

Rysunek 3.8: Główne okno programu

Aby poprawnie skonfigurować środowisko pracy dla wersji systemowej należy:

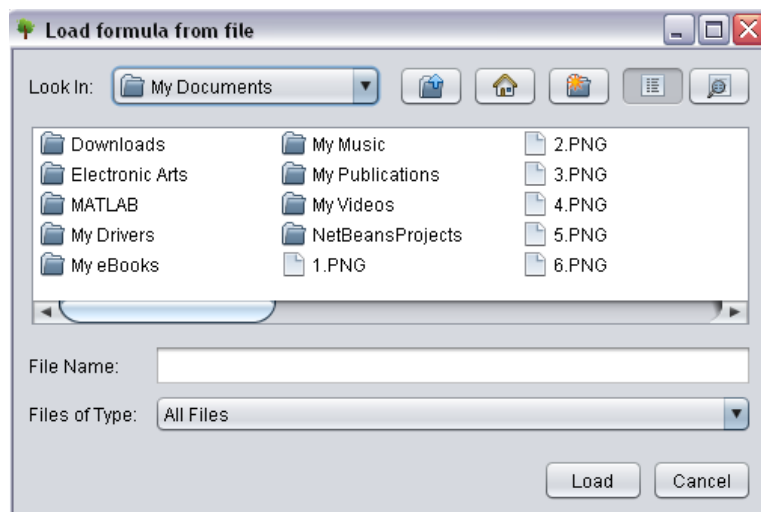
1. zainstalować środowisku uruchomieniowe Javy - Java JRE
2. rozpakować Prover z archiwum zip
3. program można uruchomić za pomocą
  - poprzez uruchomienie jara *The-Muffin-Project.jar*
  - poprzez uruchomienie skryptu *The-Muffin-Project.bat*
  - poprzez wykonanie skryptu *java -jar The-Muffin-Project.jar* w folderze programu

Poprawna konfiguracja wersji internetowej obejmuje jedynie rozpakowanie archiwum z Proverem. Należy zadbać aby użytkownicy zewnętrzni mieli odpowiednie prawa do katalogu, zawartych w nim plików.

## 3.6. Instrukcja użytkownika

### 3.6.1. Wersja systemowa

Po uruchomieniu programu wyświetla się okno z paskiem z trzema zakładkami jak na rysunku 3.8. Pracę z programem warto zacząć od wpisania formuły w odpowiednią sekcję, zaznaczoną numerem 1. Wpisaną formułę można zapisać do pliku tekstowego, aby użyć jej później, używając przycisku „Save in file”, umieszczonego w tej samej sekcji. Wyrażenie można też prosto usunąć używając przycisku „Clear”. Kolejną opcją jest wczytanie formuły z pliku, używając przycisku „Load from file”, znajdującym się w sekcji 1. Po jego naciśnięciu wyświetli się okno jak na rysunku 3.9, w którym należy wybrać plik



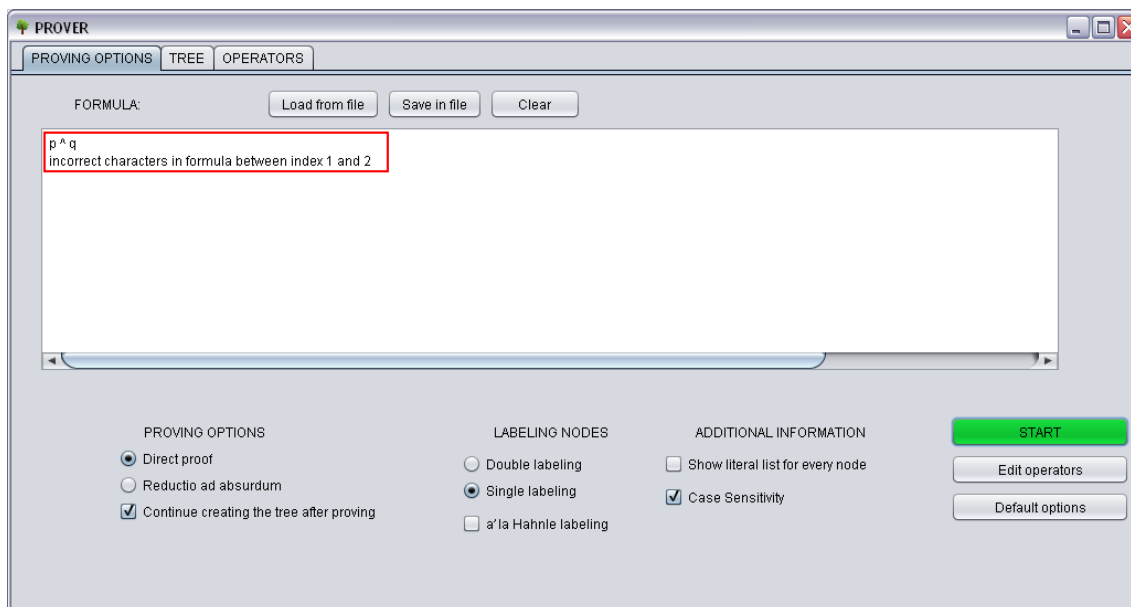
Rysunek 3.9: Okno ładowania plików

tekstowy z zapisaną wcześniej formułą i potwierdzić przyciskiem „load”. Po poprawnym załadowaniu formuła powinna pokazać się w odpowiedniej sekcji. W ten sam sposób działają wszystkie okna wyboru pliku umieszczone w prezentowanej aplikacji.

Program umożliwia przeprowadzenie dowodu wprost i dowodu nie wprost (apagogicznego), opcje wyboru znajdują się w sekcji drugiej. Domyślnie program przeprowadza dowód wprost. W tej samej sekcji znajduje się również opcja poprawiająca optymalizację tworzenia drzewa wnioskowania. Przy zaznaczonej opcji „Continue creating the tree after proving” algorytm dekomponuje całą formułę, przy odznaczeniu jej dekompozycja gałęzi zostaje przerwana w momencie znalezienia sprzeczności. Po znalezieniu sprzeczności w gałęzi znany jest jej stan, więc dalsza dekompozycja nie jest konieczna do określenia spełnialności formuły jak opisano w rozdziale 2.3.

W sekcji trzeciej znajdują się opcje etykietowania węzłów tworzonego grafu. Program oferuje etykietowanie pojedyncze („Single labeling”), w którym każdy węzeł ma swój numer porządkowy, podwójne („Double labeling”), gdzie węzeł ma zarówno swój numer porządkowy jak i numer rodzica w postaci:  $[i, j]$  gdzie  $i$  to numer węzła, a  $j$  to numer rodzica. Można również użyć etykietowania światów („a’la Hahnle labeling”). Sekcja czwarta zawiera dodatkowe opcje. Pierwsza („Show literal list for every node”) po zaznaczeniu umieszcza w każdym węźle listę aktualnych literalów. Druga („Case sensitivity”) rozpoznaje wielkość liter, czyli sprawia że atomy takie jak np.  $P$  i  $p$  są traktowane jak różne literały. Ostatnia, piąta sekcja, zawiera przyciski, pierwszy „Start”, dla ułatwienia w kolorze zielonym, uruchamia proces udowadniania formuły, drugi „Edit operators” umożliwia przejście do edycji operatorów, a ostatni „Default options” pozwala użytkownikowi powrócić do domyślnych opcji programu.

Przed uruchomieniem algorytmu dowodzenia wyrażenie jest sprawdzane w poszukiwaniu błędów syntaktycznych takich jak: niedomknięte nawiasy, złe symbole operatorów. Na ilustracji 3.10 pokazano przykład takiego błędu, użyto niedozwolonego symbolu operatora, program zwrócił informację o błędnej formule i podał indeks błędnego znaku. Użytkownik następnie powinien poprawić formułę i ponownie kliknąć przycisk „Start”.



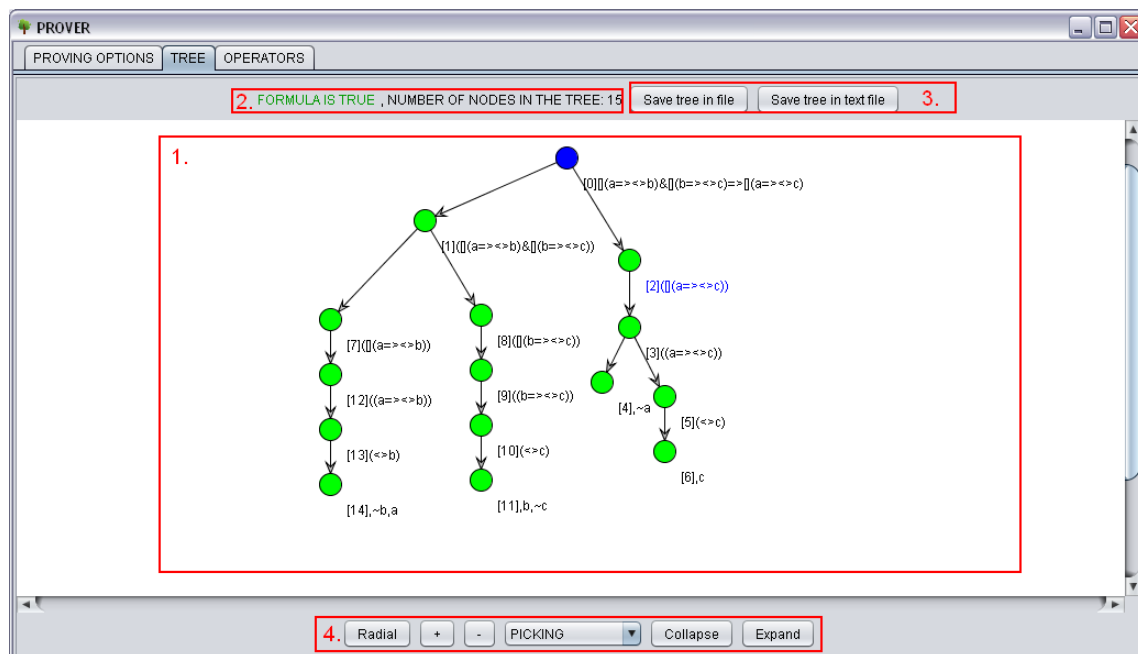
Rysunek 3.10: Okno programu z niepoprawną formułą

Po kliknięciu przycisku „Start” rozpoczyna się analiza formuły za pomocą metody tablic semantycznych, po jej zakończeniu program automatycznie przechodzi do zakładki drugiej, w której przedstawione jest drzewo wnioskowania i powiązane z nim opcje. Ilustracja 3.11 przedstawia drzewo wnioskowania utworzone dla formuły:

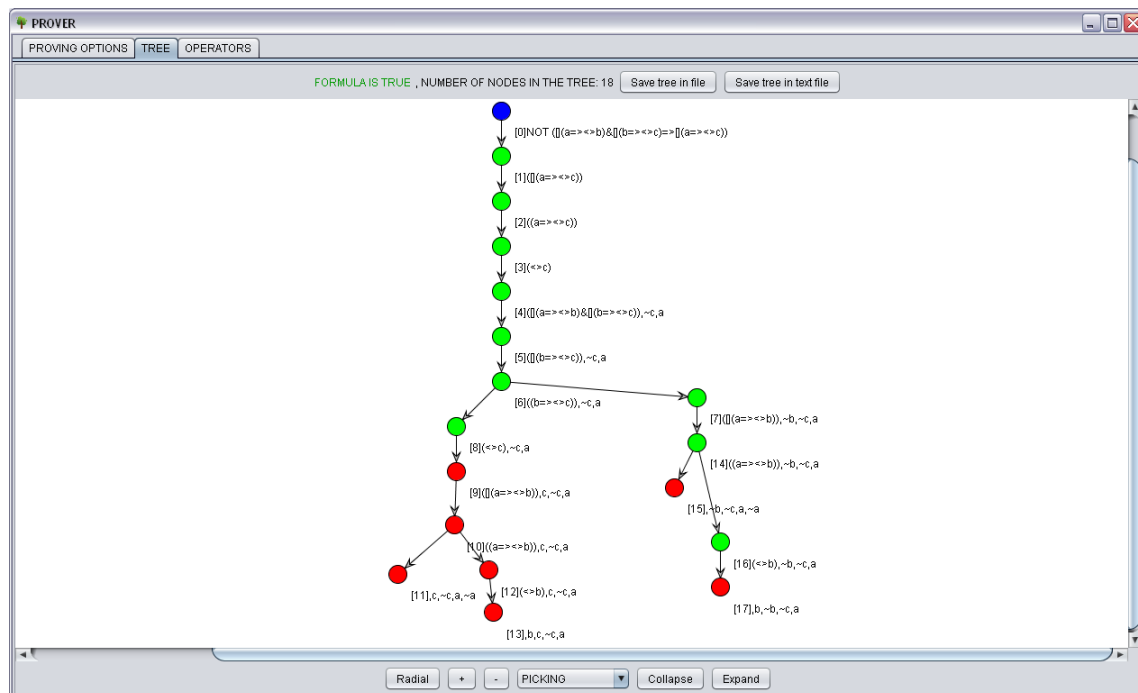
$$\Box(a \Rightarrow \Diamond b) \wedge \Box(b \Rightarrow \Diamond c) \Rightarrow \Box(a \Rightarrow \Diamond c)$$

przy pomocy dowodu wprost a ilustracja 3.12 przy pomocy dowodu nie wprost. Sekcja zaznaczona numerem 2 na ilustracji 3.11 informuje użytkownika o wyniku dowodzenia, czyli czy formuła jest poprawna czy niepoprawna oraz o ilości węzłów w zbudowanym drzewie. Stworzone drzewo, widoczne w sekcji pierwszej, można zapisać w formacie GraphML, używając przycisku „Save tree in file” albo w pliku tekstowym za pomocą przycisku „Save tree in text file”, które umieszczone są w sekcji trzeciej. Węzły grafu mogą być w trzech kolorach: niebieskim, czerwonym i zielonym. Węzeł niebieski oznacza korzeń drzewa, czerwony kolor zaznacza zamknięte gałęzie, zielone są pozostałe węzły. Sekcja czwarta obejmuje opcje wyglądu drzewa. Przycisk „Radial” zmienia widok drzewa ze standardowego na koncentryczny, który jest przedstawiony na ilustracji 3.13. Naciśnięcie go drugi raz powoduje powrót do widoku standardowego. Aby ułatwić analizę drzewa ma ono zwijalne wierzchołki. Po zaznaczeniu wierzchołka i kliknięciu przycisku „Collapse” wszystkie węzły potomne zostaną zwinięte, tą zmianę można cofnąć klikając przycisk „Expand”. Przyciski „+” i „-” służą do powiększania i pomniejszania drzewa.

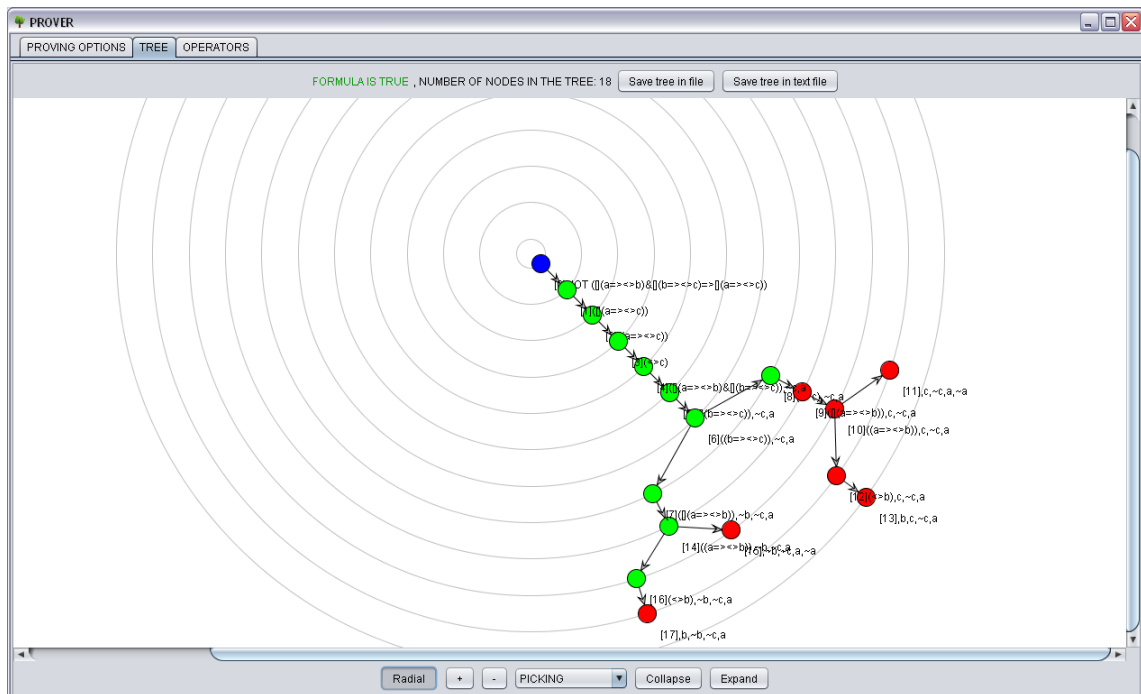
Po kliknięciu na przycisk „Edit operators” lub po wyborze trzeciej zakładki na pasku zakładek przechodzimy do widoku edycji operatorów, pokazanym na rysunku 3.14. W sekcji pierwszej widoczna jest nazwa pliku, w którym zapisane są aktualnie używane operatory. Plik ten można zmienić, co umożliwia korzystanie z kilku różnych plików, ze zdefiniowanymi odmiennymi operatorami, i wybieranie ich w zależności od tego, które operatory są aktualnie potrzebne do pracy. W sekcji oznaczonej numerem dwa umieszczona jest lista aktualnych operatorów, widoczne są ich nazwy. Operatory posortowane są w kolej-



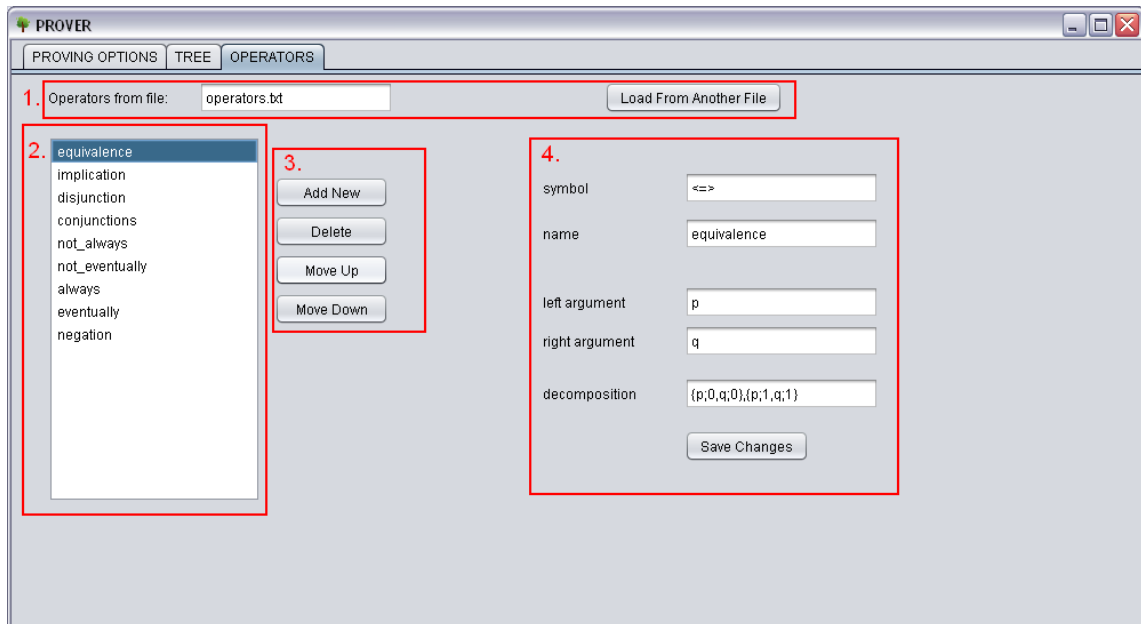
Rysunek 3.11: Zakładka z drzewem wnioskowania dla dowodu wprost



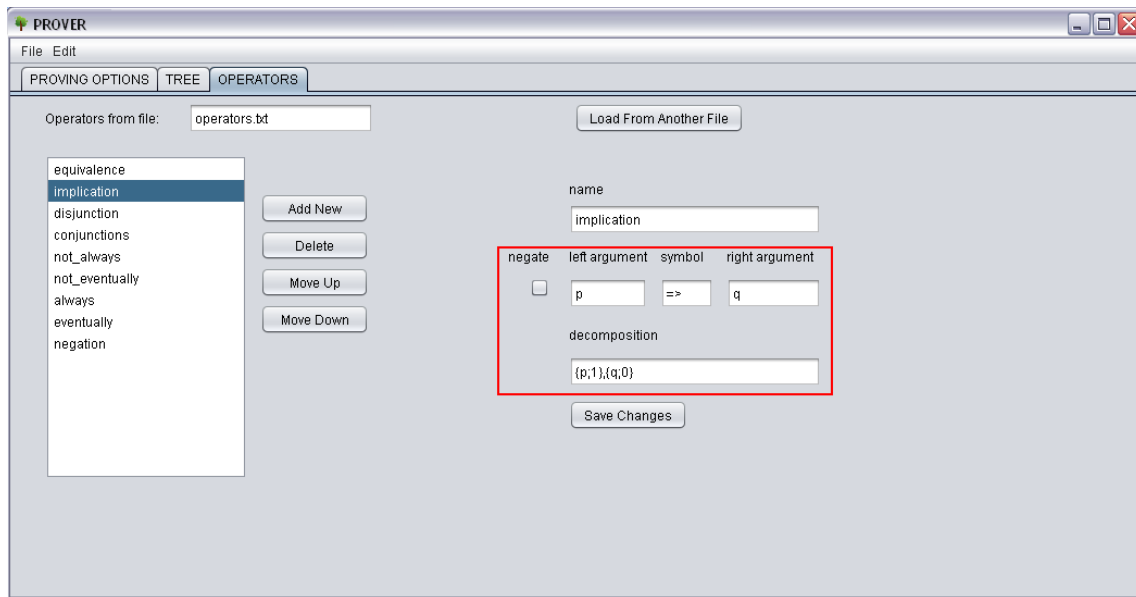
Rysunek 3.12: Drzewo wnioskowania przy wybraniu dowodu nie wprost



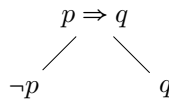
Rysunek 3.13: Widok koncentryczny drzewa wnioskowania



Rysunek 3.14: Zakładka z edycją operatorów



Rysunek 3.15: Zakładka z edycją operatorów



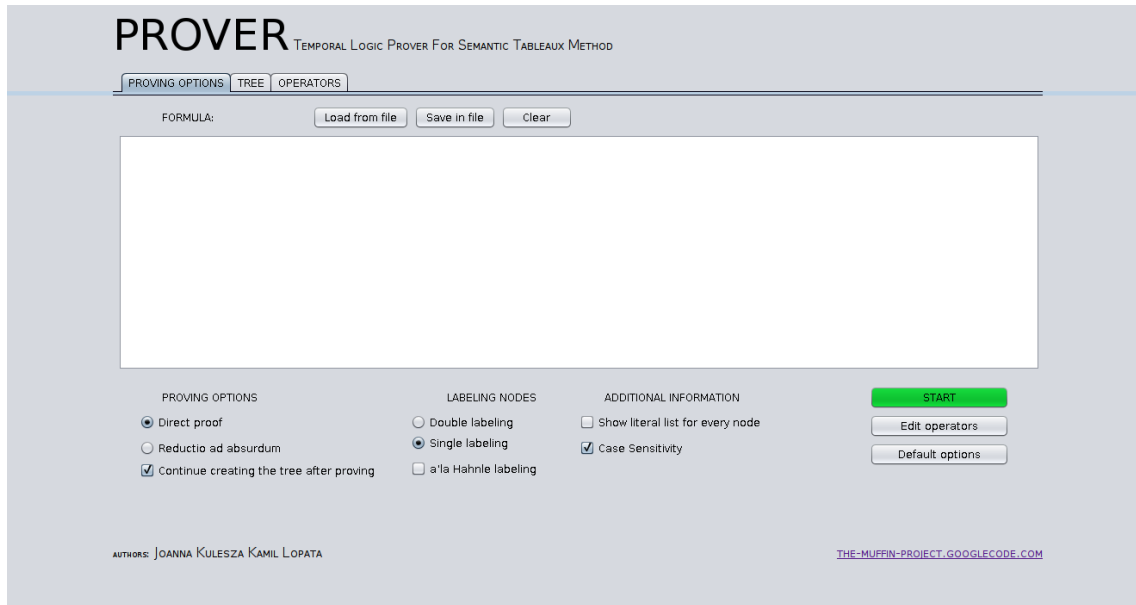
Rysunek 3.16: Dekompozycja implikacji

ności od tych o najwyższym priorytecie do tych o najniższym. Priorytet decyduje o tym, który operator zostanie najpierw poddany dekompozycji. Kolejność tą można zmieniać używając przycisków „Move Up” i „Move Down” z sekcji drugiej. Po wybraniu nazwy operatora, którego chcemy przesunąć klikamy „Move Up” i operator przesunie się na wyższe miejsce na liście, a co za tym idzie otrzyma wyższy priorytet. W celu usunięcia operatora zaznaczamy go i klikamy na przycisk „Delete”. Po zaznaczeniu nazwy operatora zostaje on wyświetlony w sekcji czwartej. Wyświetlony zostaje jego symbol, nazwa i sposób dekompozycji. W tej sekcji można dokonywać zmian we właściwościach operatora, które później należy potwierdzić przyciskiem „Save Changes”. Nowy operator dodaje się naciskając przycisk „Add New”, umieszczony w sekcji trzeciej. Wpisywanie nazwy i symbolu operatora jest prostym zadaniem, jedyny problem może stanowić zapisywanie reguły jego dekompozycji, co zostanie omówione poniżej.

Jak wspomniano w rozdziale 2.3 usuwanie spójników logicznych z formuły wiąże się z umieszczaniem jej części w odpowiednich węzłach drzewa. Przykładowy zapis reguły dekompozycji umieszczono na rysunku 3.15. Przedstawia on dekompozycję implikacji, przy pomocy własności logicznej:

$$p \Rightarrow q \longleftrightarrow \neg p \vee q$$

Efektom zastosowania takiej reguły jest stworzenie dwóch nowych węzłów o tym samym przodku (alternatywa), pierwszy argument jest zanegowany a drugi jest w formie podstawowej, jak widać na poglądowej ilustracji 3.16. Można zauważyć że w polu „decomposition” zapisuje się utworzone węzły, gdzie zapis  $\{p;0\},\{q;0\}$  tworzy dwa węzły ( $p$  i  $q$ ) o tym samym przodku (przykładem jest de-



Rysunek 3.17: Wizualizacja internetowej wersji Provera

kompozycja alternatywy) a  $\{p;0,q;0\}$  tworzy dwa węzły  $(p,q)$  w jednej gałęzi (jak przy dekompozycji koniunkcji).  $P, q$  są symbolicznym oznaczeniem dla prawego i lewego argumentu operatora. Po symbolu argumentu, po średniku zapisuje się czy jest on zanegowany czy nie, 0 symbolizuje argument niezanegowany a 1 zanegowany. Przykładowo,  $\{p;1\}, \{q;0\}$  stworzy zanegowany węzeł  $p$  i niezanegowany węzeł  $q$ . Ustalając regułę dla zanegowanych operatorów należy zaznaczyć opcję „negate”. Przykładem takiego operatora będzie zanegowana implikacja, którą można dekomponować za pomocą własności:

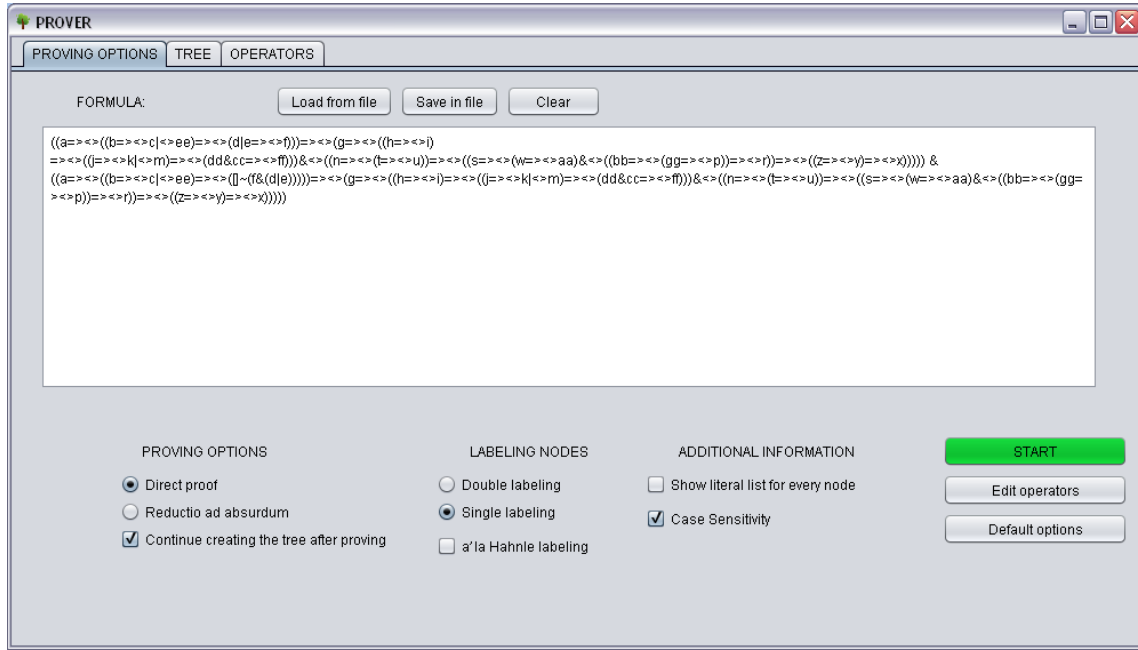
$$\neg(p \Rightarrow q) \leftrightarrow p \wedge \neg q$$

### 3.6.2. Wersja przeglądarkowa

Zgodnie z założeniami przy wyborze technologii wersja internetowa jest zbudowana w oparciu o aplikację. Sposób wygląd tej wersji programu jest rozwinięciem wersji systemowej o nagłówek zawierający nazwę, oraz stopkę zawierającą hiperłącze do strony projektu, co jest przedstawione na rysunku 3.17.

Funkcjonalność oraz sposób obsługi pozostała ta sama co wersji systemowej, z wyjątkiem pliku operatorów. Użytkownik nie może nadpisać przykładowego zestawu operatorów, ale może wybrać własny plik z zestawem operandów, który będzie mógł w pełni modyfikować. Wynika to z bezpieczeństwa - aby nierównoważny klient aplikacji nie utrudnił innym użytkownikom pracy z systemem.

Wersja internetowa podobnie jak przeglądarkowa wymaga do działania zainstalowanej maszyny wirtualnej Javy. W wypadku niespełnienia tego wymogu przeglądarka wyświetli link do pobrania Javy lub poinformuje o braku odpowiedniej wtyczki.



Rysunek 3.18: Ekran startowy z wpisaną formułą

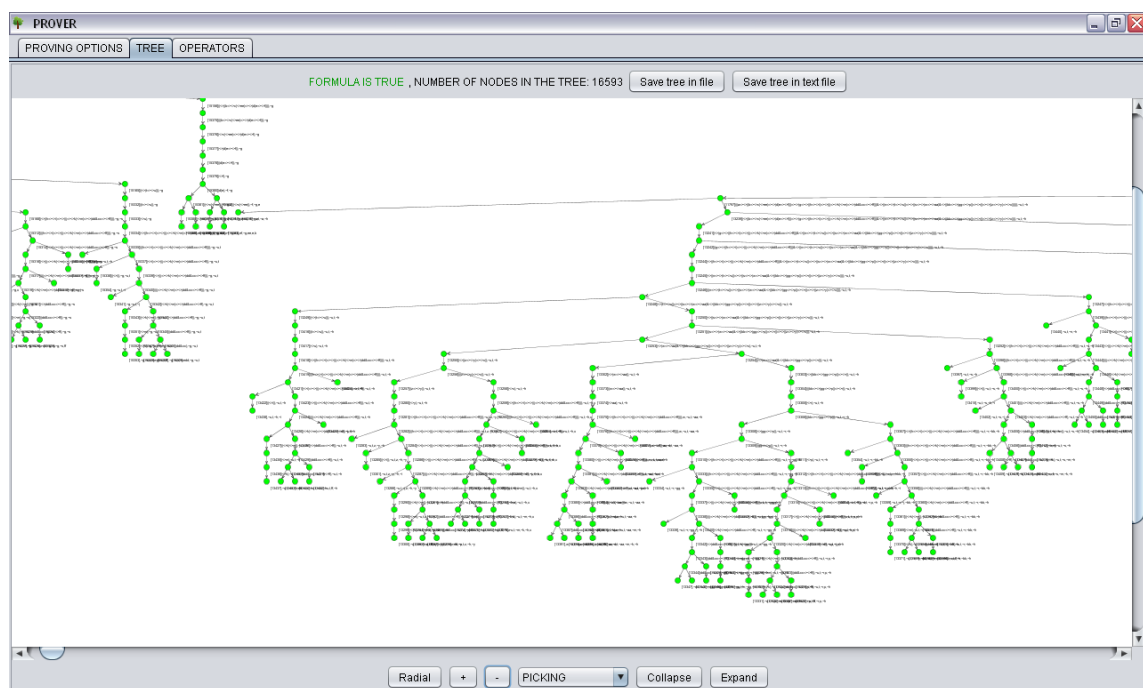
### 3.7. Przykład użycia programu

Pracę z programem zaprezentowano na przykładzie udowadniania długiej formuły, czyli takiej o której spełnialności ciężko jest wnioskować bez użycia aplikacji automatycznego dowodzenia. Używana formuła zawiera operatory logiki klasycznej i temporalnej.

$$\begin{aligned}
 & ((a \Rightarrow \Diamond((b \Rightarrow \Diamond c \vee \Diamond ee) \Rightarrow \Diamond(d \vee e \Rightarrow \Diamond f))) \Rightarrow \Diamond(g \Rightarrow ((h \Rightarrow \Diamond i) \Rightarrow \\
 & \Diamond((j \Rightarrow \Diamond k \vee \Diamond m) \Rightarrow (dd \wedge cc \Rightarrow \Diamond ff))) \wedge \Diamond((n \Rightarrow (t \Rightarrow \Diamond u)) \Rightarrow \Diamond((s \Rightarrow \\
 & \Diamond(w \Rightarrow aa) \wedge \Diamond((bb \Rightarrow \Diamond(gg \Rightarrow \Diamond p)) \Rightarrow \Diamond r)) \Rightarrow \Diamond((z \Rightarrow \Diamond y) \Rightarrow \Diamond x)))) \wedge \\
 & ((a \Rightarrow \Diamond((b \Rightarrow \Diamond c \vee \Diamond ee) \Rightarrow \Diamond(\Box \neg(f \wedge (d \vee e)))))) \Rightarrow \Diamond(g \Rightarrow \Diamond((h \Rightarrow \Diamond i) \Rightarrow \\
 & \Diamond((j \Rightarrow \Diamond k \vee \Diamond m) \Rightarrow \Diamond(dd \wedge cc \Rightarrow \Diamond ff))) \wedge \Diamond((n \Rightarrow \Diamond(t \Rightarrow \Diamond u)) \Rightarrow \Diamond((s \Rightarrow \\
 & \Diamond(w \Rightarrow \Diamond aa) \wedge \Diamond((bb \Rightarrow \Diamond(gg \Rightarrow \Diamond p)) \Rightarrow \Diamond r)) \Rightarrow \Diamond((z \Rightarrow \Diamond y) \Rightarrow \Diamond x))))
 \end{aligned}$$

Na początek wpisano formułę w odpowiednie miejsce w programie i ustawiono opcje jak widać na ilustracji 3.18. Wybrano dowód wprost i pojedyncze etykietowanie węzłów. Po naciśnięciu przycisku start uruchamia się procedura udowadniania. Po jej zakończeniu następuje przejście do zakładki drugiej, w której jest widoczny rezultat badania spełnialności formuły oraz stworzone drzewo wnioskowania jak na rysunku 3.19. Formuła okazała się poprawna, gdyż przy zastosowaniu dowodu wprost drzewo formuł miało przynajmniej jedną gałąź otwartą. Utworzone drzewo miało 16593 węzłów, w związku z tym na ilustracji 3.19 przedstawiono jedynie jego część. Na tym etapie można zapisać graf w formacie GraphML używając przycisku „Save tree in file”, albo zmienić wygląd drzewa na widok koncentryczny.





Rysunek 3.19: Drzewo wnioskowania dla formuły