Dziesiętny – dwójkowy:

0	-	0000
1	-	0001
2	-	0010
3	-	0011
4	-	0100
5	-	0101
6	-	0110
7	-	0111
8	-	1000
9	-	1001
10	-	1010
11	-	1011
12	-	1100
13	-	1101
14	-	1110
15	-	1111

ab \ cd	00	01	11	10
00	0	1	3	2
01	4	5	7	6
11	12	13	15	14
10	8	9	11	10

ab \ c	0	1
00	0	1
01	2	3
11	6	7
10	4	5

ab \ cde	000	001	011	010	110	111	101	100
00	0	1	3	2	6	7	5	4
01	8	9	11	10	14	15	13	12
11	24	25	27	26	30	31	29	28
10	16	17	19	18	22	23	21	20

abc \ de	00	01	11	10
000	0	1	3	2
001	4	5	7	6
011	12	13	15	14
010	8	9	11	10
110	24	25	27	26
111	28	29	31	30
101	20	21	23	22
100	16	17	19	18

a \ bc	00	01	11	10
0	0	1	3	2
1	4	5	7	6

METODA KARNAUGHA:

$y = \Sigma(1,(-))$, $y = \Pi(0,(-))$

I sp. - zakreślanie 1 i −, II sp - zakreślanie 0 i − a potem wypisać rozwiązanie jako sumę iloczynów.

Bezpośrednie konstruowanie wyrażenia boolowskiego - dla wierszy o tej samej wartości funkcji wypisać: sumę iloczynów argumentów, gdy wartość jest 1, iloczyn sum argumentów, gdy wartość jest 0.

Obliczanie MKZ: wypisać iloczyn sum par sprzecznych, przekształcić do sumy iloczynów, MKZ to uzupełnienia zbiorów tych składników.

— : relacje zgodności - dowolna wartość (0 lub 1), Karnaugha - nieokreślone, ekspansja (macierze F i R) - dowolna wartość (0 lub 1), algebra podziałów - dowolna wartość (0 lub 1), na wejściu i wyjściu to samo.

METODA EKSPANSJI: Dane są macierze F i R

1) Macierz blokująca: $B(k_n, R)$ - n-ty wiersz (kostka) z F porównany z R, w B 0 - gdy bez różnicy, 1 - gdy jest różnica
2.) Wypisać obok B miejsca, gdzie są 1, przedstawić jako iloczyn sum, przekształcić do sumy iloczynów.
3.) Zapisać rozwiązania dla wszystkich kostek za pomocą −, 1 i 0 oraz literek.
4.) Oznaczyć niepowtarzające się implikanty, tzw. implikanty proste.
5.) Stworzyć tablicę implikantów prostych, wpisując: 1 - gdy I_n pokrywa k_i oraz 0 - gdy I_n nie pokrywa k_i (patrz na listę z pkt. 3)

	I_1	$I_{..}$	I_n
k_1	⋮	⋮	⋮
k_2	⋮	⋮	⋮
k_i	⋮	⋮	⋮

6.) Wypisać obok tablicy implikantów prostych, gdzie są 1, przedstawić jako iloczyn sum, przekształcić do sumy iloczynów. (pokrycie kolumnowe)
7.) Zapisać wynik jako sumę implikantów prostych (z tablicy) i/lub sumę iloczynów literek, wziętych z listy z pkt. 3.

REDUKCJA ARGUMENTÓW:

Zmienne niezbędne - obliczanie przy pomocy macierzy blokującej, jeżeli w danym wierszu jest tylko jedna 1 to wskazuje ona zmienną niezbędną bezpośrednio, ew. sprawdzić, czy można bezkarnie wykreślić jakąś kolumnę, by wiersze nie stały się sprzeczne, jeśli nie można to jest to właśnie zmienna niezbędna (zważaj dane w zadaniach!)

1.) Dla danych zmiennych niezbędnych wypisać: $P_F = P(y) = \left(\frac{0}{\text{nr. wierszy}}; \frac{1}{\text{nr. wierszy}}\right)$ oraz $P_N = P(\text{zmiennych niezbędnych}) = P(z.n._1) \cdot P(z.n._2) \cdot \ldots = \left(\frac{00}{\text{nr. wierszy}}; \frac{01}{\text{nr. wierszy}}; \frac{10}{\text{nr. wierszy}}; \frac{11}{\text{nr. wierszy}}\right)$ elementy wspólne!
2.) Wykonać tzw. podział ilorazowy, czyli $P_N | P_F = ((\text{zgodne z } P_N \text{ i } P_F)(\text{zgodne z } P_N\text{, ale niezgodne z } P_F))$
3.) Wypisać wszystkie sprzeczności z podziału ilorazowego (wypisać z danej tabelki, na których pozycjach różnią się dane wiersze), wykreślić powtarzające się sprzeczności (zazwyczaj te, mające więcej elementów), resztę zapisać jako iloczyn sum, przekształcić do sumy iloczynów.
4.) Wypisać rozwiązania: {roz.$_1$} {roz.$_2$} .... ∩ {zmienne niezbędne}, czyli minimalny zbiór argumentów.
5.) Wypisać zbiory o najmniejszej liczności: roz$_i$, zmienne niezbędne (czyli te, co mają najmniej argumentów) → sprawdzić do wyrażenia boolowskiego, np. za pomocą tablicy Karnaugha.
6.) Stworzyć tablicę Karnaugha, znajdując odpowiednie wartości w tablicy pierwotnej
7.) Rozwiązać standardowo tablicę Karnaugha.

0 - 0000	8 - 1000
1 - 0001	9 - 1001
2 - 0010	10 - 1010
3 - 0011	11 - 1011
4 - 0100	12 - 1100
5 - 0101	13 - 1101
6 - 0110	14 - 1110
7 - 0111	15 - 1111

	$X_1 \ldots X_n$	y
R		0 0 0 ⋮
F		1 1 1 ⋮
D		− −

Ekspansja (F,R)
Karnaugha (F,D)
(R,D)

oś symetrii

	000	001	011	010	110	111	101	100
00								
01								
11								
10								

Zasady zakreślania pętelek:

- 2^n elementów, jak największe, jak najmniejsza ich liczba, mogą się nakładać i być otwarte, muszą pokryć wszystkie 0/1 i ew. część kresek.

DEKOMPOZYCJA KLASYCZNA:

Wypisać kolumny zgodne; obliczyć MKZ (metodą par zgodnych); wskazać maksymalne pokrycie rozłączne, czyli zbiory z MKZ obejmujące wszystkie pierwotnie występujące kolumny (łączone kolumny nie mogą się powtarzać!); połączyć kolumny zgodnie z maksymalnym pokryciem rozłącznym i zapisać w nowej tabelce z nowym kodowaniem kolumn; zmienić kodowanie kolumn z pierwszej tablicy (pierwotnej) do drugiej (ze sklejonych kolumn). Ew. rozpisać na dwie tablice jednowyjściowe → jeśli w drugiej tablicy kolumny są zakodowane więcej niż jednym bitem. ROZWIĄZANIE: schemat z wejściami i wyjściem.

MINIMALIZACJA LICZBY STANÓW AUTOMATU:

Wyznaczyć pary stanów zgodnych/sprzecznych/warunkowo zgodnych z tablicy; wypisać pary zgodne i sprzeczne; obliczyć MKZ; utworzyć nowy automat, używając MKZ (musi zawierać wszystkie stany z pierwotnego automatu!); sprawdzić zamkniętość nowego automatu → dla użytych MKZ wypisać z pierwotnej tablicy stany następne i sprawdzić, czy zawierają się w użytych MKZ; napisać nowy automat wraz z wyjściami.

REALIZACJA AUTOMATU NA PRZERZUTNIKACH:

$Q \to Q'$	D	T
$0 \to 0$	0	0
$0 \to 1$	1	1
$1 \to 0$	0	1
$1 \to 1$	1	0

Zakodować binarnie automat i zapisać w nowej tabeli; napisać tabelki Karnaugha dla przerzutników danego typu (1 - dla pierwszej kolumny z tabeli pierwotnej, 2 - dla drugiej kolumny itd.); metoda Karnaugha → równania dla przerzutników; analogicznie dla wyjść automatu pierwotnego (tabelka Karnaugha → równanie dla wyjścia)

Pojęcia:

Tablica decyzyjna	Uogólnianie reguł decyzyjnych	Redukcja atrybutów	Obiekt	Atrybut	Rdzeń
Tablica prawdy	Ekspansja	Redukcja argumentów	Kostka	Argument	Zmienne niezbędne

UOGÓLNIANIE REGUŁ DECYZYJNYCH:

1) Obliczyć macierz blokującą B dla obiektów o tym samym wyjściu. (Macierz R = zawartość pierwotnej tablicy - wszystkie wiersze z rozpatrywanym wyjściem)
2) Napisać obok B miejsca, gdzie są 1, przedstawić jako iloczyn sum, przekształcić do sumy iloczynów.
3.) Jeżeli rozwiązanie pokrywa wszystkie obiekty o tym samym wyjściu, to wpisujemy je w uogólnionej tabeli reguł decyzyjnych. Jeśli nie ma takiego rozwiązania to należy uwzględnić w uogólnionej tabeli reguł decyzyjnych indywidualne rozwiązania dla każdego obiektu.

REDUKCJA ATRYBUTÓW: (Zmienne niezbędne dane w zadaniu) - czyli rdzenie.
1) Dla danych rdzeniów wypisać: $P_f = P(y) = \left(\frac{n_1}{\text{nr. wierszy}}; \frac{n_2}{\text{nr. wierszy}}\text{ itd.}\right)$, gdzie n przedstawia wartości z tabeli pierwotnej dane na wyjściu. oraz $P_N = P(r_1)\cdot P(r_2)\ldots = \left(\frac{\text{konf. kodu w danych wejściowych}}{\text{nr. wierszy, w których występuje}}; \ldots\right)$ elementy wspólne!
2.) Wykonać podział ilorazowy, czyli $P_N | P_f = (\overline{(\text{zgodne z } P_N; P_f)(\text{zgodne z } P_N, \text{ale niezgodne z } P_f)})$
3.) Wypisać wszystkie sprzeczności z podziału ilorazowego (wypisać z tabeli pierwotnej, na których pozycjach różnią się dane wiersze), wykreślić powtarzające się sprzeczności, resztę zapisać jako iloczyn sum, po czym przekształcić do sumy iloczynów.
4.) Wypisać rozwiązania, dodając do nich dane rdzenie.

OBLICZANIE MKZ:

I metoda - par sprzecznych
Wypisać iloczyn sum par sprzecznych, przekształcić do sumy iloczynów, MKZ to uzupełnienia zbiorów tych składników.

II metoda - par zgodnych
Wypisać pary zgodne, uporządkować pary (mniejszy element na pierwszej pozycji w parze), zbiory S - tyle zbiorów, ile kolumn (indeks S traktować jakby jak drugą pozycję w parze) i porównujemy, (∅ - przepisujemy indeks, nie ∅ piszemy indeks razem z tym co tworzy parę zgodną), otrzymujemy MKZ

KOD GRAY'A: 000, 001, 011, 010, 110, 111, 101, 100

Podstawy układów logicznych - przygotowanie do egzaminu:

Bramka OR – operacja sumy logicznej.
Bramka AND – operacja iloczynu logicznego.
Bramka NOT – operacja negacji.

Kanoniczna postać sumacyjna – suma iloczynów.
Kanoniczna postać iloczynowa – iloczyn sum.

Metoda Quine'a McCluskey'a – generacja implikantów prostych, selekcja implikantów (tzw. pokrycie)
Metoda Espresso: duża liczba różnorodnych procedur, procedury heurystyczne, iteracyjne poprawianie wyniku.

Automaty (funkcja przejść: σ: S x X → S)
Typu Mealy'ego – funkcja wyjść: λ: S x X → Y
Typu Moore'a - funkcja wyjść: λ: S → Y

Przerzutniki – tablice przejść:

Q\D	0	1
0	0	1
1	0	1

Q\T	0	1
0	0	1
1	1	0

Multiplekser – posiada 2^N wejść informacyjnych i n wejść adresowych, wejście zezwalające i jedno wyjście.

Demultiplekser – Jedno wejście informacyjne, n wejść adresowych, wejście zezwalające i 2^N wyjść.

Sumator – dwa wejścia binarne, jedno wyjście binarne plus dane z bloku poprzedniego i przejście do bloku następnego.

Komparator – dwa wejścia binarne i trzy wyjścia, zależące od wyniku porównania.

Rejestry – buduje się w przerzutników typu D. Najprostszy rejestr – ładowanie i pamiętanie.
Rejestr przesuwający – ładowanie, przesuwanie i zapamiętanie.

Licznik – ładowanie, pamiętanie i zliczanie.

Kod BCD – każda cyfra liczby zapisanej w kodzie dziesiętnym jest przedstawiana czterobitową liczbą binarną.

1	2	3	4	5	Σ

ID PUL – egz. 4.02.2004
Uwaga. Rozwiązania testu zapisujemy wyłącznie na tej kartce,
(również na odwrocie)!!

1 a) Zapisać w kodzie U2 liczby –11, –15:

b) Jaką liczbę dziesiętną całkowitą reprezentuje liczba binarna: 10101 jeżeli jest ona dana
b1) w kodzie naturalnym binarnym:
b2) w kodzie U2:

c) Liczba 5B dana jest heksadecymalnie, zapisać tę liczbę w naturalnym kodzie binarnym oraz w kodzie BCD.

2. Podaj tablicę przejść przerzutnika typu JK i wyprowadź jego funkcję charakterystyczną.

3. W zbiorze S = {1, 2, 3, 4, 5, 6, 7, 8} następujące pary są zgodne: (1,3), (1,7), (2,5), (2,8), (3,4), (3,5), (3,6), (4,5), (4,6), (5,7), (5,8), (6,7), (6,8). Obliczyć (sensowną metodą) wszystkie maksymalne klasy zgodności.

4. Jaką minimalną liczbę linii iloczynu musi mieć układ a) PAL, b) PLA, aby zrealizować zespół funkcji.

$$y_1 = \overline{a}bc + a\overline{c} + bd$$

$$y_2 = a\overline{c} + \overline{b}d$$

$$y_3 = \overline{a}bc + a\overline{c}$$

5. Dla tablicy poniżej obliczyć wszystkie minimalne uogólnienia reguł decyzyjnych **(zapisz rozwiązanie na odwrocie kartki)**.

U	*a*	*b*	*c*	*d*	*e*
1	0	1	0	1	0
2	0	1	0	0	0
3	0	0	0	0	1
4	1	1	0	1	1
5	1	1	0	2	2
6	2	2	0	2	2
7	2	2	2	2	2

Zadanie zaliczeniowe

Dla funkcji F opisanej tablicą zmienne niezbędne są x_4 oraz x_6. Należy wyznaczyć wszystkie minimalne zbiory argumentów, od których zależy ta funkcja oraz jej minimalne wyrażenie boolowskie z najmniejszą liczbą argumentów.

	x_1	x_2	x_3	x_4	x_5	x_6	x_7	F
1	0	1	1	0	1	0	0	1
2	1	1	1	0	0	1	1	1
3	1	0	0	1	0	1	0	1
4	1	1	0	1	1	0	0	0
5	1	0	1	0	0	1	1	1
6	0	1	1	1	0	0	0	1
7	1	0	0	0	0	1	0	0
8	1	1	0	0	1	0	1	1
9	1	1	0	1	1	1	0	1
10	1	0	0	0	0	0	1	0
11	0	1	1	0	1	1	0	1
12	0	1	1	0	0	1	0	1