

Posudek bakalářské práce

Matematicko-fyzikální fakulta Univerzity Karlovy

Autor práce Arkadiusz Martin Antoniewicz
Název práce Tokenization-aware Diff and Patch
Rok odevzdání 2021
Studijní program Informatika
Studijní obor IPSS

Autor posudku Miroslav Kratochvíl Vedoucí
Pracoviště Katedra softwarového inženýrství

K celé práci

lepší OK horší nevyhovuje

	lepší	OK	horší	nevyhovuje
Obtížnost zadání		×		
Splnění zadání			×	
Rozsah práce <i>... textová i implementační část, zohlednění náročnosti</i>			×	×
<p>Cílem práce je vyprodukovat nástroj pro jednodušší porovnávání textových souborů, zohledňující komplikovanější strukturu než řádky. Výsledkem je alternativa k tradičním UNIXovým utilitám <code>diff</code> a <code>patch</code>.</p> <p>Druhá odevzdaná verze práce bohužel oproti původní verzi nepřináší mnoho vylepšení — autor zvolil spíše minimalistickou metodu reakce na nedostatky zmiňované v posudcích první verze, ze kterých je dostatečně opravená jen opravdu malá podmnožina.</p> <p>Ačkoliv práce popisuje zajímavou novou konstrukci a poměrně inovativní a potenciálně užitečný výsledek, kvalita prezentace výsledků a vzniklého software je nedostatečná, místy až horší, než ve verzi první.</p>				

Textová část práce

lepší OK horší nevyhovuje

	lepší	OK	horší	nevyhovuje
Formální úprava <i>... jazyková úroveň, typografická úroveň, citace</i>		×		
Struktura textu <i>... kontext, cíle, analýza, návrh, vyhodnocení, úroveň detailu</i>		×	×	
Analýza				×
Vývojová dokumentace			×	
Uživatelská dokumentace			×	
<p>Text práce nedoznal oproti původní verzi větších změn, kromě úplného odstranění několika sekcí (původně zkopírovaných z jiných zdrojů) a několika nových odstavců textu (cca tři). Odstraněním teoretických základů pro mergování a patchování textu bohužel zmizel prakticky všechny netriviální obsah řešeršní části.</p> <p>Popis hlavní výhody algoritmu (schopnost nezpůsobit kolizi na prázdných znacích, §2.2) zůstává poněkud hůře pochopitelný, podobně jako velmi implicitní definice textové podoby REDFA automatů (§2.1.1).</p> <p>Autor v závěru přidal poněkud vytrženou zmínku o integraci s <code>gitem</code>, čímž sice technicky splnil zadání práce, ale diskuze integrace s <code>gitem</code> není nijak rozvinutá. Očekávat by se dala např. ukázka jak nástroj z <code>gitu</code> vlastně spustit, nebo jak se projeví konflikt.</p> <p>Kvalita uživatelské a vývojové dokumentace se prakticky nezměnila. Užitečné command-line nástroje je nutné dokumentovat vyčerpávajícím popisem funkcionality aspoň v README, ideálně pak ve standardních manuálových stránkách.</p>				

Implementační část práce

lepší OK horší nevyhovuje

Kvalita návrhu	<i>... architektura, struktury a algoritmy, použité technologie</i>		×		
Kvalita zpracování	<i>... jmenné konvence, formátování, komentáře, testování</i>			×	
Stabilita implementace			×	×	

Kromě opravení několika potenciálních zdrojů nedefinovaného chování v programu nedošlo k větším změnám. Největším ‘vylepšením’ je jednoduchá detekce nekonečných smyček v REDFA automatu při tokenizaci. Program s uživatelem bohužel opět nekomunikuje nijak obsáhle, např. není jasné, co znamená chybová hláška ‘Cycle detected. Fix automaton needed’, a proč se zobrazuje při zadání automatu obsahujícího jediný triviální regex `.*`, který by celý text měl podle specifikace reprezentovat jako jediný token.

Program při některých chybách (např. při použití adresářů podle návodu v `--helpu`) skončí bez jakékoliv akce nebo chybové hlášky a s nulovým exit-kódem hlásícím úspěch.

Dodaný systém sestavení programu je výrazně nedokonalý — mírně nestandardní `makefile` bez schopnosti konfigurace, hledání systémových knihoven nebo instalace. Očekávat by se dal alespoň CMake nebo jednoduché použití autotools.

Celkové hodnocení Neprospěl(a)

Práci navrhuji na zvláštní ocenění Ne

Datum

Podpis