2: "Логика и доказателства" - Резюме

Тема: Формални системи, логика и доказателства, използвани в Z нотацията.

Customer Information Control System (CICS) е фамилия от продукти за банкови трансакции;

B-notation (B method)

Формални системи

- Състоят се от **формален език** (синтаксис: азбука и граматика) и **система за извод** (семантика: аксиоми и правила за извод).

- Позволяват доказване на теореми и формализирано описание на софтуерни системи.

Логика в Z нотацията:

- Всяка софтуерна спецификация се базира на две теории: математическа логика и теория на множествата.
- Използва схеми за описание на типове данни, състояния на системата и операции, начините за промяна на състояние.
-

Пропозиционна логика:

- Работи с твърдения, които са или верни, или неверни: **Твърдението** е изявление (изказване) за предполагаем факт. То е или вярно или невярно, но никога и двете.
- Използват се логически оператори (¬ not, ∧ and/конюмкция, ∨ or/дизюнкция , ⇒ Импликация (ако-

то), ⇔ Еквивалентност/тогава и само тогава/) u uзречения(p, q, r…).

- Семантика: Стойността на истинност се определя чрез таблици на истинност.

Значението на пропозиционното изречение (wff) се дефинира като: **a)** Всеки примитивен символ (p, q, r, …) се интерпретира чрез твърдение, което се свързва със съответната си стойност на истинност: истина (t) или лъжа (f). Пр.: p - Днес е сряда. **b)** Истинността на сложните твърдения се дефинира единствено от истинността на отделните съставящи твърдения.

Методи за доказателства:

Дедукция: Доказателство чрез естествено умозаключение.

Доказателство чрез опровергаване: дтво чрез противоречие.

Доказателство чрез анализ на случаи: Разделяне на доказателството на подслучаи.

Разсъждение чрез равенства: Д-тва чрез еквивалентности (напр. закони на Де Морган, двойно отрицание).

Предикатна логика:

- **Универсални** твърдения (∀) -описват характеристика(и) на всеки обект от разглежданото множество. Заявяваме, че всички елементи от дадено множество S удовлетворяват свойство P
  ∀x : S ● Stooge(x)
  Stooge(x1) ∧ Stooge(x2) ∧ Stooge(x3) ∧ …
  Stooge( ) се нарича ПРЕДИКАТ

-**Екзистенциални** твърдения(∃): съществуването на елемент от множество S, който удовлетворява/притежава свойство P : ∃ x : S ● P(x) или P(x1) v P(x2) v P(x3) v …

Свойствата (предикатите) могат да се разглеждат като **булеви** функции: когато се приложат към аргумент връщат стойност истина или лъжа. Предикатите могат да имат n-аргумента: P(x,y,z)

символът ● обикновено служи като **разделител** или **ограничител на обхвата** - „такова, че“ или „където“.

- Описва свойства на обекти от дадено множество чрез предикати.
- Синтаксисът включва допълнителни символи и обхвати на променливите.

One-point rule -Упрощава екзистенциални изрази чрез замяна на променливи със стойности.

тавтология (tautology)- Някои твърдения винаги се интерпретират като верни

противоречие (contradiction) -Някои твърдения винаги се интерпретират като неверни

(contingency) – например случайност -Твърдение, което е нито истина, нито лъжа

Значение на доказателството- Повишава качеството на софтуера, защото:

-изясняване на изискванията: Процесът на конструиране на доказателства може да помогне в изясняването на системата, както и да идентифицира скритите допускания.

-при проектирането: доказателството може да покаже не само, че проектът е верен, но и да обясни защо е верен.

-в етапа на изпълнение: осигурява факта, че имплементираната част от кода се “държи” като нейната спецификация.

-приложима част при използване на формалните методи в практиката.

Доказателство-Пишем P |= W , твърдението W е истина, когато твърденията от списъка P са истина. Тогава W е семантично следствие на P.

Дедукция: Пропозиционни изчисления: За да завършим нашата система се нуждаем от множество от правила за извод (изчисления). Представяне на правилата: **Ако можем да докажем тези факти /Можем да направим заключение за тези факти или Истинността на заключението е следствие на истинността на предпоставката**

=> - elimination = modus ponens

¬ - introduction = proof by contradiction(assume that the conclusion is not true and derive something that we know to be false - Modus Tollens)

v - elimination = proof by cases (break proof in to separate parts and then combine)

Modus Ponens (Latin: mode that affirms (твърдя)) - Правилото се гласи: If p is true and $p \Rightarrow q$ is true then q is true.

Modus Tollens (the formal name for indirect proof or proof by contrapositive/contradiction) правило за извод гласи: If $p \Rightarrow q$ is true and q is false then p is false.

Case Analysis $(p \Rightarrow r) \wedge (q \Rightarrow r) \equiv (p \vee q \Rightarrow r)$

Simple Case Analysis $(p \Rightarrow r) \wedge (\neg p \Rightarrow r) \equiv r$

Доказателство чрез равенство/еквивалентност - Две изречения са еквивалентни (≡), ако и само ако имат равна стойност на истинност при всяка интерпретация.

$\neg (p \wedge \neg q) \equiv (q \vee \neg p)$

Връзка между синтактичната и семантичната област:

Пропозиционното изчисление е **последователно** ako:

-Всичко, което може да се докаже, е вярно: P |- W , то P |= W

Пропозиционното изчисление е **пълно** (цялостно) ако всичко, което е валидно, може да бъде доказано чрез правила за извод: Ако P |= W , то P |- W

$\Leftrightarrow$ е kamo ≡

|-P ≡ Q (синтактична доказуемост)- означава, че можем да **докажем** логическата еквивалентност P≡Q в дадена формална система, използвайки **правила на дедукция**. Това е въпрос на **формалното доказване** чрез логически правила и аксиоми.

|=P ≡ Q (семантична истинност) - означава, че логическата еквивалентност P≡Q е **вярна във всички модели**. С други думи, при всяка възможна интерпретация P и Q имат една и съща стойност (и двете са или верни, или неверни).

Определители и декларации в Z нотацията:

Ξ x : a | p • q   където:

Ξ определител (∀ или ∃)

x ограничена променлива

a обхват на x

p ограничение

q предикат

Въвеждаме един нов символ на Предикатната логика “: =“ , който да представи факта, че два обекта са еднакви.

Заместване (Substitution rule): Ако m=n, то валидното за n е валидно и за m.

The one-point rule: ako $\exists x \cdot (x=3 \wedge x>2)$ mo $3>2$