

Задание 2.

Размеченные системы переходов (LTS) (10 баллов).

Формулировка задания

Дано:

Дан текст программы на языке C с функциями f и g . (см. задание 1)

Предполагается, что функции f и g выполняются параллельно, в двух разных потоках управления (процессах) P_f и P_g соответственно. Будем считать состоянием модели программы совокупность значений счётчиков команд c_f и c_g процессов P_f и P_g , значения глобальных и локальных переменных заданной программы.

Требуется:

1. (1 балл) Построить размеченные системы переходов (LTS) M_1 , M_2 функций f и g из первого задания для следующих значений параметров функций f и g : $f.a = 1$, $f.b = 2$, $g.a = 3$, $g.b = 4$. При этом предполагается, что функции выполняются независимо и никакая другая функция не может влиять на значение глобальной переменной. Состояния LTS должны быть размечены значениями счётчика управления, глобальной переменной и локальных переменных x и y , дуги – выполняемыми операторами. Полученные LTS записать в формате dot и при помощи программного средства dot/GraphViz сгенерировать по ним картинки в формате png.
2. (3 балла) Модифицировать программу из п.3 задания 1 так, чтобы при указании параметра «-lts имя_файла» она строила и сохраняла в указанный файл систему переходов для асинхронной параллельной композиции функций f и g в формате dot (для значений входных параметров $(f.a=1, f.b=2, g.a=3, g.b=4)$). Состояния LTS должны быть размечены значениями счётчика управления, глобальной переменной и локальных переменных x и y . Дуги должны быть размечены выполняемыми операторами, для операторов ветвления дуга - истинным условием перехода (т.е. для оператора `while(x<3)` дуга, ведущая в тело цикла, размечается « $(x<3)$ », ведущая на следующий за циклом оператор – « $!(x<3)$ »). Цвет дуг должен зависеть от того, какой процесс выполнил действие.
3. (2 балла) Считая состоянием программы значение переменной h , построить LTS такой модели параллельной программы. Состояния LTS должны быть размечены значением глобальной переменной h (размечать состояния счетчиком операторов не нужно), дуги – выполняемыми операторами, приводящими к изменению переменной h (если действие не приводит к изменению h , его отображать в виде дуги не нужно). Полученную LTS записать в формате dot и при помощи программного средства dot/GraphViz сгенерировать по ней картинку в формате png.
4. (3 балла) Построить модель параллельной программы из первой задачи на языке временных автоматов и, запустив верификацию или симуляцию в среде UPPAAL, получить точное значение числа достижимых состояний этой модели для значений входных параметров $(f.a=1, f.b=2, f.c=3, f.d=4)$
5. (1 балл) Обоснуйте, почему количество состояний, подсчитанное при помощи системы UPPAAL, отличается от полученного при построении LTS программой из п.2

На выполнение задания отводится четыре недели. Последний срок сдачи задания – 18 октября.

Требования к файлам решения:

Решение задачи должно включать в себя 12 файлов:

1. Текстовый файл task.txt с описанием функций f и g .
2. Описание LTS M_1 и M_2 в формате dot в файлах с именами lts_m1.txt и lts_m2.txt, а также сгенерированные по ним картинки в файлах lts_m1.png и lts_m2.png.
3. Исходный код модифицированной согласно п.2 программы в файле group_surname_2.c, описание сгенерированной ей LTS в формате dot в файле lts_m12.dot и сгенерированную по нему картинку в файле lts_m12.png.
4. Описание LTS с состоянием, включающим только значение h , в формате dot в файле abstract_lts_m12.dot и сгенерированную по нему картинку в файле abstract_lts_m12.png
5. Текстовый файл с расширением group_surname_2.xml с моделью на UPPAAL, текстовый файл log.txt с тем, что выдал верификатор или симулятор при запуске модели, текстовый файл states.txt с 1) фразой «Согласно верификатору, у модели – N достижимых состояний», где N -- число состояний, выданных верификатором, 2) фразой «В LTS для программы – M достижимых состояний», где M – число, выдаваемое программой из п.2 данной задачи и 3) текстом, описывающим разницу между M и N объясняющим (с точностью до состояния) эту разницу.

Дополнительная информация:

1. Пример описания LTS в виде графа в формате dot:

```
digraph G {
0 [label="1,#,1"];
1 [label="2,1,1"];
2 [label="4,1,1"];
0 -> 1 [label="h = a;" color="red"];
1 -> 2 [label="!(h < a)" color="blue"];
}
```

Графический файл в формате png может быть сгенерирован следующей командой (имя_файла_1 – имя файла с картинкой, имя_файла_2 – имя файла в формате dot):

```
>dot -Tpng -o имя_файла_1 имя_файла_2
```

2. О том, как среди вывода верификатора распознать число состояний модели, можно прочесть в руководстве к системе UPPAAL (например, здесь: <http://www.mbsd.cs.ru.nl/publications/papers/fvaan/uppaaltutorial.pdf>). Обратите внимание, что в ходе построения пространства состояний UPPAAL выполняет редукцию частичных порядков, удаление незначущих переменных, слияние состояний, накладывает ограничения на порядок завершения процессов.
3. Если разница равна 0, это тоже нужно обосновать.

