Rozbudowane Notatki z Informatyki Teoretycznej

Szczegółowe omówienie zagadnień, przykłady oraz pytania do samokontroli

1 1. Znaczenie, działanie oraz najczęściej występujące typy kanałów informacyjnych

Notatki

Kanały informacyjne pełnią kluczową rolę w systemach komunikacji. Ich główną funkcją jest przekazywanie informacji między nadawcą (źródłem) a odbiorcą (miejscem przeznaczenia). W praktyce transmisji informacji rozróżniamy kilka rodzajów kanałów:

1. Kanały bezszumowe:

- Założenie: przekaz przesyłany jest bez żadnych zakłóceń.
- Przykład: idealny kanał transmisyjny używany do teoretycznego rozważania problemów kompresji i kodowania.

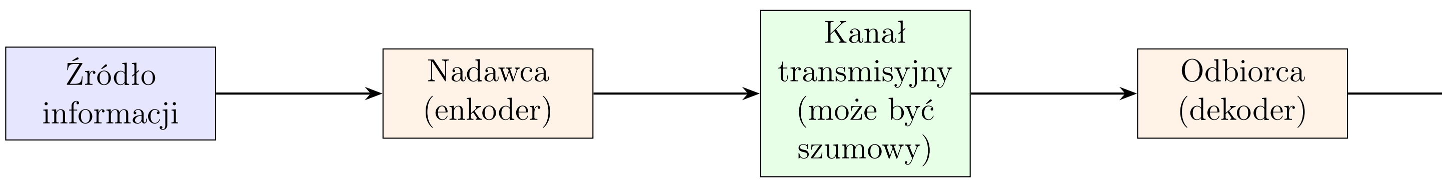
2. Kanały szumowe:

- Rzeczywiste kanały, które wprowadzają błędy, zakłócenia lub szumy.
- Przykłady zakłóceń: interferencje elektromagnetyczne, zakłócenia akustyczne, szumy termiczne.
- Mechanizmy radzenia sobie: stosowanie kodów korekcyjnych (np. kody Hamminga) i filtrów.

3. Kanały wielodostępowe:

- Umożliwiają transmisję informacji z wielu źródeł jednocześnie przez jeden wspólny kanał.
- Przykłady: systemy telefonii komórkowej, sieci komputerowe (np. Ethernet).
- Zagadnienia: kontrola dostępu, algorytmy multipleksacji.

Schemat działania kanału transmisyjnego:



W powyższym schemacie nadawca (enkoder) przekształca informacje uzyskane ze źródła na sygnał nadający się do transmisji, który następnie przesyłany jest przez kanał (który może być obarczony szumem). Odbiorca (dekoder) odbiera sygnał, dokonuje korekcji (w razie potrzeby) i przywraca pierwotną informację.

Przykładowe pytania i odpowiedzi

Pytanie: Wymień i krótko opisz trzy główne typy kanałów informacyjnych.

Odpowiedź:

- **Kanały bezszumowe:** Przekaz jest przesyłany bez zakłóceń, co umożliwia teoretycznie idealne przekazanie informacji.
- **Kanały szumowe:** Rzeczywiste kanały, w których sygnał ulega zakłóceniom. Stosowane są techniki korekcji błędów oraz adaptacyjne metody transmisji.
- **Kanały wielodostępowe:** Jednoczesne przesyłanie informacji z wielu źródeł przez wspólny kanał, wymagające algorytmów zarządzania dostępem i multipleksacji.

2 2. Typy kodów oraz wielkości charakteryzujące kody

Notatki

Definicja kodu: Kod to sposób reprezentowania informacji (danych) za pomocą ciągów symboli (np. bitów) wybranych z określonego alfabetu.

Typy kodów:

- **Kody źródłowe:** Kody służące do reprezentacji informacji w celu jej kompresji. Klasycznym przykładem jest kod Huffmana, który przydziela krótsze słowa częściej występującym symbolom.
- **Kody kanałowe:** Wykorzystywane w celu ochrony przekazywanych informacji przed błędami w kanałach szumowych. Przykłady: kody Hamminga, kody BCH.
- **Kody liniowe:** Klasa kodów, w których suma liniowa (modulo pewnej liczby) dwóch kodowych słów też należy do zbioru kodów. Współczesne systemy korekcji błędów często wykorzystują kody liniowe.

Kluczowe wielkości charakteryzujące kody:

- **Długość słowa kodowego, n:** Liczba symboli, które tworzą kod. Przykład: w kodzie binarnym o długości 7, każde słowo składa się z 7 bitów.
- **Liczba informacji, k:** Liczba symboli zawierających informacje w kodzie. W kodach liniowych zapisuje się je jako (n, k), gdzie $k \leq n$.
- **Odległość Hamminga:** Minimalna liczba pozycji, w których różnią się dwa rozłączne słowa kodowe. Odległość Hamminga decyduje o zdolności wykrywania i korekcji błędów. Przykładowo, aby wykryć jedną zmianę bitową, potrzebujemy kodu o odległości Hamminga co najmniej 2, a do korekcji — co najmniej 3.

Przykładowe pytania i odpowiedzi

Pytanie: Czym jest kod Hamminga i dlaczego odległość Hamminga jest istotna?

Odpowiedź: Kod Hamminga to przykład kodu korekcyjnego, który umożliwia wykrycie i korekcję błędów przy przekazywaniu danych.
Odległość Hamminga określa minimalną liczbę zmian bitowych pomiędzy dwoma poprawnymi słowami kodowymi. Im większa odległość, tym większa możliwość wykrywania i korekcji błędów – na przykład, dla odległości 3 możliwa jest korekcja pojedynczego błędu.

Pytanie: Wyjaśnij różnicę między kodami źródłowymi a kodami kanałowymi.

Odpowiedź:

- **Kody źródłowe** mają na celu reprezentację informacji w sposób umożliwiający jej kompresję (redukcję redundancji). Przykładem jest kod Huffmana.
- **Kody kanałowe** służą do zabezpieczenia informacji przed błędami podczas transmisji przez dodanie nadmiarowych danych (redundancji), co umożliwia wykrycie i korekcję błędów. Przykładami są kody Hamminga i kody BCH.

3 3. Źródła informacji: bezpamięciowe i Markowa

Notatki

Definicja źródła informacji: Źródło informacji generuje symbole z pewnym prawdopodobieństwem.

Źródła bezpamięciowe (Memoryless sources):

- Każdy symbol generowany jest niezależnie od poprzednich.
- Prawdopodobieństwa pozostają stałe w czasie.
- Przykład: rzut monetą – każdy rzut jest niezależny, z prawdopodobieństwem $p = 0.5$ dla orła i $p = 0.5$ dla reszki.

Źródła Markowa:

- Generowanie symboli zależy od wartości poprzednich symboli.
- Opisuje je **łańcuchy Markowa**, gdzie zmiana symbolu zależy od aktualnego stanu.
- Typowy przykład: prognoza pogody, gdzie stan pogody danego dnia (np. słońce, deszcz, pochmurno) jest zależny od stanu z poprzedniego dnia.
  Przykładowe macierze przejścia:

$$P(\text{deszcz}|\text{deszcz}) = 0.7, \quad P(\text{słońce}|\text{deszcz}) = 0.3,$$

$$P(\text{słońce}|\text{słońce}) = 0.8, \quad P(\text{deszcz}|\text{słońce}) = 0.2.$$

Przykładowe pytania i odpowiedzi

Pytanie: Co decyduje o tym, że źródło jest bezpamięciowe? Podaj przykład.

Odpowiedź: Źródło nazywamy bezpamięciowym, gdy każdy kolejny symbol jest generowany niezależnie od poprzednio wygenerowanych. Przykładem jest rzut monetą, w którym wynik każdego rzutu (orzeł lub reszka) nie zależy od wyników poprzednich.

Pytanie: Jak działają źródła Markowa? Podaj przykład macierzy przejścia dla łańcucha Markowa opisującego pogodę.

Odpowiedź: W źródłach Markowa prawdopodobieństwo wygenerowania symbolu zależy od stanu poprzedniego symbolu. Przykładowo, dla prognozy pogody możemy założyć stany: "słońce" i "deszcz". Przykładowa macierz przejścia może wyglądać następująco:

$$\begin{pmatrix} P(\text{słońce}|\text{słońce}) & P(\text{deszcz}|\text{słońce}) \\ P(\text{słońce}|\text{deszcz}) & P(\text{deszcz}|\text{deszcz}) \end{pmatrix} = \begin{pmatrix} 0.8 & 0.2 \\ 0.3 & 0.7 \end{pmatrix}.$$

Oznacza to, że jeśli poprzedni dzień był słoneczny, następnego dnia prawdopodobieństwo słonecznej pogody wynosi 0.8, a deszczu 0.2.

4 4. Ilość informacji, entropia, twierdzenie Shannona o kodowaniu kanałów bezszumowych

Notatki

Ilość informacji: Jeśli zdarzenie ma prawdopodobieństwo p, to ilość informacji (mierzone w bitach) jest dana wzorem:

$$I = -\log_2 p.$$

Im mniejsze prawdopodobieństwo zdarzenia, tym więcej informacji niesie jego zaistnienie.

Entropia źródła: Średnia ilość informacji, która jest w stanie wyrazić symbol pochodzący ze źródła. Oblicza się ją jako:

$$H(X) = -\sum_i p_i \log_2 p_i,$$

gdzie p_i to prawdopodobieństwo wystąpienia i-tego symbolu. Entropia wyraża dolną granicę średniej długości optymalnego kodowania.

Twierdzenie Shannona (o źródle) - podstawowa idea:

- Nie można skompresować przekazu ze źródła bezstratnie poniżej entropii $H(X)$ bitów na symbol.
- Dla dowolnej wartości $\epsilon > 0$ istnieje kod, dla którego średnia długość L spełnia:

$$H(X) \leq L < H(X) + \epsilon.$$

Oznacza to, że entropia jest fundamentalną granicą efektywności kompresji kodu.

Przykładowe pytania i odpowiedzi

Pytanie: Oblicz entropię źródła, które generuje dwa symbole o prawdopodobieństwach p i $1 - p$.

Odpowiedź: Entropia źródła wynosi:

$$H = -\left(p \log_2 p + (1-p)\log_2(1-p)\right).$$

Dla $p = 0.5$ dostajemy:

$$H = -\left(0.5\log_2 0.5 + 0.5\log_2 0.5\right) = -\left(0.5(-1) + 0.5(-1)\right) = 1 \text{ bit},$$

czyli dla równomiernie rozłożonych prawdopodobieństw entropia wynosi 1 bit na symbol.

Pytanie: Czym jest twierdzenie Shannona o kodowaniu kanałów bezszumowych i jakie ma implikacje praktyczne?

Odpowiedź: Twierdzenie Shannona mówi, że średnia długość słowa kodowego w optymalnym kodzie bezstratnym nie może być mniejsza od entropii źródła $H(X)$ oraz, że dla dowolnie małego $\epsilon > 0$, można skonstruować kod, którego średnia długość L spełnia:

$$H(X) \leq L < H(X) + \epsilon.$$

Implikacja: jest to dolna granica dla kompresji informacji – nie można uzyskać średniej długości mniejszej niż entropia bez utraty informacji, co stanowi fundament projektowania algorytmów kompresji.

5 5. Liczbowe systemy pozycyjne: system binarny i system U2

Notatki

System binarny:

- Składa się z dwóch cyfr: 0 i 1.
- Każda pozycja w zapisie liczby ma przypisaną wagę będącą potęgą 2.
- Przykład: liczba $(1011)_2$ to:

$$1 \cdot 2^3 + 0 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^0 = 8 + 0 + 2 + 1 = 11.$$

System U2 (Uzupełnień do 2): Służy do reprezentacji liczb całkowitych ze znakiem w systemach komputerowych.

- **Liczby dodatnie:** Są reprezentowane tak samo jak w systemie binarnym, przy czym najstarszy bit (MSB) jest 0.
- **Liczby ujemne:** Sprowadza się je do postaci uzupełnień do 2. Proces konwersji liczby $-x$ na reprezentację U2:
  1. Zapisz wartość bezwzględną x w postaci binarnej, uzupełniając zerami do żądanej liczby bitów.
  2. Odwróć każdy bit (uzyskasz tzw. uzupełnienie do 1).
  3. Dodaj 1 do otrzymanego wyniku.
- **Przykład:** Przedstawienie liczby -5 w reprezentacji U2 na 8 bitach:
  1. 5 w zapisie binarnym (8-bit): `00000101`.
  2. Odwrócenie bitów: `11111010`.
  3. Dodanie 1: `11111011`.

  Wynik `11111011` reprezentuje liczbę -5 w systemie U2.

Przykładowe pytania i odpowiedzi

Pytanie: Przelicz liczbę $(1011)_2$ na system dziesiętny.

Odpowiedź: Rozłóżmy liczbę $(1011)_2$ według wag pozycyjnych:

$$(1011)_2 = 1 \cdot 2^3 + 0 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^0 = 8 + 0 + 2 + 1 = 11.$$

Pytanie: Opisz szczegółowo, jak przekształcić liczbę -5 do 8-bitowej reprezentacji w systemie U2.

Odpowiedź: Postępuj zgodnie z poniższymi krokami:

1. **Reprezentacja bezwzględnej wartości liczby:**
   5 w systemie binarnym (8-bit): `00000101`.

2. **Odwracanie bitów:**
   Każdy bit liczby `00000101` odwracamy, otrzymując: `11111010`.

3. **Dodanie 1:**
   Dodajemy 1 do liczby `11111010`:
   `11111010` + `00000001` = `11111011`.

Wynik `11111011` to 8-bitowa reprezentacja liczby -5 w systemie U2.

6 6. Działanie maszyny Turinga

Notatki

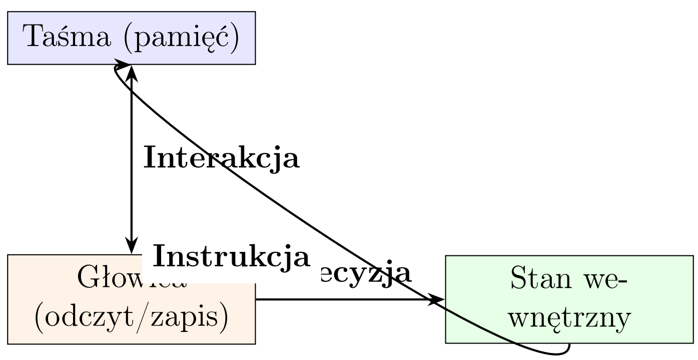
Maszyna Turinga to teoretyczny model komputera, który posłużył do zdefiniowania pojęcia obliczalności. Składa się z następujących elementów:

- **Taśma:** Nieskończony (w obu kierunkach lub jednostronny) pasek podzielony na komórki, z których każda zawiera symbol z pewnego alfabetu. Taśma pełni rolę pamięci.
- **Głowica:** Urządzenie odczytujące i zapisujące symbole na taśmie. Głowica może przesuwać się po taśmie w lewo lub w prawo.
- **Stan wewnętrzny:** Maszyna znajduje się w jednym z skończonej liczby stanów. Stan ten determinuje, jak maszyna reaguje na odczytany symbol.
- **Funkcja przejścia:** Dla danego stanu i odczytanego symbolu funkcja ta określa:
  - Jaki symbol zapisać w bieżącej komórce,
  - W którą stronę przesunąć głowicę (w lewo, w prawo lub pozostawić ją w miejscu),
  - Do którego stanu przejść.

Szczegółowy schemat działania maszyny Turinga:

1. **Inicjalizacja:** Maszyna zaczyna działanie w określonym stanie początkowym, a głowica umieszczona jest nad wybranym symbolem na taśmie.
2. **Odczyt i decyzja:** Głowica odczytuje symbol znajdujący się na taśmie. Na podstawie bieżącego stanu i odczytanego symbolu funkcja przejścia określa:
   - Jaki symbol zapisać,
   - Kierunek ruchu głowicy,
   - Następny stan.
3. **Aktualizacja:** Maszyna zapisuje symbol, przesuwa głowicę i przechodzi do nowego stanu.
4. **Iteracja:** Proces jest powtarzany aż do osiągnięcia stanu akceptującego (końcowego) lub stanu, w którym nie jest określone dalsze działanie (maszyna się zatrzymuje).

Przykładowy graficzny model działania:



Przykładowe pytania i odpowiedzi

Pytanie: Wymień wszystkie podstawowe elementy maszyny Turinga i wyjaśnij ich rolę.

Odpowiedź: Maszyna Turinga składa się z:

- **Taśmy:** Służy jako nieskończona pamięć, na której zapisywane są dane.
- **Głowicy:** Odpowiada za odczyt i zapis symboli na taśmie oraz ich modyfikację.
- **Stanów:** Określają aktualny kontekst działania maszyny.
- **Funkcji przejścia:** Reguluje sposób, w jaki maszyna reaguje na odczytany symbol i aktualny stan, określając operacje (zapis, ruch, przejście do nowego stanu).

Pytanie: Opisz krok po kroku, jak działa maszyna Turinga w procesie obliczeniowym.

Odpowiedź: Proces działania maszyny Turinga przebiega następująco:

1. **Inicjalizacja:** Maszyna znajduje się w stanie początkowym, a głowica umieszczona jest na pierwszej komórce taśmy.
2. **Odczyt:** Głowica odczytuje symbol znajdujący się w aktualnej komórce.
3. **Decyzja:** Na podstawie aktualnego stanu i odczytanego symbolu, funkcja przejścia określa, jaki symbol należy zapisać, w którą stronę przesunąć głowicę oraz jaki będzie kolejny stan.
4. **Wykonanie:** Maszyna zapisuje wyznaczony symbol, przesuwa głowicę i zmienia swój stan.
5. **Iteracja:** Proces powtarza się aż do osiągnięcia stanu końcowego lub zatrzymania maszyny, co oznacza zakończenie obliczeń.