



Semestrální práce z KIV/TI

Logické řízení - sanitace nádrží

Lukáš Runt (A20B0226P)

lrunt@students.zcu.cz

Miroslav Vdoviak (A20B0268P)

miravdov@students.zcu.cz

24. ledna 2022

Obsah

Obsah	1
1 Zadání	3
2 Analýza úlohy	4
2.1 Konečný automat	4
2.1.1 Definice	4
2.1.2 Analýza	4
3 Automatový model	5
3.1 Stavy	5
3.2 Snímače	5
3.3 Řídící signály	6
3.4 Řízení operátora	6
3.5 Přejchodový graf	6
3.6 Chybové stavy	6
4 Implementace	7
5 Uživatelská příručka	8
5.1 Spuštění programu	8
5.2 Ovládání aplikace	9
5.2.1 Výčet ovládacích kláves	9
6 Závěr	9
Seznam obrázků	10
7 Reference	10

1 Zadání

Z3. Logické řízení – sanitace nádrží:

Na <http://home.zcu.cz/~vais/> v rozšiřujícím materiálu o konečných automatech prostudujte kapitoly Logické řízení a Principy softwarové implementace.

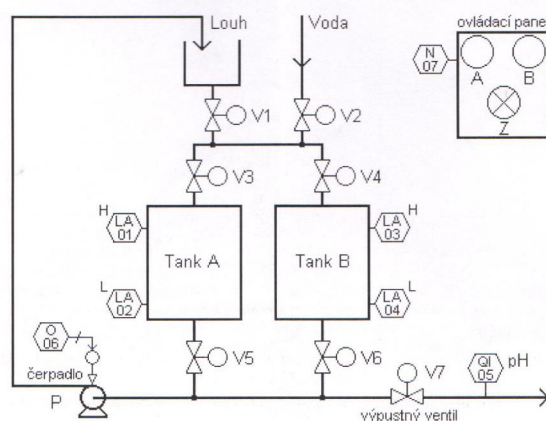
Navrhnete konečněautomatový model řídicího systému níže popsaného zařízení.

Sanitace pivovarských tanků se provádí ve dvou fázích. V první fázi se přepustí roztok louhu ze zásobní nádrže do tanku. Jakmile dosáhne hladina v tanku maxima (signál LA011 nebo LA031), tzn. že dosáhne čidla LA/01 resp. LA/03, celý obsah tanku se přečerpá pomocí čerpadla (spuštění signálem P1, vypnutí signálem P0) zpět do zásobní nádrže. Ve druhé fázi se tank naplní vodou a poté se otevře výpustný ventil (otevření ventilu i signálem Vi1, uzavření signálem Vi0) a tank je proplachován vodou tak dlouho, dokud pH na výtoku neklesne pod zadanou mez (signál Q0). Celý cyklus sanitace je ukončen když hladina v nádrži klesne pod dolní mez (LA020 nebo LA040), tzn. že klesne pod čidlo LA/02 resp. LA/04. Operátor spouští sanitaci tanku A nebo B stisknutím tlačítka A (signál A) nebo B (signál B). Jestliže tank není prázdný, nelze nezačínat sanitaci, ale výstupním signálem Z1 rozsvítit signální žárovku. Žárovka má svítit do té doby, dokud není příslušný tank vyprázdněn ručním ovládáním.

Model řídicího automatu realizujte softwarově na základě principů popsaných v materiálu. Všechny signály od čidel modelujte vstupy od klávesnice, řídicí signály a informaci o stavu vypisujte textově na obrazovku.

Automat popište přechodovým grafem. Pro zakreslení přechodového grafu použijte software JFLAP (<https://www.jflap.org/>).

Technologické schéma:



Obrázek 3: Zadání

2 Analýza úlohy

2.1 Konečný automat

2.1.1 Definice

Definice konečného automatu

$$A = (Q, \Sigma, \delta, q_0, F)$$

Q - konečná neprázdná množina stavů

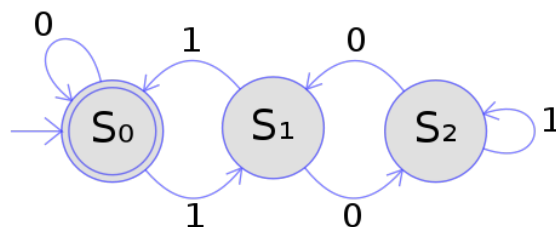
Σ - konečná neprázdná množina vstupních symbolů

$\delta - Q \times \Sigma \rightarrow$ přechodová funkce

$q_0 \in Q$ - počáteční stav

$F \subseteq Q$ - množina koncových stavů

Konečný automat je teoretický výpočetní model používaný v informatice pro studium formálních jazyků. Jedná se o počítač s konečným počtem stavů. Automat postupně čte symboly, které přicházejí ze vstupu a následně pomocí nich přechází mezi jednotlivými stavy. Počet stavů je konečný. Tento jednoduchý počítač nemá žádnou paměť a jedině, co si pamatuje je aktuální stav. V praxi se používá pro vyhodnocování regulárních výrazů.



Obrázek 2: Příklad konečného automatu se třemi stavy

2.1.2 Analýza

Automat, který budeme vytvářet by měl obsahovat

- 10 stavů (viz. stavy)
- 8 snímačů (viz. snímače)
- 6 řídicích signálů (viz. signály)

- 2 spušťecí tlačíka a kontrolní žárovku (viz. řízení operátora)

Konečný automat budeme realizovat v programovacím jazyce java a pro grafické rozhraní použijeme standardní knihovny javy `java.swing` a `java.awt`. Uživatel bude ovládat aplikaci pomocí klávesnice. Při stisku tlačítka A by se měla spustit sanitaze tanku. Při dovršení maxima se spustí signál LA011 a pomocí čerpadla P se nádrž vypustí zpět do zásobní nádrže. V druhém kroku se tank napustí vodou a proplachuje se do té doby dokud pH výtoku neklesne pod dolní mez. Obdobně by měl program fungovat i pro signál B. Pokud by uživatel chtěl zavolat sanitaci tanku, který není prázdný zobrazí se mu chybové hlášení a měla by se rozsvítit žlutá chybová kontrolka. Grafické naplňování tanků plánujeme naimplementovat pomocí časové, kdy budeme za krátký časový úsek zvětšovat či zmenšovat o pár pixelů obdelník, který budeme vykreslovat uvnitř tanku a tím se navodí iluze napouštění a vypouštění tanku. Teoreticky bychom si mohli předpřipravit obrázky formátu `.gif`, ale myslím si, že první řešení je mnohem elegantnější.

3 Automatový model

3.1 Stavý

STAV 0 - Systém není v činnosti
 STAV 1 - Tank A se napouští lihem
 STAV 2 - Tanku A se přečerpává čerpadlem
 STAV 3 - Tank A se plní vodou
 STAV 4 - Tank A se proplachuje dokud není ph v normálu
 STAV 5 - Tank A se vypouští
 STAV 6 - Tank B se napouští lihem
 STAV 7 - Tanku B se přečerpává čerpadlem
 STAV 8 - Tank B se plní vodou
 STAV 9 - Tank B se proplachuje dokud není ph v normálu
 STAV 10 - Tank B se vypouští

3.2 Snímače

LA011 - Hladina dosahuje maxima tanku A
 LA010 - Hladina nedosahuje maxima tanku A
 LA021 - Hladina dosahuje minima tanku A
 LA020 - Hladina nedosahuje minima tanku A
 LA031 - Hladina dosahuje maxima tanku B
 LA030 - Hladina nedosahuje maxima tanku B

LA041 - Hladina nedosahuje minima tanku B
 LA040 - Hladina dosahuje minima tanku B

3.3 Řídící signály

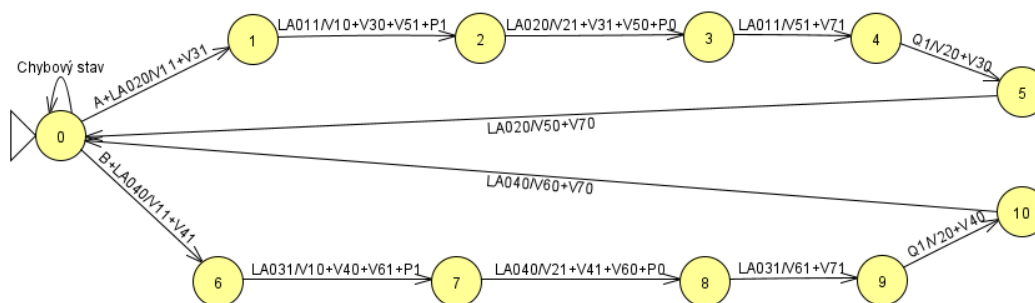
P0 - Čerpadlo vyplé
 P1 - Čerpadlo zaplé
 Vi0 - Ventil i zavřen
 Vi1 - Ventil i otevřen
 Q0 - Ph nad požadovanou mezí
 Q1 - Ph pod požadovanou mezí

3.4 Řízení operátora

A - Sanitace tanku A
 B - Sanitace tanku B
 Z - Žárovka

3.5 Přechodový graf

Na následujícím obrázku je přechodový graf konečného automatu.



Obrázek 3: Přechodový graf automatu

3.6 Chybové stavy

- Uživatel chce začít sanitací tanku, který není prázdný
- Uživatel chce začít sanitací tanku v době, kdy se sanitarizuje jiný tank
- Uživatel chce míchat líc s vodou

- Uživatel chce vylévat líh do potrubí
- Uživatel chce čerpat vodu do nádrže s lihem

4 Implementace

Program je implementován v jazyce Java. Při vytváření programu bylo zvažováno, zda nebudou nakresleny obrázky v nějakém z grafických editorů, které se budou postupně měnit. Toto nám, ale přišlo krajně nepraktické, a proto byla zvolena implementace s pomocí knihovny **Java Swing** a vektorovou grafikou, která se bude překreslovat podle aktuálního stavu.

Tato knihovna vytváří okno v metodě `main()`, které má pevně stanovenou velikost. O obsah okna se stará třída `DrawingPanel`, která je zodpovědná o vzhled modelu. Tato třída obsahuje i logiku automatu. Při spuštění se vše inicializuje tak, aby se vykreslil automat ve stavu 0, tedy v nečinnosti.

Každá komponenta má svoji proměnnou, která uchovává její stav. Tento stav se poté promítá do výsledného grafického výstupu. Stav komponent jsou reprezentovány barvami. Zelená znamená logickou 1 a červená logickou 0. Toto neplatí u žárovky, která je při logické 1 žlutá a při logické 0 šedá. Komponenta tanku je pak řešena tak, že uchovává data o její plnosti v procentech a druh jejího obsahu (voda, líh, vzduch), který je řešen pomocí enumu `Napln`.

Stavy modelu jsou realizovány pomocí proměnné `stav` datového typu `Integer`. V programu je zabudovaný `timer`, který každých 50ms spustí metodu `zjistiStav()`, která zjistí stav a následně podle toho spustí metodu příslušející aktuálnímu stavu modelu. Každý stav pak vykonává nějakou pro něj typickou činnost (plnění tanku, čerpání lihu z nádrže, atd.), dokud není splněna podmínka (např.: pro stav 1 snímač LA01 je v logické 1), pokud je potřebná podmínka splněna, model se přesouvá do dalšího stavu (uložení do proměnné `stav` příslušné hodnoty). metody stavů jsou nazvány `stav[A|B][1|2|3|4|5]`. Pro aktualizování stavu snímaču slouží metoda `aktualizujStav()`, která kontroluje, zda má snímač posílat logickou 1 nebo 0.

Reakce na uživatele je vyřešena pomocí metody `KeyListener()`, rozeznávající, která klávesa byla stisknuta nebo uvolněna. Díky této metodě lze spoštět jednotlivé sanitazace a manuálně ovládat model, avšak manuální ovládání je povoleno jen, když je konečný automat ve stavu 0. K zjištění, zda je ovládání povoleno slouží metoda `isManualniPovoleno()`. Jestliže chce uživatel spustit akci, která nemůže být spuštěna z důvodu chybového stavu, je zavolána metoda `vypisChybu()`, která vypíše ve vyskakovacím okně chybu a

rozsvítí žárovku, která zhasne po zavření okna, které se zobrazilo.

5 Uživatelská příručka

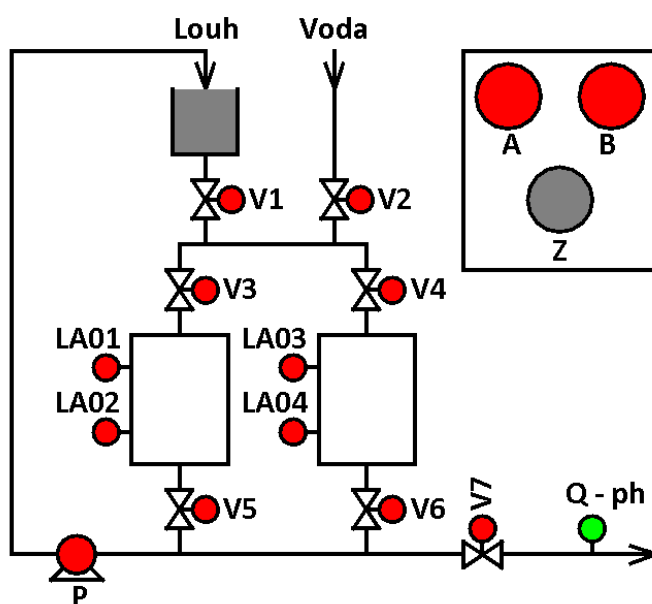
5.1 Spuštění programu

Aplikace se spoštl pomocí příkazu v příkazové řádce. Před zadáním příkazu se musíme ujistit, zda se nacházíme ve stejné složce, jako jar soubor, který se chystáme spustit (`semestralkaTI.jar`). Aplikaci poté spustíme pomocí příkazu: `java -jar semestralkaTI.jar 4`. Pro spuštění je předpokladem mít nainstalovanou Javu verze nejméně 11. Odkaz ke stažení Javy 11:

```
d:\TI>java -jar semestralkaTI.jar
```

Obrázek 4: Příklad spuštění

Pokud se program podaří spustit zobrazí se model sanitazace tanků 5.



Obrázek 5: Vzhled aplikace po spuštění

5.2 Ovládání aplikace

Po spuštění se zobrazí model ve stavu 0. Červená barva znamená logickou 0, tedy ventil je zavřené, tlačítko není stlačeno, ph není v požadované mezi, čerpadlo nečerpá líh, hladina v tanku není výš než snímač. Zelená barva znamená naopak logickou 1, tedy ventil je otevřený, tlačítko je stlačeno, atd. Žárovka má své barvy a to šedou pokud nesvítí a žlutou pokud svítí. Voda je znázorněna modrou barvou a líh barvou šedou.

Aplikace se ovládá pomocí klávesnice. Uživatel má k dispozici ovládací panel (Spouštění sanitizace nádrží) a manuální ovládání, které zahrnuje ovládání jednotlivých ventilů a čerpadla. Při implementaci byla snaha o intuitivní ovládání, tedy ventily se ovládají pomocí jejich čísla, ostatní prvky se ovládají pomocí písmenka, kterým je daný prvek pojmenovaný. Podrobný výčet ovládání je uveden níže 5.2.1.

Aplikace počítá s neobvyklým zacházením. Jsou tedy ošetřeny stavy, při kterých by mohlo dojít k chybě. Při chybovém stavu vyskočí na uživatele upozornění a v modelu se rozsvítí žárovka. Výčet chybových stavů lze najít zde: 3.6

5.2.1 Výčet ovládacích kláves

- A - Spuštění sanitizace tanku A
- B - Spuštění sanitizace tanku B
- P - Manuální spuštění čerpadla
- 1 - Manuální otevření ventilu 1
- 2 - Manuální otevření ventilu 2
- 3 - Manuální otevření ventilu 3
- 4 - Manuální otevření ventilu 4
- 5 - Manuální otevření ventilu 5
- 6 - Manuální otevření ventilu 6
- 7 - Manuální otevření ventilu 7

6 Závěr

Výsledkem semestrální práce je funkční program simulující konečněautomatový model řídicího systému sanitace pivovarských tanků. Na práci by šla udělat nějaká vylepšení, například umožnit uživateli měnit velikost okna, ovšem myslíme si, že výsledek naší práce vypadá celkem solidně. V rámci úlohy jsme si vyzkoušeli napsat konečný automat, což byl také cíl. Byl to pro

nás zážitek, který nás studijně obohatil a posunul o krok blíže k praktickým aplikacím teoreticky získaných vědomostí.

Seznam obrázků

1	Zadání	3
2	Příklad konečného automatu se třemi stavy	4
3	Přechodový graf automatu	6
4	Příklad spuštění	8
5	Vzhled aplikace po spuštění	8

7 Reference

- [1] Wikipedia, “Konečný automat — Wikipedia, the free encyclopedia,” <http://cs.wikipedia.org/w/index.php?title=Kone%C4%8Dn%C3%BD%20automat&oldid=20327947>, 2022, [Online; accessed 24-January-2022].