МИНОБРНАУКИ РОССИИ
САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ
ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)
Кафедра МО ЭВМ

ОТЧЕТ

по индивидуальному домашнему заданию
по дисциплине «Верификация распределенных алгоритмов»
Тема: Разработка контроллера светофоров и его верификация
Вариант: 4

Студент гр. 9303 _____ Игнашов В.М.

Преподаватель _____ Шошмина И.В.

Санкт-Петербург
2024

СОДЕРЖАНИЕ

ВВЕДЕНИЕ

Целью работы является разработка модели контроллера перекрестка, регулируемого светофором, на языке Promela. Модель должна обрабатывать потоки машин по нескольким пересекающимся направлениям (в случае если направления движений не пересекаются – допустимо одновременное выполнение потоков). Предполагается, что потоки машин недетерминированные, процессы модели обрабатывают их параллельно для исключения последовательной обработки потоков. Необходимо реализовать модель таким образом, чтобы она соответствовала проверяемым требованиям: безопасность движения, живость движения и справедливость движения.

1. ПОСТРОЕНИЕ МОДЕЛИ

1.1. Описание задачи

Вариант: 4.

Четыре пересечения: NS/ED, SD/WN, SD/DN, WN/DE

Схема сложного перекрестка с указанными выше пересечениями представлена на рисунке 1. Синими точками отмечены конфликтующие направления из условия, красными – конфликтующие направления вне условия задачи.

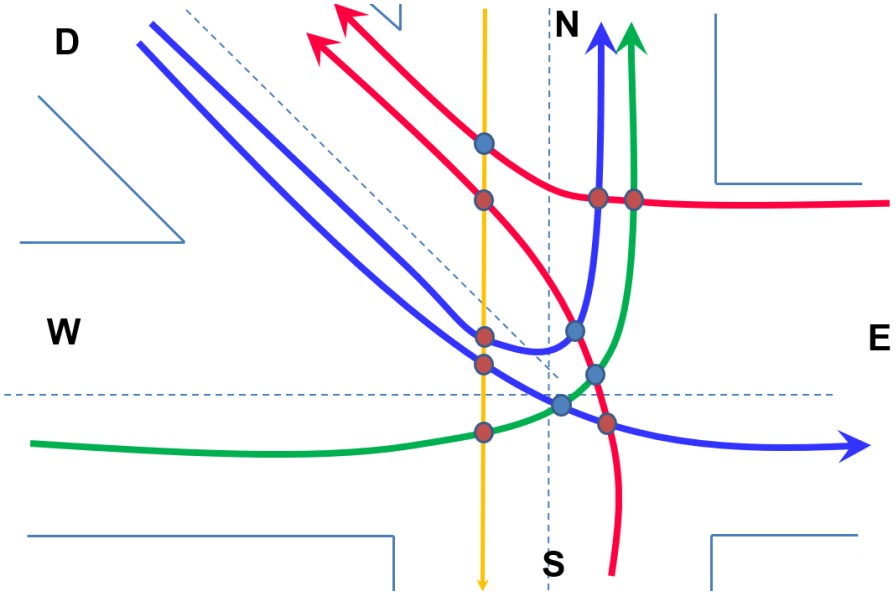

Рисунок 1. Схема перекрестка

Информация о том, какие направления должны быть закрыты при потоке машин для каждого направления, представлена в таблице 1.

Таблица 1. Конфликтующие направления

	NS	WN	SD	ED	DE	DN
NS		+	+	+	+	+
WN	+		+	+	+	
SD	+	+			+	+
ED	+	+				+
DE	+	+	+			
DN	+		+	+		

Дополнительные условия, которые необходимо учесть:

- Машины на каждом направлении движутся независимо
- Появление машины в каждом направлении регистрируется своим независимым датчиком движения

Исходный код реализованной модели представлен в приложении А.

1.2. Описание структур

Состояние каждого контроллера направления описывается реализованной структурой Direction, имеющей три булевых поля

```
typedef Direction {
	bool cars = 0;
	bool open = 0;
	bool completed = 1;
};
```

- cars – маркер, обозначающий наличие машин на этом направлении. Параметр использует логический тип данных в связи с отсутствием необходимости контролировать количество машин в потоке – независимо от их количества они все выполнят движение в случае открытого направления;
- open – маркер, обозначающий сигнал светофора (0 при закрытой дороге, 1 при открытой);
- completed – маркер, обозначающий завершение действия. Обновляется на 0 в случае, если процесс MarkAsNotCompleted определил, что процессы завершили свою работу на этой «итерации». Таким образом обеспечивается выполнение критерия справедливости и критерия живости.

Для синхронизации работы процессов используются семафоры. Его структура содержит только счетчик C. Реализованы две inline функции, позволяющие захватить и отпустить контроль.

С помощью таких семафоров выполняется блокировка конфликтующих направлений, для этого созданы переменные NS_WN, NS_SD, NS_ED, NS_DE, NS_DN, WN_SD, WN_ED, WN_DE, SD_DE, SD_DN и ED_DN для каждой пары.

1.3. Описание процессов

В решении используются различные процессы, в совокупности выполняющие функционал светофора, обеспечивающие создание движения и управляющие потоками.

1.3.1. Процесс AddCarsToDirection

Данный процесс обеспечивает создание движения путем установления параметра cars в состояниях в значение 1 при отсутствии машин в данном направлении.

```
proctype AddCarsToDirection() {
      do
      :: !NS.cars -> {NS.cars = 1;}
      :: !WN.cars -> {WN.cars = 1;}
      :: !SD.cars -> {SD.cars = 1;}
      :: !ED.cars -> {ED.cars = 1;}
      :: !DE.cars -> {DE.cars = 1;}
      :: !DN.cars -> {DN.cars = 1;}
      od
}
```

1.3.2. Процесс MarkAsNotCompleted

Процесс обновляет значения completed у всех направлений в случае, если каждый из них на этой «итерации» завершил свою работу. Таким образом, между ситуациями, когда ни одно из направлений не должно быть открыто, каждое из направлений выполнит свою работу. Процесс использует специальный маркер COMPLETED_MARKER, определяющий количество процессов, завершивших работу.

```
proctype MarkAsNotCompleted() {
    do
    :: COMPLETED_MARKER == PROCESSES_NUM -> {
            COMPLETED_MARKER = 0;
           NS.completed = 0; WN.completed = 0; SD.completed = 0;
           ED.completed = 0; DE.completed = 0; DN.completed = 0;
       }
    od
}
```

1.3.3. Процессы Direction(DIR)

Были реализованы 6 схожих процессов для каждого из направлений (NS, WN, SD, ED, DE и DN), управляющие их состоянием.

Процесс выполняет действия в бесконечном цикле в случае, если свою задачу не завершил (completed==0). Если в направлении присутствуют машины – процесс выполняет следующую последовательность действий:

1. ожидает, пока может взять контроль над пересечениями с конфликтными направлениями, последовательно берет контроль.
2. открывает направление;
3. пропускает поток машин;
4. закрывает направление.

Далее независимо от того, присутствовали машины или нет он завершает работу, изменяя соответствующие маркеры. Пример реализации процесса для направления NS представлен ниже.

```
proctype DirectionNS() {
	do
	:: (!NS.completed) -> {
		if
		:: NS.cars -> {
			acquire(NS_WN); acquire(NS_SD); acquire(NS_ED);
acquire(NS_DN); acquire(NS_DE);
			NS.open = 1;
			printf("Opened NS direction")
			NS.cars = 0;
			printf("Released cars for NS direction")
			NS.open = 0;
			printf("Closed NS direction")
			release(NS_WN); release(NS_SD); release(NS_ED);
release(NS_DN); release(NS_DE);
		}
		fi
		NS.completed = 1; COMPLETED_MARKER = COMPLETED_MARKER + 1;
		printf("Finished NS direction")
	}
	od
}
```

1.3.4. Процесс init

Процесс init отвечает за одновременное (atomic) создание всех описанных выше процессов, включая AddCarsToDirection, MarkAsCompleted и шести процессов, отвечающих за направления.

```
init {
	atomic {
		run DirectionNS();
		run DirectionWN();
		run DirectionSD();
		run DirectionED();
		run DirectionDE();
		run DirectionDN();

		run AddCarsToDirection();
		run MarkAsNotCompleted();
	}
}
```

2. ВЕРИФИКАЦИЯ МОДЕЛИ

Для верификации реализованной модели было проверено, что для каждого из направлений NS, WN, SD, ED, DE и DN соблюдаются критерии безопасности, живости и справедливости. Критерии были описаны соответствующими LTL-формулами.

2.1. Свойство безопасности

Формулировка – «Никогда не случится ситуации, что направление открыто для движения и при этом какое-либо из конфликтующих направлений также открыто для движения»

LTL формулы для направлений представлены в таблице 2.

Таблица 2. Свойство безопасности

	LTL-формула
NS	$G(\overline{NS.open \;\&\&\; (WN.open \mid SD.open \mid ED.open \mid\; DE.open \mid DN.open)})$
WN	$G(\overline{WN.open \;\&\&\; (NS.open \mid SD.open \mid ED.open \mid DE.open\;)})$
SD	$G(\overline{SD.open \;\&\&\; (NS.open \mid WN.open \mid\; DE.open \mid DN.open)})$
ED	$G(\overline{ED.open \;\&\&\; (NS.open \mid WN.open \mid DN.open)})$
DE	$G(\overline{DE.open \;\&\&\; (NS.open \mid WN.open \mid SD.open)})$
DN	$G(\overline{DN.open \;\&\&\; (NS.open \mid SD.open \mid ED.open)})$

Описание LTL-формул в Promela:

```
ltl safety0 { [] ! (NS.open && (WN.open || SD.open || ED.open || DE.open ||
DN.open))}
ltl safety1 { [] ! (WN.open && (NS.open || SD.open || ED.open || DE.open))}
ltl safety2 { [] ! (SD.open && (NS.open || WN.open || DE.open || DN.open))}
ltl safety3 { [] ! (ED.open && (NS.open || WN.open || DN.open))}
ltl safety4 { [] ! (DE.open && (NS.open || WN.open || SD.open))}
ltl safety5 { [] ! (DN.open && (NS.open || SD.open || ED.open))}
```

2.2. Свойство живости

Формулировка – «Всегда правда, что если в направлении присутствуют машины и направление закрыто – то когда-нибудь направление откроется»

LTL-формула одинакова для всех направлений (DIR - направление):

$$G((DIR.cars \;\&\&\; \overline{DIR.open}) \rightarrow F(DIR.open))$$

```
ltl liveness0 { [] ((NS.cars && !NS.open) -> <> (NS.open))}
ltl liveness1 { [] ((WN.cars && !WN.open) -> <> (WN.open))}
ltl liveness2 { [] ((SD.cars && !SD.open) -> <> (SD.open))}
ltl liveness3 { [] ((ED.cars && !ED.open) -> <> (ED.open))}
ltl liveness4 { [] ((DE.cars && !DE.open) -> <> (DE.open))}
ltl liveness5 { [] ((DN.cars && !DN.open) -> <> (DN.open))}
```

2.3. Свойство справедливости

Формулировка – «Всегда в будущем либо направление закрыто, либо на направлении нет машин»

LTL-формула одинакова для всех направлений (DIR - направление):

$$GF(\overline{DIR.open} \;\&\&\; DIR.cars)$$

```
ltl fairness0 { [] (<> (! (NS.open && NS.cars)))}
ltl fairness1 { [] (<> (! (WN.open && WN.cars)))}
ltl fairness2 { [] (<> (! (SD.open && SD.cars)))}
ltl fairness3 { [] (<> (! (ED.open && ED.cars)))}
ltl fairness4 { [] (<> (! (DE.open && DE.cars)))}
ltl fairness5 { [] (<> (! (DN.open && DN.cars)))}
```

2.4. Верификация в Spin

При верификации модели с помощью Spin были установлены параметры, представленные на рисунке 2.

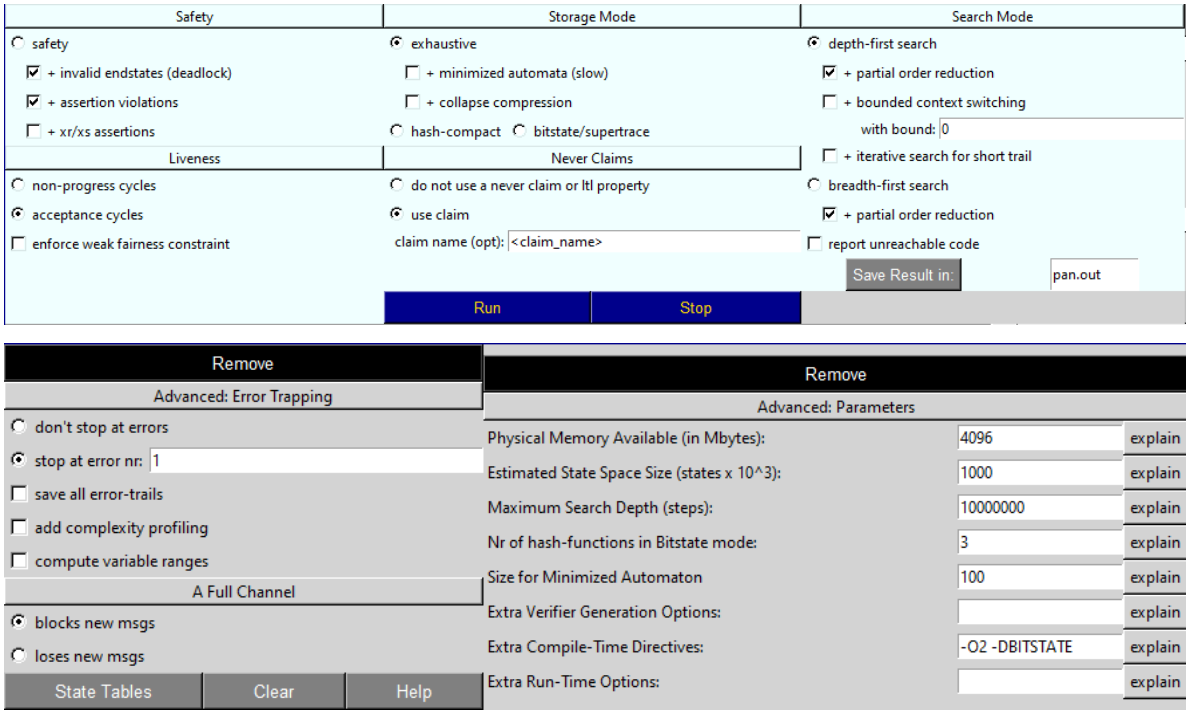

Рисунок 2. Параметры для верификации свойств

Функция «report unreachable code» отключена по причине, что предполагается, что модель работает бесконечно, а значит процессы не приходят в состояние “-end-“.

По причине того, что для верификации модели необходимо использовать большие вычислительные ресурсы, параметры используемой памяти и максимальной глубины поиска увеличены, а также использован аргумент DBITSTATE при компиляции.

Результаты верификации на примере направления NS свойств безопасности (safety0), безопасности (liveness0) и справедливости (fairness0) представлены в приложениях B, C и D соответственно.

ЗАКЛЮЧЕНИЕ

В результате выполнения данной работы была выполнена задача моделирования сложного перекрестка, содержащего конфликтующие направления. По итогам ее выполнения была реализована модель перекрестка на языке Promela, использующая 2 процесса, управляющих внешней средой и 6 процессов, управляющих направлениями, заданными в условии задачи.

Для итоговой модели была проведена верификация свойств безопасности, живости и справедливости, выраженных в LTL-формулах. По результатам верификации можно сказать, что модель корректна.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Карпов Ю.Г., Шошмина И.В. Верификация распределенных систем: учеб. пособие СПб.: Изд-во Политехнического университета, 2011. 212 с.

2. Шошмина И.В., Карпов Ю.Г. Введение в язык Promela и систему комплексной верификации Spin: учеб. пособие СПб.: Изд-во Политехнического университета, 2009

3. Карпов Ю.Г. Model checking. Верификация параллельных и распределенных программных систем. // БХВ-Петербург, 2009, 520 с.

4. Документация к Promela // Promela Manual Pages [Электронный ресурс]. URL: https://spinroot.com/spin/Man/promela.html (дата обращения: 26.04.2024)

5. Документация к Spin // Spin Online References [Электронный ресурс]. URL: https://spinroot.com/spin/Man/ (дата обращения: 26.04.2024)

ПРИЛОЖЕНИЕ А
ИСХОДНЫЙ КОД МОДЕЛИ

Название файла - traffic-light.pml

```
#define PROCESSES_NUM 6
typedef Direction {
	bool cars = 0;
	bool open = 0;
	bool completed = 1;
};
typedef Semaphore{
	byte C = 1;
}
inline acquire(sem){
	atomic{
		sem.C > 0;
		sem.C--;
	}
}
inline release(sem){
	sem.C++;
}
Semaphore NS_WN, NS_SD, NS_ED, NS_DE, NS_DN;
Semaphore WN_SD, WN_ED, WN_DE;
Semaphore SD_DE, SD_DN;
Semaphore ED_DN;
byte COMPLETED_MARKER = PROCESSES_NUM
Direction NS, WN, SD, ED, DE, DN;
// LTL-Checks
// Safety: Always (if direction is open then conflicting one is open) is
False
ltl safety0 { [] ! (NS.open && (WN.open || SD.open || ED.open || DE.open
|| DN.open))}
ltl safety1 { [] ! (WN.open && (NS.open || SD.open || ED.open ||
DE.open))}
ltl safety2 { [] ! (SD.open && (NS.open || WN.open || DE.open ||
DN.open))}
ltl safety3 { [] ! (ED.open && (NS.open || WN.open || DN.open))}
ltl safety4 { [] ! (DE.open && (NS.open || WN.open || SD.open))}
```

```
ltl safety5 { [] ! (DN.open && (NS.open || SD.open || ED.open))}
// Liveness: Always (if there are cars in direction and it is closed then
later it will open) is True
ltl liveness0 { [] ((NS.cars && !NS.open) -> <> (NS.open))}
ltl liveness1 { [] ((WN.cars && !WN.open) -> <> (WN.open))}
ltl liveness2 { [] ((SD.cars && !SD.open) -> <> (SD.open))}
ltl liveness3 { [] ((ED.cars && !ED.open) -> <> (ED.open))}
ltl liveness4 { [] ((DE.cars && !DE.open) -> <> (DE.open))}
ltl liveness5 { [] ((DN.cars && !DN.open) -> <> (DN.open))}
// Fairness: Always in the future either there won't be any cars either
the direction will close
ltl fairness0 { [] (<> (! (NS.open && NS.cars)))}
ltl fairness1 { [] (<> (! (WN.open && WN.cars)))}
ltl fairness2 { [] (<> (! (SD.open && SD.cars)))}
ltl fairness3 { [] (<> (! (ED.open && ED.cars)))}
ltl fairness4 { [] (<> (! (DE.open && DE.cars)))}
ltl fairness5 { [] (<> (! (DN.open && DN.cars)))}
// Process that adds new cars to any empty direction
proctype AddCarsToDirection() {
      do
      :: !NS.cars -> {NS.cars = 1;}
      :: !WN.cars -> {WN.cars = 1;}
      :: !SD.cars -> {SD.cars = 1;}
      :: !ED.cars -> {ED.cars = 1;}
      :: !DE.cars -> {DE.cars = 1;}
      :: !DN.cars -> {DN.cars = 1;}
      od
}
// Process that marks all the directions as not completed
proctype MarkAsNotCompleted() {
    do
    :: COMPLETED_MARKER == PROCESSES_NUM -> {
           COMPLETED_MARKER = 0;
           NS.completed = 0; WN.completed = 0; SD.completed = 0;
           ED.completed = 0; DE.completed = 0; DN.completed = 0;
       }
    od
}
```

```
proctype DirectionNS() {
      do
      :: (!NS.completed) -> {
            if
            :: NS.cars -> {
                  acquire(NS_WN); acquire(NS_SD); acquire(NS_ED);
acquire(NS_DN); acquire(NS_DE);
                  NS.open = 1;
                  printf("Opened NS direction")
                  NS.cars = 0;
                  printf("Released cars for NS direction")
                  NS.open = 0;
                  printf("Closed NS direction")
                  release(NS_WN); release(NS_SD); release(NS_ED);
release(NS_DN); release(NS_DE);
            }
            fi
            NS.completed = 1; COMPLETED_MARKER = COMPLETED_MARKER + 1;
            printf("Finished NS direction")
      }
      od
}
proctype DirectionWN() {
      do
      :: (!WN.completed) -> {
            if
            :: WN.cars -> {
                  acquire(NS_WN); acquire(WN_DE); acquire(WN_ED);
acquire(WN_SD);
                  WN.open = 1;
                  printf("Opened WN direction")
                  WN.cars = 0;
                  printf("Released cars for WN direction")
                  WN.open = 0;
                  printf("Closed WN direction")
                  release(NS_WN); release(WN_DE); release(WN_ED);
release(WN_SD);
            }
```

```
            fi
            WN.completed = 1; COMPLETED_MARKER = COMPLETED_MARKER + 1;
            printf("Finished WN direction")
      }
      od
}
proctype DirectionSD() {
      do
      :: (!SD.completed) -> {
            if
            :: SD.cars -> {
                  acquire(NS_SD); acquire(WN_SD); acquire(SD_DE);
acquire(SD_DN);
                  SD.open = 1;
                  printf("Opened SD direction")
                  SD.cars = 0;
                  printf("Released cars for SD direction")
                  SD.open = 0;
                  printf("Closed SD direction")
                  release(NS_SD); release(WN_SD); release(SD_DE);
release(SD_DN);
            }
            fi
            SD.completed = 1; COMPLETED_MARKER = COMPLETED_MARKER + 1;
            printf("Finished SD direction")
      }
      od
}
proctype DirectionED() {
      do
      :: (!ED.completed) -> {
            if
            :: ED.cars -> {
                  acquire(NS_ED); acquire(WN_ED); acquire(ED_DN);
                  ED.open = 1;
                  printf("Opened ED direction")
                  ED.cars = 0;
                  printf("Released cars for ED direction")
```

```
                ED.open = 0;
                printf("Closed ED direction")
                release(NS_ED); release(WN_ED); release(ED_DN);
            }
            fi
            ED.completed = 1; COMPLETED_MARKER = COMPLETED_MARKER + 1;
            printf("Finished ED direction")
        }
        od
}
proctype DirectionDE() {
        do
        :: (!DE.completed) -> {
            if
            :: DE.cars -> {
                acquire(NS_DE); acquire(WN_DE); acquire(SD_DE);
                DE.open = 1;
                printf("Opened DE direction")
                DE.cars = 0;
                printf("Released cars for DE direction")
                DE.open = 0;
                printf("Closed DE direction")
                release(NS_DE); release(WN_DE); release(SD_DE);
            }
            fi
            DE.completed = 1; COMPLETED_MARKER = COMPLETED_MARKER + 1;
            printf("Finished DE direction")
        }
        od
}
proctype DirectionDN() {
        do
        :: (!DN.completed) -> {
            if
            :: DN.cars -> {
                acquire(NS_DN); acquire(SD_DN); acquire(ED_DN);
                DN.open = 1;
                printf("Opened DN direction")
```

```
                DN.cars = 0;
                printf("Released cars for DN direction")
                DN.open = 0;
                printf("Closed DN direction")
                release(NS_DN); release(SD_DN); release(ED_DN);
            }
            fi
            DN.completed = 1; COMPLETED_MARKER = COMPLETED_MARKER + 1;
            printf("Finished DN direction")
        }
        od
}
init {
        atomic {
            run DirectionNS();
            run DirectionWN();
            run DirectionSD();
            run DirectionED();
            run DirectionDE();
            run DirectionDN();

            run AddCarsToDirection();
            run MarkAsNotCompleted();
        }
}
```

ПРИЛОЖЕНИЕ В

ВЕРИФИКАЦИЯ СВОЙСТВА БЕЗОПАСНОСТИ (NS)

```
./pan -m10000000  -a -n -N safety0
Pid: 13740
pan: ltl formula safety0
Depth=  392876 States=    1e+06 Transitions= 2.95e+06 Memory=   650.473
      t=    0.735 R=   1e+06
Depth=  468170 States=    2e+06 Transitions= 5.97e+06 Memory=   655.063
      t=     1.45 R=   1e+06
Depth=  605044 States=    3e+06 Transitions= 9.03e+06 Memory=   663.461
      t=     2.17 R=   1e+06
Depth=  643000 States=    4e+06 Transitions=  1.2e+07 Memory=   665.707
      t=      2.9 R=   1e+06
Depth=  648684 States=    5e+06 Transitions= 1.53e+07 Memory=   666.098
      t=     3.65 R=   1e+06
Depth=  762844 States=    6e+06 Transitions= 1.85e+07 Memory=   673.031
      t=     4.42 R=   1e+06
Depth=  909760 States=    7e+06 Transitions= 2.15e+07 Memory=   682.016
      t=     5.17 R=   1e+06
Depth= 1020252 States=    8e+06 Transitions= 2.45e+07 Memory=   688.754
      t=      5.9 R=   1e+06
Depth= 1113336 States=    9e+06 Transitions= 2.76e+07 Memory=   694.418
      t=     6.65 R=   1e+06
Depth= 1151560 States=    1e+07 Transitions= 3.07e+07 Memory=   696.762
      t=     7.46 R=   1e+06
Depth= 1162996 States=  1.1e+07 Transitions= 3.34e+07 Memory=   697.445
      t=     8.25 R=   1e+06
Depth= 1181200 States=  1.2e+07 Transitions= 3.68e+07 Memory=   698.617
      t=     9.12 R=   1e+06
Depth= 1194434 States=  1.3e+07 Transitions= 3.98e+07 Memory=   699.399
      t=     9.94 R=   1e+06
Depth= 1252952 States=  1.4e+07 Transitions= 4.32e+07 Memory=   703.012
      t=     10.8 R=   1e+06
Depth= 1278642 States=  1.5e+07 Transitions= 4.62e+07 Memory=   704.574
      t=     11.6 R=   1e+06
Depth= 1292724 States=  1.6e+07 Transitions= 4.91e+07 Memory=   705.356
      t=     12.3 R=   1e+06
Depth= 1294816 States=  1.7e+07 Transitions= 5.19e+07 Memory=   705.551
      t=     13.1 R=   1e+06
Depth= 1296834 States=  1.8e+07 Transitions= 5.53e+07 Memory=   705.649
      t=     13.9 R=   1e+06
Depth= 1300766 States=  1.9e+07 Transitions= 5.89e+07 Memory=   705.844
      t=     14.8 R=   1e+06
Depth= 1320270 States=    2e+07 Transitions= 6.19e+07 Memory=   707.113
      t=     15.6 R=   1e+06
Depth= 1337680 States=  2.1e+07 Transitions= 6.51e+07 Memory=   708.188
      t=     16.3 R=   1e+06
Depth= 1353284 States=  2.2e+07 Transitions= 6.87e+07 Memory=   709.067
      t=     17.2 R=   1e+06
Depth= 1427668 States=  2.3e+07 Transitions= 7.21e+07 Memory=   713.656
      t=     18.1 R=   1e+06
Depth= 1482768 States=  2.4e+07 Transitions= 7.52e+07 Memory=   716.977
      t=     18.9 R=   1e+06
Depth= 1488256 States=  2.5e+07 Transitions= 7.85e+07 Memory=   717.367
      t=     19.7 R=   1e+06
```

```
Depth= 1488256 States=  2.6e+07 Transitions= 8.21e+07 Memory=   717.367
     t=     20.6 R=   1e+06
Depth= 1488262 States=  2.7e+07 Transitions= 8.56e+07 Memory=   717.367
     t=     21.4 R=   1e+06
Depth= 1488262 States=  2.8e+07 Transitions=  8.9e+07 Memory=   717.367
     t=     22.2 R=   1e+06
Depth= 1488262 States=  2.9e+07 Transitions=  9.2e+07 Memory=   717.367
     t=     22.9 R=   1e+06
Depth= 1494222 States=    3e+07 Transitions= 9.56e+07 Memory=   717.660
     t=     23.8 R=   1e+06
Depth= 1496526 States=  3.1e+07 Transitions= 9.89e+07 Memory=   717.856
     t=     24.7 R=   1e+06
Depth= 1496554 States=  3.2e+07 Transitions= 1.02e+08 Memory=   717.856
     t=     25.5 R=   1e+06
Depth= 1523846 States=  3.3e+07 Transitions= 1.06e+08 Memory=   719.516
     t=     26.3 R=   1e+06
Depth= 1562614 States=  3.4e+07 Transitions= 1.09e+08 Memory=   721.860
     t=     27.2 R=   1e+06
Depth= 1574060 States=  3.5e+07 Transitions= 1.12e+08 Memory=   722.543
     t=       28 R=   1e+06
Depth= 1579782 States=  3.6e+07 Transitions= 1.16e+08 Memory=   722.934
     t=     28.8 R=   1e+06
Depth= 1588614 States=  3.7e+07 Transitions= 1.19e+08 Memory=   723.422
     t=     29.6 R=   1e+06
Depth= 1594594 States=  3.8e+07 Transitions= 1.22e+08 Memory=   723.813
     t=     30.5 R=   1e+06
Depth= 1594600 States=  3.9e+07 Transitions= 1.25e+08 Memory=   723.813
     t=     31.3 R=   1e+06
Depth= 1594600 States=    4e+07 Transitions= 1.28e+08 Memory=   723.813
     t=     32.1 R=   1e+06
Depth= 1594600 States=  4.1e+07 Transitions= 1.31e+08 Memory=   723.813
     t=     32.8 R=   1e+06
Depth= 1594600 States=  4.2e+07 Transitions= 1.34e+08 Memory=   723.813
     t=     33.7 R=   1e+06
Depth= 1594600 States=  4.3e+07 Transitions= 1.38e+08 Memory=   723.813
     t=     34.7 R=   1e+06
Depth= 1594600 States=  4.4e+07 Transitions= 1.43e+08 Memory=   723.813
     t=     35.7 R=   1e+06
Depth= 1594600 States=  4.5e+07 Transitions= 1.46e+08 Memory=   723.813
     t=     36.7 R=   1e+06
Depth= 1594600 States=  4.6e+07 Transitions= 1.51e+08 Memory=   723.813
     t=     37.7 R=   1e+06
Depth= 1594600 States=  4.7e+07 Transitions= 1.54e+08 Memory=   723.813
     t=     38.6 R=   1e+06
Depth= 1596396 States=  4.8e+07 Transitions= 1.58e+08 Memory=   723.910
     t=     39.6 R=   1e+06
Depth= 1600290 States=  4.9e+07 Transitions= 1.62e+08 Memory=   724.203
     t=     40.5 R=   1e+06
Depth= 1605000 States=    5e+07 Transitions= 1.66e+08 Memory=   724.496
     t=     41.5 R=   1e+06
Depth= 1608120 States=  5.1e+07 Transitions= 1.69e+08 Memory=   724.692
     t=     42.3 R=   1e+06
Depth= 1609400 States=  5.2e+07 Transitions= 1.73e+08 Memory=   724.692
     t=     43.3 R=   1e+06
Depth= 1609934 States=  5.3e+07 Transitions= 1.76e+08 Memory=   724.789
     t=     44.2 R=   1e+06
```

```
Depth= 1610194 States=  5.4e+07 Transitions=  1.8e+08 Memory=   724.789
      t=     45.2 R=   1e+06
Depth= 1610460 States=  5.5e+07 Transitions= 1.84e+08 Memory=   724.789
      t=     46.1 R=   1e+06
Depth= 1610464 States=  5.6e+07 Transitions= 1.88e+08 Memory=   724.789
      t=     47.1 R=   1e+06
Depth= 1610464 States=  5.7e+07 Transitions= 1.91e+08 Memory=   724.789
      t=     48.1 R=   1e+06
Depth= 1610464 States=  5.8e+07 Transitions= 1.95e+08 Memory=   724.789
      t=     49.1 R=   1e+06
Depth= 1610464 States=  5.9e+07 Transitions= 1.99e+08 Memory=   724.789
      t=     50.1 R=   1e+06
Depth= 1610464 States=    6e+07 Transitions= 2.03e+08 Memory=   724.789
      t=     51.1 R=   1e+06
Depth= 1610464 States=  6.1e+07 Transitions= 2.07e+08 Memory=   724.789
      t=     52.2 R=   1e+06
Depth= 1610464 States=  6.2e+07 Transitions= 2.11e+08 Memory=   724.789
      t=     53.3 R=   1e+06
Depth= 1610464 States=  6.3e+07 Transitions= 2.15e+08 Memory=   724.789
      t=     54.4 R=   1e+06
Depth= 1610464 States=  6.4e+07 Transitions= 2.18e+08 Memory=   724.789
      t=     55.7 R=   1e+06
Depth= 1610464 States=  6.5e+07 Transitions= 2.23e+08 Memory=   724.789
      t=       57 R=   1e+06
Depth= 1610464 States=  6.6e+07 Transitions= 2.27e+08 Memory=   724.789
      t=     58.5 R=   1e+06
Depth= 1610464 States=  6.7e+07 Transitions= 2.32e+08 Memory=   724.789
      t=       60 R=   1e+06
(Spin Version 6.5.1 -- 20 December 2019)
      + Partial Order Reduction

Bit statespace search for:
      never claim          + (safety0)
      assertion violations  + (if within scope of claim)
      acceptance   cycles   + (fairness disabled)
      invalid end states    - (disabled by never claim)

State-vector 104 byte, depth reached 1610464, errors: 0
 67527086 states, stored
1.68455e+08 states, matched
2.3598209e+08 transitions (= stored+matched)
        7 atomic steps

hash factor: 1.98761 (best if > 100.)
bits set per state: 3 (-k3)

Stats on memory usage (in Megabytes):
 7985.457  equivalent memory usage for states (stored*(State-vector +
overhead))
   16.000  memory used for hash array (-w27)
   76.294  memory used for bit stack
  534.058  memory used for DFS stack (-m10000000)
   98.399  other (proc and chan stacks)
  724.789  total actual memory usage

pan: elapsed time 61.5 seconds
No errors found -- did you verify all claims?
```

ПРИЛОЖЕНИЕ С

ВЕРИФИКАЦИЯ СВОЙСТВА ЖИВОСТИ (NS)

```
./pan -m10000000  -a -n -N liveness0
Pid: 2960
pan: ltl formula liveness0
Depth=     263 States=    1e+06 Transitions=  3.4e+06 Memory=   626.547
      t=    0.815 R=   1e+06
Depth=     263 States=    2e+06 Transitions= 6.95e+06 Memory=   626.547
      t=     1.65 R=   1e+06
Depth=     263 States=    3e+06 Transitions= 1.08e+07 Memory=   626.547
      t=     2.56 R=   1e+06
Depth=     263 States=    4e+06 Transitions= 1.45e+07 Memory=   626.547
      t=     3.42 R=   1e+06
Depth=     263 States=    5e+06 Transitions= 1.82e+07 Memory=   626.547
      t=      4.3 R=   1e+06
Depth=     263 States=    6e+06 Transitions= 2.22e+07 Memory=   626.547
      t=     5.25 R=   1e+06
Depth=     263 States=    7e+06 Transitions= 2.53e+07 Memory=   626.547
      t=     6.07 R=   1e+06
Depth=     263 States=    8e+06 Transitions= 2.86e+07 Memory=   626.547
      t=     6.94 R=   1e+06
Depth=     263 States=    9e+06 Transitions= 3.17e+07 Memory=   626.547
      t=     7.86 R=   1e+06
Depth=     263 States=    1e+07 Transitions=  3.5e+07 Memory=   626.547
      t=     8.84 R=   1e+06
Depth=     263 States=  1.1e+07 Transitions= 3.85e+07 Memory=   626.547
      t=     9.76 R=   1e+06
Depth=     263 States=  1.2e+07 Transitions= 4.18e+07 Memory=   626.547
      t=     10.6 R=   1e+06
Depth=     263 States=  1.3e+07 Transitions=  4.5e+07 Memory=   626.547
      t=     11.5 R=   1e+06
Depth=     263 States=  1.4e+07 Transitions= 4.84e+07 Memory=   626.547
      t=     12.5 R=   1e+06
Depth=     263 States=  1.5e+07 Transitions= 5.18e+07 Memory=   626.547
      t=     13.4 R=   1e+06
Depth=     263 States=  1.6e+07 Transitions= 5.55e+07 Memory=   626.547
      t=     14.6 R=   1e+06
Depth=     263 States=  1.7e+07 Transitions= 5.98e+07 Memory=   626.547
      t=       16 R=   1e+06
Depth=     263 States=  1.8e+07 Transitions=  6.4e+07 Memory=   626.547
      t=     17.2 R=   1e+06
Depth=     263 States=  1.9e+07 Transitions= 6.83e+07 Memory=   626.547
      t=     18.4 R=   1e+06
Depth=     263 States=    2e+07 Transitions= 7.25e+07 Memory=   626.547
      t=     19.7 R=   1e+06
Depth=     263 States=  2.1e+07 Transitions= 7.71e+07 Memory=   626.547
      t=       21 R=   1e+06
Depth=     263 States=  2.2e+07 Transitions= 8.16e+07 Memory=   626.547
      t=     22.3 R=   1e+06
Depth=     263 States=  2.3e+07 Transitions= 8.58e+07 Memory=   626.547
      t=     23.5 R=   1e+06
Depth=     263 States=  2.4e+07 Transitions= 8.94e+07 Memory=   626.547
      t=     24.6 R=   1e+06
Depth=     263 States=  2.5e+07 Transitions= 9.36e+07 Memory=   626.547
      t=     25.8 R=   1e+06
```

```
Depth=     263 States=  2.6e+07 Transitions= 9.73e+07 Memory=   626.547
     t=     26.8 R=   1e+06
Depth=     263 States=  2.7e+07 Transitions= 1.01e+08 Memory=   626.547
     t=     27.9 R=   1e+06
Depth=     263 States=  2.8e+07 Transitions= 1.05e+08 Memory=   626.547
     t=       29 R=   1e+06
Depth=     263 States=  2.9e+07 Transitions= 1.09e+08 Memory=   626.547
     t=     30.2 R=   1e+06
Depth=     263 States=    3e+07 Transitions= 1.13e+08 Memory=   626.547
     t=     31.3 R=   1e+06
Depth=     263 States=  3.1e+07 Transitions= 1.17e+08 Memory=   626.547
     t=     32.4 R=   1e+06
Depth=     263 States=  3.2e+07 Transitions= 1.21e+08 Memory=   626.547
     t=     33.6 R=   1e+06
Depth=     263 States=  3.3e+07 Transitions= 1.24e+08 Memory=   626.547
     t=     34.7 R=   1e+06
Depth=     263 States=  3.4e+07 Transitions= 1.28e+08 Memory=   626.547
     t=     35.8 R=   9e+05
Depth=     263 States=  3.5e+07 Transitions= 1.31e+08 Memory=   626.547
     t=       37 R=   9e+05
Depth=     263 States=  3.6e+07 Transitions= 1.35e+08 Memory=   626.547
     t=     38.1 R=   9e+05
Depth=     263 States=  3.7e+07 Transitions= 1.38e+08 Memory=   626.547
     t=     39.4 R=   9e+05
Depth=     263 States=  3.8e+07 Transitions= 1.42e+08 Memory=   626.547
     t=     40.6 R=   9e+05
Depth=     263 States=  3.9e+07 Transitions= 1.45e+08 Memory=   626.547
     t=       42 R=   9e+05
Depth=     263 States=    4e+07 Transitions= 1.49e+08 Memory=   626.547
     t=     43.3 R=   9e+05
Depth=     263 States=  4.1e+07 Transitions= 1.53e+08 Memory=   626.547
     t=     44.6 R=   9e+05
Depth=     263 States=  4.2e+07 Transitions= 1.57e+08 Memory=   626.547
     t=       46 R=   9e+05
Depth=     263 States=  4.3e+07 Transitions= 1.62e+08 Memory=   626.547
     t=     47.5 R=   9e+05
Depth=     263 States=  4.4e+07 Transitions= 1.66e+08 Memory=   626.547
     t=       49 R=   9e+05
Depth=     263 States=  4.5e+07 Transitions= 1.71e+08 Memory=   626.547
     t=     50.5 R=   9e+05
Depth=     263 States=  4.6e+07 Transitions= 1.76e+08 Memory=   626.547
     t=     52.2 R=   9e+05
Depth=     263 States=  4.7e+07 Transitions= 1.81e+08 Memory=   626.547
     t=     53.8 R=   9e+05
Depth=     263 States=  4.8e+07 Transitions= 1.86e+08 Memory=   626.547
     t=     55.1 R=   9e+05
Depth=     263 States=  4.9e+07 Transitions=  1.9e+08 Memory=   626.547
     t=     56.4 R=   9e+05
Depth=     263 States=    5e+07 Transitions= 1.94e+08 Memory=   626.547
     t=     57.7 R=   9e+05
Depth=     263 States=  5.1e+07 Transitions= 1.98e+08 Memory=   626.547
     t=     59.1 R=   9e+05
Depth=     263 States=  5.2e+07 Transitions= 2.03e+08 Memory=   626.547
     t=     60.6 R=   9e+05
Depth=     263 States=  5.3e+07 Transitions= 2.07e+08 Memory=   626.547
     t=     62.2 R=   9e+05
```

```
Depth=     263 States=  5.4e+07 Transitions= 2.11e+08 Memory=   626.547
     t=     63.6 R=   8e+05
Depth=     263 States=  5.5e+07 Transitions= 2.15e+08 Memory=   626.547
     t=       65 R=   8e+05
Depth=     263 States=  5.6e+07 Transitions= 2.19e+08 Memory=   626.547
     t=     66.5 R=   8e+05
Depth=     263 States=  5.7e+07 Transitions= 2.23e+08 Memory=   626.547
     t=     67.9 R=   8e+05
Depth=     263 States=  5.8e+07 Transitions= 2.28e+08 Memory=   626.547
     t=     69.2 R=   8e+05
Depth=     263 States=  5.9e+07 Transitions= 2.32e+08 Memory=   626.547
     t=     70.6 R=   8e+05
Depth=     263 States=    6e+07 Transitions= 2.37e+08 Memory=   626.547
     t=     71.9 R=   8e+05
Depth=     263 States=  6.1e+07 Transitions= 2.41e+08 Memory=   626.547
     t=     73.5 R=   8e+05
Depth=     263 States=  6.2e+07 Transitions= 2.46e+08 Memory=   626.547
     t=     75.2 R=   8e+05
Depth=     263 States=  6.3e+07 Transitions= 2.51e+08 Memory=   626.547
     t=     76.8 R=   8e+05
Depth=     263 States=  6.4e+07 Transitions= 2.55e+08 Memory=   626.547
     t=     78.4 R=   8e+05
Depth=     263 States=  6.5e+07 Transitions=  2.6e+08 Memory=   626.547
     t=     79.9 R=   8e+05
Depth=     296 States=  6.6e+07 Transitions= 2.65e+08 Memory=   626.547
     t=     81.6 R=   8e+05
Depth=     296 States=  6.7e+07 Transitions= 2.71e+08 Memory=   626.547
     t=     83.3 R=   8e+05
(Spin Version 6.5.1 -- 20 December 2019)
     + Partial Order Reduction

Bit statespace search for:
     never claim          + (liveness0)
     assertion violations  + (if within scope of claim)
     acceptance   cycles   + (fairness disabled)
     invalid end states    - (disabled by never claim)

State-vector 104 byte, depth reached 296, errors: 0
 32138325 states, stored (6.7932e+07 visited)
2.0856253e+08 states, matched
2.7649448e+08 transitions (= visited+matched)
        7 atomic steps

hash factor: 1.97577 (best if > 100.)
bits set per state: 3 (-k3)

Stats on memory usage (in Megabytes):
 3800.537  equivalent memory usage for states (stored*(State-vector +
overhead))
   16.000  memory used for hash array (-w27)
   76.294  memory used for bit stack
  534.058  memory used for DFS stack (-m10000000)
  626.547  total actual memory usage

pan: elapsed time 84.9 seconds
No errors found -- did you verify all claims?
```

ПРИЛОЖЕНИЕ D

ВЕРИФИКАЦИЯ СВОЙСТВА СПРАВЕДЛИВОСТИ (NS)

```
./pan -m10000000  -a -n -N fairness0
Pid: 3052
pan: ltl formula fairness0
Depth=  371042 States=    1e+06 Transitions= 2.95e+06 Memory=   649.106
      t=    0.719 R=   1e+06
Depth=  459172 States=    2e+06 Transitions= 6.02e+06 Memory=   654.477
      t=     1.42 R=   1e+06
Depth=  608416 States=    3e+06 Transitions= 9.13e+06 Memory=   663.656
      t=     2.17 R=   1e+06
Depth=  661200 States=    4e+06 Transitions= 1.23e+07 Memory=   666.879
      t=     2.93 R=   1e+06
Depth=  663254 States=    5e+06 Transitions= 1.54e+07 Memory=   666.977
      t=     3.66 R=   1e+06
Depth=  690840 States=    6e+06 Transitions= 1.87e+07 Memory=   668.637
      t=     4.41 R=   1e+06
Depth=  829934 States=    7e+06 Transitions= 2.18e+07 Memory=   677.133
      t=     5.16 R=   1e+06
Depth=  960464 States=    8e+06 Transitions= 2.49e+07 Memory=   685.141
      t=     5.91 R=   1e+06
Depth= 1052500 States=    9e+06 Transitions= 2.79e+07 Memory=   690.707
      t=     6.65 R=   1e+06
Depth= 1114118 States=    1e+07 Transitions=  3.1e+07 Memory=   694.516
      t=     7.41 R=   1e+06
Depth= 1160140 States=  1.1e+07 Transitions=  3.4e+07 Memory=   697.348
      t=     8.17 R=   1e+06
Depth= 1189952 States=  1.2e+07 Transitions= 3.73e+07 Memory=   699.106
      t=     9.02 R=   1e+06
Depth= 1203562 States=  1.3e+07 Transitions= 4.05e+07 Memory=   699.985
      t=     9.97 R=   1e+06
Depth= 1216266 States=  1.4e+07 Transitions= 4.37e+07 Memory=   700.766
      t=     10.9 R=   1e+06
Depth= 1270122 States=  1.5e+07 Transitions= 4.69e+07 Memory=   703.988
      t=     11.9 R=   1e+06
Depth= 1284158 States=  1.6e+07 Transitions= 4.99e+07 Memory=   704.867
      t=     12.8 R=   1e+06
Depth= 1308336 States=  1.7e+07 Transitions=  5.3e+07 Memory=   706.332
      t=     13.8 R=   1e+06
Depth= 1308336 States=  1.8e+07 Transitions= 5.59e+07 Memory=   706.332
      t=     14.7 R=   1e+06
Depth= 1308338 States=  1.9e+07 Transitions= 5.96e+07 Memory=   706.332
      t=     15.9 R=   1e+06
Depth= 1323110 States=    2e+07 Transitions= 6.31e+07 Memory=   707.211
      t=     16.9 R=   1e+06
Depth= 1323124 States=  2.1e+07 Transitions= 6.67e+07 Memory=   707.211
      t=       18 R=   1e+06
Depth= 1355624 States=  2.2e+07 Transitions= 6.99e+07 Memory=   709.262
      t=       19 R=   1e+06
Depth= 1387634 States=  2.3e+07 Transitions= 7.32e+07 Memory=   711.215
      t=       20 R=   1e+06
Depth= 1421410 States=  2.4e+07 Transitions= 7.63e+07 Memory=   713.266
      t=       21 R=   1e+06
Depth= 1445352 States=  2.5e+07 Transitions= 7.95e+07 Memory=   714.731
      t=       22 R=   1e+06
```

```
Depth= 1460920 States=  2.6e+07 Transitions= 8.29e+07 Memory=   715.707
     t=       23 R=   1e+06
Depth= 1474998 States=  2.7e+07 Transitions= 8.66e+07 Memory=   716.488
     t=     24.1 R=   1e+06
Depth= 1474998 States=  2.8e+07 Transitions=    9e+07 Memory=   716.488
     t=     25.2 R=   1e+06
Depth= 1474998 States=  2.9e+07 Transitions= 9.36e+07 Memory=   716.488
     t=     26.3 R=   1e+06
Depth= 1475000 States=    3e+07 Transitions= 9.72e+07 Memory=   716.488
     t=     27.3 R=   1e+06
Depth= 1475000 States=  3.1e+07 Transitions=    1e+08 Memory=   716.488
     t=     28.2 R=   1e+06
Depth= 1475000 States=  3.2e+07 Transitions= 1.04e+08 Memory=   716.488
     t=     29.1 R=   1e+06
Depth= 1478626 States=  3.3e+07 Transitions= 1.07e+08 Memory=   716.781
     t=     30.1 R=   1e+06
Depth= 1509552 States=  3.4e+07 Transitions= 1.11e+08 Memory=   718.637
     t=       31 R=   1e+06
Depth= 1515296 States=  3.5e+07 Transitions= 1.14e+08 Memory=   719.028
     t=       32 R=   1e+06
Depth= 1543378 States=  3.6e+07 Transitions= 1.18e+08 Memory=   720.688
     t=     32.9 R=   1e+06
Depth= 1543380 States=  3.7e+07 Transitions= 1.21e+08 Memory=   720.688
     t=     33.8 R=   1e+06
Depth= 1547730 States=  3.8e+07 Transitions= 1.24e+08 Memory=   720.981
     t=     34.7 R=   1e+06
Depth= 1564912 States=  3.9e+07 Transitions= 1.28e+08 Memory=   722.055
     t=     35.5 R=   1e+06
Depth= 1571194 States=    4e+07 Transitions= 1.31e+08 Memory=   722.348
     t=     36.4 R=   1e+06
Depth= 1571200 States=  4.1e+07 Transitions= 1.34e+08 Memory=   722.348
     t=     37.2 R=   1e+06
Depth= 1571200 States=  4.2e+07 Transitions= 1.37e+08 Memory=   722.348
     t=       38 R=   1e+06
Depth= 1571200 States=  4.3e+07 Transitions= 1.41e+08 Memory=   722.348
     t=       39 R=   1e+06
Depth= 1571200 States=  4.4e+07 Transitions= 1.45e+08 Memory=   722.348
     t=     40.3 R=   1e+06
Depth= 1571200 States=  4.5e+07 Transitions= 1.49e+08 Memory=   722.348
     t=     41.4 R=   1e+06
Depth= 1571448 States=  4.6e+07 Transitions= 1.53e+08 Memory=   722.445
     t=     42.5 R=   1e+06
Depth= 1571448 States=  4.7e+07 Transitions= 1.57e+08 Memory=   722.445
     t=     43.6 R=   1e+06
Depth= 1573244 States=  4.8e+07 Transitions=  1.6e+08 Memory=   722.543
     t=     44.9 R=   1e+06
Depth= 1573270 States=  4.9e+07 Transitions= 1.64e+08 Memory=   722.543
     t=     46.2 R=   1e+06
Depth= 1573512 States=    5e+07 Transitions= 1.68e+08 Memory=   722.543
     t=     47.4 R=   1e+06
Depth= 1578170 States=  5.1e+07 Transitions= 1.72e+08 Memory=   722.836
     t=     48.5 R=   1e+06
Depth= 1580044 States=  5.2e+07 Transitions= 1.76e+08 Memory=   722.934
     t=     49.6 R=   1e+06
Depth= 1580820 States=  5.3e+07 Transitions= 1.79e+08 Memory=   723.031
     t=     50.7 R=   1e+06
```

```
Depth= 1580822 States=  5.4e+07 Transitions= 1.83e+08 Memory=   723.031
      t=     51.8 R=   1e+06
Depth= 1580822 States=  5.5e+07 Transitions= 1.87e+08 Memory=   723.031
      t=     52.9 R=   1e+06
Depth= 1580822 States=  5.6e+07 Transitions= 1.91e+08 Memory=   723.031
      t=     54.1 R=   1e+06
Depth= 1580822 States=  5.7e+07 Transitions= 1.94e+08 Memory=   723.031
      t=     55.2 R=   1e+06
Depth= 1580822 States=  5.8e+07 Transitions= 1.98e+08 Memory=   723.031
      t=     56.4 R=   1e+06
Depth= 1580824 States=  5.9e+07 Transitions= 2.02e+08 Memory=   723.031
      t=     57.5 R=   1e+06
Depth= 1580824 States=    6e+07 Transitions= 2.06e+08 Memory=   723.031
      t=     58.6 R=   1e+06
Depth= 1580824 States=  6.1e+07 Transitions= 2.09e+08 Memory=   723.031
      t=     59.9 R=   1e+06
Depth= 1580824 States=  6.2e+07 Transitions= 2.13e+08 Memory=   723.031
      t=       61 R=   1e+06
Depth= 1580824 States=  6.3e+07 Transitions= 2.17e+08 Memory=   723.031
      t=     62.2 R=   1e+06
Depth= 1580824 States=  6.4e+07 Transitions= 2.21e+08 Memory=   723.031
      t=     63.5 R=   1e+06
Depth= 1580824 States=  6.5e+07 Transitions= 2.26e+08 Memory=   723.031
      t=     64.7 R=   1e+06
Depth= 1580824 States=  6.6e+07 Transitions=  2.3e+08 Memory=   723.031
      t=     66.1 R=   1e+06
Depth= 1580824 States=  6.7e+07 Transitions= 2.35e+08 Memory=   723.031
      t=     67.5 R=   1e+06
(Spin Version 6.5.1 -- 20 December 2019)
      + Partial Order Reduction

Bit statespace search for:
      never claim          + (fairness0)
      assertion violations  + (if within scope of claim)
      acceptance   cycles   + (fairness disabled)
      invalid end states    - (disabled by never claim)

State-vector 104 byte, depth reached 1580824, errors: 0
 65370006 states, stored (6.75203e+07 visited)
1.7125004e+08 states, matched
2.3877029e+08 transitions (= visited+matched)
        7 atomic steps

hash factor: 1.98781 (best if > 100.)
bits set per state: 3 (-k3)

Stats on memory usage (in Megabytes):
 7730.370  equivalent memory usage for states (stored*(State-vector +
overhead))
   16.000  memory used for hash array (-w27)
   76.294  memory used for bit stack
  534.058  memory used for DFS stack (-m10000000)
   96.641  other (proc and chan stacks)
  723.031  total actual memory usage

pan: elapsed time 68.9 seconds
No errors found -- did you verify all claims?
```