Упражнения по аксиоматическим теориям и задачам выполнимости формул

“Практические” аксиоматические теории

В задачах используются наиболее известные в программировании аксиоматические теории *с равенством*. Эти теории приведены в данном разделе.

Линейная целочисленная арифметика

Предикатные символы: $<^{(2)}$, $\leq^{(2)}$, $>^{(2)}$, $\geq^{(2)}$, $\neq^{(2)}$.
Функциональные символы: $+^{(2)}$, $-^{(2)}$, $\alpha\cdot^{(1)}$ $(\alpha \in \mathbb{Z})$.
Константы: $\mathbb{Z}$.
Интерпретация:

- предметная область: $\mathbb{Z}$;
- оценка предикатных и функциональных символов: стандартная арифметическая.

Арифметика битовых векторов

Предикатные символы: $\neq^{(2)}$, $[\alpha]^{(1)}$ $(\alpha \in \mathbb{N}_0)$.
Функциональные символы: $\wedge^{(2)}$, $|^{(2)}$, $\oplus^{(2)}$, $+^{(2)}$, $-^{(2)}$, $\cdot^{(2)}$, $/^{(2)}$, $\%^{(2)}$, $\mathbf{rev}^{(1)}$, $\gg^{(2)}$, $\ll^{(2)}$, $\circ^{(2)}$, $[\alpha : \beta]^{(1)}$ $(\alpha, \beta \in \mathbb{N}_0)$.
Константы: v_i $(v \in \mathbb{Z},\ i \in \mathbb{N})$.
Интерпретация:

- предметная область: множество всех конечных слов в алфавите $\{0, 1\}$ (то есть конечных битовых векторов);
- оценка констант: константа v_i интерпретируется как битовый вектор длины i, содержащий двоичную запись числа v_i;
  пример: $\overline{6_5} = 00110$;
- оценка функциональных символов:
  - $\wedge^{(2)}$, $|^{(2)}$, $\oplus^{(2)}$: побитовые конъюнкция, дизъюнкция и сумма по модулю 2 векторов равной длины;
    пример: $6_5 \oplus 3_5 = 5_5$;
  - $+^{(2)}$, $-^{(2)}$, $\cdot^{(2)}$, $/^{(2)}$, $\%^{(2)}$: операции целочисленной арифметики *с переполнением* для векторов равной длины;
    пример: $6_5 - 2_5 = 4_5$; $6_5 + 30_5 = 4_5$;
  - $\mathbf{rev}^{(1)}$: операция получения обратного кода числа;
    пример: $\mathbf{rev}(6_5) = 25_5$;
  - $\gg^{(2)}$, $\ll^{(2)}$: сдвиги вправо и влево с заполнением нолями;
    пример: $6_5 \gg 2_2 = 1_5$;
  - $\circ^{(2)}$: операция конкатенации векторов;
    пример: $6_5 \circ 2_2 = 26_7$;
  - $[\alpha : \beta]^{(1)}$: операция выделения подвектора;
    пример: $26_7[2 : 5] = 6_4$;
- оценка предикатных символов:
  - $\neq^{(2)}$: неравенство битовых векторов;
  - $[\alpha]^{(1)}$: выделение бита по смещению;
    пример: $6_5[2] = \mathbf{true}$; *пример*: $6_5[3] = \mathbf{false}$.

Теория массивов

Функциональные символы: $\cdot\{\cdot \leftarrow \cdot\}^{(3)}$, $\cdot[\cdot]^{(2)}$.
Аксиомы:

1. $\forall \mathtt{A}\ \forall \mathtt{i}\ \forall \mathtt{x}\ \mathtt{A}\{\mathtt{i} \leftarrow \mathtt{x}\}[\mathtt{i}] = \mathtt{x}$;
2. $\forall \mathtt{A}\ \forall \mathtt{i}\ \forall \mathtt{j}\ \forall \mathtt{x}\ (\neg \mathtt{i} = \mathtt{j} \rightarrow \mathtt{A}\{\mathtt{i} \leftarrow \mathtt{x}\}[\mathtt{j}] = \mathtt{A}[\mathtt{j}])$;
3. $\forall \mathtt{A}\ \forall \mathtt{B}\ (\forall \mathtt{i}\ \mathtt{A}[\mathtt{i}] = \mathtt{B}[\mathtt{i}] \rightarrow \mathtt{A} = \mathtt{B})$.

Примечание. В зависимости от своего назначения предметы модели для теории массивов называются *массивами*, *индексами* b *элементами*. Массив записывается в первые аргументы оценок функциональных символов теории. Индекс означает смещение в массиве и записывается во вторые аргументы. Элемент массива записывается в третий аргумент оценки символа $\cdot\{\cdot \leftarrow \cdot\}^{(3)}$. Значение оценки символа $\cdot\{\cdot \leftarrow \cdot\}^{(3)}$ — массив, получающийся из заданного массива записью заданного элемента по заданному смещению. Значение оценки символа $\cdot[\cdot]^{(2)}$ — элемент заданного массива по заданному смещению.

Теория равенства с неинтерпретируемыми функциями

Функциональные символы: любые
Аксиомы: отсутствуют

Упражнения

Упражнение 1

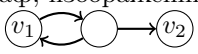
Для предложенной задачи выбрать подходящую *бескванторную* теорию [или логику высказываний] и

- описать алгоритм построения формулы φ в выбранном синтаксисе, такой что:
  - формула φ выполнима тогда и только тогда, когда задача имеет решение;
  - для любой модели $\mathcal{I}$ выбранной теории и любого набора предметов $\widetilde{d}^n$ верно: если $\mathcal{I} \models \varphi[\widetilde{d}^n]$, то по предметам $\widetilde{d}^n$ можно *легко* восстановить ответ к задаче;
  - [для любой интерпретации $\mathcal{I}$ логики высказываний верно: если $\mathcal{I}(\varphi) = \mathbf{true}$, то по значениям пропозициональных переменных в $\mathcal{I}$ можно *легко* восстановить ответ к задаче;]
- записать формулу, получаемую в результате работы алгоритма для предложенных входных данных задачи.

1. Старый пират после налёта на корабль забрёл в трактир выпить несколько бутылок рома. С собой пират принёс *достаточно много* награбленных монет достоинствами $n_1, \ldots, n_k$. Когда весь ром был выпит и хозяин потребовал заплатить (на сумму m), оказалось, что у хозяина таверны нет никакой сдачи, а пират настолько жаден, что не хочет платить больше положенного. Задача: сказать пирату, сколько каких монет он может отдать хозяину таверны за ром.
   Входные данные: количество номиналов k; номиналы $n_1, \ldots, n_k$; стоимость выпитого m.
   Конкретные входные данные: $k = 3$, $n_1 = 3$, $n_2 = 7$, $n_3 = 19$, $m = 103$.
2. В точно такую же таверну зашёл менее удачливый пират, имеющий *немного* монет каждого достоинства: $p_1, \ldots, p_k$. Задача: точно так же помочь расплатиться пирату с хозяином таверны.
   Дополнительные входные данные: $p_1, \ldots, p_k$ — количество монет номинала $n_1, \ldots, n_k$.
   Дополнительные конкретные входные данные: $p_1 = 13$, $p_2 = 10$, $p_3 = 11$.
3. **Принцип Дирихле**. Имеется n клеток и k кроликов. Задача: придумать, как рассадить кроликов по клеткам так, чтобы в каждой клетке сидело не более одного кролика.
   Входные данные: количество клеток n, количество кроликов k.
   Конкретные входные данные: $n = 6$, $k = 4$.
4. **Верификация**. Задача: для приведённой ниже функции **f** в синтаксисе языка Си найти значения аргументов функции, такие что для заданных значений k, n_i, n_j, m_i, m_j типа uint_32 **не** выполнено следующее требование: значение, возвращаемое функцией, кратно k.

```
uint_32 f(uint_32 i, uint_32 j) {
  if(i >= n_i || j >= n_j) return k;
  i *= k * m_i;
  j *= k * m_j;
  return i + j;
};
```

   Примечание: "uint_32" — это тип целых чисел в диапазоне $[0, 2^{32})$
   Входные данные: числа k, n_i, n_j, m_i, m_j.
   Конкретные входные данные: $k = 3$, $n_i = 5000$, $n_j = 10000$, $m_i = 1$, $m_j = 2$.
5. Для того же фрагмента кода найти значения аргументов функции **f**, такие что в процессе работы функции происходит ходя бы одно переполнение.
6. **Достижимость в графе**. В ориентированном графе G выделены вершины v_1, v_2. Задача: проверить достижимость v_2 из v_1 по маршруту, длина которого ограничена заданным числом k.
   Входные данные: множество вершин V и дуг E графа G; вершины v_1, v_2; число k.
   Конкретные входные данные: $k = 2$, граф, изображённый ниже.

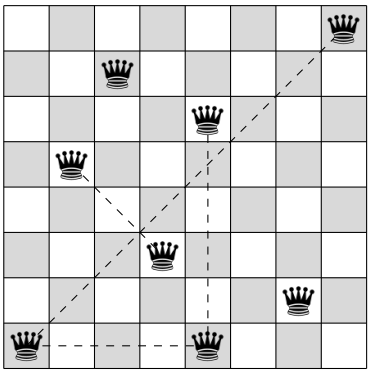


7. Задача: то же, что и в предыдущем пункте, но с длиной маршрута в точности равной k.
8. Задача: то же, что и в предыдущем пункте, но без ограничения на длину маршрута.
   Разница во входных данных: отсутствует число k.

9. **Судоку**. Дано поле, замощённое девятью квадратами (9×9), квадраты разбиты на группы (3×3). Квадраты частично заполнены цифрами от 1 до 9. Задача: дозаполнить квадраты поля цифрами от 1 до 9 так, чтобы цифры в каждой группе, каждой горизонтали и каждой диагонали не повторялись.
Входные данные: цифры и их расположение на поле.
Конкретные входные данные:

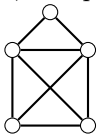
5	3			7				
6			1	9	5			
	9	8					6	
8				6				3
4			8		3			1
7				2				6
	6					2	8	
			4	1	9			5
				8			7	9

10. **Восемь ферзей**. Дано шахматное поле. Расположить на этом поле восемь ферзей так, чтобы ни один из них не находился под ударом других.
Пример: ферзей нельзя располагать так, как показано ниже. Все пары ферзей, находящихся под взаимным ударом, соединены пунктиром.

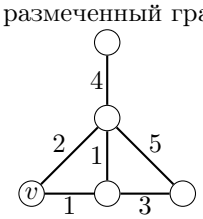


11. N **ферзей**. То же, что и в предыдущем пункте, но для поля размера $N \times N$ и N ферзей.
Входные данные: число N.

12. **Поиск клики в графе**. Задача: найти клику заданного размера k в заданном графе G.
Входные данные: множество вершин V и рёбер E графа G; число k.
Конкретные входные данные: $k = 3$; граф G, изображённый ниже.



13. Путешественник находится в городе v и хочет объехать все города из списка на машине, заправленной N литрами бензина, зная, между какими городами есть дорога и сколько литров бензина тратится на каждый переезд. Задача: составить маршрут путешествия.
Входные данные: число N; множества вершин V и рёбер E графа, описывающего города и дороги между ними; город v, в котором находится путешественник ($v \in V$); разметка **cost** рёбер графа числами, описывающими затраты бензина в литрах на переезд.
Конкретные входные данные: $N = 18$; размеченный граф G, изображённый ниже.



Упражнение 2

Очередь может быть представлена как массив A, в котором выделены смещения hd головы очереди и tl хвоста очереди. Расширим сигнатуру теории массивов следующими функциональными и предикатными символами:

- функциональный символ $\langle\cdot,\cdot,\cdot\rangle^{(3)}$: $\langle A, hd, tl\rangle$ — очередь описанного выше вида;
- функциональный символ $\mathbf{S}^{(1)}$: $\mathbf{S}(\mathtt{i})$ — смещение, следующее за смещением $\mathtt{i}$;
- функциональный символ $\mathbf{head}^{(1)}$: $\mathbf{head}(\mathtt{q})$ — элемент в голове очереди $\mathtt{q}$;
- функциональный символ $\mathbf{append}^{(2)}$: $\mathbf{append}(\mathtt{q},\mathtt{e})$ — очередь, получаемая из очереди $\mathtt{q}$ добавлением элемента $\mathtt{e}$.
- предикатный символ $\mathrm{Empty}^{(1)}$: $\mathrm{Empty}(\mathtt{q}) = \mathbf{true}$ тогда и только тогда, когда очередь пуста;

Задача: предоставить набор аксиом, адекватно описывающий содержательный смысл добавленных символов, и проверить общезначимость следующих формул в полученной теории:

$$\begin{aligned}
\forall\mathtt{q}\,\forall\mathtt{e} \quad & \neg\mathrm{Empty}(\mathbf{append}(\mathtt{q},\mathtt{e})) \\
\forall\mathtt{q}\,\forall\mathtt{e} \quad & (\neg\mathrm{Empty}(\mathtt{q}) \to \mathbf{head}(\mathtt{q}) = \mathbf{head}(\mathbf{append}(\mathtt{q},\mathtt{e}))) \\
\forall\mathtt{q} \quad & (\neg\mathrm{Empty}(\mathtt{q}) \to \exists\mathtt{p}\,\exists\mathtt{e}\;\mathtt{q} = \mathbf{append}(\mathtt{p},\mathtt{e})) \\
\forall\mathtt{q}\,\exists\mathtt{p}\,\exists\mathtt{e} \quad & \mathtt{q} = \mathbf{append}(\mathtt{p},\mathtt{e})
\end{aligned}$$