

AKADEMIA GÓRNICZO-HUTNICZA IM. STANISŁAWA
STASZICA

KRAKÓW

Generator specyfikacji logicznej

Autorzy:

Marcin JĘDRZEJCZYK
Paweł OGORZAŁY

Prowadzący:

Dr inż. Radosław KLIMEK

26 kwietnia 2016

1 Wstęp

2 Cel projektu

Celem projektu jest wytworzenie programu, który dla podanego diagramu będzie w stanie go sparsować do formatu pozwalającego na wygenerowanie specyfikacji logicznej.

3 Powód tworzenia generatora

- Ręczne tworzenie specyfikacji logiki jest trudne dla niedoświadczonych w tym użytkowników.
- Formalna weryfikacja modelu oprogramowania pozwala obniżyć koszty i zwiększyć niezawodność.
- Brak takich narzędzi.

4 Ważne

- Diagram aktywności musi składać się z wcześniej zdefiniowanych wzorców, zagnieżdżanie jest dozwolone.
- Diagram aktywności składa się tylko z atomicznych aktywności, zidentyfikowanych podczas tworzenia scenariuszy przypadków użycia.
- Generator musi działać automatycznie, usuwa to błąd ludzki.

5 Algorytmy

Przykładowe wzorce:

- Sekwencja, sequence
- Współbieżność, concurrent fork/join
- Pętla while, loop while
- Rozgałęzienie, branching

Wyrażenie logiczne W_L jest strukturą stworzoną według poniższych zasad:

- każdy elementarny zbiór $pat(a_i)$, gdzie $i > 0$ i każde a_i jest formułą atomiczną, jest wyrażeniem logicznym,
- każde $pat(A)$, gdzie $i > 0$ i każde A_i jest albo
 - atomiczną formułą lub
 - logicznym wyrażeniem $pat()$

także jest wyrażeniem logicznym.

Wstępny algorytm:

1. Analiza diagramów aktywności w celu wyciągnięcia z nich wcześniej zdefiniowanych wzorców przepływu.
2. Przetłumaczenie wyłuskanych wzorców na wyrażenia logiczne W_L .
3. Generowanie specyfikacji logicznej L z wyrażeń logicznych,

Algorytm generujący specyfikację logiczną:

1. Na początku specyfikacja jest pusta, np. $L = \emptyset$;
2. Najbardziej zagnieżdżone wzorce są przetwarzane jako pierwsze, a następnie mniej zagnieżdżone;
3. Jeśli obecnie analizowany wzorec składa się wyłącznie z formuł atomicznych, specyfikacja logiczna jest rozszerzana, poprzez sumowanie zbiorów, których formuły są złączone z obecnie analizowanym wzorcem $pat()$, np. $L = L \cup pat()$;
4. Jeżeli jakiś argument jest wzorem sam w sobie to:
 - po pierwsze formuła $f1 ??$, a potem
 - formuła $f4 ??$

tego wzoru(jeśli jakiegoś), lub w innym wypadku wziąć pod uwagę tylko najbardziej zagnieżdżony daleko? na lewo lub prawo, odpowiednio, są podstawiane osobno w miejsce wzorca jako argument.

6 Technologie

Wydaje mi się, że języki funkcyjne mogłyby się tu dobrze sprawdzić, może Erlang. Chociaż w Javie mogłoby się to łatwiej napisać(lepiej znamy Jave niż Erlanga).

7 Literatura

Radosław Klimek: From Extraction of Logical Specifications to Deduction-Based Formal Verification of Requirements Models. Strony 61-75.