Тема2: Формални системи, логика и доказателства, използвани в Z нотацията. - Резюме

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/конюнкция, ∨ or/дизюнкция, ⇒ Импликация (ако-то), ⇔ Еквивалентност/тогава и само тогава/) и изреч.(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: Д-тво чрез противоречие.

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

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

Предикатна логика: Универсални твърдения (∀**)** -описват характеристиkи на всеки обект от разглежданото множество - всички елементи от дад. множество S удовлетворяват свойство P

$\forall x : S \bullet Stooge(x)$
$Stooge(x1) \land Stooge(x2) \land Stooge(x3) \land \dots$
Stooge( ) се нарича ПРЕДИКАТ

Екзистенциални твърдения(∃): съществуването на елемент от множество S, който удовлетворява/притежава свойство P :

$\exists x : S \bullet P(x)$ uлu $P(x1) \lor P(x2) \lor P(x3) \lor \dots$

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 дад. множество чрез предикати.

-Синтаксисът вкл. gопълн. символи и обхвати на променливите.

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

тавтология (tautology)- твърдения винаги верни

противоречие (contradiction) - твърдения винаги неверни

(contingency) – напр.случайност **-** нито истина, нито лъжа

Значение на **доказателството-** Повишава качеството на софтуера**:**

-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 формал. методи в практиката

Доказателство: **P |= W** , твърдението W е истина, когато твърденията от списъка P са истина. **W е семантично следствие на P**.

Дедукция: Пропозиционни изчисления: За да завършим нашата система се нуждаем от множество от **правила за извод** (изчисления): **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 $(p \Rightarrow r) \land (q \Rightarrow r) \equiv (p \lor q \Rightarrow r)$

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

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

$\neg (p \land \neg q) \equiv (q \lor \neg p)$

Пропозиционното изчисление е **последователно** ako: Всичко, което може да се докаже, е вярно: $P \vdash W$, то $P \models W$

Пропозиционното изчисление е **пълно** (цялостно) ако ßсичко, което е валидно, може да бъде доказано чрез правила за извод: Ако $P \models W$, то $P \vdash W$

$\Leftrightarrow$ е kamo ≡

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

|=P ≡ Q (семантична истинност) - логическата еквивалентност P≡Q е вярна във всички модели т.е. при всяка възм. интерпретация P и Q имат една и съща ст-ст(верни/ неверни).

Определители и декларации в Z нотацията:

Ξ x : a | p • q   където: **Ξ** значи (∀ **или** ∃)

x ограничена промен.; **a** обхват на x; **p** ограничение; **q** предикат

: = два обекта са еднакви

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

The one-point rule: е концепция в Z-нотацията, която се използва за опростяване на изрази, свързани с квантори (например ∀ и ∃). ako $\exists x \cdot x=a \land P(x)$ то $P(a)$