

5. Z нотацията - схеми- Резюме

Z нотацията е формален метод за спецификация на софтуерни системи(СС), който комбинира математическата теория на множествата и математическата логика с естествения език. Целта е да се създаде ясна, точна и разбираема спецификация на системата. Начин за представяне на **машини на състоянието**: Z глобални декларации; Z схеми (schemas)

Схемите са именувани записи на: • **тип**; • **декларация**; • **предикат**;

Схемите са ключов елемент на Z нотацията и представляват:

Всяка схема се състои от две части:

1. **Декларативна част**: списък на променливите
2. **Предикатна част**: ограничения върху стойностите на променливите

Форми на запис: **Хоризонтално** [declaration| predicate]; Вертикално;
 $\text{name} \triangleq [\text{declaration} | \text{predicate}]$

Схемите са осн. средство за структуриране на формалните спецификации: структури, които описват променливи, чиито стойности са ограничени по някакъв начин. Схемата предоставя възможност за описание на: **структури** от данни; **състояние** на системата; **операции**/действия на системата

Предимства на схемите: Повторното използване; създаването и поддържането а добър стил на спецификацията;

Еквивалентност - Две схеми са еквивалентни, ако имат **едни и същи променливи и едни и същи ограничения** върху тях. Някои от ограниченията могат да бъдат скрити в декларативната част.

Четимост: всяка декларация на отделен ред с конюнктивна връзка.

За системи без ограничения: предикатната част се пропуска.

Схема като тип: композира се тип от различни компоненти.

Компонентите на схемите се запазват не като позиция, а като имена. За да се отнесем до конкретен компонент използваме оператора за отделяне (selection operator) “**_.**”.

Нормализация: Декларативната част е редуцирана до уникална (единствена), канонична форма.

Ако S е нормализирана схема, то отрицанието и е: $\neg S$

Преименуване: Schema[**new/old**]

Общи схеми (Generic schemas): запазваме структурата на дад. схема, но декларираме различни типове за елементи \dot{y} - позволява **схемата да бъде по-гъвкава и да се използва с различни типове данни.**

Операции със схеми:

-Конюнкция (AND): комбинира условия, които трябва да бъдат изпълнени едновременно **- $S \wedge T$** - нова схема, в която **декларативните части на S и T са слети, а предикатните са конюнктивно свързани**

-Дизюнкция (OR): комбинира условия, където поне едно трябва да **бъде изпълнено**

-Отрицание (NOT): изразява противоположността на дад. условие

-Композиция: обединява различни схеми, за да се изрази взаимодействието между тях. Ако OpOne и OpTwo са две схеми на операции, всяка от които включва недекорирано и декорирано състояние State, тогава композицията OpOne ; OpTwo описва промяната в състоянието State, която е резултат от действието на операция OpOne, последвана от операция OpTwo

Композиция на схеми • Схемите, които се композират, описват различни операции на едно и също състояние. • Така при композицията на схеми за всяко декорирано име в едната операция трябва да съществува не декорирано име във втората операция. • Въвежда се междинно състояние State. • При композицията на схеми, състоянието „след“ на първата операция се идентифицира със състоянието „преди“ на втората операция.

Всяка операция, която променя състоянието на системата, може да се разглежда като релация между обектите на схемата - операцията изразява връзката между обектите на състоянието преди и след прилагането ѝ.

Декорация: • За да се опише операция върху състояние се правят две копия на схемата: състоянието преди и състоянието след операцията. • Различимост на двете състояния – чрез декорация на компонентите на втората схема и модификация на предикатната ѝ част

? – вход o! – изход

Δ- операцията води до промяна в състоянието

Ξ - декларация, че състоянието не се променя /търсене, справка/

Предимства на подхода: Модулност; Reuse; Възможност за абстрактно описание; Математическа точност; Лесно композиране на системни елементи

Подходът има известни ограничения: Не се справя добре с **глобални променливи**; Трудности при работа с **безкрайни състояния**; Ограничения при работа с **параметри и изключения**

Z е особено полезна за **прости машини на състоянието** и системи, които изискват висока точност и формалност.

Z нотацията и машините на състоянието:

- Инвариантата на състоянието се приема да е валидна по време на операциите –so don't have to prove as it is maintained
- Операциите извън предварителните условия водят до “хаос” –so always want final precondition to be "true"
- **Pre- и post-условията не са разгран. синтактично** –but may be derived
- Фокусира се **върху състояния**, а не върху трасета
- Входовете, изходите, изключенията са просто нотирани конвенции