

## **Zadání 2. projektu ATA LS 2020/21**

Cílem projektu je praktické cvičení dynamické analýzy na základě sledování událostí. Cvičení by mělo pokrýt runtime monitoring programů a sledování platností formulí za běhu.

Vytvořte dynamický analyzátor, který bude sledovat porušení vlastností daného programu na základě aktuálního záznamu událostí. Dynamický analyzátor bude také reportovat základní pokrytí využití vozíku.

### **Popis testovaného produktu**

Testovacím subjektem je řídicí software vozíku v hypotetické továrně. Specifikace chování (následující 2 odstavce nezměněny od 1. projektu):

#### **Řízení vozíku v robotické továrně**

V robotické továrně je motorizovaný vozík pro převoz materiálu potřebného k výrobě. Vozík má předem určené trasy a více zastávek. Řídicí systém vozíku má na starost plánování další zastávky a monitorování zátěže a využití kapacity vozíku. Vozík zpracovává požadavky o přesun materiálu. Materiál, resp. požadavek na přesun, je určený svojí zdrojovou a cílovou stanicí, váhou a binární prioritou (přesun je nebo není prioritní). Každý materiál má při převozu alokovan jeden slot vozíku. Pokud je materiál pro převoz větší než jeden slot nebo je příliš těžký, bude rozdělen do více slotů. O toto rozdělení se však kontrolní systém vozíku nestará. Uvažujte, že vozík je používán v ideálním prostředí (systémy s ním komunikující splňují veškeré předpoklady, data odpovídají reálným stavům). Vozík má pro převoz materiálu celkem 1 až 4 sloty. Vozíky jsou 3 druhů, každý má jinou maximální nosnost: 50 kg, 150 kg a 500 kg. Vozíky s nejmenší nosností mají nejméně 2 sloty, vozíky s největší nosností mají maximálně 2 sloty.

#### **Specifikace chování**

Pokud je požadováno přemístění nákladu z jednoho místa do druhého, vozík si materiál vyzvedne do 1 minuty. Pokud se to nestihne, materiálu se nastavuje prioritní vlastnost. Každý prioritní materiál musí být vyzvednutý vozíkem do 1 minuty od nastavení prioritního požadavku. Pokud vozík nakládá prioritní materiál, přepíná se do režimu pouze-vykládka. V tomto režimu zůstává, dokud nevyloží všechny takový materiál. Normálně vozík během své jízdy může nabírat a vykládat další materiály v jiných zastávkách. Na jednom místě může vozík akceptovat nebo vyložit jeden i více materiálů. Pořadí vyzvednutí materiálů nesouvisí s pořadím vytváření požadavků. Vozík neakceptuje materiál, pokud jsou všechny jeho sloty obsazené nebo by jeho převzetím byla překročena maximální nosnost.

## Záznam událostí

Vozík s každou aktivitou vypisuje na standardní výstup svůj aktuální stav. Následují formáty hlášení:

Žádost o naložení (v čase `TIME` ze stanice `SRC` do stanice `DST` s nákladem `CONTENT` o váze `W`):

`TIME requesting SRC DST CONTENT W`

Počátek nakládání (v čase `TIME` s vozíkem ve stanici `POS` nákladu `CONTENT` o váze `W` do slotu `SLOT`):

`TIME loading POS SLOT CONTENT W`

Počátek vykládání (v čase `TIME` s vozíkem ve stanici `POS` nákladu `CONTENT` o váze `W` ze slotu `SLOT`):

`TIME unloading POS SLOT CONTENT W`

Pohyb vozíku (v čase `TIME` ze stanice `POS1` do sousední stanice `POS2`):

`TIME moving POS1 POS2`

Oznámení o nečinnosti vozíku (v čase `TIME` ve stanici `POS`):

`TIME idle POS`

## Kritérium pokrytí

Aby bylo možné vyhodnotit testovací sadu, implementujte vyhodnocení kritéria pokrytí zaměřené na obsazenost jednotlivých slotů vozíku a požadavků z jednotlivých stanic. Vámi implementovaný analyzátor na konci testování reportuje pokrytí. Kritérium pokrytí vozíku a požadavků je založeno na ADC (All-Defs Coverage), ve kterém proměnnou definujeme jako dvojici zdrojové stanice požadavku nákladu a slotu pro naložení nákladu. Jinými slovy je kritérium pokrytí 100%, pokud ve všech existujících stanicích byl alespoň jednou naložen náklad do každého slotu vozíku. Pro 4 sloty vozíku a 4 existující stanice je to celkem 16 testovacích požadavků.

## Monitorované vlastnosti

Dynamický analyzátor bude monitorovat platnost následujících vlastností (předpokládá se vozík se 4 sloty a nosností 150):

1. Vozík nesmí nakládat na obsazený slot.
2. Vozík nesmí vykládat z volného slotu.
3. Náklad se musí vyložit, pokud je vozík v cílové stanici daného nákladu.
4. (volitelné) Každý požadavek o přesun musí někdy způsobit nakládku.

5. Vozík nesmí nakládat ve stanici, pokud na to neexistovala žádost.
6. Nesmí být naloženo více než 4 náklady.
7. Vozík nesmí být přetížen.
8. (volitelné) Každý naložený náklad se musí vyložit.
9. (volitelné) Vozík nesmí být nečinný, pokud existuje žádost o přesun.
  - Nečinnost vozíku znamená, že je ve stavu idle po nenulovou dobu. Jinými slovy, že neexistuje následující akce vozíku se stejným časem, ve kterém vozík přešel do stavu idle.

## Úkoly

1. Vytvořte program v jazyku Python, který čtením standardního vstupu průběžně analyzuje správné chování vozíku.
2. Přeložte monitorované vlastnosti do monitorovacích automatů.
  - Omezený počet parametrů eliminujte rozvinutím.
  - Neomezený počet parametrů ve vlastnostech řešte instanciací monitorů za běhu.
  - Automaty implementujte v souboru `cart_monitor.py`.
  - V případě porušení vlastní vypíšte na standardní výstup základní report: `TIME:error: popis porušené vlastnosti`
  - Automaty nakreslete a odevzdejte v souboru `automaty.pdf`.
3. Implementujte výše definované kritérium pokrytí. Po skončení analýzy vypíšte výsledné kritérium pokrytí (v procentech): `Coverage XX%`
4. Rozšiřte testovací vstupy v souboru `requests.csv` o náhodně vygenerované požadavky přesunu nákladu:
  - řádek obsahuje (čárkou oddělený seznam): čas příchodu požadavku, zdrojová stanice, cílová stanice, váha, popis nákladu

## Odevzdání

Odevzdejte archiv `proj2.zip` obsahující následující soubory:

```
proj2.zip
+- automaty.pdf          - obrázky automatů, případné komentáře a popis
|                        - implementace, reportování nalezených chyb
+- cartctl_monitor.py    - implementovaný dynamický analyzátor
`- requests.csv          - vámi vygenerované testovací vstupy
```

## Poznámky a hodnocení

V souboru `bad_log.txt` je příklad sekvence událostí, která v čase 12 porušuje první vlastnost.

K dobrému hodnocení přispěje:

- správná implementace všech (nevolitelných) vlastností,
- výpočet pokrytí,
- reportování chyby v případě porušení vlastnosti.

Primárním cílem projektu je vytvoření dynamického analyzátoru. Generování vstupních testovacích dat je nepovinné a slouží pouze jako pomůcka pro demonstraci analyzátoru nebo reprodukci chybového stavu. Plného počtu bodů je možné dosáhnout i bez implementace monitorů vlastností označené jako “volitelné”.

## Příklady a spuštění

```
$ cat requests.csv
10,A,B,20,helmet
45,C,A,40,heart
25,D,C,40,braceletR
30,D,C,40,braceletL

$ ./cartctl_test.py requests.csv
10 requesting A B helmet 20
10 requesting A B braceletR 40
10 loading A helmet 20 0
12 loaded A helmet
12 idle A
12 loading A braceletR 40 1
14 loaded A braceletR
14 idle A
14 moving A B
34 idle B
34 unloading B helmet 20 0
36 unloaded B helmet
36 idle B
36 unloading B braceletR 40 1
38 unloaded B braceletR
38 idle B
38 stop

$ ./cartctl_test.py requests.csv | ./cart_monitor.py
All properties hold.
CartCoverage 42%

$ ./cart_monitor.py <bad_log.txt
12:error: loading into an occupied slot #0
CartCoverage 6%
```