

Posudek diplomové práce

Matematicko-fyzikální fakulta Univerzity Karlovy

Autor práce Bc. Tomáš Hurt
Název práce Framework pro vývoj optimalizačních algoritmů
Rok odevzdání 2020
Studijní program Informatika **Studijní obor** Softwarové systémy

Autor posudku RNDr. Ing. Otakar Trunda **Role** Vedoucí
Pracoviště Katedra teoretické informatiky a matematické logiky

Text posudku:

Práce se zabývá vytvořením frameworku pro podporu vývoje algoritmů kombinatorické optimalizace, zejména plánování. V první části je představena oblast plánování a základní techniky pro reprezentaci plánovacích problémů a jejich řešení. Druhá část pak popisuje návrh frameworku a analyzuje související otázky z hlediska softwarového inženýrství.

Úvod do problematiky je podán přehledně a v adekvátní míře detailu. Vzhledem k tomu, že analýza fungování algoritmů není cílem práce, je většina technik popsána jen stručně spíše s ohledem na způsob jejich použití. V popisu plánovacích technik jsem objevil pouze dvě drobné nepřesnosti: algoritmus Hill climbing je popsán jako maximalizační, přestože v daném kontextu se používá jeho minimalizační verze, a výraz "relaxace" na str. 70-72 je použitý ve významu "delete relaxace", což by mělo být explicitně uvedeno. Úvodní část také definuje několik pojmů a technik, které nejsou v dalším textu odkazovány - např. plánování v prostoru plánů. Popis takových pojmů by bylo vhodné zkrátit nebo vypustit. Naopak použitý algoritmus pro výběr vzorů při konstrukci PDB heuristiky (str. 85) není všeobecně známý a zasloužil by si podrobnější popis.

Hlavní část práce popisuje proces návrhu a implementace frameworku. Jsou popsány požadavky na systém a analyzované možné způsoby jejich naplnění. Na základě analýzy jsou pak zvolené vhodné implementační techniky. Výsledkem je poměrně rozsáhlé softwarové dílo s robustní architekturou, které poskytuje podporu pro mnoho typických činností spojených s plánováním: načítání vstupů v nejčastěji používaných formátech, reprezentace dat a metody pro efektivní práci s nimi, automatizace experimentů a podobně. Je zde implementováno i několik prohledávacích algoritmů a heuristik. Oceňuji zejména fakt, že framework plně podporuje formát PDDL 3.1 v celé jeho šíři a také implementaci některých pokročilých technik, např. liftovaného zpětného prohledávání.

K implementační části mám dvě výhrady. Na některých místech je architektura poměrně komplikovaná a pro nového uživatele zpočátku těžko pochopitelná. Není např. na první pohled zřejmé, jakou výhodu přináší použití "ISuccessor" namísto "IState" a podobně. Navíc při řešení úloh jednoduššího typu může složitá architektura výpočet zpomalovat, jak autor sám upozorňuje.

Druhou věcí je nulové využití generických typů, přestože by to mohlo na některých místech přinášet výhodu. Např. heuristiky i prohledávací algoritmy pracují s objektem obecného typu ISearchNode. Při implementaci doménově specifické heuristiky, která je schopna pracovat pouze se stavy konkrétního typu je pak potřeba při každém volání objekt přetypovat. Navíc

při použití nevhodného typu stavu se toto pozná až za běhu. Návrh ve stylu

IHeuristic<T> where T : ISearchNode

by mohl oba tyto problémy vyřešit.

K celkovému zpracování mám následující připomínky:

- srovnání s existujícími programy podobného typu bych zařadil před návrh výsledného řešení, ne až za něj. Navíc srovnání by mohlo být obsáhlejší.
- práce neobsahuje žádné experimentální srovnání efektivity navrženého systému s jinými nástroji. Např. srovnání rychlosti parsování s existujícím parserem v Javě, srovnání počtu expandovaných vrcholů za sekundu a podobně. Tento fakt považuju celkově za nejslabší část práce.
- naopak oceňuji návrh i kvalitu zpracování ilustrací a diagramů

Textová úroveň práce je velmi dobrá. Neobjevil jsem žádné pravopisné chyby a po stylistické stránce je text čtivý a současně exaktní na místech, kde je to třeba. Mám výhrady ke dvěma anglicismům: "traverzovat" a české skloňování slova "interface". Z formálního hlediska práce splňuje všechny náležitosti.

Řešitel se dokázal dobře zorientovat v algoritmicky náročné oblasti automatizovaného plánování a prokázal schopnost efektivně využívat nástroje programovacího jazyka a znalosti principů OOP pro vytvoření architektury rozsáhlejšího systému. Práci považuju celkově za kvalitní a doporučuji ji k obhajobě s navrhovanou známkou 1-2.

Práci doporučuji k obhajobě.

Práci nenavrhuji na zvláštní ocenění.

Pokud práci navrhuje na zvláštní ocenění (cena děkana apod.), prosím uveďte zde stručné zdůvodnění (vzniklé publikace, významnost tématu, inovativnost práce apod.).

Datum 27.1.2020

Podpis