

## **Zadání 1. projektu ATA LS 2020/21**

Cílem projektu je praktické cvičení testování založené na datech. Cvičení by mělo pokrýt testování založené na požadavcích dle CEG, kombinační testování při tvorbě dat a automatizaci testů.

Na základě specifikace testované komponenty v přirozeném jazyce navrhnete automatizovanou testovací sadu.

### **Popis testovaného produktu**

#### **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.

## Úkoly

1. Upravte specifikaci (ze sekce “Specifikace chování”) dle ambiguity review. Zaměřte se na podmínky, příčiny bez důsledků a neurčité nebo víceznačné výrazy. Pro chybějící části nebo jejich upřesnění si stanovte vlastní vysvětlení (toto není běžná praxe, je to čistě z důvodu kontroly projektu, tj. bez účelné zpětné vazby zadavatele).
2. Z upravené specifikace vytvořte graf příčin a důsledků (CEG graf). Při stanovení časových závislostí můžete definovat časová pásma, náležitosti příčin a důsledků a jejich souvislosti.
3. Identifikujte vstupní parametry testu. Parametry mohou být přímé (jejich hodnota se přímo objeví v testu) nebo nepřímé (další data z nich budou odvozena), např. “čas vytvoření požadavku č. 3” nebo “maximální váha všech požadavků”.
4. Identifikujte charakteristiky parametrů a definujte jejich bloky. Zaměřte se převážně na charakteristiky plánu požadavků a přesunu materiálu (místa, časy požadavků, délky převozu, ale také závislosti mezi těmito aspekty). Specifikujte omezení kombinace bloků různých charakteristik. Navrhněte minimální (suboptimální) kombinace dvojic charakteristik (kritérium Pairwise Coverage).
5. Navrhněte testovací sadu pro *plánovač se zpětnými voláními* (viz soubor `cartctl_test.py`) zahrnující testovací scénáře z rozhodovací tabulky CEG grafu z bodu 2 (měly by být pokryty všechny příčiny a důsledky, obdoba Condition coverage a Decision coverage) a testy odpovídající kombinacím dvojic z bodu 4.

## Odevzdání

Odevzdejte archiv `proj1.zip` obsahující následující strukturu:

```
proj1.zip
+- specifikace.md      - upravená specifikace (formát Github
|                      Markdown, formátování textu se pokud
|                      možno vyhněte),
+- ceg.txt            - graf příčin a důsledků (formát
|                      akceptovaný ceg.testos.org)
+- combine.json        - nastavení pro nástroj combine (volitelné)
+- testy.md           - dokumentace testů (formát Github Markdown)
|
`- cartctl/           - adresář obsahující testovací scénáře vč.
|                      kódu SUT a jeho okolí
|   +- cartctl_test.py - Hlavní testovací soubor
|   |   ...
|   `-- *test*.py      - volitelné pomocné testovací soubory
```

## Poznámky

### System Under Test

V informačním systému, sekce Soubory / Projekty je v archivu `cartctl.zip` k dispozici příklad implementace vozíku, resp. jeho řídicího kontroleru. V archivu je k dispozici ukázkový testovací soubor. SUT je `cartctl.py`.

### Úprava specifikace (dle Ambiguity review)

Report z úpravy bude mít následující formát:

Jedna věta, která byla zkopírována z původní specifikace.

- typ odhalené chyby ve větě, krátké vysvětlení.
- typ další chyby, včetně krátkého popisu.

\*Upravená věta opravující odhalené chyby. Doplnující věta, pokud je třeba informaci přidat.\*

Další věta z původní specifikace.

- typ chyby v další větě...

## Dokumentace testů

V souboru `testy.md` musí být dokumentace testů. Skládá se z několika částí:

1. CEG graf (obrázek) a výsledná rozhodovací tabulka (může být screenshot tabulky z `ceg.testos.org`). Dále nechť je test z rozhodovací tabulky označován “testovací scénář”.
2. Identifikace vstupních parametrů, které budou tvořit test, ve formě tabulky dvou sloupců (název/identifikátor parametru, stručný popis).
3. Popište charakteristiky parametrů ve formě tabulky vč. omezení, příklad níže.
4. Identifikujte požadavky (testy) na kombinace všech dvojic bloků (může být screenshot tabulky z nástroje `combine`).
5. (volitelně) Charakteristiky a omezení uložte do souboru `combine.json`. Obsah tohoto souboru by měl být použitelný pro nástroj `combine.testos.org`. Získat ho můžete odchycením HTTP požadavku z webového prohlížeče (Např. Chrome / Developer tools / Network / endpoint `generate` po stisku tlačítka `Generate` / Headers, část Request payload).

Příklad formátu popisu charakteristiky:

Cx	Popis charakteristiky Cx
1	popis prvního bloku
2	popis druhého bloku
...	...
n	popis bloku n

- Cx.1 -> !Cy.3

## Hodnocení

K dobrému hodnocení přispěje:

- správné použití nástrojů ([ceg.testos.org](http://ceg.testos.org) a [combine.testos.org](http://combine.testos.org)),
- použití omezení v CEG grafu,
- správná definice charakteristik,
- použití omezení u definice charakteristik a bloků.