Модели на софтуерни системи

доц. Олга Георгиева

СУ, ФМИ

катедра “Софтуерни технологии”

Лекция 2:

Z нотация. Логики и доказателства

ФС - дефиниция

Формалната система *се дефинира чрез два основни елемента:*

1) ***формален език (синтаксис)****:*
азбука + граматика

2) ***система за извод/заключение (семантика)****:*
аксиоми + правила за извод,

Така можем да

- доказваме теореми;

- чрез доказателство, умозаключение.

Софтуерна нотация

Изграждането на ясна, точна спецификация е в основата на всяко формализирано описание.

*Една формална спецификация би трябвало да описва голямо количество проза**. Тя трябва да съпоставя математическите обекти към особеностите на проектираната система:***

състояния на системата,

структури от данни,

техни свойства и

операции с тях.

Z нотация - 1

Нотациите, в частност Z нотацията, се базират на две теории:

математическата логика и
теорията на множествата

Математическите обекти и техните характеристики се събират в структури, наречени ***схеми***. Езикът на ***схемите*** се използва за структурни и композиционни описания:

- събиране на части от информацията,
- формулиране на общи описания и
- доказателства, необходими при следващо приложение.

Езикът на схемите може да се използва **за описание на:**

- **тип (данни)**
- ***на състоянието*** на системата и начините
- **за *промяна на състояние***.

Чрез схемите може да се *изследват характеристиките* на системата.

Z нотация - 2

- **Z нотацията се базира на**:
  математическата логика и теорията на множествата

- **Схеми** - Начин на структуриране на математиката чрез
  "именувани" структури – записи с декларативна и ограничаваща част

- **Изчисления със схеми**
  Математически оператори за построяване на по-големи схеми чрез по-малки

- **Синтактични конвертори за целите на:**

  - описание на характеристиките/функциите на системата

  - разсъждения с цел *пречистване (refinements)* на проектирането с цел достигне на формално описание, което е най-лесно и точно за програмиране.

Customer Information Control System (CICS)

> **CICS е фамилия от продукти за банкови трансакции, която осигурява достъп до данни, комуникация, цялостност, сигурност т.е. управлява информация...Има около 30000 лиценза.**

***Производител*: IBM UK Laboratories at Hursley Park.**

***История*:**

- от средата на 1970 регулярно обновяване на версиите
- началото на 80-те и академичната идея на Оксфорд (Tony Hoare)
- адаптация на формалния метод към индустриално приложение
- юни, 1989 – първата версия на CICS, реализирана чрез Z.
- 1992 – Награда за технологични постижения.

> **B-notation (B method)**

B is a tool-supported formal method based around Abstract Machine Notation. It was originally developed by Jean-Raymond Abrial in France and the UK. **B** is related to the Z notation and supports development of programming language code from specifications. It is attracting increasing interest in industry. It has robust, commercially **available tool suppor**t for specification, design, proof and code generation.

Compared to Z, B is slightly more low-level and more focused on refinement to code rather than just formal specification — hence it is easier to correctly implement a specification written in B than one in Z.

Математическата логика

Две формални системи, които са в основата на

- ***Пропозиционна логика (****Propositional logic)*
- ***Предикатна логика*** *(Predicate logic)*

Основни методи за доказателство, използвани за тези системи

- *Дедукция (Natural deduction*)
- *Доказателство чрез опровергаване (Proof by contradiction)*
- *Доказателство чрез анализ на случаи (Proof by case analysis)*
- *Разсъждение чрез равенства (Equational reasoning)*

Пропозиционна логика*: Твърдения (propositions)*

Обхваща езика на логиката, ***базирана на традиционните (пропозиционни) твърдения****.*
Представя се рамката на мислене чрез твърдения: правила и условия за получаване на резултат/заключение.

Твърдението е изявление (изказване) за предполагаем факт. То е или вярно или невярно, но никога и двете.

Примери*:*

Портокалът е плод.

Картофът е плод.

Портокалите не са единствените плодове.

Паролата е осемзнакова символна поредица.

…

Твърденията могат да бъдат изразени по различни начини.

Пример**:**

Пет е по-голямо от четири

Четири е по-малко от пет

5>4

Пропозиционна логика: Съюзи

Твърденията могат да бъдат свързани с различни съюзи*:*

$\neg$	negation	not
$\wedge$	conjunction	and
$\vee$	disjunction	or
$\Rightarrow$	implication	implies
$\Leftrightarrow$	equivalence	if and only if

Приоритет на операциите

Зад. Използвайки приоритета на операциите напишете еквивалентна скобова версия на израза $\neg p \wedge q \vee r \Leftrightarrow q \Rightarrow p \wedge r$

Отг. $(((\neg p) \wedge q) \vee r) \Leftrightarrow (q \Rightarrow (p \wedge r))$

С помощта на съюзите можем да изразим сложни твърдения:

Пример 1:

- $\neg$(jaffa cakes are biscuits)
- your cat is rich $\wedge$ your dog is good looking
- the economic recovery has started $\vee$ the minister is lying
- Jim is thirty-something $\Rightarrow$ Jim is under forty
- Jim is thirty-something $\Leftrightarrow$ Jim is under forty

Пример 2: Система за даване на бонуси …

Пропозиционна логика: синтаксис

Език:

Азбука: { p, q, r, ..., ¬, ^, v, $\Rightarrow$, $\Leftrightarrow$ }

Граматика:

изречение = “p” | “q” | “r” | ...
| “¬”, изречение
| “(”, изречение, “^“, изречение, “)”
| “(”, изречение, “v“, изречение, “)”
| “(”, изречение, “$\Rightarrow$“, изречение, “)”
| “(”, изречение, “$\Leftrightarrow$“, изречение, “)”

Ако няма двусмислие, обикновено скобите не се пишат.

Пропозиционна логика: семантика

Значението на пропозиционното изречение (wff) се дефинира като:

А) Всеки примитивен символ (**p, q, r**, ...) се интерпретира чрез твърдение, което се свързва със съответната си стойност на истинност.

Пр.: **p** - Днес е сряда.

Б) ***Истинността на сложните твърдения се дефинира единствено от истинността на отделните съставящи твърдения***.

таблици на истинност

Конюнкция

p	q	$p \wedge q$
t	t	t
t	f	f
f	t	f
f	f	f

Дизюнкция

p	q	$p \vee q$
t	t	t
t	f	t
f	t	t
f	f	f

Импликация (ако-то)

p	q	$p \Rightarrow q$
t	t	t
t	f	f
f	t	t
f	f	t

Еквивалентност
(тогава и само тогава)

p	q	$p \Leftrightarrow q$
t	t	t
t	f	f
f	t	f
f	f	t

Зад.: Дайте пример за твърдение, което е нито истина, нито лъжа.

Специални случаи:

1. Някои твърдения винаги се интерпретират като верни: **тавтология (tautology)**

Пр.: p $\vee$ ¬ p ; p => p ; p => (q => p))

2. Някои твърдения винаги се интерпретират като неверни: **противоречие (contradiction)**

Пр.: p $\wedge$ ¬ p ; p $\Leftrightarrow$ ¬ p ; ¬ (p => (q => p))

*3. T*върдение, което е нито истина, нито лъжа (**contingency**) – например случайност

Примери:

Как може да докажете еквивалентност за следното твърдение?

(¬ p => p ^ q) $\Leftrightarrow$ p

Покажете:
b) p ^ (p => q) => q

Пр.

S1 и S2 – сигнали на железопътни релси;

Как можем да осигурим безопасност на движението?

отг: (¬(S1 ^ S2))

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.

- **то е приложима част при използване на формалните методи в практиката.**

note*!*: “The trick of using formal methods effectively is to know *when proofs are* ***worth doing*** and ***when they are not***.”

Основни методи за доказателство, използвани за тези системи

- *Дедукция (Natural deduction*)

- *Доказателство чрез опровергаване (Proof by contradiction)*

- *Доказателство чрез анализ на случаи (Proof by case analysis)*

- *Разсъждение чрез равенства (Equational reasoning)*

Доказателство

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

Зад. Докажете: (p ∧ q) |- (p v q)

Отговор:

p ∧ q	premise
p	∧ - elimination
p ∨ q	∨ -introduction

Съществуват различни стилове (техники) на доказателство.

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 $\dfrac{premiss_1 \quad \ldots \quad premiss_n}{conclusion}$ [name]

Истинността на заключението е следствие на истинността на предпоставката !

***Начин*!** The rules come in two flavours: the *op*-elimination rule and the *op*-introduction rule. Using *these rules to introduce and eliminate* different operators, we can start from a set of propositions, or hypotheses, and derive another proposition.

If the set of hypotheses is empty, then we call the derived proposition a theorem.

Пропозиционни изчисления – правила за заключение

Пр.: Конюнкция

$$\frac{p \quad q}{p \wedge q}\ [\wedge\text{–intro}] \qquad \frac{p \wedge q}{p}\ [\wedge\text{–elim1}] \qquad \frac{p \wedge q}{q}\ [\wedge\text{–elim2}]$$

Дизюнкция – за “elim” използват се предположения

$$\frac{p}{p \vee q}\ [\vee\text{–intro1}] \qquad \frac{q}{p \vee q}\ [\vee\text{–intro2}] \qquad \frac{p \vee q \qquad \begin{matrix}[p]^{[i]} \\ r\end{matrix} \qquad \begin{matrix}[q]^{[i]} \\ r\end{matrix}}{r}\ [\vee\text{–elim}^{[i]}]$$

Импликация (предпоставката е по-силна от следствието)

$$\frac{\begin{matrix}[p]^{[i]} \\ q\end{matrix}}{p \Rightarrow q}\ [\Rightarrow\text{–intro}^{[i]}] \qquad \frac{p \Rightarrow q \quad p}{q}\ [\Rightarrow\text{–elim}]$$

Пропозиционни изчисления – правила за заключение

- Три правила за заключение за отрицанието:

$$\frac{\begin{array}{c}[p]^{[i]}\\ false\end{array}}{\neg p}\ [\neg\text{–intro}^{[i]}] \qquad \frac{p \quad \neg p}{false}\ [\neg\text{–elim}] \qquad \frac{\begin{array}{c}[\neg p]^{[j]}\\ false\end{array}}{p}\ [false\text{–elim}^{[j]}]$$

- Тавталогията $a \Leftrightarrow b$ кореспондира с две правила за извод:

$$\frac{b}{a}\ [a \Leftrightarrow b] \qquad \frac{a}{b}\ [a \Leftrightarrow b]$$

1. Доказателство чрез Дедукция

Доказателството се конструира отзад напред - използва се формулираната цел, за да се избере подходящо правилото за извод.

Пр: if have to prove **p** $\Rightarrow$ **q**, then reach for $\Rightarrow$ -introduction

Пр. Докажете, че ***p*** **^** ***q*** $\Leftrightarrow$ ***q*** **^** ***p***. т.е. можем да получим правило:

$$\frac{p \wedge q}{q \wedge p}$$

Много от правилата за заключение на Дедукцията са по-добре познати като техника за доказателство, с която те се асоциират:

=> - 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)

2. Доказателство чрез разглеждане на различни случаи

Стил: Прекъсваме доказателството на части, които след това комбинираме.

$$\begin{array}{c}[p]^{[i]} \\ r\end{array}$$

p е предположение, което може да бъде направено, за да докажем ***r***.

Зад. Да се докаже, че дизюнкцията е комутативна т.е. **p v q ⇔ q v p**:

$$\dfrac{p \vee q \qquad \dfrac{[p]^{[1]}}{q \vee p}\,[\vee\text{–intro2}] \qquad \dfrac{[q]^{[1]}}{q \vee p}\,[\vee\text{–intro1}]}{q \vee p}\,[\vee\text{–elim}^{[1]}]$$

Специални форми: Case analysis

- Derives from the rule for **v**-elimination

Find the cases **p** and **q** such that p **v** q holds. Then show that r holds in each case **p⇒r** and **q⇒r**. The justication of this proof technique relies on this property of implication:

(p ⇒ r) ^ (q ⇒ r) ≡ (p v q ⇒ r) **Case Analysis**

Notice the common and simpler version of this property when **q ≡ ¬p:**

(p ⇒ r) ^ (¬ p ⇒ r) ≡ r **Simple Case Analysis**

This proof technique generalizes to a case analysis of more than two cases in the obvious way

•Example:

show |a| > 0,

where |a| = a if a > 0, and |a| = -a if a < 0

3. Доказателства чрез еквивалентност/равенства

• Ще използваме и доказателство чрез равенства (**"equational reasoning"**). Техниката е много обща и с широко приложение:

"Equations are often a natural way of expressing mathematical knowledge and lie at the heart of a class of formal specification methods for software known as *algebraic specification*.

The properties of the equality relation allow us to reason by **'replacing equals by equals',** a very powerful and general technique." (от "Using Z")

• Съществуват и други стилове. Много от тях се изграждат на базата на конкретно практическо правило за извод (заключение).

3. Доказателство чрез равенство/еквивалентност

Увеличаваме символите на нашата формална система с още един:

Две изречения са еквивалентни (≡), ако и само ако имат равна стойност на истинност при всяка интерпретация.

Пр: **(p => q) ≡ (¬ p v q)**

Съществуват различни свойства на еквивалентността. Те могат да бъдат използвани при доказателства, реализирани чрез еквивалентност. Тези свойства включват *комутативност*, *асоциативност и дистрибутивност* на операциите, *закони на Де Морган, закон за двойното отрицание*.

Зад. Докажете чрез еквивалентности: ¬ (p ^ ¬q) ≡ (q v ¬p)

Д-во: ¬ (p ^ ¬q)

≡ DeMorgan

¬ p v ¬ ¬q

≡ Double Negation

¬ p v q

≡ Commutative

q v ¬ p

Qn: Why won't this form of proof work for all theorems?

Връзка между синтактичната и семантичната област

- Пропозиционното изчисление е *последователно* ако:
  - Всичко, което може да се докаже, е валидно/вярно:
  - Ако **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**
- Това ни позволява да преминаваме от единия в другия “свят”.
- Практически можем да заместваме ⇔ с ≡ .

За да покажем **|- P ≡ Q** можем да демонстрираме **|= P ≡ Q**

Предикатна логика (Predicate logic)

Езикът на пропозиционната логика ни позволява да формулираме твърдения за специфични обекти, но не ни дава възможност да формализираме изказвания от типа: “Всеки портокал е оранжев”. Тази формализация не разкрива вътрешната структура на твърдението.

Пр. Примери за универсални твърдения:

- Всеки студент има факултетен номер.

- Each student must hand in course work.
- Nobody knows the trouble I seen.
- Jim doesn’t know anybody who can sign his bail application.

Пр. Примери за екзистенциални твърдения

- I heard it from one of your friends.
- A mad dog has bitten Andy.
- Some people prefer logic.

Предикатна логика - формализация

Универсални твърдения - описват характеристика(и) на всеки обект от разглежданото множество. Заявяваме, че всички елементи от дадено множество **S** удовлетворяват свойство **P.**

Пр. Ако **S == {Larry, Joe, Moe}** искаме да заявим, че всички елементи на **S** са актьори, то:

(Larry is a stooge) ^ (Joe is a stooge) ^ (Moe is a stooge)

(stooge – актьор, клоун)

Алтерантивно:

създаваме шаблон (template): **Stooge( )**

Stooge(Larry)^ Stooge (Joe)^ Stooge (Moe)

Тогава:

Stooge( ) се нарича ПРЕДИКАТ.

Предикатна логика - формализация

За големи множества това записване е трудно (или невъзможно).

Алтернативно **използваме съкращение от вида**:

1) Универсално твърдение:

$$\forall \mathbf{x : S \bullet Stooge(x)}$$

което е съкращение на

Stooge(x1) ^ Stooge(x2) ^ Stooge(x3) ^ …

Предикатна логика - формализация

Екзистенциални твърдения: съществуването на елемент от множество **S**, който удовлетворява/притежава качество **P** :

2) ∃ **x : S ● P(x)**

което е съкращение на

P(x1) v P(x2) v P(x3) v ...

Предикатите могат да се разглеждат като *булеви функции*: когато се приложат към аргумент връщат стойност *истина* или *лъжа*.

Предикатите могат да имат *n* аргумента.

Пр: **P(x, y, z)**

Предикатна логика: Синтаксис

Азбука: Включва елементите на пропозиционната логика, но допълнително и определителите ∀ ∃ и ● .

∀ - универсален определител,

∃ - екзистенциален определител;

Граматика:

***Забележка*.! Изразите използват знака ● .**

Гнездови определители

∀ **x:A** ● (∀**y**:B ● (∃ **z:C** ● P(x,y,z)))

- обикновено се пропускат скобите,
- комбинират се близките променливи с някой определител и тогава:

∀ **x:A; y**:B; ∃ **z:C** ● P(x,y,z)

Определители и декларации в Z нотацията

В Z нотацията двата типа определители имат подобен синтаксис:

$$\Theta\, x : a \mid p \bullet q$$

където

Θ **определител (**$\forall$ **или** $\exists$**)**

x ограничена променлива

a обхват на x

p ограничение

q предикат

The existentially quantified predicate

$$\exists x : a \mid p \bullet q$$

is pronounced ‘there exists an x in a satisfying p, such that q’. The universally quantified predicate

$$\forall x : a \mid p \bullet q$$

is pronounced ‘for all x in a satisfying p, q holds’.

The optional constraint p restricts the set of objects under consideration; only those objects in a that satisfy p are to be considered. The constraint takes on the role of a conjunction or an implication, depending upon the quantifier concerned, as may be seen from the following equivalences:

$$(\exists x : a \mid p \bullet q) \Leftrightarrow (\exists x : a \bullet p \wedge q)$$

$$(\forall x : a \mid p \bullet q) \Leftrightarrow (\forall x : a \bullet p \Rightarrow q)$$

Пример:

Let *Friends* stand for the set of all your friends, and let *x told y* mean that *x* has told *y*.

$$\exists f : Friends \bullet f \ told \ me$$

Предикатна логика: Семантика

Включва повече отколкото пропозиционната логика. Основната идея:

$\forall x \bullet$ **P(x) is true iff P is true of all values of x,**

$\exists x \bullet$ **Q(x) is true iff Q is true for some value of x**

Предикатна логика: Изчисления

• Въвеждат се дедуктивни правила за въвеждане (intro) и елиминиране (elim) аналогично на пропозиционната логика:

- може да се използва дедуктивния стил като се предполага, че **q** е валидно за произволен елемент от **a**

Пр. За универсалния определител

$$\frac{[x \in a]^{[i]} \quad q}{\forall x : a \bullet q}\ [\forall\text{-intro}]^{[i]}$$

За екзистенциалния определител:

$$\frac{t \in a \quad p[t/x] \quad q[t/x]}{\exists x : a \mid p \bullet q}\ [\exists\text{-intro}]$$

Равенство (еднаквост)

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

Това е еквивалентно да имаме отделен (специален) предикат “equals”, но ние използваме по-общото означение.

Нови правила: към нашия дедуктивен апарат за доказателство.

Най-важното от тях е:

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

$$\frac{m = n,\ S(n)}{S[m/n]}$$

Примери:

$$(x \leq y + 2)[0 / x] \Leftrightarrow (0 \leq y + 2)$$

$$(\exists x : \mathbb{N} \bullet x \leq y + 2)[5 / y] \Leftrightarrow (\exists x : \mathbb{N} \bullet x \leq 5 + 2)$$

The one-point rule

Идеята за равенство позволява да се манипулира екзистенциалния определител.

Ако идентичността на две променливи е показана, то в рамките на определителя може да се заместят всички стойности на тази променлива, а определителят да се премахне.

Ako $\exists\, x : a \bullet p \wedge x = t$, **то one-point rule** води до следния еквивалентен израз:

$$(\exists\, x : a \bullet p \wedge x) \;=\; t \Leftrightarrow\; t \in a \wedge p[\, t /x \,]$$

Полза:

• ако знаем, че обектът *x* съществува с определена характеристика *p* и ако сме идентифицирали *x* като *t* , то правим заключение, че *p* е характеристика и на *t*.

• по този начин се премахва променливата и екзистенциалния определител без да променяме силата на предиката. (опростяване на израза)

Пр. Предикатът

$\exists\, n : \mathrm{N} \bullet 4 + n = 6 \wedge n = 2$

е еквивалентен на

$2 \in \mathrm{N} \wedge 4 + 2 = 6$