Санкт-Петербургский Политехнический университет
Факультет Технической кибернетики

Ю.Г.Карпов

Задачи по курсу

“Верификация параллельных и распределенных программных систем”

Санкт-Петербург, 2010

Предисловие

Сборник будет включать основные типовые задачи по курсу "*Верификация параллельных и распределенных программных систем*", который читается на факультете технической кибернетики Санкт-Петербургского Политехнического университета. Задачи разбиты по разделам. Рекомендуется решать задачи каждого раздела сразу после прослушивания соответствующей лекции.

Задачи, требующие размышления или некоторого времени для их решения, помечены звездочкой (*). Двумя звездочками помечены задачи, которые представляют собой программный проект.

Сборник будет пополняться, в частности, задачами из книги Ю.Г.Карпова "Model checking"

По курсу проводится письменный экзамен, который состоит в решении задач. Экзаменационные варианты составляются из задач данного сборника, возможно, с измененными данными или формулами. Поэтому умение самостоятельно решать задачи сборника гарантирует студенту положительную оценку на экзамене.

Бинарные решающие диаграммы для представления двоичных функций

1.8. BDD

1.8.1. Постройте BDD булевой функции $f=\neg((z\oplus x)\Rightarrow t\vee y) \Rightarrow xy$ для упорядочения $x<y<z<t$ по таблице истинности и по формуле, используя синтаксическое дерево этой формулы.

1.8.2. Постройте BDD булевой функции $x\downarrow(y\Rightarrow z)\oplus(t\equiv\neg x)$ для двух различных упорядочений ее аргументов: $x<y<z<t$ и $t<z<y<x$.

1.8.3. Постройте BDD функции, являющейся дизъюнкцией двух БФ, заданных в форме BDD с одной и той же упорядоченностью аргументов этих функций.

1.8.4. Пусть наборы значений двоичных аргументов $x_1, \ldots, x_5$ задаются соответствующими целыми, например, набор 00010 задается номером 2, а набор 11011 задается номером 27.

а) Постройте BDD булевой функции от 5 аргументов, которая равна 1 на следующем множестве двоичных наборов: {9, 11, 17, 19, 21, 23, 25, 27}, а на остальных наборах равна 0.

б) Постройте BDD булевой функции от 5 аргументов, которая равна 1 на следующем множестве двоичных наборов: {0, 1, 4, 5, 8, 9, 12, 13, 17, 19, 21, 23}, а на остальных наборах равна 0.

1.8.5. Для двух БФ f_1 и f_2, заданных формулами, постройте их BDD и BDD функции $f_1\Rightarrow f_2$.

1.8.6. Постройте списочное представление BDD для функции $(q\Rightarrow p)\&r \Rightarrow(p\Rightarrow r)\&q$.

1.8.7. БФ, заданную формулой, представьте в виде BDD и в базисе {***ite***, 0, 1}

1.8.8. Сколько вершин содержит бинарная решающая диаграмма функции $(x\vee y\Rightarrow z)\oplus t$ при порядке переменных $x<y<z<t$?

1.8.9. Постройте алгоритм вычисления значения функции $(r\Rightarrow p)\&s \Rightarrow\neg(p\Rightarrow r\&q)$ по BDD этой функции.

1.8.10. Постройте BDD двоичной функции от переменных $x_1, x_2, \ldots, x_n$, которая равна единице только на таких наборах значений этих переменных, в которых ровно одна переменная не 0, а остальные равны 0 (для порядка переменных $x_1>x_2> \ldots>x_n$).

1.8.11. Для двоичных функций $f_1 = p \Rightarrow q\oplus r$ и $f_2 = p\vee q\Leftrightarrow r\wedge p$ постройте совместную BDD этих функций с двумя выходными вершинами.

1.8.12. Для двух заданных подмножеств конечного множества постройте

а) их представление в BDD;
б) представление в BDD дополнения первого подмножества;
в) представление в BDD пересечения и объединения этих двух множеств с помощью операций над соответствующими BDD.

1.8.13**. Используя одну из библиотек функций для манипуляций с BDD, постройте программу для решения следующей задачи.
На выставке три девушки рекламируют три новых автомобиля разных марок, разного цвета и разной стоимости.
Известно, что:
1) самый дорогой из всех - черный автомобиль;
2) Хонда, которую рекламирует Вера, дороже красной машины;
3) Люба рекламирует синий автомобиль, который дороже Ниссана.
Кто рекламирует Тойоту, и машину какого цвета рекламирует Надя?

1.8.14**. Используя одну из библиотек функций для манипуляций с BDD, постройте программу для решения задачи Эйнштейна.

1.8.15**. Используя одну из библиотек функций для манипуляций с BDD, решите проблему расстановки ферзей.

Верификация императивных программ индуктивным методом Флойда-Хоара

2.10.

2.10.1. График какой функции получится, если график функции $y = 2x^2 - 5x + 7$ сдвинуть на две единицы влево вдоль оси x и на одну единицу вверх по оси y?

2.10.2. Если график функции $y = x^3 + 2x + 4$ сдвинуть на одну единицу вправо вдоль оси x и на две единицы вниз по оси y, то график какой функции получится?

2.10.3. Докажите корректность двумя способами: построением сильнейшего постусловия *sp* и построением слабейшего предусловия *wp*:

$\{X=x \wedge Y=y\}$ `X:=X+Y; Y:=Y-X; X:=X+Y` $\{Y=x \wedge X=y\}$

$\{X=R+Y*Q\}$ `R:= R+Y; Q:= Q-1` $\{X= R+Y*Q\}$

2.10.4. Для приведенных ниже фрагментов программ S проверьте их корректность по отношению к предусловию I и постусловию R двумя способами: построением сильнейшего постусловия *sp*(S, I) и построением слабейшего предусловия *wp*(S, R):

Предусловие I	Фрагмент программы S	Постусловие R

$\{z+a*b <c\}$
$\{z = b – a\}$
$\{i>0 \ \& \ s = \Sigma_{0\le j<i} b[i] \}$
	`z:=2z+a; a:= 3a-b;`
	`a:=2a+1; b:= b-1;`
	`s:=s+b[i]; i:= i+1;`
		$\{z+a <0 \}$
		$\{z + a – b = 3 \}$
		$\{i>0 \ \& \ s = \Sigma_{0\le j<i} b[i] \}$

2.10.5. Для приведенных ниже фрагментов программ S проверьте их корректность по отношению к предусловию I и постусловию R двумя способами: построением сильнейшего постусловия *sp*(S, I) и построением слабейшего предусловия *wp*(S, R):

Предусловие I	Фрагмент программы S	Постусловие R

$\{z + a = c\}$
$\{True\}$
$\{True\}$
	`z:= 2*z; a := a+1;`
	`i :=0; s :=1;`
	`x :=2; y := 3+x;`
		$\{z+a > c+2\}$
		$\{ s>2*i \}$
		$\{x+y \ge 0 \}$

2.10.6. Для приведенных ниже фрагментов программ S найдите такие выражения E, при которых приведенные ниже фрагменты корректны по отношению к предусловию I и постусловию R:

Предусловие I	Фрагмент программы S	Постусловие R

$\{z + 2a = c\}$
$\{True\}$
$\{i>0 \ \& \ s = \Sigma_{0\le j<i} a[i] \}$
	`z:= E; a := a+1;`
	`i :=0; s :=E;`
	`i :=1+1; s := E;`
		$\{ z + 2a = c \}$
		$\{ s = \Sigma_{0\le j<i} b[i] \}$
		$\{i>0 \ \& \ s = \Sigma_{0\le j<i} a[i] \}$

2.10.7. Докажите корректность следующей программы индуктивным методом Флойда:

{X = x & Y = y }

```
X :=X+Y;
Y:= Y-X;
X := X-Y;
```

{X = x & Y = y }

2.10.8. Докажите корректность следующей программы индуктивным методом Флойда:

{ X = R+Y*Q }

```
R :=R - Y;
Q:= Q+1;
X := X-Y;
```

{ X = R+Y*Q }

2.10.9. Постройте программу нахождения максимума из трех чисел и докажите ее корректность двумя способами: построением сильнейшего постусловия sp и построением слабейшего предусловия wp.

2.10.10. Проведите доказательство корректности программы вычисления НОД с помощью построения слабейшего предусловия wp.

2.10.11. Проведите доказательство корректности программы целого деления с помощью построения слабейшего предусловия wp.

2.10.12*. Постройте программу нахождения максимального элемента массива чисел и докажите ее корректность двумя способами: построением сильнейшего постусловия sp и построением слабейшего предусловия wp.

2.10.13*. Постройте программу нахождения номера максимального элемента массива чисел и докажите ее корректность двумя способами: построением сильнейшего постусловия sp и построением слабейшего предусловия wp.

2.10.14*. Постройте программу суммирования элементов массива и докажите ее корректность.

2.10.15*. Постройте программу определения минимального элемента массива и докажите ее корректность двумя способами: построением сильнейшего постусловия sp и построением слабейшего предусловия wp.

2.10.16*. Рассмотрите программу вычисления факториала:

```
begin
  f = 1;
  z = 0;
```

```
while z != x
      do
        f = f*z;
        z = z+1;

      od
end
```

с предусловием {x>=0} и постусловием {f=x!}. Проверьте корректность этой программы.

2.10.17*. Докажите корректность следующей программы возведения *x* в степень *y* двумя способами: построением сильнейшего постусловия sp и построением слабейшего предусловия wp:

```
function exponential (x,y);
  int x,y;
  begin
     int i, z;
       z=1; i=1;
       while (i≠y) do
          z*=x; i=i+1;
       return z;
  end
```

Темпоральные логики

4.1. Логика LTL

4.1.1*. Выразите в LTL следующие свойства:

а) *Если произойдет событие p, то в будущем, после этого, событие q не произойдет никогда*;

б) *Атомарные предикаты p и q выполняются попеременно: (в одном состоянии p и q не встречаются, если выполнится p, то после этого p не будет выполняться, пока не выполнится q, и наоборот)*;

в) *Если произойдет событие p, то когда-нибудь в будущем выполнится событие q, а сразу за этим произойдет событие r*;

г) *Если случится событие p, то в будущем обязательно встретится событие q, а между ними не случится события r*;

д) *В будущем событие p может случиться не более одного раза*;

е) *В будущем событие p может случиться точно один раз*;

ж) *В будущем событие p может случиться точно два раза*.

4.1.2. Выразите в LTL следующее свойство: "*Я сейчас живу, но когда-нибудь я умру*".

4.1.3. Выразите в LTL следующее свойство вычислений: "*Переменная* `enabled` *на вычислении системы будет истинной бесконечное число раз*".

4.1.4. Выразите в LTL следующее свойство протокола: "*Посылка запроса* `req` *всегда в конце концов приведет к получению разрешения* `ack`".

4.1.5. Постройте структуру Крипке, удовлетворяющую формуле LTL $p \Rightarrow \mathbf{XG}\neg q$.

4.1.6. Постройте структуру Крипке, удовлетворяющую формуле LTL $p \& \mathbf{FG} r$.

4.1.7. Постройте структуру Крипке, удовлетворяющую формуле LTL $\mathbf{F}(p \oplus q) \Rightarrow \mathbf{GX}q$.

4.1.8. Постройте структуру Крипке, удовлетворяющую формуле LTL $\mathbf{G}(p \Rightarrow q)$.

4.2. Логика CTL и верификация

4.2.1. Выразите в CTL следующее свойство программы: *"Из любого состояния программа может быть переведена в начальное состояние"*.

4.2.2. Выразите в CTL следующее свойство протокола передачи данных: *"Существуют такие траектории вычисления, что следующий пакет данных посылается в канал до получения подтверждения о доставке предыдущего пакета"*.

4.2.3. Выразите в CTL свойство свободы от блокировок в параллельной программе: *“Для каждого достижимого состояния существует возможность продолжения функционирования”*.

4.2.4. Выразите в CTL свойство управляющей программы: “*После того, как сигнал p стал активным, он не будет активным до тех пор, пока не станет активным r*”.

4.2.5. Выразите в CTL свойство частичной корректности программы: “*Если при запуске программы (атомарный предикат* `at_Start`*), программные переменные удовлетворяют утверждению φ, то, по какому бы пути программа ни пришла в заключительное состояние (*`at_Finish`*), программные переменные будут удовлетворять утверждению ψ*”.

4.2.6. Выразите в CTL свойство локального инварианта программы: “Если программа придет в состояние s (`at_s`), то утверждение φ, связанное с этим состоянием, станет истинным”.

4.2.7. Выразите в CTL свойство взаимного исключения параллельных процессов: “*В любом достижимом состоянии два параллельных процесса не могут находиться одновременно в своих критических интервалах (процесс P1 в состоянии* $crint_1$*, а процесс P2 в состоянии* $crint_2$*)*”.

4.2.8. Выразите все комбинации *<квантор пути, темпоральный оператор>*

а) через базис { **EX**, **AU**, **EU**},

б) через базис {**EX**, **EU**, **EG**}.

4.2.9. Следующие формулы CTL представьте с помощью операторов базиса {**EX**, **EU**, **EG**}:

а) **A**p**U(EX**p);

б) **AG**(p ∨ **EX**q);

в) **AX**(**EF**p ∨ **A**(**Ex**q)**U**p).

4.2.10. Следующие формулы CTL представьте через операторы базисов {**EX**, **AF**, **EU**}, {**AX**, **AU**, **EU**}:

а) **AG**(p∨**EX**q);

б) **AX**(**EF**p∨**A**((**EX**q)**U**p));

в) **A**p**UEX**q.

4.2.11. Для следующих формул постройте синтаксические деревья, определяющие их структуру в соответствии с грамматикой формул CTL:

а) **A**p**UEX**(p⇒**AG**q);

б) **AG**(p∨**EX**q).

4.2.12. Пусть p означает "*Я люблю Машу*", а q – "*Я люблю Дашу*". Каким высказываниям соответствуют следующие формулы CTL:

а) **AF EG**p;
б) **EF AG**p;
в) **A**(p**U**q);
г) **E**[(**EX**p)**U**(**AG**q)].

4.2.13. Постройте процедуры проверки модели для формул **AX**β, **EF**β, **A**[β**U**γ], а также графическое представление соответствующих процедур.

4.2.14. Пусть M=(S, S_0, R, AP, L) – такая структура Крипке:
S={s_0, s_1, s_2, s_3}, S_0={s_1, s_3}, R={(s_2, s_3), (s_2, s_0), (s_0,s_1), (s_1,s_2), (s_3,s_0), (s_2,s_2)}, AP={a, b}, L(s_0) ={a}, L(s_1) ={a, b} L(s_2) ={b}, L(s_3) =∅.

a) В каких состояниях M выполняются формулы CTL: **EXAX**b, **A**(**EX**b**U**a)?
b) Выполняется ли для M формула **EGA**(b**U**a)?

4.2.15. Пусть M=(S, S_0, R, AP, L) – такая структура Крипке:
S={s_0, s_1, s_2, s_3}, S_0={s_0}, R={(s_0, s_1), (s_2, s_0), (s_0,s_2), (s_2,s_3), (s_3,s_0)}.
Известно, что формула **EF**φ выполняется в состоянии s_3. Будет ли **EF**φ выполняться в s_0? в s_1? в s_2? в M?

4.2.16. Пусть M=(S, S_0, R, AP, L) – такая структура Крипке:
S = {p, q, r, s}; S_0 = {p,r}; R = {(p,s), (q, r), (s, s), (q, s), (s,q), (r,p), (p,r)};
AP={a, b, c}, L: S→2^{AP}; L(p) ={a, b}; L(q) = {c}; L(r) = {}; L(s) = {b};
а) постройте структуру Крипке M;
б) проверьте выполнимость:
M |= **AFEX**¬a,
M |= **EXAF**(a⇒b),
M |= **AG**(c⇒**EF**b).

4.2.17. На основе формального определения семантики операторов **F** и **G** докажите справедливость тождеств:

а) **F**(p∨q) ≡ **F**p ∨ **F**q;
б) **G**(p∧q) ≡ **G**p ∧ **G**q.

4.2.18. На основе формального определения семантики операторов **U**, **X**, **F** и **G** докажите правильность рекурсивного определения этих операторов:

а) φ**U**ψ ≡ ψ ∨ **X**(φ**U**ψ);
б) **F** ψ ≡ ψ ∨ **XF** ψ;
в) **G** ψ ≡ ψ ∧ **XG** ψ.