Часть I

Лекция 1 (14.02.15) “Основы теории булевых функций (функций алгебры логики)”

Преподаватель: Воронцов Александр Алексеевич

1 Теория множеств

1. M - множество
2. $x \in M$ - элемент х принадлежит множеству М
3. $N \subseteq M$ - множество N является подмножеством М
4. $M = \{a_1, a_2, ..., a_n\}$ - конечное множество
5. $|M| = n$ - мощность множества
6. $A \cup B = \{x \in U / x \in A$ или $x \in B\}$
7. $A \cap B = \{x \in U / x \in A$ и $x \in B\}$
8. $\overline{A} = \{x \in U / x \notin A\}$
9. $P(U)$ - множество всех подмножеств множества U
   (a) $U = \{a_1, a_2\}$
   (b) $P(U) = \{\varnothing, \{a_1\}, \{a_2\}, U\}$
      i. $\{a_1\} \in P(U)$
      ii. $\{a_2\} \subseteq U$
   (c) $|P(U)| = 2^n$

1.1 Свойства операций

1. Коммутативность
   (a) $A \cup B = B \cup A$
   (b) $A \cap B = B \cap A$
2. Ассоциативность
   (a) $(A \cup B) \cup C = A \cup (B \cup C)$
   (b) $(A \cap B) \cap C = A \cap (B \cap C)$

3. Дистрибутивность
   - (a) $(A \cup B) \cap C = (A \cap C) \cup (B \cap C)$
   - (b) $(A \cap B) \cup C = (A \cup C) \cap (B \cup C)$
4. Свойство констант (существование универсальных границ)
   - (a) $A \cap \oslash = \oslash$
   - (b) $A \cup \oslash = A$
   - (c) $A \cap U = A$
   - (d) $A \cup U = U$
5. Закон поглощения
   - (a) $A \cup A = A$
   - (b) $A \cap A = A$
6. Свойство инвалентивности
   - (a) $\overline{(\overline{A})} = A$
7. Свойство дополнимости
   - (a) $A \cup \overline{A} = U$
   - (b) $A \cap \overline{A} = \oslash$
8. Законы двойственности (законы Де-Моргана)
   - (a) $\overline{(A \cup B)} = \overline{A} \cap \overline{B}$
   - (b) $\overline{(A \cap B)} = \overline{A} \cup \overline{B}$

2 Булева алгебра

2.1 Определение булевой алгебры

Булева алгебра — множество M, в котором выделено два элемента $\oslash, I$ и введены две бинарных $x \Delta y, x \nabla y$ операции и одна унарная x^*. Это множество называется булевой алгеброй, если введенные операции обладают разобранными выше свойствами.

1. $P(U)$ является булевой алгеброй, если: $\cup \leftrightarrow \nabla$, $\cap \leftrightarrow \Delta$, $\bar{\ } \leftrightarrow *$, $U \leftrightarrow I$, $\oslash \leftrightarrow \oslash$

2.2 Булевы вектора

1. $U = \{a_1, ..., a_n\}$
2. $S \in P(U)$, $S \subseteq U$
3. $S \leftrightarrow \vec{\alpha_s} = \{\alpha_1, ..., \alpha_n\}$
4. $\alpha_i = \begin{cases} 0, \text{ если } \alpha_i \notin S \\ 1, \text{ если } \alpha_i \in S \end{cases}$
5. B_n - множество всех векторов размерности n

2.3 Логические операции

α	β	$\alpha \cup \beta$	$\alpha \cap \beta$	$\overline{\alpha}$
0	0	0	0	1
0	1	1	0	1
1	0	1	0	0
1	1	1	1	0

1. $\alpha = (\alpha_1, ..., \alpha_n)$, $\beta = (\beta_1, ..., \beta_n)$
   (a) $\alpha \cup \beta = (\alpha_1 \cup \beta_1, ..., \alpha_n \cup \beta_n)$
   (b) $\alpha \cap \beta = (\alpha_1 \cap \beta_1, ..., \alpha_n \cap \beta_n)$
   (c) $\overline{\alpha} = (\overline{\alpha_1}, ..., \overline{\alpha_n})$

- Операции объединения множеств соответствует операция логического сложения их характеристических векторов.
- Операции пересечения подмножеств соответствует операция логического умножения их характеристических векторов.
- Операции дополнения к множеству соответствует операция отрицания к его характеристическому вектору.

2.4 Булева алгебра в высказываниях об элементах конечного множества U

$U = \{a_1, a_2, ..., a_n\}$

Под высказываниями мы будем понимать любое утверждение об элементах множества U, которые для каждого элемента множества U либо истинно, либо ложно.

1. $U = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$
2. A = “Число x делится на 2”, $S_A = \{0, 2, 4, 6, 8\}$
3. B = “Число x четно”, $S_B = \{0, 2, 4, 6, 8\}$

4. $S_A = S_B \implies$ A и B эквивалентные высказывания

Все эквивалентные выскаывания объединяются между собой в один класс эквивалентности.

2.5 Изоморфизм булевых алгебр

Две булевых алгебры $(M, \overline{M})$ называются изоморфными, если между их элементами можно установить такое однозначное соответствие, при котором операции одной булевой алгебры переходят в операции другой.

Часть II

Лекция 2 (21.02.15) "Определение и способы задания Булевых функций"

3 Булевы функции

3.1 Определение

1. $f(x_1, ..., x_n) = \begin{cases} 0 \\ 1 \end{cases}$

2. $x_i = \begin{cases} 0 \\ 1 \end{cases}$

3. $B^n \overset{t}{\curvearrowright} B^1$

3.2 Способы задания

3.2.1 Табличный способ задания

1. $N = 2^n$

2.

x_1	x_2	...	x_n	f
0	0	...	0	γ_1
0	0	...	1	γ_2
...	...	...	...	...
1	1	...	1	γ_n

3.2.2 Векторный способ задания

1. $f = (\gamma_1, \gamma_2, ..., \gamma_n)$

3.2.3 Геометрический способ задания

1. **Единичный вектор** для функции — вектор, где данная функция $= 1$.

2. **Носитель функции** — множество всех векторов данной функции.

   (a) $N_f = \{(0,0,1),(0,1,0),...,(1,1,1)\}$

   (b) $N_f = \{1,2,...,7\}$

3. Функция $f(x_1,...,x_n)$ зависит от x_i **фиктивно**, если для любых двух наборов значение аргументов отличается только значением переменной x_i, а значения функции совпадают.

   (a) $f(x_1,...,x_n), \forall \alpha_i \Rightarrow$
   $\Rightarrow f(\alpha_1,...,\alpha_{n-1},0,\alpha_{n+1},...,\alpha_n) =$
   $= f(\alpha_1,...,\alpha_{n-1},1,\alpha_{n+1},...,\alpha_n)$

4. Переменная x_i называется **существенной,** если существуют такие два набора значений аргумента, которые отличаются только значением переменной x_i и такие, что значения функции не совпадают.

   (a) $f(\alpha_1,...,\alpha_n) \neq f(\alpha_1,...,\alpha_n) \Longrightarrow \exists \alpha_n$

5. Фиктивные переменные можно исключать или добавлять.

6. Две Булевы функции **равные**, если одну из них можно получить из другой путем добавления или изъятия фиктивных переменных.

   (a)

x_1	x_2	x_3	$f(x_1,x_2,x_3)$
0	0	0	0
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	0

   x_1 – существенная
   x_2 – фиктивная
   x_3 – существенная

3.3 Элементарные Булевы функции

- Основные Булевы функции: $\hat{0}, \hat{1}, \overline{x}, x \wedge y, x \vee y$ (удовлетворяют аксиомам Булевой алгебры)

- Выражаются через основные элементарные функции:

  – $x/y = \overline{x \wedge y} = \overline{x} \vee \overline{y}$
  – $x \downarrow y = \overline{x \vee y} = \overline{x} \wedge \overline{y}$
  – $x \oplus y = (\overline{x} \cdot \overline{y}) \vee (x \cdot \overline{y})$

- $x \sim y = (\overline{x} \cdot \overline{y}) \vee (x \cdot y)$
- $x \to y = \overline{x} \vee y$

- Операцией элементарных суперпозиций называется одно из следующих действий:
  - Переименование аргументов функции, в том числе отождествление.
  - Подстановка функции в какую-либо другую функцию.

Часть III

Лекция 3 (28.02.15) “Нормальные формы”

4 Нормальные формы

4.1 Дизъюнктивная нормальная форма (ДНФ)

1. Обозначение: $x^\alpha = \begin{cases} x & \text{если } \alpha = 1 \\ \overline{x} & \text{если } \alpha = 0 \end{cases}$
2. Элементарная конъюнкция (ЭК) — выражение вида $x_1^{\alpha_1} \& x_2^{\alpha_2} \& ... \& x_k^{\alpha_k}$
3. Правильная элементарная конъюнкция — элементарная конъюнкция, в которой все входящие элементы различны.
4. Правильная элементарная конъюнкция называется **полной** относительно данного набора переменных, если в нее входят все эти переменные.
5. Элементарная дизъюнкция (ЭД) — выражение вида $x_1^{\alpha_1} \vee x_2^{\alpha_2} \vee ... \vee x_k^{\alpha_k}$
6. Дизъюнктивная нормальная форма — дизъюнкция разных правильных элементарных конъюнкций (или одна правильная элементарная конъюнкция)
7. Совершенная дизъюнктивная нормальная форма (СДНФ) — ДНФ, у которой все элементарные конъюнкции полны по отношению к данному набору переменных.

4.2 Конъюнктивная нормальная форма (КНФ)

1. Конъюнктивная нормальная форма (КНФ) — конъюнкция разных правильных элементарных дизъюнкций (или одна правильная элементарная конъюнкция).

2. Совершенная конъюктивная нормальная форма (СКНФ) — КНФ, у которой все элементарные правильные дизъюнкции полны по отношению к данному набору переменных.

4.3 Теорема 1

Любая Булева функция, тождественно не равная нулю, представима и при том единственным образом в виде совершенной дизъюнктивной нормальной формы.

1. $f(x_1, ..., x_n) = \bigvee (x_1^{\alpha_1} \cdot x_2^{\alpha_2} \cdot ... \cdot x_n^{\alpha_n})$

2. $\overline{\alpha} = (\alpha_1, \alpha_2, ..., \alpha_n) \in N_f$

3.

x_1	x_2	x_3	f	СДНФ
0	0	0	1	$\overline{x_1} \cdot \overline{x_2} \cdot \overline{x_3}$
0	0	1	0	
0	1	0	0	
0	1	1	1	$\overline{x_1} \cdot x_2 \cdot x_3$
1	0	0	1	$x_1 \cdot \overline{x_2} \cdot \overline{x_3}$
1	0	1	0	
1	1	0	1	$x_1 \cdot x_2 \cdot \overline{x_3}$
1	1	1	0	

4.3.1 Доказательство

1. Обозначим правую часть формулы за функцию $g(x_1, x_2, ..., x_n)$.

2. $N_f = N_g$

3. $g(\alpha_1, ..., \alpha_n) = \alpha_1^{\alpha_1} \cdot ... \cdot \alpha_n^{\alpha_n} \vee ... = 1 \cdot ... \cdot 1 \vee ... = 1 \Longrightarrow \overline{\alpha} \in N_g$

4. $\forall \overline{\beta} = (\beta_1, ..., \beta_n) \in N_g$

5. $g(\beta_1, ..., \beta_n) = 1$

6. $\beta_1^{\alpha_1} \cdot ... \cdot \beta_n^{\alpha_n} = 1$

4.4 Теорема 2

Любая Булева функция, тождественно не равная единице, представима, и при том единственным образом, в виде совершенной конъюктивной нормальной формы.

1. $f(x_1, ..., x_n) = \bigwedge(x_1^{\overline{\alpha_1}} \vee x_2^{\overline{\alpha_2}} \vee ... \vee x_n^{\overline{\alpha_n}})$

2. $\overline{\alpha} = (\alpha_1, ..., \alpha_n) \notin N_f$

3.

x_1	x_2	x_3	f	СКНФ
0	0	0	1	
0	0	1	0	$x_1 \vee x_2 \vee \overline{x_3}$
0	1	0	0	$x_1 \vee \overline{x_2} \vee x_3$
0	1	1	1	
1	0	0	1	
1	0	1	0	$\overline{x_1} \vee x_2 \vee \overline{x_3}$
1	1	0	1	
1	1	1	0	$\overline{x_1} \vee \overline{x_2} \vee \overline{x_3}$

4. Доказательство этой теоремы следует из принципа двойственности.

4.5 Двойственная функция

Двойственной к функции $f(x_1, ..., x_n)$ называется функция $f^*(x_1, ..., x_n) = \overline{f}(\overline{x_1}, ..., \overline{x_n})$.

Примеры:

1. $(x_1 \wedge x_2)^* = \overline{(\overline{x_1} \wedge \overline{x_2})} = \overline{\overline{x_1}} \vee \overline{\overline{x_2}} = x_1 \vee x_2$

2. $(x_1 \vee x_2)^* = \overline{(\overline{x_1} \vee \overline{x_2})} = x_1 \wedge x_2$

3. $(\overline{x})^* = \overline{\overline{\overline{x}}} = \overline{x}$

4.6 Принцип двойственности

Если функция $F(x_1, ..., x_n)$ является некоторой суперпозицией функции

$$f(f_1(x_1, ..., x_n), ..., f_k(x_1, ..., x_n))$$

, тогда двойственная к ней функция $F^*(x_1, ..., x_n)$ будет являться точно такой же суперпозицией, только с двойственной структурой.

$$F^*(x_1, ..., x_n) = f^*(f_1^*(x_1, ..., x_n), ..., f_k^*(x_1, ..., x_n))$$

4.7 Алгоритм преобразования формулы к виду СДНФ (ДНФ)

1. Заменить в формуле все Булевые функции на выражения через дизъюнкцию, конъюнцию и отрицание.

2. Применив закон двойственности, опустить все отрицания на переменные.

3. Провести упрощения в формуле, используя свойства констант и свойства упрощений.

4. Используя свойство дистрибутивности конъюнкции относительно дизъюнкции, преобразовать формулу так, чтобы конъюнкции выполнялись раньше, чем дизъюнкции.

5. Используя свойства поглощения и свойства констант, упростить выражение.

6. Дополнив каждую элементарную конъюнкций до полной и упростив полученные выражения, получим формулу СДНФ.

4.7.1 Примеры

1. $f(x,y,z) = \overline{(x \oplus \overline{y})} \sim (\overline{z} \to \overline{x}) =$

   (a) $= \overline{(\overline{x} \cdot \overline{y} \vee x \cdot \overline{\overline{y}})} \sim (\overline{\overline{z}} \vee \overline{x}) =$
   $= \overline{(\overline{x} \cdot \overline{y} \vee x \cdot y)} \wedge (z \vee \overline{x}) \vee (\overline{x} \cdot \overline{y} \vee x \cdot y) \wedge \overline{(z \vee \overline{x})} =$
   $= [\{(x \vee y) \cdot (\overline{x} \cdot \overline{y})\} \cdot (z \vee \overline{x})] \vee (\overline{x} \cdot \overline{y} \vee x \cdot y) \cdot \overline{z} \cdot x =$

   (b) $= [\{x \cdot \overline{x} \vee y \cdot \overline{x} \vee x \cdot \overline{y} \vee y \cdot \overline{y}\} \cdot (z \vee \overline{x})] \vee (\overline{x} \cdot \overline{y} \cdot \overline{z} \cdot x \vee x \cdot y \cdot \overline{z} \cdot x) =$
   $= [\{y \cdot \overline{x} \vee x \cdot \overline{y}\} \cdot (z \vee \overline{x})] \vee x \cdot y \cdot \overline{z} =$
   $= y \cdot \overline{x} \cdot z \vee x \cdot \overline{y} \cdot z \vee y \cdot \overline{x} \cdot \overline{z} \vee x \cdot y \cdot \overline{z}$ — СДНФ

Часть IV

Лекция 4 (07.03.15) “Основные понятия минимизации ДНФ”

4.8 Тривиальный алгоритм минимизации

1. Ранг элементарной конъюнкции

   (a) Пусть:

      i. Правильная элементарная конъюнкция имеет вид: $k = x_{i_1}^{\alpha_1} \cdot x_{i_2}^{\alpha_2} \cdot ... \cdot x_{i_n}^{\alpha_n}$

   (b) Тогда:

      i. n — ранг элементарной конъюнкции
      ii. Рангом ДНФ называется сумма рангов элементарных конъюнкций в ДНФ.
      iii. $N = 3^n - 1$ — число разных элементарных конъюнкций, входящих в ДНФ.
      iv. $2^N - 1 = 2^{3^n-1} - 1$ — число различных ДНФ от n переменных.

   (c) Примеры:

      i. $x_1 \to x_2 = \overline{x_1} \vee x_2 (r = 2) = \overline{x_1} \cdot \overline{x_2} \bigvee \overline{x_1} \cdot x_2 \bigvee x_1 \cdot x_2 (r = 6)$

2. Минимальной ДНФ для данной функции называется ДНФ, которая равна данной функции и имеет наименьший ранг среди всех ДНФ, равных этой функции.

3. Тривиальный алгоритм минимизации

   (a) Выписываем все возможные ДНФ в порядке возрастания их рангов.

   (b) Начинаем последовательно сравнивать функцию с выписанными ДНФ.

   (c) Первая ДНФ, которая будет равна функции и будет минимальной.

4.9 Геометрическая интерпретация задачи построения минимальной ДНФ для данной функции

1. Носитель элементарной конъюнкции — интервал ранга n.

   (a) $\begin{Bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \end{Bmatrix} \longleftrightarrow \overline{x_2} \cdot \overline{x_3}$

   (b) $\begin{Bmatrix} 0 & 1 & 1 \\ 0 & 1 & 0 \end{Bmatrix} \longleftrightarrow \overline{x_1} \cdot x_2$

   (c) $\begin{Bmatrix} 0 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 0 & 1 \end{Bmatrix} \longleftrightarrow x_3$

4.9.1 Теорема

1. Носитель дизъюнкции двух функций равняется объединению носителей этих функций. $N_{f \vee g} = N_f \bigcup N_g$

2. $\forall \overline{\alpha} \in N_{f \vee g} \Longrightarrow f(\overline{\alpha}) \vee g(\overline{\alpha}) = 1 \Longrightarrow \begin{Bmatrix} f(\overline{\alpha}) = 1 \\ \text{или} \\ g(\overline{\alpha}) = 1 \end{Bmatrix} \Longleftrightarrow \overline{\alpha} \in N_f \vee N_g$

4.9.2 Определения

Все следующие определения будут соответствовать функции $f(x_1, ..., x_n)$.

1. Интервал называется **допустимым** для функции f, если он целиком принадлежит носителю N.

2. Соответствующая этому интервалу элементарная конъюнкция называется **импликантой** (I).

3. Мы будем рассматривать только допустимые интервалы, которые целиком не содержатся в других допустимых интервалах. Такие допустимые интервалы называются **максимальными**.

4. Импликанта, соответствующая максимальному интервалу, называется **простой** импликантой.

5. Представление носителя функции в виде объединения максимальных интервалов называется **покрытием** носителя интервала.

   (a) $N_{\text{ДНФ}} = I_1 \cup I_2 \cup ... \cup I_n = N_f$

   (b) Каждому покрытию носителя соответствует формула ДНФ, равная данной функции и построенная из дизъюнкции соответствующих простых импликант.

   (c) $f = \text{ДНФ} = k_1 \vee k_2 \vee ... \vee k_n$

6. Покрытие носителя максимальными интервалами называется **неприводимым**, если удаление любого из интервалов приводит к нарушению покрытия.

7. Соответствующая неприводимому покрытию ДНФ называется **тупиковой**.

8. ДНФ, соответствующая покрытию носителя всеми возможными максимальными интервалами, называется **сокращенной** ДНФ $\text{Д}_{\text{сокр}}$.

   (a) Сокращенная ДНФ равна дизъюнкции всех простых импликант и равняется данной функции.

9. Минимальная ДНФ содержится среди тупиковых ДНФ данной функции.

10. Максимальный интервал называется **ядровым**, если он покрывает хотя-бы одну вершину носителя функции, не покрытую никакими другими максимальными интервалами.

    (a) Соответствующая простая импликанта также называется **ядровой**.

    (b) Любой ядровый интервал обязательно содержится в любом неприводимом покрытии, следовательно любая ядровая импликанта входит во все тупиковые ДНФ.

11. Ядро функции — объединение всех ядровых интервалов.

    (a) Ядровой ДНФ или ядром функции будем называть дизъюнкцию всех ядровых импликант.

4.9.3 Алгоритм нахождения минимальной ДНФ

1. Выделяем носитель функции и находим все максимальные интервалы. Выписываем все соответствующие импликанты.
2. Выделяем ядровые импликанты, интервалы и выписываем ядро функции.
3. С помощью комбинации ядра и неядровых интервалов находим все возможные неприводимые покрытия и для каждого выписываем тупиковую ДНФ.
4. Среди тупиковых ДНФ находим минимальную.

4.9.4 Пример

x_1	x_2	x_3	f
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	1

1. Выделяем носитель функции: $f = 1$
2. Находим интервалы:
   (a) $I_1 = \begin{Bmatrix} 0 & 0 & 1 \\ 0 & 1 & 1 \end{Bmatrix} \longleftrightarrow \pi_1 = \overline{x_1} \cdot x_3$
   (b) $I_2 = \begin{Bmatrix} 1 & 1 & 1 \\ 0 & 1 & 1 \end{Bmatrix} \longleftrightarrow \pi_2 = x_2 \cdot x_3$
   (c) $I_3 = \begin{Bmatrix} 1 & 1 & 1 \\ 1 & 1 & 0 \end{Bmatrix} \longleftrightarrow \pi_3 = x_1 \cdot x_2$
   (d) $I_4 = \begin{Bmatrix} 1 & 1 & 0 \\ 1 & 0 & 0 \end{Bmatrix} \longleftrightarrow \pi_4 = x_1 \cdot \overline{x_3}$
3. Выделяем ядровые интервалы I^*:
   (a) $I_1^* = \begin{Bmatrix} 0 & 0 & 1 \\ 0 & 1 & 1 \end{Bmatrix} \longleftrightarrow \pi_1 = \overline{x_1} \cdot x_3$
   (b) $I_4^* = \begin{Bmatrix} 1 & 1 & 0 \\ 1 & 0 & 0 \end{Bmatrix} \longleftrightarrow \pi_4 = x_1 \cdot \overline{x_3}$
   (c) Ядро: $I_1 \cup I_4$
   (d) $Д_{\text{ядр}} = \pi_1 \vee \pi_4$

4. $N_f = I_1 \cup I_4 \cup I_2$ — неприводимое покрытие

   (a) $Д_1 = \pi_1 \vee \pi_4 \vee \pi_2$ — **минимальная ДНФ**

      i. $r_1 = 6$

Часть V

Лекция 5 (14.03.15) "Метод карт Карно и Метод Квайна—Мак-Клоски"

4.10 Метод карт Карно [162]

1. Интервалы

   (a) Ранг 1: две соседние строки или столбца ($\overline{x_4}$)

   (b) Ранг 2: четыре соседние клетки ($x_2 \cdot x_4$)

   (c) Ранг 3: две соседние клетки ($\overline{x_2} \cdot x_3 \cdot x_4$)

   (d) Ранг 4: отдельно взятая клетка ($\overline{x_1} \cdot \overline{x_2} \cdot \overline{x_3} \cdot x_4$)

2. Пример

   (a) f = (1011 1001 1100 1101) — функция 4 переменных, заданная векторно

   (b) $f = \begin{array}{cccc} 1 & & 1 & 1 \\ 1 & & 1 & \\ 1 & 1 & 1 & \\ 1 & 1 & & \end{array}$ — носитель функции в форме Карно

   (c) Интервалы

      i. $I_1^* \longleftrightarrow \pi_1^* = \overline{x_3} \cdot \overline{x_4}$

      ii. $I_2^* \longleftrightarrow \pi_2^* = x_1 \cdot \overline{x_3}$

      iii. $I_3 \longleftrightarrow \pi_3 = x_1 \cdot x_2 \cdot x_4$

      iv. $I_4 \longleftrightarrow \pi_4 = x_2 \cdot x_3 \cdot x_4$

      v. $I_5 \longleftrightarrow \pi_5 = \overline{x_1} \cdot x_3 \cdot x_4$

      vi. $I_6 \longleftrightarrow \pi_6 = \overline{x_1} \cdot \overline{x_2} \cdot x_3$

      vii. $I_7 \longleftrightarrow \pi_7 = \overline{x_1} \cdot \overline{x_2} \cdot \overline{x_4}$

   (d) $Д_{сокр} = \pi_1 \vee \pi_2 \vee ... \vee \pi_7 = f(x_1, x_2, x_3, x_4)$

   (e) $Д_{ядр} = \pi_1 \vee \pi_2$

   (f) Ядро: $I_1 \cup I_2$

   (g) $N_f = I_1 \cup I_2 \cup I_4 \cup I_6$

   (h) $Д_1 = \pi_1 \vee \pi_2 \vee \pi_4 \vee \pi_6$

4.11 Метод Квайна—Мак-Клоски

1. Импликанта K_1 целиком покрывает импликанту K_2, если каждая переменная, входящая в K_1, точно также входит и в K_2.

2. Общие идеи алгоритма Квайна:

   (a) Функцию представляем в виде СДНФ

   (b) Проводим все возможные склеивания между элементарными конъюнкциями, входящими в СДНФ

   (c) Выписываем все полученные новые элементарные конъюнкции меньших рангов

   (d) Снова проводим все возможные склеивания до тех пор, пока это возможно

   (e) Из всех полученных элементарных конъюнкций (импликант) оставляем только те, которые не покрываются другими элементарными конъюнкциями

   (f) В результате остаются все простые импликанты данной функции ($Д_{сокр}$)

   (g) Формализация Мак-Клоски: $\overline{x_1} \cdot x_2 \cdot \overline{x_3} \cdot x_4 \longleftrightarrow 0101$

3. Алгоритм Мак-Клоски на примере функции f

   (a) Выписываем все импликанты, входящие в СДНФ функции в формализованном виде, разделив их на классы по числу единиц, входящих в эти вектора.

      i.
$$
\begin{array}{ccccc}
0000 & * & 00-0(\pi_7) & & --00(\pi_1) \\
0010 & * & 0-00 & * & 1-0-(\pi_2) \\
0100 & * & -000 & * & \\
1000 & * & 001-(\pi_6) & & \\
0011 & * & -100 & * & \\
1100 & * & 1-00 & * & \\
1001 & * & 100- & * & \\
0111 & * & 0-11(\pi_5) & & \\
1101 & * & 110- & * & \\
1111 & * & 1-01 & * & \\
 & & -111(\pi_4) & & \\
 & & 11-1(\pi_3) & &
\end{array}
$$

   (b) Проводим все возможные склеивания между векторами. Вектора, участвующие в склеивании отмечаем звездочками, а результат склеивания записываем в новый столбец справа. Участвовать в склеивании могут только вектора из соседних классов.

   (c) Те элементарные конъюнкции, которые остались непомеченными звездочками, являются простыми импликантами.

(d) Построение таблицы Квайна:

i.

	k_1	k_2	k_3	k_4	k_5	k_6	k_7	k_8	k_9	k_{10}
π_1	+		+	+		+				
π_2				+		+	+		+	
π_3									+	+
π_4								+		+
π_5					+			+		
π_6		+			+					
π_7	+	+								

(e) Ядровые импликанты: Ищем столбцы, куда входит только один $+$. Строка, соответствующая этому плюсу соответствует ядровой импликанте.

i. $Д_{ядр} = \pi_1 \vee \pi_2$

(f) Сокращенная таблица Квайна:

i. Вычеркиваем вся ядровые строки

ii. Вычеркиваем все столбцы, в которых есть вычеркнутые плюсы

iii.

	k_2	k_5	k_8	k_{10}
π_3				+
π_4			+	+
π_5		+	+	
π_6	+	+		
π_7	+			

iv. Находим наименьшие сочетания строк, которые содержат хотя-бы один плюс из каждого столбца:

A. $Д_1 = Д_{ядр} \vee \pi_4 \vee \pi_6$

B. $Д_2 = Д_{ядр} \vee \pi_3 \vee \pi_5 \vee \pi_7$

Часть VI

Лекции 6,7 (21,28.03.15) “Функциональная полнота системы”

5 Функциональная полнота системы функций

1. Пусть:

   (a) $\Sigma = \{f_1, ..., f_k\}$ — система функций — конечное множество Булевых функций.

2. Тогда:

   (a) Система Σ называется **функционально полной**, если любую Булеву функцию можно представить в виде суперпозиции функций этой системы.

   (b) $\forall f = F_\Sigma = f_{i_1}(f_{i_2}, ..., f_{i_n})$

5.1 Теорема 1 (о функциональной полноте)

1. Пусть:

   (a) Есть две системы функций: Σ и Σ_1

   (b) Σ_1 — функционально полная система

2. Тогда:

   (a) Если любая функция из системы Σ_1 представима формулой $\forall f_i^{(1)} = F_\Sigma$, то и система Σ также будет функционально полной.

5.2 Теорема 2

Если система $\Sigma = \{f_1, ..., f_k\}$ — функционально полная, то система $\Sigma^* = \{f_1^*, ..., f_k^*\}$, состоящая из двойственных функций, также будет функционально полной.

5.3 Основные классы функционально полных систем

1. $\Sigma_1 = \{\wedge, \vee, -\}$ — СДНФ

   (a) $0 = x \wedge \overline{x}$

2. $\Sigma_2 = \{\&, -\}$

   (a) $x \vee y = \overline{(\overline{x} \wedge \overline{y})}$

3. $\Sigma_3 = \{\vee, -\}$

   (a) $x \wedge y = \overline{(\overline{x} \vee \overline{y})}$

4. $\Sigma_4 = \{x/y\}$

   (a) $x/y = \overline{x \wedge y}$ — конъюнкция с отрицанием

   (b) $x/x = \overline{x \wedge x} = \overline{x}$ — отрицание

   (c) $(x/y)/(x/y) = x \wedge y$ — конъюнкция

5. $\Sigma_5 = \{x \downarrow y\}$

(a) $x \downarrow y = \overline{x \vee y}$

(b) $x \downarrow x = \overline{x \vee x} = \overline{x}$ — отрицание

(c) $(x \downarrow y) \downarrow (x \downarrow y) = x \vee y$ — дизъюнкция

6. $\Sigma_6 = \{x \oplus y, x \wedge y, 1\}$ — система Жегалкина

   (a) $\overline{x} = x \oplus 1$

   (b) $x \vee y = ((x \oplus 1) \wedge (y \oplus 1)) \oplus 1 =$
   $= (x \wedge y \oplus y \oplus x \oplus 1) \oplus 1 =$
   $= x \wedge y \oplus y \oplus x$

5.4 Полином Жегалкина

Полином Жегалкина — сумма по модулю 2 различных правильных элементарных конъюнкций или отдельно взятая константа 1 без отрицаний переменных и константы 0.

5.5 Теорема 3

Любая Булевая функция, тождественно неравная 0 представима единственным образом в виде полинома Жегалкина.

5.6 Метод неопределенных коэффициентов для получения полинома Жегалкина

$f(x_1, ..., x_n) = c_0 \cdot 1 \oplus c_1 x_1 \oplus ... \oplus c_n x_n \oplus c_{n+2} x_1 x_2 \oplus ... \oplus c_N x_1 x_2 ... x_n$

5.6.1 Пример

x_1	x_2	x_3	f
0	0	0	1
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	0

1. $f(x_1, x_2, x_3) = c_0 1 \oplus c_1 x_1 \oplus c_2 x_2 \oplus c_3 x_3 \oplus$
   $\oplus c_4 x_1 x_2 \oplus c_5 x_1 x_3 \oplus c_6 x_2 x_3 \oplus c_7 x_1 x_2 x_3$

   (a) $1 = c_0$

   (b) $0 = c_0 \oplus c_3 = 1 \oplus c_3 \Longrightarrow c_3 = 1$

   (c) $0 = 1 \oplus c_2 \Longrightarrow c_2 = 1$

(d) $1 = c_0 \oplus c_2 \oplus c_3 \oplus c_6 = 1 \oplus 1 \oplus 1 \oplus c_6 \Longrightarrow c_6 = 0$

(e) $1 = 1 \oplus c_1 \Longrightarrow c_1 = 0$

(f) $0 = 1 \oplus c_1 \oplus c_3 \oplus c_5 \Longrightarrow c_5 = 0$

(g) $1 = 1 \oplus 0 \oplus 1 \oplus c_4 \Longrightarrow c_4 = 1$

(h) $0 = 1 \oplus 0 \oplus 1 \oplus 1 \oplus 1 \oplus 0 \oplus 0 \oplus c_7 \Longrightarrow c_7 = 0$

2. $f(x_1, x_2, x_3) = 1 \oplus x_2 \oplus x_3 \oplus x_1x_2$

5.7 Линейная функция

1. Булева функция называется **линейной**, если полином Жегалкина, соответствующий ей, не содержит ни одной элементарной конъюнкции ранга выше 1.

   (a) $f(x_1, ..., x_n) = c_0 \oplus c_1x_1 \oplus ... \oplus c_nx_n$

2. Множество L всех линейных функций называется **классом** линейных функций.

5.7.1 Необходимое условие линейности функции

1. Если функция линейна, то на половине своих наборов значений аргументов она равна единице (т.е. в векторе значений функции число нулей и единиц совпадает).

2. Следствие: Если в векторе значений функции число нулей и единиц разное, то это обязательно нелинейная функция. А если число нулей и единиц совпадает, тио эта функция может быть как линейной, так и нелинейной.

5.8 Класс функций

1. Пусть:

   (a) B — класс каких-либо функций.

2. Тогда:

   (a) **Замыканием** класса B будем называть множество всех Булевых функций, представимых в виде суперпозиции функций класса B.

   (b) Класс B называется **замкнутым**, если его замыкание совпадает с ним самим: $[B] = B$

   (c) Класс L является замкнутым: $[L] = L$

5.9 Класс самодвойственных функций

1. Функция f называется **самодвойственной**, если $f^* = f$
2. У самодвойственной функции на всех противоположных наборах противоположные значения, а у несамодвойственной функции существует хотя-бы одна пара противоположных значений аргументов, на которой значения функции одинаковы.
3. S — множество всех самодвойственных функций.
4. Класс S функционально замкнут.

5.10 Монотонность функции

1. Пусть:
   (a) $\vec{\alpha} = (\alpha_1, ..., \alpha_n)$
   (b) $\vec{\beta} = (\beta_1, ..., \beta_n)$
2. Тогда:
   (a) Вектор $\vec{a}$ предшествует вектору $\vec{\beta}$, если: $\vec{\alpha} \preceq \vec{\beta}$
   (b) Функция является монотонной, если:
      i. $\forall \vec{\alpha} \preceq \vec{\beta} \Longrightarrow f(\vec{\alpha}) \leq f(\vec{\beta})$
      ii. $\alpha_1 \leq \beta_1, ..., \alpha_n \leq \beta_n$
   (c) Если на нулевом векторе $f = 1$ или на единичном векторе $f = 0$ и эта функция не является константой, то эта функция обязательно немонотонна.
   (d) Константы являются монотонными функциями.

5.11 Класс функций, сохраняющих константу 0 или 1

1. Пусть:
   (a) $T_0 = f(0, ..., 0) = 0$
   (b) $T_1 = f(1, ..., 1) = 1$
2. Тогда:
   (a) Классы T_0 и T_1 монотонны

5.12 Критерий полноты системы функций

1. Пусть:
   (a)

x_1	x_2	x_3	f
0	0	0	1
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	0

2. Тогда:
   (a)

	T_0	T_1	S	M	L
f	–	–	–	–	–

   (b) Система функций не должна полностью принадлежать какому-либо из замкнутых классов.

5.13 Лемма 1

Если функция f_1 несамодвойственная, то из нее путем подстановки вместо аргументов переменной x можно получить одну из констант.

1. $f(\alpha_1, ..., \alpha_n) = f(\overline{\alpha_1}, ..., \overline{\alpha_n}) = a^{\overline{\alpha}*}$
2. $f(x_1^{\alpha_1}, ..., x_n^{\alpha_n}) = \phi(x)$
3. $\phi(0) = f(0^{\alpha_1}, ..., 0^{\alpha_n}) = f(\overline{\alpha_1}, ..., \overline{\alpha_n})$
4. $\phi(1) = f(1^{\alpha_1}, ..., 1^{\alpha_n}) = f(\alpha_1, ..., \alpha_n)$

5.14 Лемма 2 (о немонотонной функции)

1. Если:
   (a) Функция не является монотонной
2. Тогда:
   (a) Из нее путем подстановки вместо переменных констант и функции x можно получить функцию $\overline{x}$.
   (b) $\exists\overline{\alpha} = (\alpha_1, ..., \alpha_n) \preceq \overline{\beta} = (\beta_1, ..., \beta_n)$
   (c) $\alpha_i \leq \beta_i$
   (d) $f(0, 0, x) = \phi(x) = \overline{x}$

5.15 Лемма 3 (о нелинейной функции)

1. Если:

   (a) Функция не является линейной

2. Тогда:

   (a) Из нее путем подстановки вместо аргументов констант 0, 1, x, $\overline{x}$, y, $\overline{y}$ можно получить $x \cdot y$, $x \vee y$, $\overline{x \cdot y}$, $\overline{x \vee y}$.

   (b) $f(x_!, ..., x_n) = x_1 \cdot x_2 \cdot f_1(x_3, ..., x_n) \oplus f_2(x_3, ..., x_n) \oplus x_2 f_3(x_3, ..., x_n) \oplus f_4(x_3, ..., x_n)$

   (c) $f(x_1, ..., x_4) = x_1x_2x_3 \oplus x_1x_2\overline{x_4} \oplus x_1x_3 \oplus x_1x_4 \oplus x_2x_3 \oplus x_3\overline{x_4} \oplus x_4 =$
   $= x_1x_2(x_3 \oplus \overline{x_4}) \oplus x_1(x_3 \oplus x_4) \oplus x_2x_3 \oplus x_3\overline{x_4} \oplus x_4$

   (d) $\exists \alpha_3, ..., \alpha_n : f_1(\alpha_3, ..., \alpha_n) = 1$

   (e) $f(x_1, x_2, \alpha_3, ..., \alpha_n) = x_1x_2 \oplus ax_1 \oplus bx_2 \oplus c$

   (f) $a = f_2(\alpha_3, ..., \alpha_n)$, $b = f_3(\alpha_3, ..., \alpha_n)$, $c = f_4(\alpha_3, ..., \alpha_n)$

   (g) $a = 1$, $b = 1$

      i. $f(x_1, x_2, \alpha_3, ..., \alpha_n) = x_1x_2 \oplus x_1 \oplus x_3 \oplus c =$
      $= (x_1 \vee x_2) \oplus c = \begin{cases} x_1 \vee x_2 & c = 0 \\ \overline{(x_1 \vee x_2)} & c = 1 \end{cases}$

   (h) $a = 1$, $b = 0$

      i. $f(x_1, x_2, \alpha_3, ..., \alpha_n) = x_1x_2 \oplus x_1 \oplus c =$
      $= x_1(x_2 \oplus 1) \oplus c = x_1\overline{x_2} \oplus c = x_1\overline{x_2} \oplus c =$
      $= x_1\overline{x_2} \oplus c = \begin{cases} x_1\overline{x_2} & c = 0 \\ \overline{x_1\overline{x_2}} & c = 1 \end{cases}$

   (i) $a = 0$, $b = 0$

      i. $f(x_1, x_2, \alpha_3, ..., \alpha_n) = x_1x_2 \oplus c = \begin{cases} xy & c = 0 \\ \overline{xy} & c = 1 \end{cases}$

5.16 Теорема Поста

Система функций $\sum = \{f_1, ..., f_k\}$ будет функционально полной только тогда, когдаона не содержится полностью ни в одном замкнутом контуре

	T_0	T_1	S	M	L
f_1	$-$				
f_2		$-$			
f_3			$-$		
f_4				$-$	
f_5					$-$
f_6					

1. Пусть:

   (a) $\sum$ — функционально полная система

2. Доказать:

   (a) Система не входит полностью ни в один из основных замкнутых классов

3. Доказательство:

   (a) $\sum = \{f_1, ..., f_k\}$
   (b) $\sum \subset B \Longrightarrow [\sum] \subset [B] = B$
   (c) ...

Часть VII

Лекция 8 (04.04.15) "Схемы из функциональных элементов"

6 Схемы из функциональных элементов

1. **Функциональным элементом** [164] будем называть устройство с n упорядоченными входами и одним выходом такое, что при подаче на входы любой комбинации двоичных сигналов $x_1, ..., x_n$, на выходе возникает сигнал равный значению функции $f(x_1, ..., x_n)$. Функциональный элемент реализует функцию $f(x_1, ..., x_n)$.

2. Элементарные функциональные элементы:

   (a) Конъюнктор
   (b) Дизъюнктор
   (c) Инвертор

3. **Минимальной** функциональной схемой для функции $f(x_1, ..., x_n)$ называется схема на конъюнкторах, дизъюнкторах и инверторах, реализующая данную функцию и содержащая наименьшее число L_f (сложность функции f) функциональных элементов.

4. **Функция Шеннона** $L(n) = \max L_f \sim \frac{2^n}{n}$ — максимальная сложность функции от n аргументов.

5. Алгоритм построения минимальной функциональной схемы:

   (a) Находим все минимальные ДНФ (D_{min}) функции

(b) Каждую минимальную ДНФ преобразуем вынесением операций за скобки так, чтобы количество операций было наименьшим

(c) Строим схему, реализующую полученную формулу

6. **Сумматор** [165]

   (a) $X = X_n X_{n-1} ... X_2 X_1$

   (b) $Y = Y_n Y_{n-1} ... Y_2 Y_1$

   (c) $Z = X + Y = Z_{n+1} Z_n Z_{n-1} ... Z_2 Z_1$

   (d) $\begin{cases} q_{i+1} = x_i \cdot y_i \bigvee q_i \cdot x_i \bigvee q_i \cdot y_i \\ z_i = x_i y_i q_i \bigvee (x_i \vee y_i \vee q_i) \cdot \overline{(x_i \cdot y_i \vee q_i \cdot x_i \vee q_i \cdot y_i)} \end{cases}$

Часть VIII

Лекция 9 (11.04.15) "Теория графов"

7 Теория графов

1. $G = \{V, X, \theta\}$ — граф

   (a) $|V| = m$, $V = \{v_1, ..., v_m\}$

   (b) $X = \{x_1, ..., x_n\}$

   (c) $\theta(x_i) = (v_j, v_k)$ — функция инцидентности

2. Если в функции инцидентности допускается менять местами вершины, то граф называется **неориентированным**, иначе — **ориентированным**.

   (a) $(v_i, v_k) \neq (v_k, v_i)$ — ориентированный граф (**орграф**)

3. Если какой-то паре вершин инцидентны несколько ребер, то они называются **кратными**.

4. Если на каком-то ребре пара состоит из одной и той же вершины, то такое ребро называется **петлей**.

   (a) $\theta(x_{i_0}) = (v_{j_0}, v_{j_0})$, x_{i_0} — петля

5. Граф называется **простым**, если в нем нет кратных ребер и петель.

6. Простой граф называется **полным**, если каждой паре вершин инцидентно ровно одно ребро.

7. Граф называется **двудольным**, если его множество вершин можно разбить на два таких непересекающихся подмножества, что ребра соединяют вершины только из разных подмножеств.

8. Двудольный граф называется **полным**, если каждая вершина одного подмножества соединяется одним ребром с каждой вершиной другого подмножества. [167]

9. **Маршрутом** в графе будем называть чередующуюся последовательность вершин и ребер $v_{i_1}, x_{i_1}, ..., v_{i_l}, x_{i_l}, v_{i_l+1}$, начинающуюся и заканчивающуюся вершинами, такую, что каждое ребро в ней соединяет две соседние вершины.

   (a) Маршрут, в котором нет повторяющихся ребер, называется **цепью**.

   (b) Маршрут, в котором нет повторяющихся вершин, называется **простой цепью**.

   (c) Если в простой цепи повторяются только первая и последняя вершины, то данная цепь называется **циклом**.

10. Граф называется **связным**, если любая его вершина достижима из любой другой вершины. В противном случае граф называется **несвязным** и он разбивается на несколько частей, каждая из которых является связным графом. Эти части называются **компонентами связности**.

11. Ребро называется **циклическим**, если оно принадлежит хотя-бы одному циклу графа. В противном случае, ребро называется **ациклическим**.

12. Утверждение: Если из связного графа удалить циклическое ребро, то вновь полученный граф останется связным. А если удалить ациклическое ребро, то граф распадется на две компоненты связности.

13. **Деревом** называется связный граф, все ребра которого — ациклические. **Лес** — это несвязный граф, все ребра которого — ациклические.

   (a) Чтобы простой связный граф был деревом, необходимо и достаточно, чтобы у него число вершин было на 1 больше числа ребер.

   (b) Чтобы граф был деревом, необходимо и достаточно, чтобы любые две его вершины соединялись единственным маршрутом.

   (c) Граф будет деревом тогда и только тогда, когда добавление в него любого нового ребра приводит к возникновению ровно одного цикла.

14. **Степенью** вершины называется число ребер, инцидентных этой вершине.

15. **Эйлеров цикл** — маршрут в графе, который проходит по каждому ребру ровно один раз (и начинается и заканчивается в одной и той же вершине).

    (a) Те графы, где возможен эйлеров цикл, называются **эйлеровыми графами**.

    (b) Критерий эйлеровости графа: Для того, чтобы связный граф без петель был эйлеровым, наобходимо и достаточно, чтобы степень каждой его вершины была четным числом.

16. **Гамильтоновым циклом** называется маршрут, который проходит через все вершины и каждая вершина проходится всего один раз.

17. **Укладкой** графа называется такое его геометрическое представление, при котором ребра не пересекаются (соединяются только в вершинах).

    (a) Если существует укладка графа на плоскости, то граф называется **планарным**.

    (b) Каждый плоский граф делит ребрами плоскость на несколько частей.

    (c) Для любого графа существует укладка в трехмерном пространстве.

    (d) Если существует укладка графа на сфере, то граф будет планарным.

    (e) Граф любого выпуклого многогранника планарен.

18. Теорема Эйлера о плоских графах: Для плоского связанного графа справедливо следующее соотношение $m - n + p = 2$, где:

    (a) m — число вершин

    (b) n — число ребер

    (c) p — число граней

    (d) Следствия:

        i. Для плоского связного простого графа, у которого болльше двух вершин, справедливо следующее соотношение $n \leq 3(m-2)$

        ii. Для плоского связного простого графа, у которого нет треугольных граней, справедливо следующее соотношение $n \leq 2(m-2)$

19. Два графа называются **гомеоморфными**, если они получены из одного и того же графа путем применения конечного числа операций разбиения ребер.

20. Критерий планарности: Граф будет **планарен** тогда и тогда, когда он содержал подграфов, гомеоморфных графуK_5.

21. **Остов графа** — дерево графа, содержащее все вершины исходного графа и какое-то количество его ребер.

    (a) Требуется найти остов, сумма длин ребер которого будет наименьшей среди всех остовов этого графа.

    i. Переобозначим ребра в порядке возрастания их длины $l(x_1) \leq l(x_2) \leq l(x_3)$

    ii. Составим остов, выбирая ребра до тех пор, пока они не образуют петлю или цикл.

22. **Парасочетанием** двудольного графа называется любое множество несмежных вершин этого графа.

    (a) Парасочетание называется **максимальным** для данного графа, если оно содержит наибольшее число ребер среди всех парасочетаний этого графа.

    (b) Парасочетание называется **совершенным** из v в w, если число ребер в нем совпадает с числом вершин в множестве v.

    (c) Совершенное парасочетание обязательно максимально.

    (d) Теорема Холла: Для того, чтобы в двудольном графе существовало **совершенное** парасочетание, необходимо и достаточно, чтобы для любого подмножества S выполнялось $|S| \leq |\phi(S)|$. И наоборот, если совершенного парасочетания нет, то существует хотя-бы одно подмножество S такое, что $|S| > |\phi(S)|$.

    (e) Парасочетание называется **полным**, если к нему нельзя добавить ни одного ребра графа, не нарушив условия несмежности ребер. [169]

    (f) Ребра, входящие в полные парасочетания будем называть **толстыми**, а вершины, соединенные такими ребрами будем называть **насыщенными**.

    (g) **Чередующейся** цепью будем называть цепь в графе, в которой тонкие и толстые ребра чередуются.

    (h) **Тонкой** чередующейся цепью будем называть чередующуюся цепь, начинающуюся и заканчивающуюся тонкими ребрами.

    (i) Венгерский алгоритм нахождения максимального парасочетания: [169]

    i. Находим полное парасочетание $\Pi = \{(v_1, w_1), (v_2, w_2)\}$

    ii. Для данного парасочетания ищем тонкую чередующуюся цепь (маршрут, соединяющий две ненасыщенные вершины, в котором тонкие и толстые ребра чередуются). Если такого пути нет, то алгоритм закончен и имеющееся парасочетание максимально. $v_3 w_1 v_1 w_3$

iii. В имеющемся парасочетании убираем те толстые ребра, которые входят в путь и заменяем их теми тонкими ребрами, которые входят в найденный путь. $\Pi = \{(v_3, v_1), (v_1, w_3), (v_2, w_2)\}$. В новом парасочетании ребер будет на 1 больше, чем в старом.

iv. С новым найденным парасочетанием идем в пункт 2.

7.1 Способы задания графов

1. Аналитический:
   (a) $x_1 - (v_{j_1}, v_{k_1})$
   (b) ...
   (c) $x_n - (v_{j_n}, v_{k_n})$
   (d) $v_{l_1}, ..., v_{l_p}$ — изолированные вершины
2. Геометрический [166]
3. С помощью матрицы инцидентности:
   (a) $A_{mxn} = a_{ij}$
   (b) Для неориентированных графов:
      i. $a_{ij} = 0$, если вершина v_1 не инцидентна ребру x_j
      ii. $a_{ij} = 1$, если вершина v_1 инцидентна ребру x_j
   (c) Для орграфов:
      i. $a_{ij} = 0$, если вершина v_i не инцидентна ребру x_j
      ii. $a_{ij} = -1$, если v_i — начало ребра x_j
      iii. $a_{ij} = 1$, если v_i — конец ребра x_j
   (d) Данный способ работает только для графов без петель
   (e) $A_{4x5} = \begin{pmatrix} 0 & 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 \\ ... & ... & ... & ... & ... \\ ... & ... & ... & ... & ... \end{pmatrix}$
4. Матрица смежности:
   (a) $B_{mxm} = b_{ij}$
   (b) b_{ij} = числу ребер, инцидентных паре вершин (v_i, v_j)
   (c) $B_{4x4} = \begin{pmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 1 \\ ... & ... & ... & ... \\ ... & ... & ... & ... \end{pmatrix}$
   (d) Если граф неориентированный, то матрица смежности **симметрична**.

7.2 Минимальные пути в графах

1. Когда требуется найти путь в графе, соединяющий две вершины и содержащий наименьшее число ребер. [168]

   (a) $\Gamma(v)$

   (b) $\Gamma^{-1}(v)$ — множество вершин, из которох можно попасть в вершину v, пройдя ровно по одному ребру.

2. Для графа, у которого каждому ребру ставится в соотсветствие число, называемое его **длиной**. $l(v_i, v_j)$. [168] Требуется найти пусть такой, что сумма длин ребер у этого пути минимальна для всех путей, идущих из a в b.

   (a) Начальной вершине a даем метку $\lambda_a = 0$. Всем остальным вершинам $\lambda_i = +\infty$.

   (b) Для каждого ребра проверяем: если $\lambda_j - \lambda_i > l(v_i, v_j)$, то $\lambda_j = \lambda_i + l(v_i, v_j)$

   (c) Та метка, которая останется у вершины b будет равна длине минимального пути. Сам путь ищется обратным ходом от вершины b назад к вершине a.

Часть IX

Лекция 10 (2.05.15) "Алгоритм оптимального назначения (задача об оптимальном назначении)"

8 Алгоритм оптимального назначения

1. $A = (a_{ij}) = \begin{vmatrix} 1 & 2 & 3 & 4 \\ 1 & 4 & 4 & 2 \\ 2 & 5 & 6 & 3 \\ 1 & 6 & 4 & 4 \end{vmatrix}$ - матрица эффективности

   (a) v_i - работник

   (b) w_i - работа

   (c) a_{ij} - эффективность

2. Требуется распределить работы между работниками таким образом, чтобы выполнялись все работы, каждый работник выполнял только одну работу и суммарная эффективность выполнения работ была бы максимальной среди всех распределений работ.

3. В терминах матрицы эффективность это означает, что надо найти среди элементов матрицы n элементов в разных строках и столбцах так, чтобы их **сумма** была максимальной.

4. В рамках двудольных графов это означает, что есть полный двудольный граф с n вершинами в каждом подмножестве и каждому ребру $v_i w_j$ соответствует элемент a_{ij}. Требуется найти совершенное паросочетание, сумма длин ребер которого максимальна среди всех совершенных паросочетаний.

8.1 Алгоритм

1. Каждой вершине v_i ставим метку a_i. $a_i = \max_j a_{ij}$, $b_j = 0$

2. Выделяем все элементы a_{ij}, для которых выполняется условие $a_i + b_i = a_{ij}$

3. Строим двудольный граф, в который входят все вершины v_i и w_j и только те ребра, которые удовлетворяют условию 2.

4. Ищем максимальное паросочетание в построенном графе. Если найденное паросочетание будет совершенным, то оно дает решение задачи. А если оно не будет совершенным, то переходим к шагу 5.

5. Находим такое подмножество S, чтобы число смежных ему вершин было строго меньше.

   (a) $S = \{v_1, v_2, v_3, v_4\}$

   (b) $\phi(S) = \{w_2, w_3, w_4\}$

6. У всех найденных вершин $v_i \in S$ **уменьшаем** метку a_i на единицу. У всех найденных вершин $w_j \in \phi(S)$ метку b_j **увеличиваем** на единицу.

7. С новыми метками переходим к шагу 2.

8.2 Потоки в транспортных сетях

1. $M^-(v)$ - множество ребер, входящих в вершину
   $M^+(v)$ - множество ребер, выходящих из вершины

2. **Транспортной сетью** называется свзяный ориентированный граф без петель, для которого выполнены следующие условия:

   (a) Существует единственная вершина a, называемая **истоком**, для которой $M^-(a) = 0$

   (b) Существует единственная вершина b, называемая **стоком**, для которой $M^+(b) = 0$

(c) Каждому ребру x поставлено в соответствие целое неотрицательное число, которое называется **пропускной способностью** ребра.

3. **Потоком** в транспортной сети называется целочисленная функция, определенная на всех ребрах графа и удовлетворяющая следующим условиям:

   (a) Значение функции потока $\phi(x)$ на каждом ребре x не превосходит пропускной способности ребра. $\phi(x) \leq C(x)$

   (b) Для каждой внутренней вершины $v \neq a$, $v \neq b$

4. $\sum_{x \in M^-(v)} \phi(x) = \sum_{x \in M^+(v)} \phi(x)$

5. **Величиной** потока $|\phi(x)| = val\phi = \sum_{x \in M^+(a)} \phi(x)$ называется сумма значений потока на всех ребрах, выходящих из истока или входящих в сток.

6. Поток в данной транспортной сети называется **максимальным**, если для любого другого потока выполнено условие $|\phi| \leq |\phi^*_{max}|$

7. **Разрезом** ребра, отделяющим исток от стока, называется множество вершин, для которого $a \in S$, а $b \notin S$, из которого выходят ребра, входящие в вершины множества S'.

   (a) **Пропускно способностью разреза** называется сумма пропускных способностей ребер этого разреза.

   (b) Разрез в транспортной сети называется **минимальным**, если его пропускная способность наименьшая среди всех возможных разрезов.

8. Утверждение (**теорема Форда-Фалкерсона**): В транспортной сети величина максимального потока равна пропускной способности минимального разреза.

Часть X

Лекция 11 (23.05.15) “Подготовка к экзамену”

9 Подготовка к экзамену

1. Аксиомы Булевой алгебры

   (a) Булевы вектора и функции

2. Способы задания Булевых функций

   (a) Специальные формы: ДНФ (дизъюнктивная нормальная форма) и КНФ (конъюнктивная нормальная форма)
   (b) $Д_{min}$ методом Квайна или картой Карно
   (c) Полином Жегалкина

3. Функционально полная система

   (a) Определение: Система функций называется функционально полной, если любая функция может быть представлена формулой $\forall f = F_{\Sigma}$, где F_{Σ} - функции, находящиеся в системе.
   (b) Критерий полноты (теорема Поста): Система будем функционально полной, если она не входит полностью ни в один из замкнутых классов:

      i. T_0 - класс функций, сохраняющих константу 0. $f(0,...,0) = 0$
      ii. T_1 - класс функций, сохраняющих константу 1. $f(1,...,1) = 1$
      iii. S - самодвойственность функции
      iv. M - монотонность функции
      v. L - линейность функции: число нулей и единиц в таблице истинности должно совпадать.

   (c) Алгоритм:

      i. Берем функцию, не принадлежащую классу T_0 и в ней отождествляем аргументы $g(x,...,x)$.
      ii. Если $g(x,...,x) = 1$, то берем функцию не из класса T_1 и подставляем единицу в нее, получая константу 0.

4. Основное правило комбинаторики: Если первую часть можно выбрать n способами, а вторую m способами, то общее количество комбинаций равно $n \cdot m$.

5. Графы

   (a) Дерево - связный граф без циклов

      i. Между любыми двумя вершинами есть всего лишь один маршрут.
      ii. Если добавить любое ребро к дереву, то возникнет ровно один цикл.
      iii. Число вершин на 1 больше числа ребер.

   (b) Ориентированность
   (c) Планарность: граф можно нарисовать на плоскости без пересечения ребер.

   i. Критерий планарности

(d) Парасочетание: множество попарно несмежных ребер.

(e) Алгоритм оптимального назначения:

   i. $A = (a_{ij})$ - матрица эффективности

   ii. (v_i, w_j)

(f) Транспортная сеть - ориентированный граф, в котором есть одна вершина, из которой только выходят ребра, и одна вершина, в которую только входят ребра, а каждому ребру сопоставлена пропускная способность.

6. Ядровый интервал - интервал, содаржащий вершину, пренадлежащую только ему.