

# Západočeská univerzita v Plzni

## Fakulta aplikovaných věd

Automat na řízení křižovatky

Jméno: Xuan Toan

Příjmení: Dinh

Osobní číslo: A19B0027P

Datum odevzdání: 12.12.2020

## Obsah

1. Zadání.....	3
2. Analýza úlohy .....	3
a. Teoretický rozbor .....	3
b. Řešení.....	3
3. Automatový model .....	4
4. Implementace .....	7
5. Uživatelská příručka .....	7
6. Závěr.....	7

## 1. Zadání

- Křižovatka
- řízení složité silniční křižovatky (alespoň šest semaforů pro auta, semaforey pro chodce)
- činnost automatu musí jít nějak ovlivnit (tlačítka pro chodce, detekci čekajících aut, ...)

## 2. Analýza úlohy

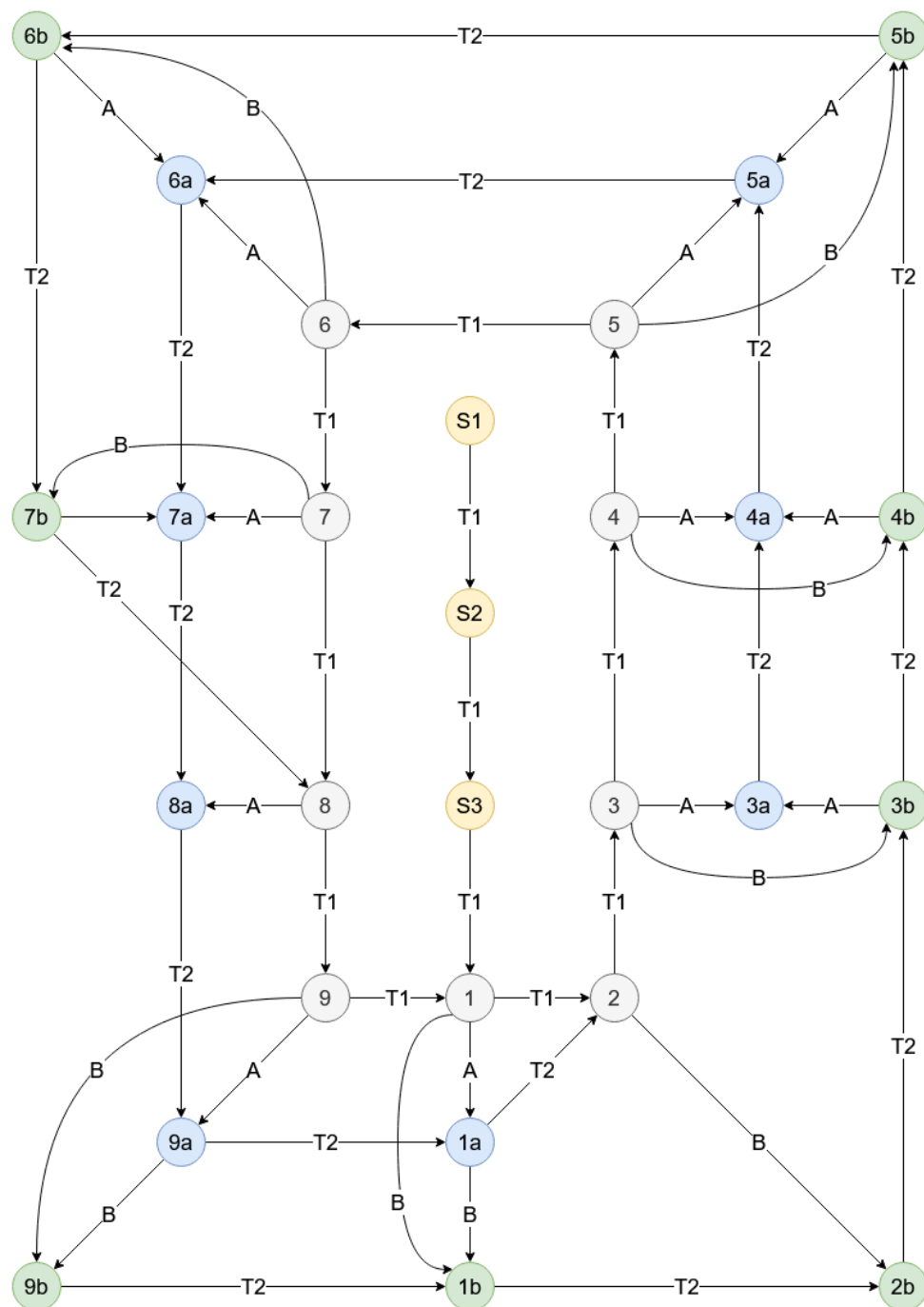
### a. Teoretický rozbor

Pro moji semestrální práci jsem si vybral automat na řízení křižovatky, konkrétně křižovatky U Plazy. Jednotlivý stavy budou odpovídat rozsvícení jednotlivých semaforů. Jednotlivý stavy může uživatel měnit pomocí tlačítek pro chodce. Stisknutím tlačítka pro chodce se zkrátí doba čekání pro chodce.

### b. Řešení

Jelikož mě nenapadlo, jak jednotlivé stavy rozumně popsat v příkazové řádce, tak jsem se rozhodl jednotlivé stavy nakreslit jako png obrázky a potom je načíst do programu. Při vytváření jednotlivých stavů jsem si všiml, že úloha se dala vyřešit několika různými způsoby. Ale všechny řešení měli stejný koncept a to, že po stisknutí tlačítka pro chodce by se měla doba mezi jednotlivými stavy zkrátit. Otázka byla, jak moc by se měl čas mezi jednotlivými stavy zkrátit. Samozřejmě šlo udělat to, že by se po stisknutí tlačítka ihned posunul automat do stavu, kde chodce, který zmáčknul tlačítko měl zelenou. Ale tohle se mi nezdálo jako vhodné řešení dané problematiky. Místo toho jsem vytvořil automat, který zkrátí dobu mezi jednotlivými stavy. Pokud nebude stisknuto žádné tlačítko pro chodce tak se automat chová lineárně, tedy automat začne ve stavu jedna přejde do stavu a potom do stavu tři a na konci se zase vrátí do stavu jedna.

### 3. Automatový model



Obrázek 1: Přechodový graf

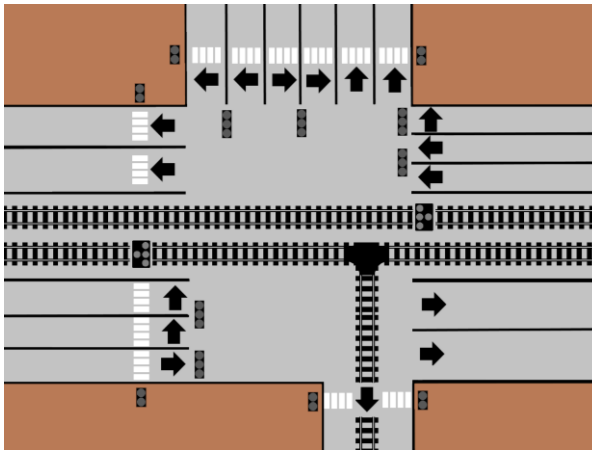
Legenda:

A – Tlačítko pro chodce na vodorovném přechodu.

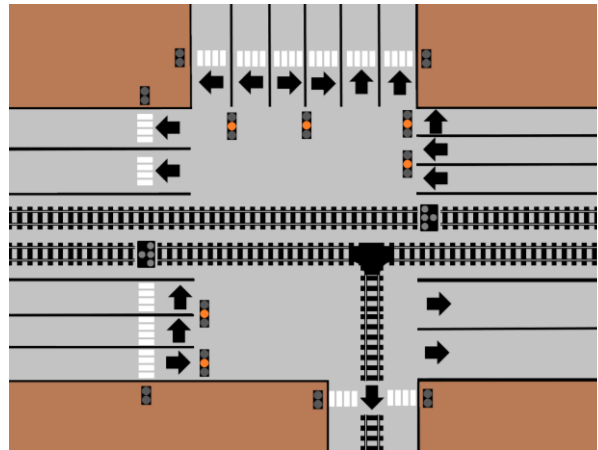
B – Tlačítko pro chodce na svislém přechodu.

T1 – čas mezi stavy – 4000 ms

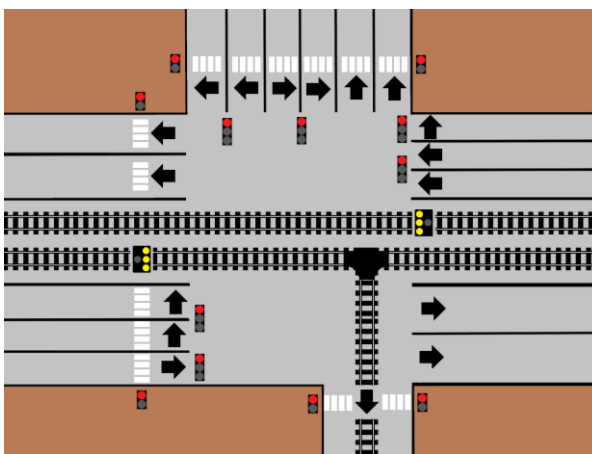
T2 – čas mezi stavy – 2000 ms



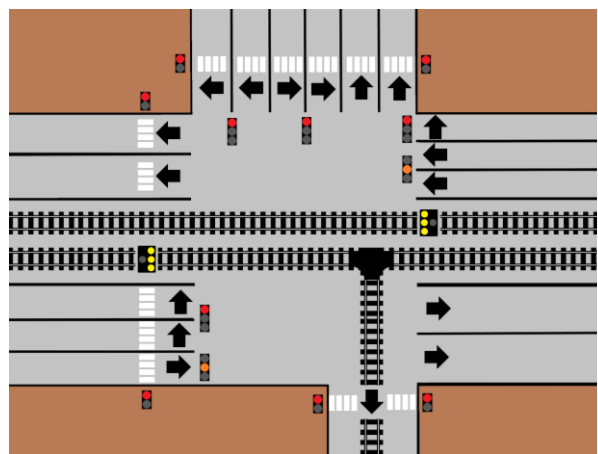
Obrázek 2: Stav S1



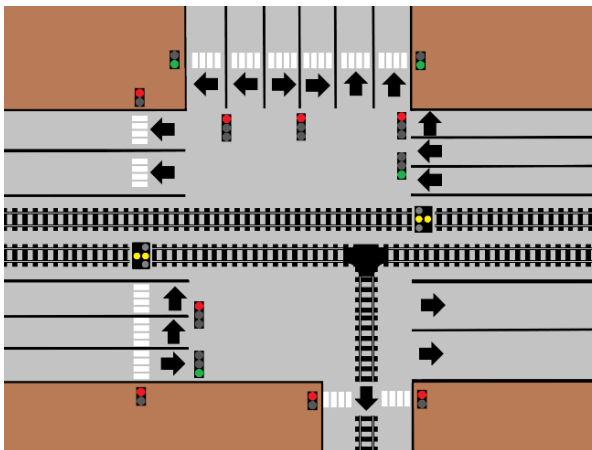
Obrázek 3: Stav S2



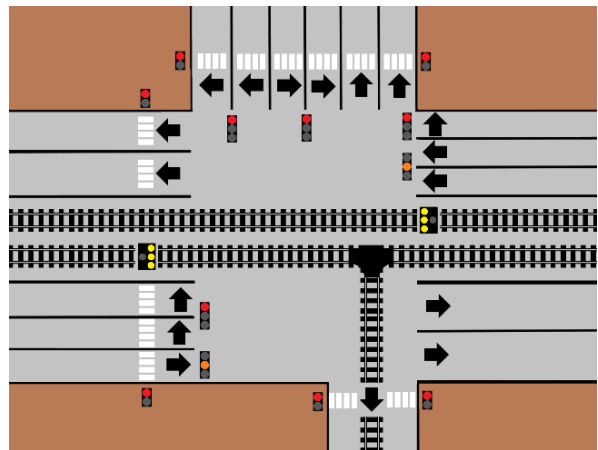
Obrázek 4: Stav S3



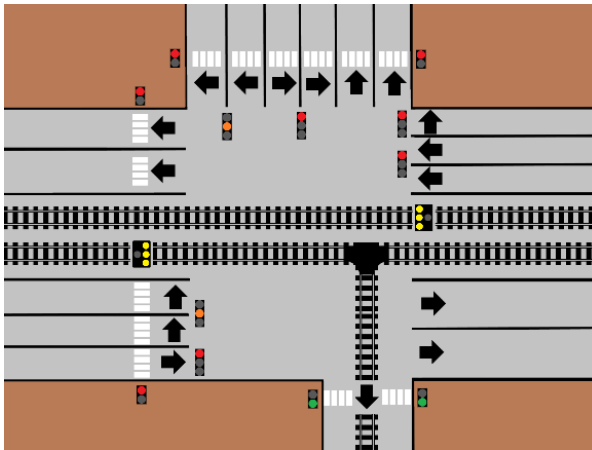
Obrázek 5: Stav 1



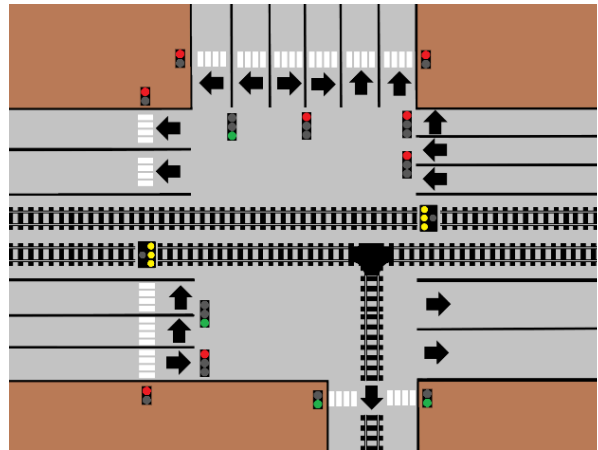
Obrázek 6: Stav 2



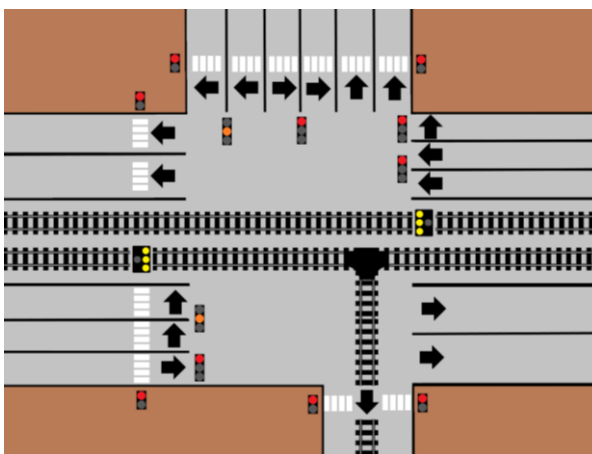
Obrázek 7: Stav 3



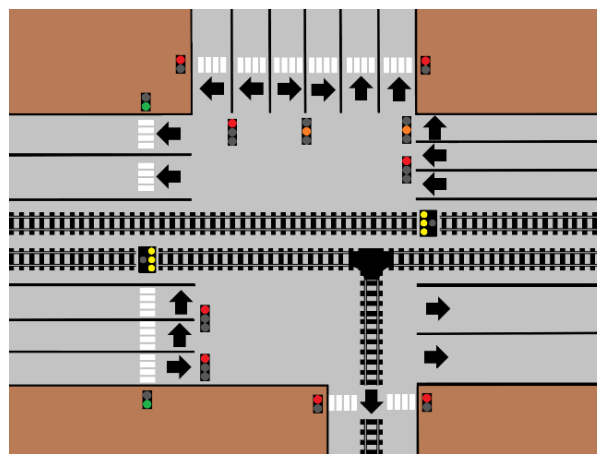
Obrázek 8: Stav 4



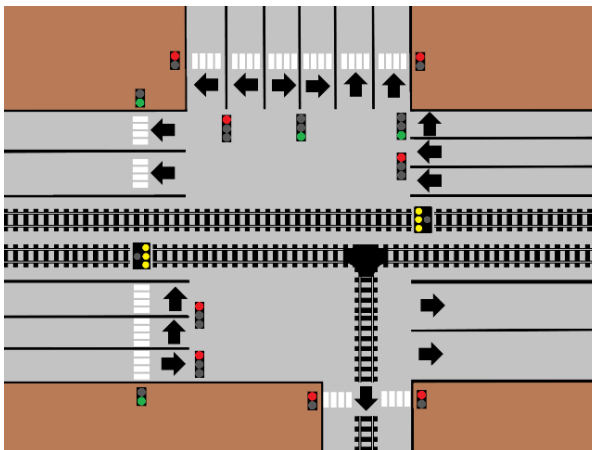
Obrázek 9: Stav 5



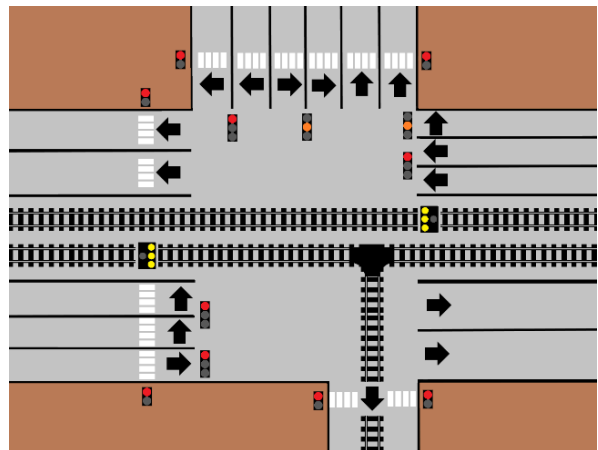
Obrázek 10: Stav 6



Obrázek 11: Stav 7



Obrázek 12: Stav 8



Obrázek 13: Stav 9

## 4. Implementace

Semestrální práci jsem vytvořil v programovacím jazyku Java. Pro zobrazení jednotlivých stavů jsem si vytvořil obrázky, které představují jednotlivé stavy. Jednotlivé obrázky jsou načteny do ArrayListu jako BufferedImage. Velikost plátna se nastaví pomocí velikosti obrázku. Okno se vytvoří pomocí JFrame a do okna se potom vloží jednotlivé komponenty. Po vytvoření okna a načtení všech dat se spustí metoda process(), která kontroluje každou sekundu jestli nebylo stisknuto tlačítko start. Po stisknutí tlačítka start se zavolá for-cyklus, který se stará o první tři stavy. Poté se zavolá poslední while cyklus, který běží do nekonečna. Ve while cyklu je for-cyklus, který se stará o vykreslení jednotlivých stavů pomocí metody changeState(). Ve for-cyklu se testuje, jestli bylo stisknuto tlačítko pro chodce, pokud byla tak se zkrátí doba mezi stavy. Zkrácená doba mezi stavy zůstane tak dlouho, dokud se nerozsvítí zelená pro daného chodce.

## 5. Uživatelská příručka

Program se dá jednoduše spustit pomocí souboru „antBuild“. Po spuštění programu se zobrazí okno, na kterém je vykreslena křižovatka. Program zde čeká až uživatel stiskne tlačítko start pro zahájení simulace. Plátno v okně se postupně mění podle jednotlivých stavů. Jednotlivé stavy se dají zrychlit pomocí tlačítek A a B. Program běží dále dokud ji uživatel neukončí pomocí tlačítka Exit.

## 6. Závěr

Program představuje jednoduchý model křižovatky U Plazy v Plzni. Z pohledu praktického využití program toho moc nenabízí. Model je jednoduchý a nedá se použít v realitě, jelikož jednotlivé stavy nejsou efektivní. V reálném životě by takhle křižovatka hodně zdržovala, jelikož pouští jenom málo aut najednou. Jediná výhoda tohoto modelu je, že nedochází ke kolizi aut ani lidí. Což má mnohem více nevýhod než výhod. Program je jednoduchý a vytvořen pro tuto konkrétní křižovatku. Tudíž by se další rozšíření programu těžce realizovalo.