Prover logiki temporalnej

Stanisław Maciąg, Piotr Mitana

2014

1 Założenia projektowe

W ramach projektu zaprojektowana oraz zaimplementowana zostanie aplikacja sprawdzająca spełnialność formuły logicznej zawierającej operatory klasyczne oraz temporalne, danej w postaci ciągu znaków wprowadzonego przez użytkownika lub wczytanego z pliku. Do realizacji tego zadania wykorzystana zostanie metoda tablic semantycznych dla logiki temporalnej (ang. *the Tableau Method for Temporal Logic*).

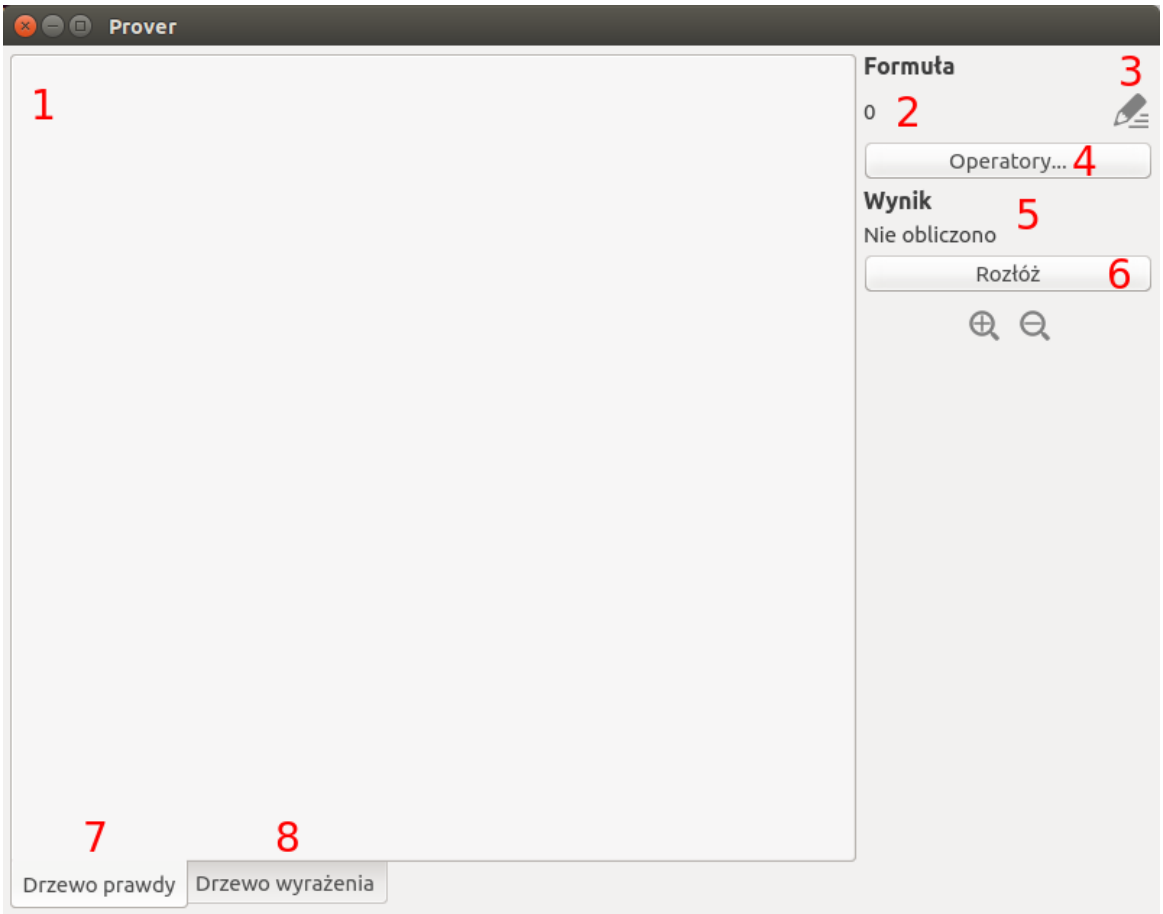
1.1 Środowisko

1. System operacyjny Linux
2. Języki programowania C++ oraz D
3. Framework Qt w wersji 5

1.2 Implementowane funkcjonalności

1. Graficzny interfejs użytkownika (*GUI*), umożliwiający wygodną obsługę aplikacji
2. Dane wejściowe (formuły logiczne) wprowadzane ręcznie w postaci ciągu znaków, lub z pliku tekstowego (składnia wyrażeń opisana w 3.1)
3. Wizualizacja formuły wejściowej w postaci drzewa
4. Wizualizacja przeprowadzonego procesu dekompozycji formuły w postaci drzewa
5. Interpretacja wynikowego drzewa dostarczająca informacji o spełnialności formuły
6. Możliwość edycji stosowanych operatorów

2 Obsługa aplikacji

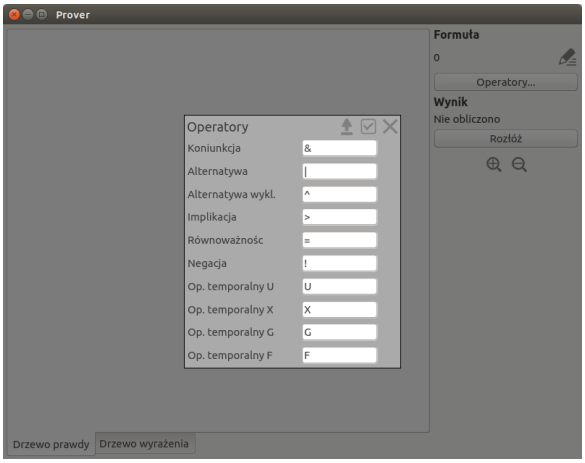


Rysunek 1: Widok głównego okna programu

1. Obszar rysowania drzewa
2. Bieżąca formuła logiczna
3. Przycisk otwierający okno wprowadzania formuły
4. Przycisk definicji operatorów
5. Informacja o wyniku działania algorytmu
6. Przycisk inicjalizacji dekompozycji
7. Widok na drzewo dekompozycji
8. Widok na drzewo formuły

2.1 Przykład dekompozycji

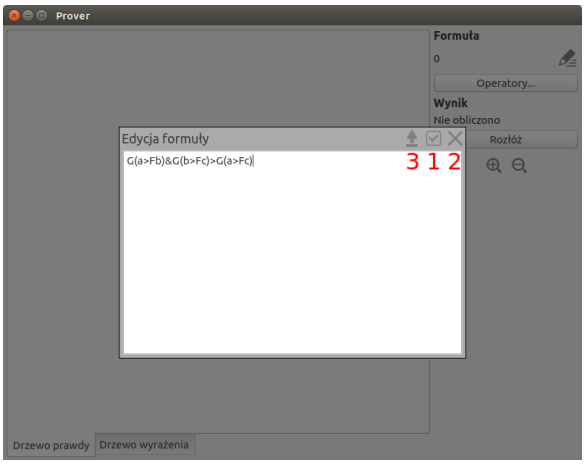
2.1.1 Wprowadzenie formuły



Rysunek 2: Okno definicji operatorów

Program posiada możliwość edycji stosowanych operatorów logicznych i temporalnych. Powyżej przedstawiono zestaw domyślnych operatorów.

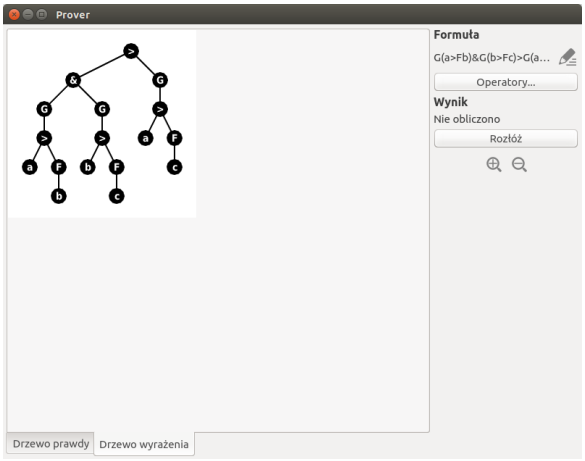
2.1.2 Wprowadzenie formuły



Rysunek 3: Okno wprowadzania formuły

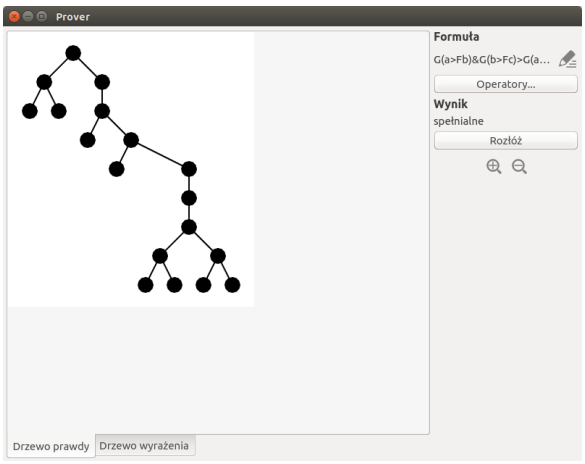
Wprowadzana formuła powinna zawierać symbole operatorów zgodne ze zdefiniowanymi. Po zakończeniu edycji należy zaakceptować (1) lub odrzucić (2)

zmiany. Istnieje także możliwość wczytania formuły z pliku (3). Po zdefiniowaniu formuły wejściowej użytkownikowi zostanie zaprezentowana jej wizualizacja w postaci drzewa:



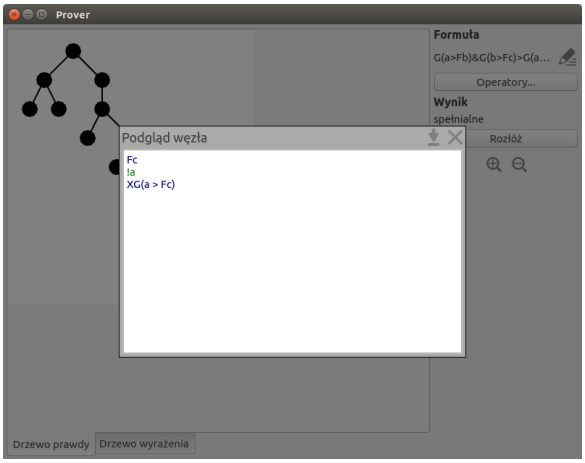
Rysunek 4: Wizualizacja formuły logicznej

2.1.3 Dekompozycja



Rysunek 5: Przedstawienie wyników

Po zakończeniu działania algorytmu program wyświetli uzyskane drzewo dekompozycji oraz informację o spełnialności formuły wejściowej. Kliknięcie na węzeł powoduje wyświetlenie jego zawartości:



Rysunek 6: Zawartość przykładowego węzła w drzewie dekompozycji

3 Analiza leksykalna

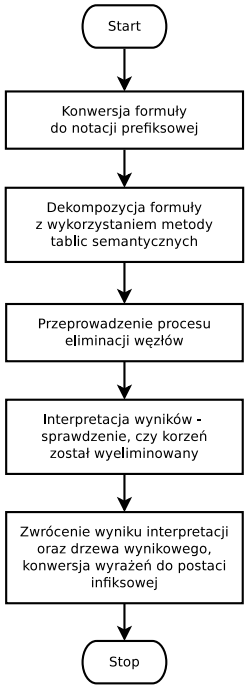
3.1 Składnia formuły wejściowej

Ciąg znaków	**Znaczenie**
Ciąg znaków alfanumerycznych	Zmienna logiczna rozumiana jako formuła atomowa
![formuła]	Jednoargumentowy operator negacji
[formuła] & [formuła]	Dwuargumentowy operator koniunkcji logicznej
[formuła] | [formuła]	Dwuargumentowy operator alternatywy logicznej
[formuła] ^ [formuła]	Dwuargumentowy operator alternatywy wykluczającej
[formuła] > [formuła]	Dwuargumentowy operator implikacji logicznej
[formuła] = [formuła]	Dwuargumentowy operator ekwiwalencji logicznej
X[formuła]	Jednoargumentowy operator temporalny *Next*
F[formuła]	Jednoargumentowy operator temporalny *Finally*
G[formuła]	Jednoargumentowy operator temporalny *Globally*
[formuła] U [formuła]	Dwuargumentowy operator temporalny *Until*
([formuła])	Nawiasy okrągłe - grupowanie wyrażeń

3.2 Hierarchia operatorów

Operator	Priorytet
Negacja, jednoargumentowe operatory temporalne	Najwyższy
Dwuargumentowy operator temporalny *Until*	Wysoki
Koniunkcja, alternatywa, alternatywa wykluczająca	Średni
Implikacja, Ekwiwalencja	Najniższy

4 Algorytm działania



Rysunek 7: Przebieg głównego algorytmu

4.1 Dekompozycja formuły

4.1.1 Reguły dekompozycji

Postać dekompozycji	**Nazwa**
$p \wedge q$ $\vert$ p q	Koniunkcja
$p \vee q$ $\diagup\diagdown$ $p \quad q$	Alternatywa
$p \Rightarrow q$ $\diagup\diagdown$ $\neg p \quad q$	Implikacja
$p \Leftrightarrow q$ $\diagup\diagdown$ $p \quad \neg p$ $q \quad \neg q$	Ekwiwalencja
$\neg(p \wedge q)$ $\diagup\diagdown$ $\neg p \quad \neg q$	Negacja koniunkcji
$\neg(p \vee q)$ $\vert$ $\neg p$ $\neg q$	Negacja alternatywy
$\neg(p \Rightarrow q)$ $\vert$ p $\neg q$	Negacja implikacji
$\neg(p \Leftrightarrow q)$ $\diagup\diagdown$ $p \quad \neg p$ $\neg q \quad q$	Negacja ekwiwalencji
$\neg\neg p$ $\vert$ p	Negacja negacji

$\neg Xp$ $\mid$ $X\neg P$	Negacja operatora *Next*
Fp $p \quad XFp$	Operator *Finally*
$\neg Fp$ $\mid$ $\neg p$ $\neg XFp$	Negacja operatora *Finally*
pUq $q \qquad p$ $X(pUq)$	Operator *Until*
$\neg(pUq)$ $\mid$ $\neg q$ $\neg p \vee \neg X(pUq)$	Negacja operatora *Until*
Gp $p \quad XGp$	Operator *Globally*

4.1.2 Algorytm metody tablic semantycznych

1. Tworzenie korzenia drzewa zawierającego formułę wejściową (lub jej zaprzeczenie)

2. Dekompozycja następnego w kolejności wyrażenia logicznego. Kolejność dekomponowania wyrażeń znajdujących się w węzłach drzewa jest dowolna, jednak ze względu efektywność algorytmu najlepiej przyjąć następujący porządek:

    1 Dekomponowanie wyrażeń niepowodujących rozgałęziania

    2 Dekomponowanie wyrażeń powodujących rozgałęzianie oraz tworzących dwa pod-wyrażenia w węzłach potomnych

    3 Dekomponowanie wyrażeń powodujących rozgałęzianie oraz tworzących jedno wyrażenie w węzłach potomnych

3. W przypadku węzła, który zawiera wyrażenia z operatorem *Next* - rozszerzenie drzewa, przez przyłączenie potomka zawierającego wyrażenia nieatomiczne, z usuniętym zewnętrznym operatorem *Next*, lub w przypadku gdy takie wyrażenie już występowało w nadrzędnym węźle, utworzenie ścieżki do tego węzła

4. Dekompozycję powtarza się, w każdym węźle nie zostaną sprawdzone wszystkie wyrażenia

5. Eliminacja węzłów - węzeł oznacza się jako usunięty, jeśli:

    1 Zawiera on dwie osobne formuły, które są zmienną logiczną i jej zaprzeczeniem

    2 Wszyscy potomkowie węzła zostali usunięci

    3 Jeśli węzeł zawiera operatory temporalne Fp lub qUp i jeśli nie istnieje w drzewie ścieżka prowadząca do węzła zawierającego formułę p

6. Interpretacja wyników - jeżeli korzeń drzewa został usunięty, to formuła jest niespełnialna (w przypadku, gdy w korzeniu zostało podane jej zaprzeczenie jest spełnialna)