

Jméno:

UČO:



líst

učo

body

Oblast strojově snímaných informací. Svě učo a číslo lístu vyplňte zleva dle vzoru číslic. Jinak do této oblasti nezasahujte.

0 1 2 3 4 5 6 7 8 9

### 1. [0,7 bodu]

Váš poslední úkol bude opět programovací. Půjde ale pouze jen o *bonusový úkol*. I když máte bodů už dost, stejně se úkol vyřešit vyplatí, jelikož bude zaměřený na **redukce**, a to je koncept se kterým se určitě ještě setkáte! Jak víte už z přednášek a cvičení, redukce nám umožňují řešit nové problémy transformací na problémy, které už umíme řešit.

### Motivace a představení zadání

V moderní době se setkáváme s nějakým systémem na každodenní bázi. Jedním z takových systémů je například výtah nebo klapky na letadlech. Mělo by být zřejmé, proč bychom od takových systémů chtěli mít určitou záruku jejich chování. Jak takovou záruku získat? Odpovědí je **Model Checking**.

Model checking je soubor metod určených k ověřování, jestli model  $\mathcal{M}$  nějakého systému odpovídá jeho specifikaci  $\mathcal{S}$ , neboli  $\mathcal{M} \stackrel{?}{\subseteq} \mathcal{S}$ . V praxi jsou systémy modelovány komplexnějšími formalismy než klasickými konečnými automaty; pro náš úkol ale postačí.

Vášim úkolem bude implementovat redukci, která pro systém  $\mathcal{A}_{system}$  reprezentovaný deterministickým DFA a specifikaci  $\mathcal{A}_{specification}$ , taktéž reprezentovanou DFA, problém model checkingu transformuje na problém  $co-L(\mathcal{A}_{specification}) \cap L(\mathcal{A}_{system}) \stackrel{?}{=} \emptyset$ .

Budete programovat pět funkcí a vašim cílem bude je nejen naprogramovat, ale i správně poskládat, aby skutečně řešily zmíněný problém. Konkrétněji půjde o:

- redukční funkci,
- funkci komplementu DFA,
- funkci ztotálnění DFA,
- funkci která provede paralelní synchronní kompozici dvou DFA,
- funkci, která zjistí, zda daný DFA akceptuje prázdný jazyk.

Mimo to se v kostře nachází funkce `check_model(system_model: DFA, specification: DFA) -> bool`. Tahle funkce je zde z čistě didaktických důvodů a to aby bylo zřetelné, že v skutečnosti řešíme jiný problém.

### Reprezentace automatů

Automat bude reprezentovaný stejně jako v minulé programovací úloze. Budeme uvažovat fixní abecedu (malá písmena anglické abecedy a mezeru) a množina stavů bude daná implicitně pomocí přechodové funkce. Přechodovou funkci budeme reprezentovat slovníkem (`dict`). Klíči budou dvojice (`stav, symbol`) a hodnotami stavu DFA. Mimo to si třída drží množinu akceptujících stavů a iniciální stav.

Do definice tříd **není povoleno zasahovat**.

### Zadání funkcí

`reduction_function(system_model: DFA, specification: DFA) -> bool`

Musí platit `reduction_function( $\mathcal{A}_{system}, \mathcal{A}_{specification}$ ) = True  $\iff \mathcal{A}_{system} \subseteq \mathcal{A}_{specification}$ .`

Oblast strojově snímaných informací, nezasahujte. **Druhá strana se neskenuje.**

Jméno:

UČO:

0007

líst

2

učo

body

Oblast strojově snímaných informací. Svě učo a číslo lístu vyplňte zleva dle vzoru číslic. Jinak do této oblasti nezasahujte.

0123456789

`DFA_complete(automaton: DFA) -> DFA`

Funkce vrací vstupní automat po ztotálnění.

`DFA_complement(automaton: DFA) -> DFA`

Funkce vrací automat rozpoznávající doplněk jazyka vstupního automatu *automaton*.

`DFA_product(a1: DFA, a2: DFA) -> DFA`

Funkce vrátí automat provádějící paralelní synchronní kompozici automatů *a1* a *a2*.

`check_emptyiness(automaton: DFA) -> bool`

Funkce vrací `True`, pokud  $L(\text{automaton}) = \emptyset$ . Jinak vrací `False`.

## Hodnocení

- K získání nenulového hodnocení musíte implementovat všechny funkce.
- **Není povolena** úprava kostry s výjimkou implementace funkcí požadovaných zadáním. Konkrétně tedy nezasahujte do žádné z definice tříd `GenericAutomaton` a `DFA`, ani do implementace `check_model`. Neměňte ani hlavičky žádných existujících funkcí, konkrétně nepřidávejte ani neubírejte argumenty.
- Můžete si implementovat pomocné funkce (vzorové řešení ale žádné pomocné funkce nepoužívá).
- Využívání balíčků s výjimkou `typing`, `itertools` a `collections` **není povoleno** bez explicitního schválení na diskusním fóru. Vzorové řešení si vystačí se zmíněnými moduly. V případě opodstatněného požadavku na nový balíček skrz diskusní fórum je možné povolit i ten.
- Řešení **nemusí** projít kontrolou `mypy`, i když silně doporučujeme, aby prošlo.
- Všechny implementované funkce musí být čisté. Speciálně **není povoleno**, aby něco vypisovaly na standardní nebo standardní chybový výstup.

## Odevzdávání

Vypracovaný úkol (jediný `.py` soubor s definicemi požadovaných funkcí) odevzdávejte do příslušné odevzdáárny v ISu.

Úkol se bude vyhodnocovat ve stejném režimu jako textové domácí úkoly, tedy hromadně po konci odevzdání. Nebudete tedy mít k dispozici průběžné výsledky od vyhodnocovací služby (toto je bohužel důsledek technických omezení daných současnou organizací předmětu).

V kostře máte dodáno několik základních testů a jistě si snadno dokážete dopsat vlastní. Svě vlastní testy volejte z `if __name__ == "__main__"` bloku.

Řešení budeme testovat pomocí Pythonu verze 3.10.12.

Jméno:

UČO:

0007

list

3

učo

body

Oblast strojově snímaných informací. Svě ucho a číslo listu vyplňte  
zleva dle vzoru číslic. Jinak do této oblasti nezasahujte.

0123456789

Příklad musíte vypracovávat samostatně, bez sdílení kódu mezi sebou, bez přebírání kódu z internetu (myšlenku algoritmů přebírat můžete, implementaci však nikoli) a bez využití jazykových modelů.

Případné dotazy směřujte jako vždy do diskusního fóra.