

Západočeská univerzita v Plzni  
Fakulta aplikovaných věd  
Katedra informatiky a výpočetní techniky



KIV/UZI  
ZNALOSTNÍ SYSTÉM PRO ŘEŠENÍ LINEÁRNÍCH ROVNIC

Faiz Suleimanov  
13. prosince 2024

# 1 Formulace úlohy

## 1.1 Specifikace

Cílem úlohy je vytvořit program, který řeší množinu lineárních podmínek postupnou eliminací proměnných. Vstupními daty je množina  $N$  proměnných a množina  $M$  lineární rovností nad těmito proměnnými. Program má implementovat symbolické řešení podobné metodám Gaussovy eliminace nebo Fourierovy eliminace.

## 1.2 Výchozí podmínky, vstupní data

- **Počet proměnných:**  $N$ .
- **Počet podmínek:** Množina  $M$  lineárních rovností a nerovností. Podmínky mohou být zadány uživatelem v textové podobě, např.:

$$2 * x + 3 * y - z = 5$$

- **Formát vstupních dat:**
  - Proměnné zadávané odděleně, např.  $x$ ,  $y$ ,  $z$ .
  - Rovnice zadávány uživatelem ve standardní analytické podobě, např.  $2*x + y = 5$ .

## 1.3 Požadavky na výstupy

- **Krokové řešení:** Program musí zobrazovat postupný proces eliminace proměnných, včetně transformací jednotlivých rovnic a nerovnic.
- **Výsledný formát:** Po dokončení eliminace musí být systém převeden na jednodušší tvar (např. trojúhelníkovou formu v případě rovnic), a uživatel musí obdržet výsledné hodnoty proměnných nebo informace o neřešitelnosti systému.
- **Výstupní médium:** Výstup je zobrazován v konzoli (textové prostředí), kde jsou jednotlivé kroky prezentovány přehledně pro snadné sledování.
- **Podpora vysvětlení:** Každý krok eliminace by měl být doprovázen stručným vysvětlením, aby uživatel pochopil aplikované transformace.

# 2 Analýza úlohy

## 2.1 Rozbor možných postupů

Při řešení úlohy bylo možné zvažovat různé přístupy a metody pro dosažení požadovaných výsledků za daných podmínek:

- **Ruční implementace algoritmu eliminace proměnných:** Tento přístup zahrnuje implementaci algoritmu eliminace proměnných (např. Gaussovy eliminace) přímo v kódu. Program by musel ručně manipulovat rovnicemi, odstraňovat proměnné a postupně transformovat systém na řešitelný tvar.
- **Využití knihoven pro matematické výpočty (např. SymPy):** Použití již existujících knihoven, které poskytují nástroje pro práci s rovnicemi a jejich manipulaci. Tyto knihovny umožňují snadné řešení lineárních rovnic a obsahují optimalizované algoritmy pro eliminaci proměnných.

- **Použití znalostního systému:** Tento přístup zahrnuje vytvoření systému, který uchovává fakta (rovnice) a pravidla (transformace rovnic). Pomocí logického odvozování systém aplikuje pravidla na danou množinu rovnic a postupně odstraňuje proměnné.
- **Použití expertního systému:** Expertní systém by simuloval proces řešení rovnic podobný tomu, jak by ho řešil lidský expert. Tento přístup by poskytoval vysvětlení jednotlivých kroků, ale implementace by byla složitější a náročnější na zdroje.

## 2.2 Výběr nejvhodnější alternativy

Po zvážení výhod a nevýhod jednotlivých přístupů bylo rozhodnuto použít znalostní systém s podporou knihovny SymPy. Tento výběr byl zdůvodněn následujícími faktory:

- **Modularita a rozšiřitelnost:** Znalostní systém umožňuje oddělené zpracování faktů (rovnice) a pravidel (operace s rovnicemi). To zajišťuje snadnou rozšiřitelnost systému o další pravidla nebo metody řešení.
- **Využití osvědčených nástrojů:** Knihovna SymPy poskytuje optimalizované algoritmy pro manipulaci s rovnicemi a jejich řešení. To eliminuje potřebu implementovat základní matematické operace ručně.
- **Transparentnost a srozumitelnost:** Znalostní systém umožňuje zobrazit jednotlivé kroky výpočtu, což odpovídá požadavku na transparentní řešení. Uživatel může sledovat postup eliminace proměnných a pochopit jednotlivé kroky.
- **Složitost implementace:** Ruční implementace algoritmů by byla časově náročná a složitá. Naopak kombinace znalostního systému a knihovny SymPy poskytuje robustní řešení s menšími nároky na implementaci.
- **Podpora vysvětlení:** Znalostní systém umožňuje snadno přidat podporu pro vysvětlení jednotlivých kroků, což zvyšuje edukativní hodnotu programu.

Tento přístup byl vybrán jako nejvhodnější, protože kombinuje efektivitu, transparentnost a modularitu, což jsou klíčové požadavky zadání.

## 3 Popis algoritmu řešení

### 3.1 Formulace algoritmu

Algoritmus řešení soustav lineárních rovnic pomocí znalostního systému je rozložen na následující dílčí kroky a moduly:

1. **Inicializace programu:** Program se spustí a uživatel je vyzván k zadání proměnných a rovnic.  
Modul: `UserInterface`.
2. **Vstup proměnných:** Uživatel zadá seznam proměnných, například  $x$ ,  $y$ ,  $z$ .  
Modul: `DataManager`.
3. **Vstup rovnic:** Uživatel zadá rovnice ve formátu, kde všechny proměnné jsou na levé straně a konstanty na pravé, například  $2x + y - z = 5$ .  
Modul: `DataManager`.

4. **Eliminace proměnných:** Program aplikuje na systém pravidla eliminace proměnných, například Gaussovu eliminaci. Pravidla jsou implementována jako součást znalostního systému.

Modul: KnowledgeBase, Rules.

5. **Aplikace pravidel:** Znalostní systém postupně aplikuje pravidla eliminace, modifikuje rovnice a přidává nové fakty do znalostní báze.

Modul: KnowledgeBase.

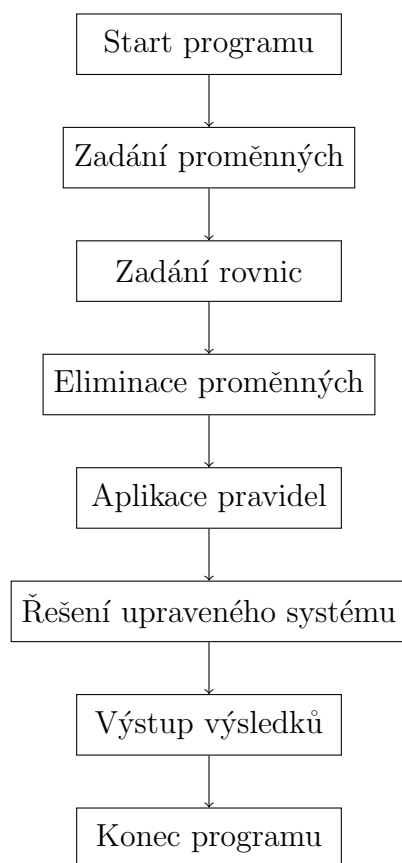
6. **Řešení upraveného systému:** Po eliminaci všech proměnných je systém převeden na trojúhelníkovou formu a vyřešen pomocí zpětného dosazení.

Modul: Solver.

7. **Prezentace výsledků:** Program zobrazí uživateli konečný výsledek a poskytne vysvětlení jednotlivých kroků.

Modul: UserInterface.

## 3.2 Schématické znázornění návazností



Obrázek 1: Schématické znázornění algoritmu řešení

## 4 Popis programu

### 4.1 Volba programovacího jazyka a technologie

Pro implementaci programu byl zvolen jazyk Python z následujících důvodů:

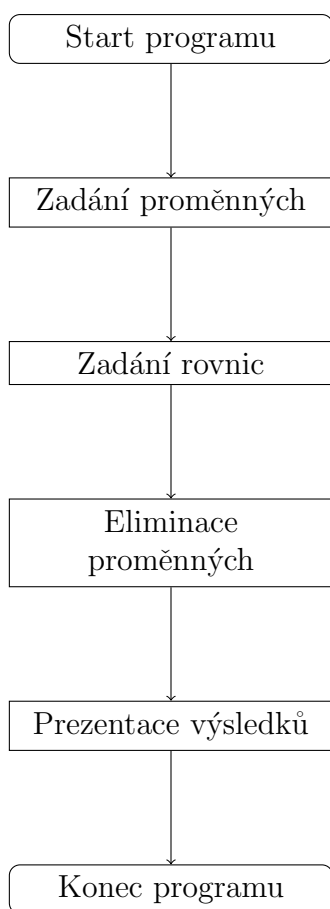
- **Čitelnost a srozumitelnost:** Python je známý svou jednoduchou syntaxí, která umožňuje snadnou údržbu a rozšiřování kódu.
- **Bohatá knihovna pro matematické výpočty:** Knihovna `SymPy` poskytuje nástroje pro manipulaci s algebraickými výrazy, řešení rovnic a eliminaci proměnných.
- **Flexibilita:** Python umožňuje snadno kombinovat různé přístupy, například použití znalostních systémů a pokročilých algoritmů.
- **Podpora pro vývoj znalostních systémů:** Python nabízí flexibilní datové struktury a snadnou práci s pravidly a fakty v kontextu znalostních systémů.

## 4.2 Volba datových typů a struktur

V programu byly použity následující datové typy a struktury:

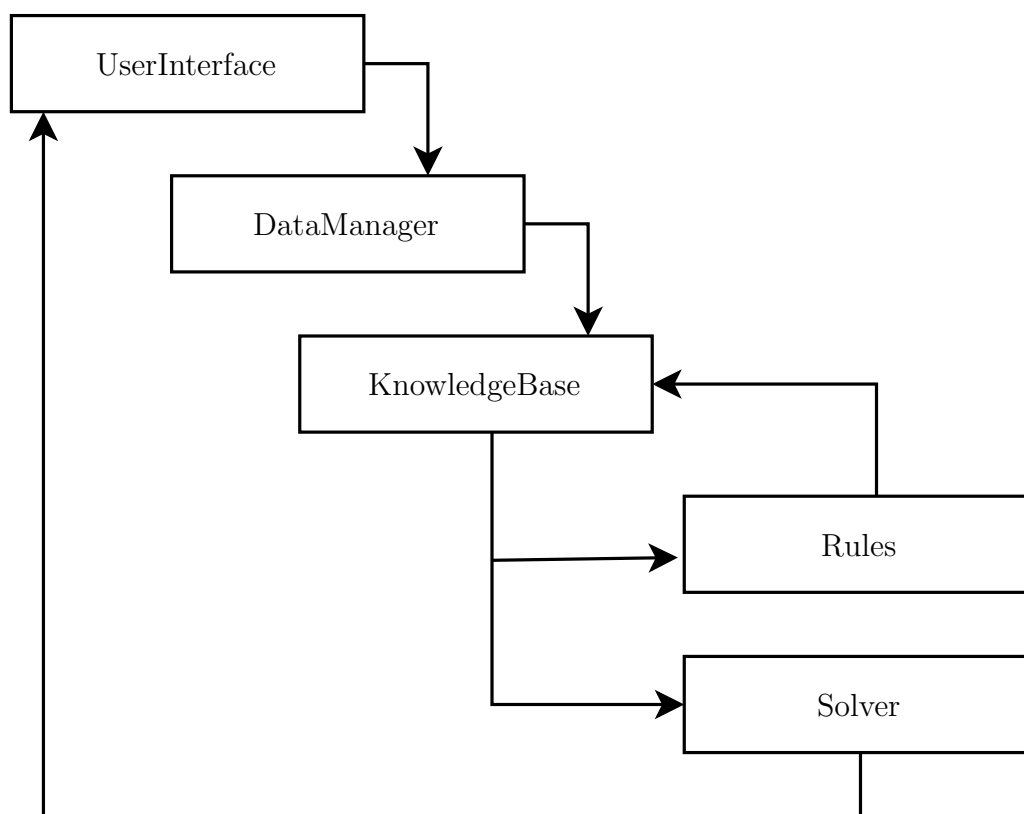
- **Seznamy (list):** Použity pro uložení rovnic a pravidel. Seznam umožňuje dynamickou manipulaci s položkami.
- **Symbolické výrazy (SymPy):** Proměnné a rovnice jsou reprezentovány pomocí symbolických výrazů z knihovny `SymPy`, což umožňuje snadné manipulace s algebraickými objekty.
- **Pravidla (custom class):** Každé pravidlo eliminace proměnné je implementováno jako samostatná třída, což zajišťuje modularitu a čitelnost.
- **Znalostní báze (KnowledgeBase):** Fakta a pravidla jsou uchovávány ve znalostní bázi, která umožňuje aplikaci pravidel a eliminaci duplicit.

### 4.3 Blokové schéma programových kroků



Obrázek 2: Blokové schéma programových kroků

## 4.4 Vývojové diagramy modulů



Obrázek 3: Vývojové diagramy modulů a jejich propojení

## 4.5 Popis použitých symbolů

- $x, y, z$ : Proměnné systému rovnic.
- $=$ : Operátory rovností.
- `Eq`: Symbolická reprezentace rovnic v knihovně SymPy.
- `linsolve`: Funkce knihovny SymPy pro řešení lineárních rovnic.
- `KnowledgeBase`: Datová struktura pro uchovávání faktů a pravidel.

## 4.6 Specifické vlastnosti modulů

- **DataManager**: Spravuje proměnné a rovnice, komunikuje se znalostní bází.
- **KnowledgeBase**: Uchovává fakta a pravidla, zajišťuje jejich aplikaci a eliminaci duplicit.
- **Rules**: Implementuje pravidla pro manipulaci s rovnicemi, například eliminaci proměnných.
- **Solver**: Řeší finální systém rovnic pomocí Gaussovy eliminace.
- **UserInterface**: Zajišťuje komunikaci s uživatelem, sbírá vstupy a prezentuje výsledky.

## 5 Popis obsluhy programu

### 5.1 Instalace a spuštění

Program je vytvořen v jazyce Python, a proto vyžaduje, aby byl Python nainstalován na vašem počítači. Pro instalaci a spuštění postupujte podle následujících kroků:

1. Stáhněte si zdrojové soubory programu a rozbalte je do preferované složky.
2. Ujistěte se, že máte nainstalovanou vhodnou verzi Pythonu (doporučuje se verze 3.8).
3. Otevřete příkazový řádek nebo terminál a přejděte do složky se soubory programu.
4. Pro instalaci potřebných knihoven spusťte následující příkaz:

```
pip install -r req.txt
```

5. Spusťte program příkazem:

```
python main.py
```

Pokud jsou všechny kroky provedeny správně, program bude spuštěn a připraven k použití.

### 5.2 Zadávání vstupních dat

Pro zadání vstupních dat následujte instrukce zobrazené přímo v uživatelském rozhraní programu:

- **Zadání proměnných:** Při spuštění programu budete vyzváni k zadání názvů proměnných oddělených mezerou. Například:

```
x y z
```

Zadejte názvy proměnných odpovídající vašemu problému. V případě potřeby můžete zadat příkaz `help`, abyste zobrazili nápovědu.

- **Zadání rovnic:** Po zadání proměnných budete vyzváni k zadání rovnic. Rovnice musí být ve formátu:

$$2 * x + 3 * y - z = 5$$

Ujistěte se, že všechny proměnné jsou na levé straně rovnice a konstanty na pravé straně. Pro ukončení zadávání rovnic napište `done`.

- **Validace vstupů:** Program automaticky zkontroluje správnost zadaných proměnných a rovnic. V případě chyby budete upozorněni a budete mít možnost opravit svůj vstup.



## 6 Rozbor výsledků, zhodnocení

### 6.1 Rozbor dosažených výsledků

Výsledkem této práce je program, který efektivně řeší systémy lineárních rovnic a nerovnic pomocí znalostní báze a aplikace pravidel eliminace. Program splňuje zadání tím, že umožňuje:

- Zadávání proměnných a rovnic uživatelem ve srozumitelném formátu.
- Postupnou eliminaci proměnných pomocí pravidel z modulu Rules.
- Řešení systému rovnic v modulu Solver pomocí metody Gaussovy eliminace.
- Interaktivní uživatelské rozhraní, které uživateli umožňuje sledovat průběh řešení a dostat vysvětlení jednotlivých kroků.

Program je navržen modulárně, což umožňuje jeho snadnou rozšiřitelnost a údržbu. Díky použití znalostní báze je možné přidávat nové typy pravidel nebo rozšířit možnosti zpracování vstupních dat. Kladem řešení je rovněž jeho přehlednost a snadné použití díky implementovanému uživatelskému rozhraní.

### 6.2 Kritika nedostatků

I přes dosažené výsledky je v programu několik oblastí, které by mohly být vylepšeny:

- **Řešení složitějších nerovnic:** Program se zaměřuje primárně na řešení rovnic. V budoucnu by mohl být rozšířen o pokročilejší algoritmy pro řešení složitých nerovnic.
- **Uživatelské rozhraní:** Přestože je rozhraní interaktivní, mohlo by být vizuálně atraktivnější a obsahovat více vizualizací (například grafické znázornění eliminace proměnných).

Navrhovaná vylepšení by mohla zvýšit univerzálnost programu, zlepšit jeho použitelnost a umožnit jeho aplikaci na širší spektrum problémů.

## 7 Závěr

### 7.1 Splnění zadání

Hlavním cílem této práce bylo navrhnout a implementovat systém, který efektivně řeší lineární podmínky pomocí eliminace proměnných, přičemž důležitou součástí zadání bylo zvolit vhodný přístup – znalostní systém nebo expertní systém. Na základě analýzy zadání a požadavků bylo rozhodnuto implementovat znalostní systém. Tento přístup umožnil modulární návrh programu a snadné přizpůsobení pravidel specifickým potřebám řešení.

Program splňuje všechny klíčové požadavky:

- Implementace znalostní báze pro uložení a správu faktů a pravidel.
- Modulární architektura, která umožňuje snadnou rozšiřitelnost a přizpůsobitelnost.
- Uživatelsky přívětivé rozhraní, které interaktivně vede uživatele při zadávání proměnných a rovnic.
- Efektivní eliminace proměnných prostřednictvím pravidel definovaných v modulu Rules.

- Řešení rovnic pomocí Gaussovy eliminace v modulu **Solver**.

Volba znalostního systému se ukázala jako správná, protože umožnila jasnou definici pravidel a jednodušší rozšiřování systému v budoucnu. Díky modularitě programu je systém přehledný a snadno udržitelný.

## 7.2 Možnosti dalšího rozvoje

Projekt vytvořil pevný základ pro další rozvoj a rozšíření funkcionality systému. Zde jsou některé možné oblasti zlepšení:

- **Podpora různých typů systémů:** Přidání podpory pro nerovnice složitějších struktur.
- **Rozšíření znalostní báze:** Možnost přidávat další pravidla, která by umožnila řešení více typů rovnic nebo integraci s dalšími matematickými metodami.
- **Vylepšení uživatelského rozhraní:** Rozšíření vysvětlujících funkcí, například o interaktivní vizualizace průběhu eliminace proměnných.
- **Optimalizace výkonu:** Vylepšení algoritmů pro řešení větších systémů rovnic s cílem zvýšit rychlost a efektivitu.
- **Vizualizace procesu:** Zavedení grafických výstupů, které by znázorňovaly průběh eliminace proměnných a řešení systému rovnic.

Tento projekt poskytuje základ pro akademické i praktické aplikace. Modularita a znalostní přístup umožňují flexibilní rozšiřování systému v souladu s novými požadavky a technologiemi.