.

[6]Предполагается, что все дефекты, обнаруженные при использовании режима полной подстановки, являются истинными.

Таблица 1. Результаты тестирования для трех стратегий реализации межпроцедурного анализа

Номер примера	Аппроксимация		Без подстановки		Подстановка	
	Время, мс	Кол-во дефектов	Время, мс	Кол-во дефектов	Время, мс	Кол-во дефектов
1	890	1	790	3	1140	1
2	26610	13	28030	13	—	—
3	4650	6	4550	6	7450	6
4	21710	2	21560	2	21630	2
5	4210	8	4510	9	6200	8
6	250	0	200	2	315	0
7	1980	1	730	1	940	1
8	2590	1	830	1	860	1
9	1700	0	200	2	315	0
10	2070	0	450	8	2410	0
11	2530	1	790	1	7050	1
12	2650	4	810	4	3960	4

При сравнении производительности сперва рассмотрим примеры № 1–6. Видно, что использование интерполяции Крейга напрямую при генерации аппроксимаций позволяет повысить производительность по сравнению со стратегией с полной подстановкой. Причем для тестового примера № 2 стратегия с полной подстановкой функций не позволяет провести анализ за приемлемое время, и Borealis завершает работу аварийно по тайм-ауту. По сравнению со стратегией без подстановки использование аппроксимаций привело к снижению производительности. Однако для примеров № 2 и 5 использование аппроксимаций повысило производительность по сравнению со стратегией без подстановки за счет снижения числа возможных степеней свободы SMT-солвера при проведении анализа.

При рассмотрении примеров № 7–12 наблюдается значительное снижение производительности аппроксимаций по сравнению со стратегией без подстановки за счет появления затрат на проведение случайной выборки моделей. Сравнение производительности стратегий с использованием аппроксимаций и с использованием полной подстановки функций показывает, что для небольших примеров (№ 7–9) использование выборки моделей приводит к снижению производительности. Основной причиной этого является то, что в данных примерах содержится относительно малое количество вызовов функций и накладные расходы на построение аппроксимаций значительно превосходят выигрыш от их использования. Примеры № 10–12, напротив, содержат достаточное количество вложенных вызовов аппроксимируемых функций, в результате наблюдается повышение производительности от использования аппроксимаций.

В итоге можно сказать, что использование аппроксимаций позволяет добиться компромисса между производительностью и точностью для стратегий реализации межпроцедурного анализа с полной подстановкой функций и без подстановки.

4. Обзор предметной области

Идея использования интерполяции Крейга для извлечения аппроксимаций функций активно исследуется последние годы [13, 10, 11]. Принципиальным отличием предложенного в данной работе подхода является то, что область возможного применения интерполяции Крейга расширяется на случай совместной пары формул (B, Q) за счет использования алгоритма случайной выборки моделей. Классический подход к использованию интерполяции Крейга в подобном случае использует подстановку тел функций в места вызовов, либо пока интерполяция не может быть применена, либо пока не будет найден подходящий контрпример.

Предложенный подход имеет некоторые общие черты с подходом к синтезу составных доказательств при помощи абдукционного вывода [9, 7]. Описанный алгоритм случайной выборки моделей вместе с интерполяцией Крейга может рассматриваться как метод решения задачи абдукции при извлечении пред- и постусловий для функций программы. Данный подход может быть обобщен на аппроксимацию произвольных фрагментов программы (а не только функций), и в будущем интересно было бы сравнить эти два подхода друг с другом. Данный подход также можно рассматривать как альтернативный способ решения задачи уточнения абстракции на основе контрпримеров (CEGAR) [5] на уровне функций программы без необходимости явного извлечения контрпримера. Получаемые аппроксимации являются уточнением тела функции и соответствующих ему предикатов в контексте метода ограниченной проверки моделей, однако данный процесс выполняется явно, а не лениво, как в оригинальном CEGAR.

В дальнейшем планируется улучшить реализацию алгоритма случайной выборки моделей и провести более детальные экспериментальные исследования на наборе тестов NECLA [8]. Также возможно применить различные техники улучшения случайной выборки из смежной области случайного тестирования для повышения эффективности предложенного подхода.

Кроме того, интерес представляет исследование двух открытых вопросов, которые возникли в ходе разработки описанного подхода. Во-первых, предложенный подход требует большого количества дополнительных вызовов SMT-решателя и преобразований между программной моделью и SMT-формулой. В случае если размер аппроксимируемой функции мал или она редко используется в программе, то эти дополнительные затраты могут существенно снизить производительность анализа по сравнению со случаем, если данная функция просто подставляется в места ее вызова. Определение граничных значений для данного перехода (между использованием подстановки и применением аппроксимации) является одним из перспективных направлений дальнейших исследований по повышению производительности предложенного подхода.

Второй вопрос, представляющий особенный интерес в контексте обнаружения дефектов в ПО, — это то, как комбинировать интересующие нас свойства безопасности в случае необходимости одновременной проверки нескольких свойств. Возможны два варианта решения данной проблемы: раздельные аппроксимации для каждого свойства или одна общая аппроксимация для их объединения. В зависимости от выбранного решения будут отличаться либо размер аппроксимации, либо число вызовов SMT-решателя. Выбор более подходящего решения в зависимости

от вида аппроксимируемой функции является весьма интересной задачей, которую также было бы интересно изучить в дальнейшем.

Заключение

В данной статье предложен подход к аппроксимации функций при помощи интерполяции Крейга, который может использоваться, когда интересующее свойство безопасности выполняется не всегда. Он основывается на использовании случайной выборки моделей в применении к пространству входных аргументов анализируемой функции, после чего полученные модели могут быть обобщены при помощи интерполяции. Несмотря на то, что данный подход является чисто эвристическим, он может быть успешно применен на практике; предварительные эксперименты подтвердили возможность решения некоторых практически полезных задач аппроксимации при помощи предложенного подхода.

В дальнейшем планируется улучшить реализованный прототип и провести более детальные исследования показателей данного подхода. Кроме того, для улучшения показателей алгоритма случайной выборки моделей возможно использовать некоторые результаты из области случайного тестирования.

Список литературы

1. Marat Akhin, Mikhail Belyaev, and Vladimir Itsykson. Yet another defect detection: Combining bounded model checking and code contracts // PSSV'13. 2013. P. 1–11.

2. Armin Biere, Alessandro Cimatti, Edmund M. Clarke, and Yunshan Zhu. Symbolic model checking without BDDs // TACAS'99. 1999. P. 193–207.

3. Jürgen Christ, Jochen Hoenicke, and Alexander Nutz. SMTInterpol: An interpolating SMT solver // SPIN'12. 2012. P. 248–254.

4. Alessandro Cimatti, Alberto Griggio, Bastiaan Joost Schaafsma, and Roberto Sebastiani. The MathSAT5 SMT solver // TACAS'13. 2013. P. 93–107.

5. Edmund Clarke, Orna Grumberg, Somesh Jha, Yuan Lu, and Helmut Veith. Counterexample-guided abstraction refinement for symbolic model checking // *The Journal of ACM*. Sep. 2003. 50(5). P. 752–794.

6. William Craig. Three uses of the Herbrand-Gentzen theorem in relating model theory and proof theory // *The Journal of Symbolic Logic*. Sep 1957. 22(3). P. 269–285.

7. Isil Dillig, Thomas Dillig, Boyang Li, and Ken McMillan. Inductive invariant generation via abductive inference // OOPSLA'13. P. 443–456. New York, USA, 2013. ACM.

8. Franjo Ivančić and Sriram Sankaranarayanan. NECLA static analysis benchmarks. `http://www.nec-labs.com/research/system/systems_SAV-website/benchmarks.php`

9. Boyang Li, Isil Dillig, Thomas Dillig, Ken McMillan, and Mooly Sagiv. Synthesis of circular compositional program proofs via abduction // TACAS'13. 2013. P. 370–384.

10. K. L. McMillan. Applications of Craig interpolants in model checking // TACAS'05. 2005. P. 1–12.

11. K. L. McMillan. Lazy abstraction with interpolants // CAV'06. 2006. P. 123–136.

12. K. L. McMillan. Interpolants from Z3 proofs // FMCAD '11. 2011. P. 19–27.

13. Ondrej Sery, Grigory Fedyukovich, and Natasha Sharygina. Interpolation-based function summaries in bounded model checking // HVC'11. 2012. P. 160–175.

Random Model Sampling: Making Craig Interpolation Work When It Should Not

Marat Akhin, Sam Kolton, Vladimir Itsykson

Saint-Petersburg Polytechnic University
Polytechnicheskaya street, 29, Saint-Petersburg 195251 Russia

Keywords: bounded model checking, static program analysis, Craig interpolation, function summaries, satisfiability modulo theories

One of the most serious problems when doing program analyses is dealing with function calls. While function inlining is the traditional approach to this problem, it nonetheless suffers from the increase in analysis complexity due to the state space explosion. Craig interpolation has been successfully used in recent years in the context of bounded model checking to do function summarization which allows one to replace the complete function body with its succinct summary and, therefore, reduce the complexity. Unfortunately this technique can be applied only to a pair of unsatisfiable formulae.
In this work-in-progress paper we present an approach to function summarization based on Craig interpolation that overcomes its limitation by using random model sampling. It captures interesting input/output relations, strengthening satisfiable formulae into unsatisfiable ones and thus allowing the use of Craig interpolation. Preliminary experiments show the applicability of this approach; in our future work we plan to do a full evaluation on real-world examples.

Сведения об авторах:
Ахин Марат Халимович, СПбПУ, исследователь;
Колтон Семен Леонидович, СПбПУ, студент;
Ицыксон Владимир Михайлович, СПбПУ, доцент.