Oponentský posudek diplomové práce

Peter Cipov: Návrh spolehlivých systémů s využitím formálních metod

Diplomová práce se týká poměrně náročného tématu formální verifikace software. Cílem bylo ověřit možnosti začlenění některého takového přístupu do nástrojové sady používané evropským projektem CORDET pro tvorbu družicového řídícího software. Tento kontext vysvětluje, proč je text psán v anglickém jazyce; bohužel jeho gramatická i stylistická úroveň je dosti nízká i přes to, že není rozsahem velký a je psán značně (místy až příliš) úsporně.

Diplomant v práci (kapitola 3) stručně vysvětluje hlavní přístupy k formální verifikaci založené na analýze modelů (*model checking*). Chybí ale vyústění v podobě popisu, jak tyto formalismy získat z konkrétních struktur softwarové implementace (UML modelů, zdrojového kódu apod.) s ohledem na využití příslušnými nástroji. V kap. 4 diplomant vhodně, byť bez odkazu na kontext projektu, definuje kritéria pro praktické použití některého takového přístupu v kontextu zmíněného projektu. Po vyloučení průmyslově nevhodných prototypů analyzuje dva hlavní kandidáty a spíše stručně zdůvodňuje výběr nástroje Java PathFinder doplněného o LTL modul. Závěr kapitoly však v tomto ohledu vyznívá nejednoznačně, neboť autor zároveň konstatuje, že modul nesplňuje kvalitativní požadavky a jeho vývoj nepokračuje.

Kapitola 5 je nejslabším místem práce. Jejím cílem je zřejmě popis výsledku autorovy snahy dovést LTL modul do funkční podoby a vyhodnotit jeho vlastnosti, ale tento cíl není v úvodu kapitoly deklarován. Funkčnost i způsob použití modulu jsou popsány příliš stručně na to, aby mohly být návodem pro uživatele (chybí ilustrativní příklad), a omezení nejsou vysvětlena systematicky. V neposlední řadě není vůbec jasné, zda popisovaný LTL modul je kompletně autorovým vlastním dílem – což by byl výsledek hodný spíše disertační práce – nebo zda jde o úpravy realizace ze zdroje [31]. Až přečtení závěru diplomové práce potvrdí druhou variantu.

Nakonec je v kapitolách 6 a 7 popsána integrace Java PathFinder s LTL modulem do vývojového prostředí Eclipse, používaného v CORDET projektu, a výsledný produkt je vyhodnocen na typovém příkladu. Postrádám zde poněkud architektonický nadhled – popis koncepce integrace a technických detailů (struktura Eclipse pluginu, způsob získání výstupu Java PathFinderu, apod.) a v sekci 7.3 vysvětlení, že jde o vyhodnocení nástroje, nikoli příkladu.

Implementace integračního Eclipse pluginu je vhodně strukturovaná a kód je srozumitelný, pouze (opět) chybí dokumentační komentář s uvedením autora kódu a nerozumím použití jmenného prostoru „com.singularity“. Výsledný nástroj je funkční jak co do ověřování LTL vlastností, tak při zobrazování automatu zapsaného LTL formulemi, ale nespolehlivý – v některých případech ověření či zobrazení končí nejasnou chybou.

Přes všechny poměrně kritické připomínky k práci bych chtěl na závěr zdůraznit, že oceňuji diplomantovu schopnost zjevně velmi fundovaně a metodicky i technicky dobře zvládnout netriviální problém. Ve svém souhrnu práce představuje dobrý příklad přenosu výsledků výzkumu do prakticky použitelné podoby. Práci proto **doporučuji k obhajobě**, během níž žádám diplomanta o zodpovězení několika otázek, a navrhuji její hodnocení známkou **velmi dobře**.

Otázky k obhajobě:

1. Informoval jste o opravách LTL modulu jeho původního vývojáře či obecně výzkumnou komunitu, např. formou blogu, publikace či diskuse na sociálních sítích? Pokud ano, jaké byly získané ohlasy?
2. Jaká je praktická škálovatelnost obou nástrojů (model checker, zobrazování formulí temporální logiky), tedy jak složité podmínky a v jakém počtu je možné ověřovat resp. zobrazovat, jaká je použitelná velikost ověřované aplikace, a jaké má ověření nároky na paměť a strojový čas?

V Plzni, dne 27.8.2012

Doc. Ing. Přemysl Brada, Ph.D.