

АВТОМАТИЧЕСКОЕ ПОСТРОЕНИЕ МОДЕЛИ ДЛЯ ВЕРИФИКАТОРА SPIN СРЕДСТВ СИНХРОНИЗАЦИИ БИБЛИОТЕКИ PTHREAD

Уваричева Е.А.

**Научный руководитель: старший преподаватель Удалова Ю. В.
Сибирский федеральный университет**

Задача параллельного программирования становится все более актуальной с распространением многоядерных систем. Параллельное программирование требует специализированные инструменты для создания, тестирования, верификации и отладки. Верификация, тестирование и отладка параллельных программ в настоящее время являются перспективными и развивающимися направлениями информатики.

Сложность отладки и верификации параллельных программ заключается в следующем. Параллельная программа содержит участки кода, которые могут выполняться одновременно, потоки или процессы. Они взаимодействуют между собой заранее неизвестным образом, команды из различных параллельных участков выполняются асинхронно, поэтому существует значительное число вариантов выполнения параллельной программы. Это может привести к различным вычислительным результатам при нескольких запусках программы над одними и теми же начальными данными. Дополнительную сложность составляют ошибки синхронизации параллельных участков, тупики, критические секции.

Верификация параллельной программы может быть выполнена автоматически с помощью специальной программы-верификатора, например с помощью Spin, Bogor, SMV, TLC, VeriSoft. Для верификации многопоточных программ, составленных с помощью библиотеки Pthread, был выбран верификатор Spin. Библиотека Pthread предоставляет инструменты для работы с потоками и механизмами синхронизации и защиты критических секций, таких как мьютексы, семафоры, блокировки чтения-записи, условные переменные и другие. Библиотека Pthread входит в любой дистрибутив операционной системы Linux.

Верификатор на основе проверки моделей Spin является свободно распространяемым и мультиплатформенным программным продуктом. Он может применяться для верификации многопоточных или многопроцессных параллельных программ, распределенных программных систем и протоколов коммуникации. Верификатор Spin получает текст программы на встроенном языке Promela, строит модель программы и производит множественный обход модели. Таким образом исследуются варианты выполнения параллельной программы, на основе этого выдается информация о корректности программы относительно требований программиста, о завершимости программы и недостижимых состояниях в программе.

Программы-верификаторы, как правило, работают со своим встроенным языком описания программ. Верификатор Spin не является исключением. Поэтому для верификации настоящей параллельной программы требуется самостоятельно составить аналог такой программы на языке верификатора, при этом язык верификатора является ограниченным средством, как правило, он оперирует малым набором операторов и понятий. Создание автоматической системы построения модели многопоточной параллельной Pthread программы позволит автоматически перевести ее на язык Promela и, впоследствии, автоматически верифицировать ее с помощью Spin. Такой подход не потребует от пользователя специальных знаний верификатора Spin и языка Promela, если верифика-

ция параллельной программы ограничится проверкой синхронизации потоков, наличия тупиков и завершенности программы.

Для создания такой автоматической системы были составлены модели для верификатора Spin средств синхронизации библиотеки Pthread, построены аналоги почти всех функций библиотеки Pthread на языке Promela. Дополнительно разработаны аналоги части операторов языка C в языке Promela, такие как команды ветвления и циклы. Необходимо заметить, что не все операторы языка C можно смоделировать в Promela: так, работа с указателями, файлами, некоторыми типами данных и многими библиотечными функциями поддерживаться не будет.

Модели функций механизмов синхронизации в библиотеке Pthread для языка Promela составлены в двух вариантах: с активным и пассивным ожиданием доступа к разделяемому ресурсу. Это позволит уменьшить различия между параллельной программой и верифицируемой моделью параллельной программы.

Рассмотрим модель блокировки чтения-записи Pthread в языке Promela. Блокировка чтения-записи – это средство синхронизации, предназначенное для защиты критических секций, таким образом, что читать данные могут несколько потоков, а изменять только один поток, при этом читатели и писатель не могут работать одновременно. Такая блокировка может находиться в трех состояниях: закрытой для чтения, закрытой для записи и открытой.

Описание блокировки pthread_rwlock_t rwlock в языке Promela будет аналогично объявлению целочисленной переменной int rwlock. Функция инициализации блокировки чтения-записи аналогична обнулению rwlock = 0, это означает, что блокировка создается в открытом состоянии. Значение переменной rwlock задается следующим образом: если rwlock = 0, то блокировка открыта, rwlock = -1, то закрыта для записи, rwlock > 0, то закрыта для чтения известным числом потоков. Функция закрытия блокировки для чтения pthread_rwlock_rdlock (pthread_rwlock_t *rwlock) имеет такой аналог в Promela:

```
с пассивным ожиданием:
    (rwlock >=0); rwlock++;
с активным ожиданием:
    if
    :: (rwlock >=0) -> rwlock++;
    :: (rwlock <0) ->
        do
        :: (rwlock >=0) -> rwlock=++; break ;
        :: (rwlock <0) ->;
        od
    fi.
```

Благодаря правилу выполнимости, запись многих операций ожидания в Promela упрощается. Цикл активного ожидания в Promela можно заменить единственным условием, пример (rwlock >=0). Такой оператор выполняется только тогда, когда условие становится истинным. Если условие ложно, то процесс входит в режим ожидания.

Функция открытия блокировки pthread_rwlock_unlock (pthread_rwlock_t *rwlock) выглядит так:

```
    if
    :: (rwlock ==0) ->;
    :: (rwlock ==-1) -> rwlock=0;
    :: (rwlock >0) -> rwlock--;
    fi.
```

Блокировку, закрытую для чтения, открывают, вызывая функцию `unlock` столько раз, сколько эта блокировка была закрыта. Для открытия блокировки, закрытой для записи, достаточно одного вызова функции `unlock`. Открытие уже открытой блокировки не приведет к остановке потока.

Программа для автоматического построения модели параллельной Pthread программы и ее дальнейшей автоматической верификации в Spin имеет текстовый интерфейс. Входные данные программы – это имя файла с параллельной программой и имя файла для записи результата работы программы. Файл с текстом программы отправляется на проверку компилятору `g++`. Если компилятор обнаружил ошибки, они выводятся на экран, и программа заканчивает свою работу, так как не имеет смысла верифицировать синтаксически неправильно написанную программу. Если ошибок нет, производится трансляция программы на язык Promela. На текущий момент при трансляции учитываются только функции работы с потоками и механизмами синхронизации потоков, другие операторы программы не рассматриваются. Такая ситуация является удовлетворительной только для тех параллельных программ, в которых использование механизмов синхронизации не зависит от вычислений программы. Доработка транслятора ведется в настоящее время. Полученная модель программы на языке Promela передается верификатору Spin, который автоматически генерирует информацию о завершимости программы и недостижимых операторах в каждом потоке.

В результате работы создан прототип транслятора параллельных Pthread программ во входной формат верификатора Spin, построены аналоги почти всех функций библиотеки Pthread и часто используемых операторов языка C. Это позволит реализовать систему автоматического построения модели многопоточных параллельных Pthread программ для дальнейшей автоматической верификации синхронизации параллельных участков и завершимости программы.