Informatik-Praktikum für Fortgeschrittene

Fachbereich Mathematik und Informatik

Philipps-Universität Marburg

Im Wintersemester 2003/2004

Erweiterung des Benutzerhandbuchs zum

Java Program Verifier Version 2.0

von

Uli Schäfer, Christian Hohmann, Silvia Ockenfels und Manuela Viehmeyer

Basiert auf:

Benutzerhandbuch zum

Java Program Verifier Version 1.0

von

Markus Hampel und Manuel Werner

betreut von:

Professor Dr. H. Peter Gumm und Jörn Abels

Inhaltsverzeichnis

1 Installation und Programmstart

Der *Java Program Verifier* (JPV) liegt in Form des ausführbaren jar-Files „JPV.jar“ vor. Dieses kann auf Systemen mit installiertem *Java Runtime Environment* ausgeführt werden, also sowohl auf Windows als auch auf Unixsystemen. JPV benötigt keine Installation, das Programm kann direkt mit dem Kommandozeilenbefehl

java -jar JPV.jar

gestartet werden. Danach präsentiert JPV das folgende Fenster:

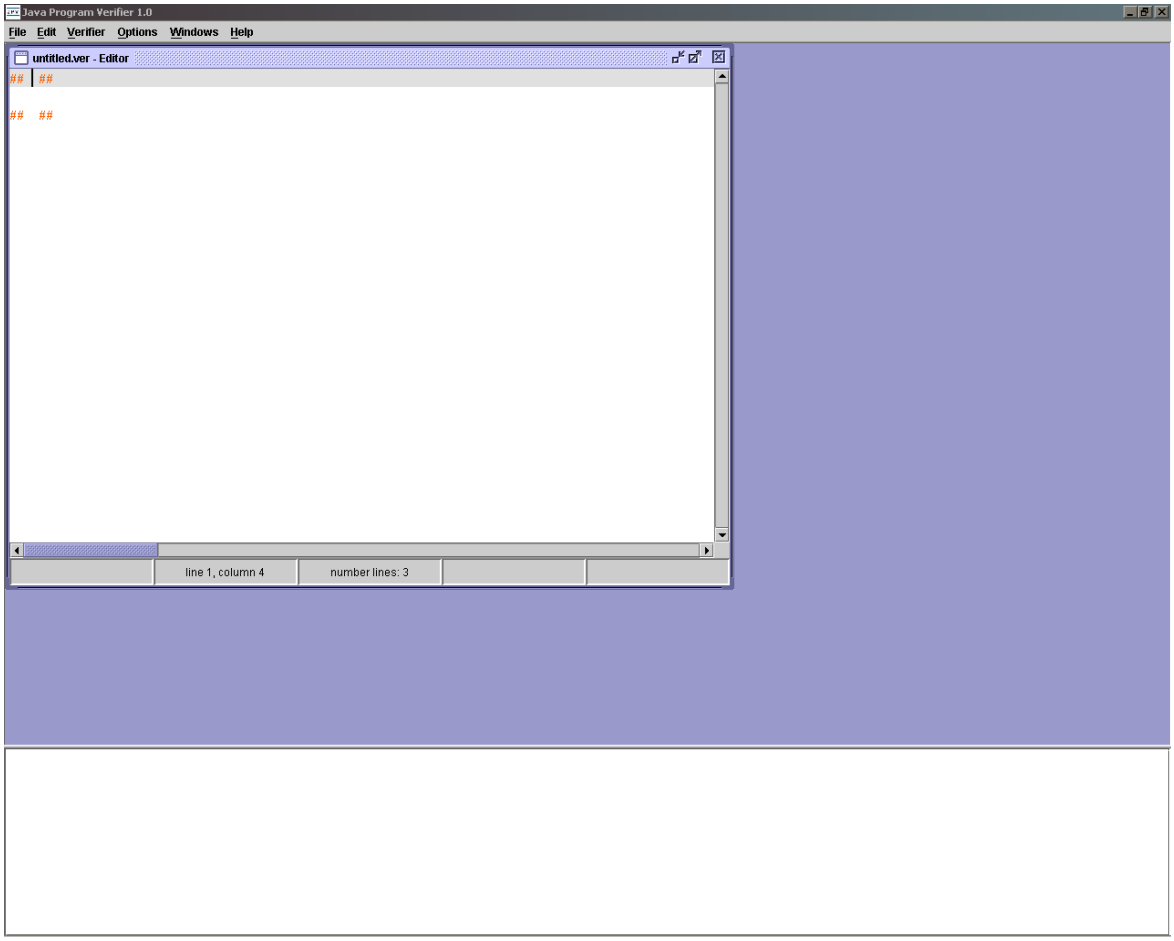


Abbildung 1: Startfenster des *JPV*

2 Die graphische Benutzerschnittstelle

Das in Abbildung 1 dargestellte Hauptfenster des *JPV* besteht im Wesentlichen aus den folgenden Bestandteilen:

- Menü
- Arbeitsbereich
- Ausgabefeld

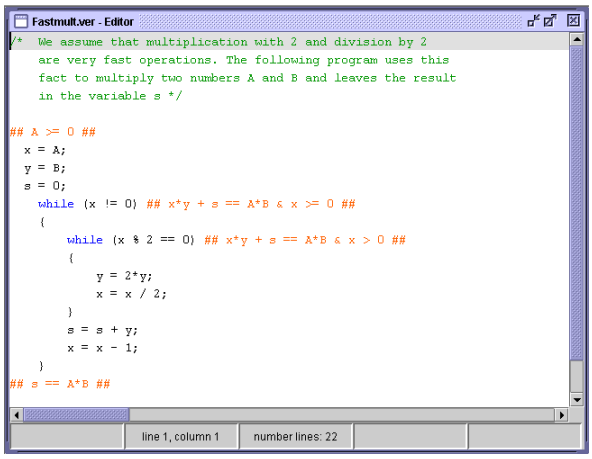
Das Menü und der Arbeitsbereich werden in den beiden nachstehenden Unterkapiteln beschrieben.

Das Ausgabefeld dient zum Anzeigen von Meldungen. Abbildung 2 zeigt dieses mit einigen beispielhaften Meldungen.

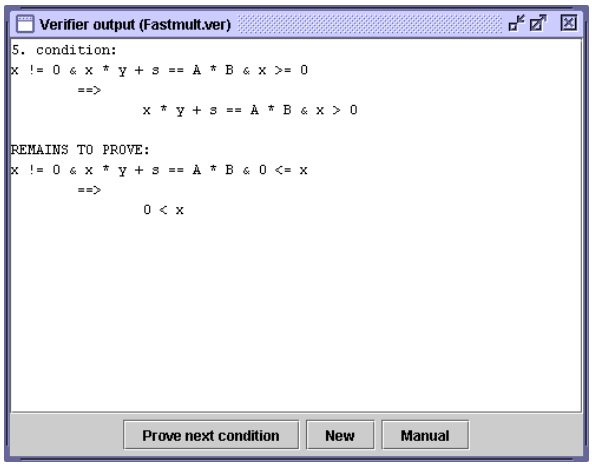
Abbildung 2: Ausgabefeld

2.1 Der Arbeitsbereich

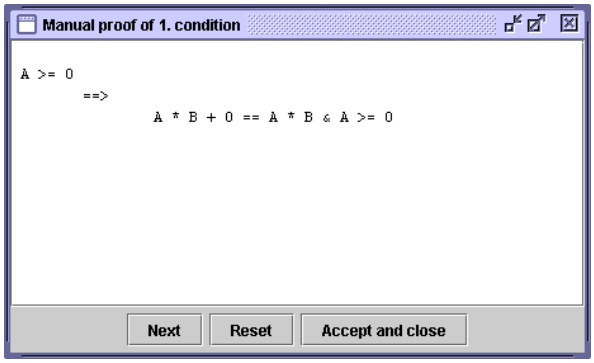
Der Arbeitsbereich befindet sich im Hauptfenster zwischen der Menüleiste und dem Ausgabefeld. Er beherbergt verschiedene Typen interner Fenster. Es folgt nun eine Beschreibung dieser Typen.



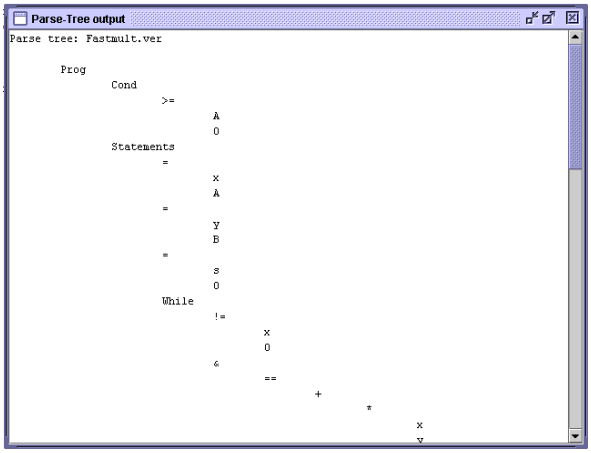
Das Editorfenster ermöglicht das Bearbeiten und Eingeben von Programmen. Die Schlüsselwörter der Eingabesprache, wie zum Beispiel `while`, werden dabei farblich hervorgehoben. Der Editor unterstützt die gängigen Features, wie Ausschneiden, Kopieren und Einfügen. Des Weiteren beinhaltet er eine Statusleiste, welche die aktuelle Position des Cursors im Text und die Gesamtzeilenzahl anzeigt.



Das Ausgabefenster des Verifizierers (*Verifier output*) zeigt die vom automatischen Verifizierer produzierten Ergebnisse an. Außerdem stellt es drei Knöpfe zur Steuerung des Beweisprozesses bereit. Der Knopf *Prove Next Condition* erlaubt das sukzessive Beweisen der einzelnen Bedingungen. Der zweite Knopf *New* startet den gesamten Prozess von neuem, d.h. der Verifizierer springt zurück zur ersten Verifikationsbedingung. Ist der Knopf *Manual* aktiv, so konnte eine Bedingung nicht automatisch verifiziert werden und steht jetzt zur manuellen Bearbeitung zur Verfügung.



Das Fenster zur manuellen Bearbeitung von Verifikationsbedingungen stellt eine einzelne Verifikationsbedingung dar, die mit der weiter unten beschriebenen Vorgehensweise vereinfacht werden kann. Es stehen dem Anwender entweder zwei oder drei Knöpfe zur Verfügung, je nachdem, ob er sich im rein manuellen Beweismodus befindet, oder die manuelle Verifikation aus dem automatischen Modus heraus gestartet hat. Der Knopf *Next* ist nur im rein manuellen Modus verfügbar. Durch Betätigen dieses Knopfes wird die aktuelle Bedingung verworfen und, sofern vorhanden, die nächste Bedingung dargestellt. Der *Reset*-Knopf macht sämtliche manuell vorgenommene Veränderungen rückgängig, während der Knopf *Accept and close* das Fenster schließt.



Der Fenstertyp *Parse-Tree output* dient der Ausgabe des Parse-Tree und stellt die zugrunde liegenden Grammatik der Eingabesprache anschaulich dar. Dieses Fenster erhält man durch Auswahl der Option *Display Parse-Tree* im Menü *Verifier* bei ausgewähltem Ausgabefenster des Verifizierers.

2.2 Das Menü

Abbildung 3: Menüleiste

Die Menüleiste ist das zentrale Bedienelement des Verifizierers. Mittels dieser lassen sich beispielsweise zu verifizierende Programme öffnen, speichern und beweisen.

Das Menü File

Abbildung 4: Menü File

New	Erzeugt ein Editorfenster mit leerem Programmrumpf.
Open...	Zum Öffnen von Eingabeprogrammen, die in Dateien mit Endung *.ver* enthalten sind.
Save	Speichert das in dem markierten Editorfenster enthaltene Eingabeprogramm in einer bereits vorhandenen *ver*-Datei. Ist kein Editorfenster selektiert, so wird eine entsprechende Meldung im Ausgabefeld angezeigt.
Save As...	Zum Speichern eines in einem markierten Editorfenster enthaltenen Eingabe-programms in einer beliebigen *ver*-Datei. Ist kein Editorfenster selektiert, so wird eine entsprechende Meldung im Ausgabefeld angezeigt.
Exit	Beendet JPV.

Das Menü Edit

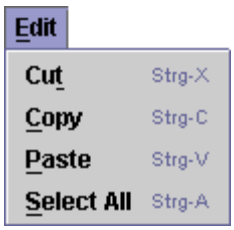


Abbildung 5: Menü Edit

Sämtliche im Menü Edit enthaltenen Menüpunkte funktionieren nur bei selektiertem Editorfenster. Sollte kein Editorfenster markiert sein, wird eine Meldung im Ausgabefeld angezeigt.

Cut	Schneidet markierten Text aus einem Editorfenster aus.
Copy	Kopiert markierten Text aus einem Editorfenster in die Zwischenablage.
Paste	Fügt Text aus der Zwischenablage in ein Editorfenster ein.
Select All	Markiert das gesamte Eingabeprogramm.

Das Menü Verifier

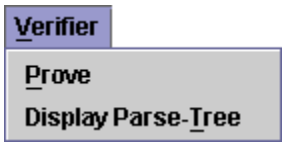


Abbildung 6: Menü Verifier

Prove	Startet zunächst den Parser, der das Eingabeprogramm auf syntaktische Korrektheit überprüft. Danach erzeugt der Verifizierer die Verifikationsbedingungen. Im automatischen Beweismodus wird nun direkt versucht, diese zu verifizieren. Ist hingegen im Menü *Options* der Eintrag *Manual Proof* aktiviert, werden die Verifikationsbedingungen unverändert zur rein manuellen Bearbeitung ausgegeben. Dies ist nur bei selektiertem Editorfenster möglich. Die Ausgaben des Parsers werden im Ausgabefeld angezeigt. Für die Ausgaben des Verifizierers hingegen wird ein separates Fenster geöffnet.
Display Parse-Tree	Zeigt den *Parse-Tree* eines erfolgreich geparsten Eingabeprogramms an. Dazu muss das zum gewünschten Eingabeprogramm gehörende Ausgabefenster des Verifizierers selektiert sein.

Das Menü Options

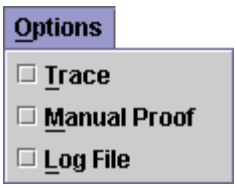


Abbildung 7: Menü Options

Trace	Bei aktiviertem *Trace* gibt der Verifizierer jeden Vereinfachungsschritt im Ausgabefenster an. Ist *Trace* deaktiviert, erfährt der Benutzer nur, ob die Verifizierung geglückt ist bzw., falls dies nicht der Fall ist, was noch zu beweisen ist. Eine Auswahl der *Trace*-Option während eines Verifikationsprozesses hat keine Auswirkung auf die derzeit dargestellte Verifikationsbedingung, d.h. ist beim Beweis einer Bedingung die *Trace*-Funktion deaktiviert, so wirkt sich eine Aktivierung der Funktion erst auf der Verifikation der nächsten Bedingung aus.
Manual Proof	Durch das Aktivieren von *Manual Proof* wechselt der Anwender in den rein manuellen Modus. Wird anschließend bei selektiertem Editorfenster der Eintrag *Prove* im *Verifier*-Menü gewählt, so werden die erzeugten Verifikations-bedingungen nicht durch das automatische Vereinfachungsverfahren verändert, sondern können der Reihe nach manuell bearbeitet werden.
Log File	Zum Speichern der Ausgaben des Verifizierers in einer Datei. Der Name dieser Datei stimmt mit dem Namen der *ver*-Datei überein, besitzt jedoch das Postfix *log* und wird im aktuellen Arbeitsverzeichnis gespeichert. Der Inhalt der Log-Datei wird unabhängig von der Auswahl dieser Option erzeugt. Die Ausgaben des Verifizierers werden in die Log-Datei übernommen. Bei eingeschalteter „Log File“ - Option wird die Datei beim Schließen des Ausgabefensters des Verifizierers erzeugt. Eine entsprechende Meldung wird im Ausgabefeld angezeigt.

Das Menü Windows

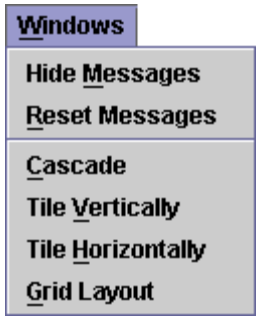


Abbildung 8: Menü Windows

Show/Hide Messages	Zum Ein- und Ausblenden des Ausgabefeldes.
Reset Messages	Löscht den Inhalt des Ausgabefeldes.
Cascade	Kaskadiert die internen Fenster.
Tile Vertically	Richtet die internen Fenster vertikal aus.
Tile Horizontally	Richtet die internen Fenster horizontal aus.
Grid Layout	Richtet die internen Fenster an einem Gitter aus.

Das Menü Help

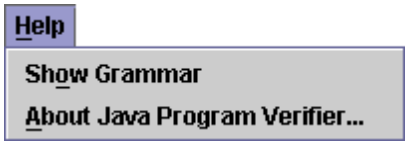


Abbildung 9: Menü Help

Show Grammar	Zeigt die zugrunde liegende Grammatik der Eingabesprache an.
About Java Program Verifier	Zeigt Programminformationen an.

3 Automatischer und manueller Verifikationsmodus

Wie aus der obigen Darstellung bereits klar geworden sein dürfte, ist die Verifikation eines gegebenen Programms in zweierlei Modi möglich.

Im automatischen Modus werden bei Auswahl des Eintrages *Prove* im Menü *Verifier* bei selektiertem Editorfenster die Verifikationsbedingungen erzeugt. Das Programm wendet sofort das automatischen Vereinfachungsverfahren auf die erste Verifikationsbedingung an und gibt anschließend aus, ob sie bewiesen werden konnte bzw., falls dies nicht möglich war, inwieweit sie vereinfacht werden konnte. Eine manuelle Bearbeitung der Bedingung ist in diesem Fall nur möglich, falls eine Bedingung nicht vollständig verifiziert werden konnte.

Der rein manuelle Modus wird durch Auswahl des Eintrages *Manual Proof* im *Options*-Menü aktiviert. Wird in diesem Modus der Eintrag *Prove* im Menü *Verifier* bei selektiertem Editorfenster ausgewählt, so wird die erste Verifikationsbedingung unverändert in einem speziellen Fenster ausgegeben und kann vom Anwender selbständig vereinfacht werden. Dieser kann nach Belieben die einzelnen Verifikationsbedingungen bearbeiten und zur Vereinfachung auch auf die automatische Routine zurückgreifen.

Zur Vereinfachung der Bedingungen kann der Anwender bestimmte Teilausdrücke der Formel markieren. Man markiert den zu vereinfachen Ausdruck anhand des entsprechenden Operators und drückt anschließend die rechte Maustaste. Es öffnet sich das PopUp-Menü *Anwendbare Formeln*, welches eine Liste derjenigen Ausdrücke enthält, zu denen der markierte Ausdruck umgewandelt werden kann.

Man beachte, dass sich nach Drücken der rechten Maustaste die Markierung automatisch an einen sinnvoll bearbeitbaren Ausdruck anpasst. Markiert der Anwender in der Formel

$$5 + k * 1 > 0 \Longrightarrow k >= -5$$

das *, so dehnt sich die Markierung nach einem Drücken der rechten Maustaste auf den Ausdruck $k * 1$ aus:

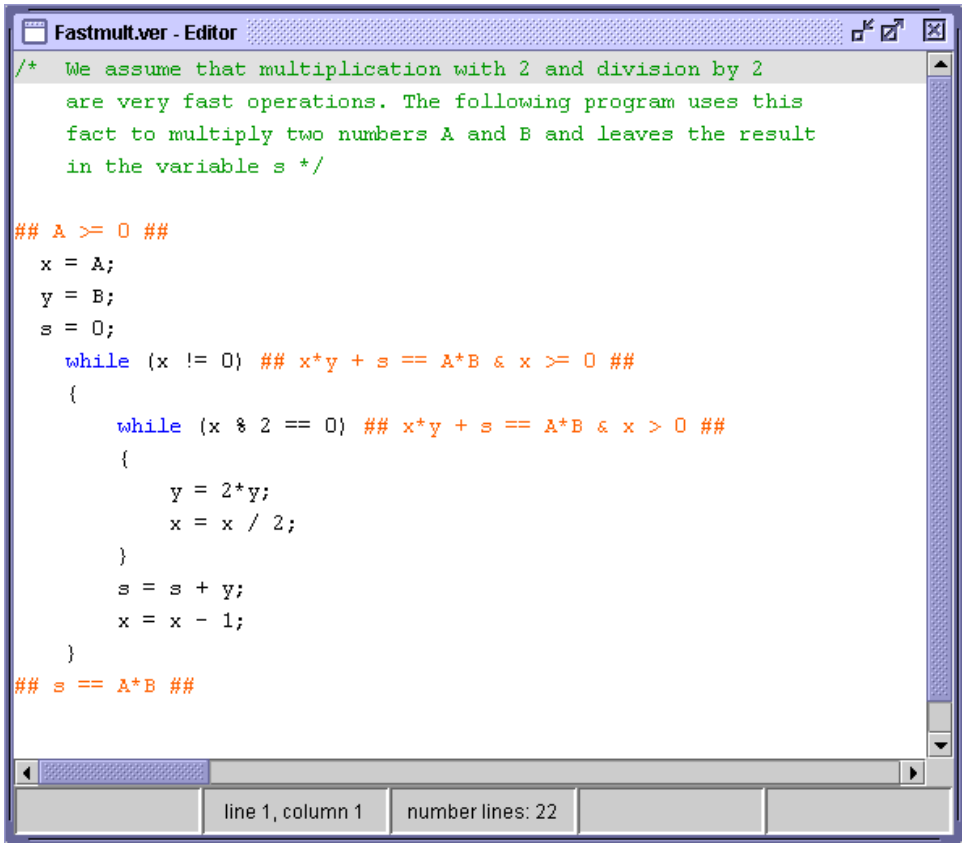
$$5 + k * 1 > 0 \Longrightarrow k >= -5$$

Sollte die angepasste Markierung nicht mit dem beabsichtigten Ausdruck übereinstimmen, dann sollte der Anwender den Vorgang wiederholen und darauf achten, nur die Operation des gewünschten Ausdruck mit der niedrigsten Präzedenz auszuwählen.

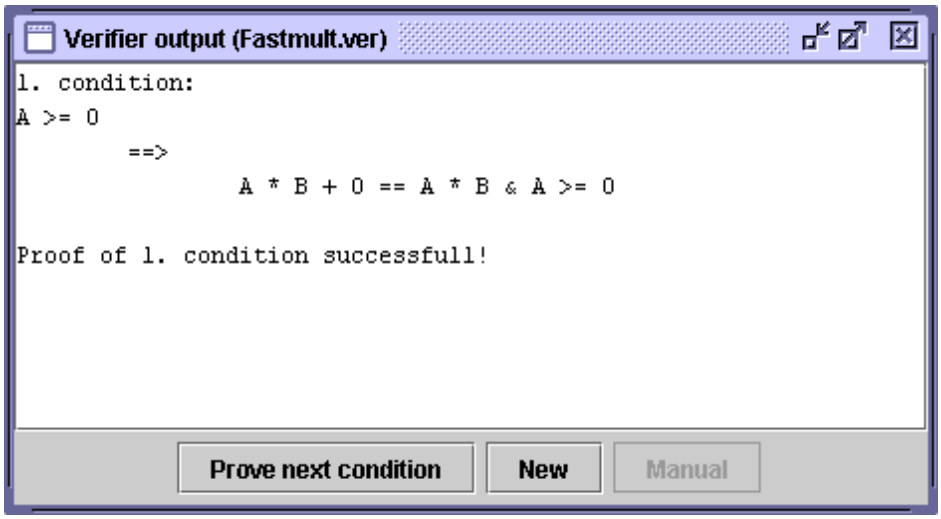
Das PopUp-Menü enthält immer den Menüpunkt *Evaluate*. Wird dieser ausgewählt, so wird das automatische Beweisverfahren auf den markierten Ausdruck angewendet, d.h. das Programm vereinfacht den Ausdruck selbständig mit den zur Verfügung stehenden Regeln (findet keine Veränderung statt, so konnte keine Regeln angewendet werden).

4 Eine Beispielsitzung

In diesem Abschnitt wird anhand des beiliegenden Beispielprogramms `fastmult.ver` die Funktionsweise des *Java Program Verifier* erläutert.

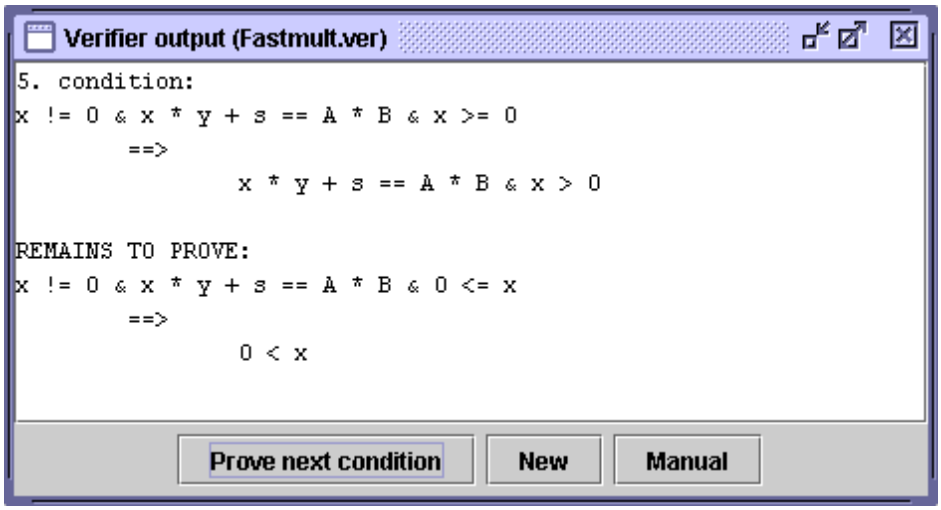


Nach dem Öffnen des Beispielprogramms `fastmult.ver` wird zunächst der automatische Verifikationsalgorithmus durch Auswahl des Eintrags *Prove* im Menü *Verifier* gestartet. Die ersten vier Verifikationsbedingungen werden vollständig automatisch bewiesen:

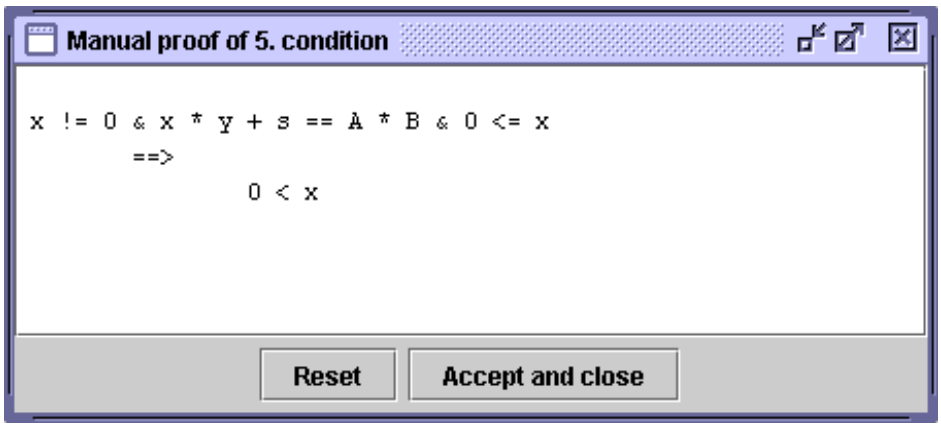


Durch Betätigung des *Prove next condition*-Knopfes kann man auf diese Art und Weise die folgenden drei Bedingungen abarbeiten, die sich alle automatisch verifizieren lassen

Die fünfte Verifikationsbedingung wird durch den automatischen Verifikationsalgorithmus zwar vereinfacht, kann aber nicht vollständig verifiziert werden:



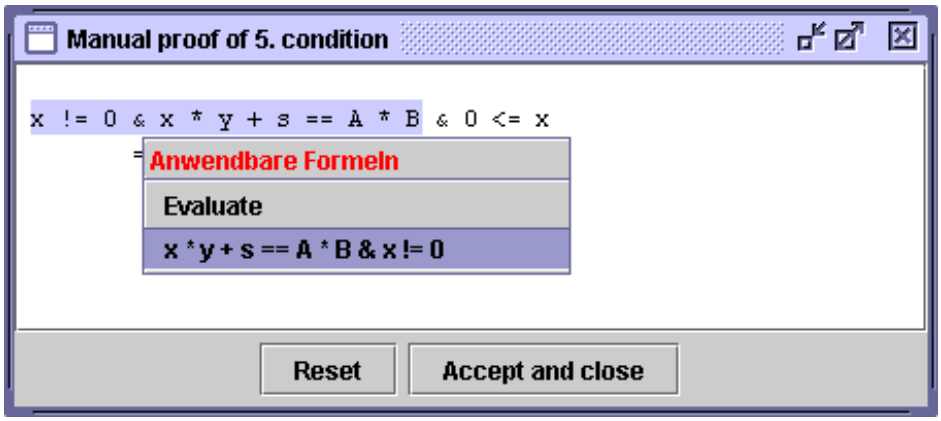
Der Button *Manual*, der bei den vorherigen Bedingungen jeweils deaktiviert war, ist nun aktiv. Wird er betätigt, so öffnet sich das Fenster *Manual proof of 5. condition*. Es enthält die vereinfachte Bedingung, die automatisch nicht weiter reduziert werden konnte.



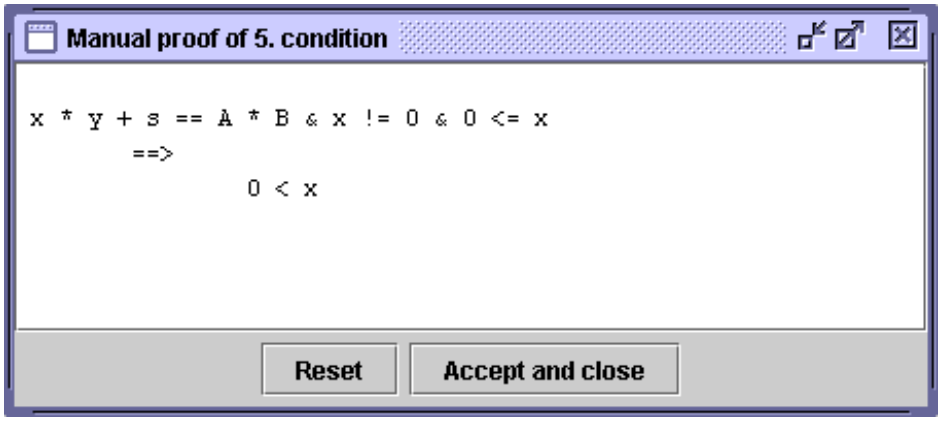
Betrachtet man in der ersten Zeile den ersten und letzten der drei mit & verknüpften booleschen Ausdrücke, so folgt aus `x != 0 & 0 <= x` natürlich direkt `0 < x`. Zunächst kommutiert man die &. Hierzu markiere das erste & und drücke die rechte Maustaste.

Es öffnet sich das PopUp-Menü *Anwendbare Formeln*, das in diesem Fall zwei Einträge aufweist: Zum einen den Standardeintrag *Evaluate*, der die Möglichkeit anbietet, den markierten Ausdruck so weit wie möglich durch das automatische Vereinfachungsverfahren zu reduzieren, zum anderen ein Vertauschen der beiden betroffenen Terme:

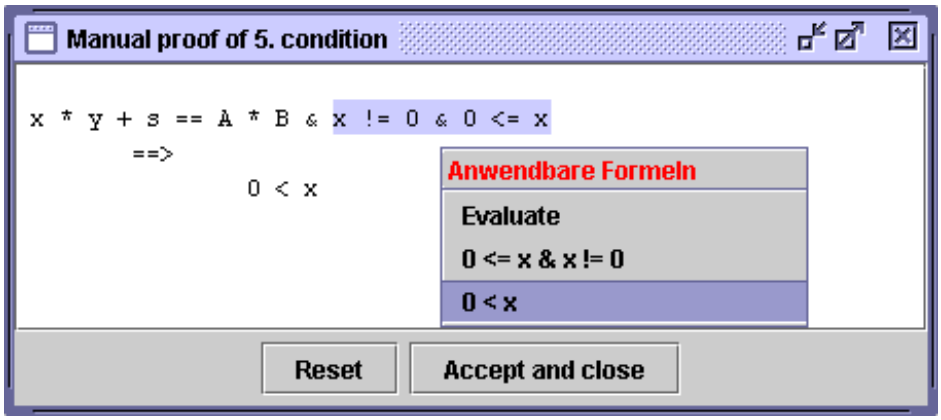
```
x * y + s == A * B & x != 0
```



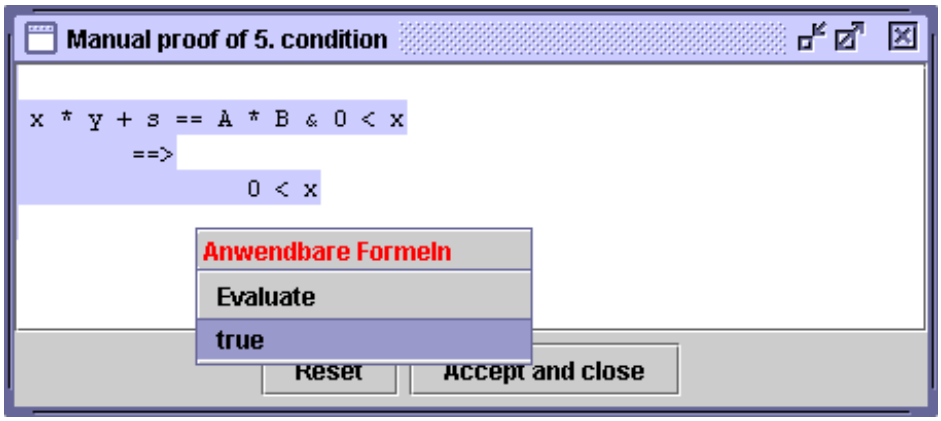
Wählt man den zweiten Eintrag, so kommutieren wie angegeben die ersten beiden booleschen Ausdrücke:



Nun lassen sich die beiden hinteren Ausdrücke `x != 0 & 0 <= x` zusammenfassen. Markiert man das verknüpfende & und drückt die rechte Maustaste, so öffnet sich erneut ein PopUp-Menü, das neben dem obligatorischen Eintrag *Evaluate* die Möglichkeit anbietet, entweder zu kommutieren oder `x != 0 & 0 <= x` zu `0 < x` zu vereinfachen.



Wählt man den letzten Eintrag, so ist es offensichtlich, dass die Bedingung wahr ist. Durch ein Markieren der Implikation (und damit der gesamten Bedingung) und ein erneutes Aufrufen des PopUp-Menüs mit der rechten Maustaste, lässt sich die gesamte Bedingung zu *true* vereinfach, d.h. sie wurde verifiziert.



Durch das Betätigen des Buttons *Accept and close* kann nun das Fenster zur manuellen Verifizierung geschlossen werden, und man kann mit den restlichen Verifikationsbedingungen auf analoge Weise fortfahren.

Die Bedienung des rein manuelle Beweismodus erfolgt auf dieselbe Art und Weise, weshalb in diesem Handbuch nicht weiter darauf eingegangen wird.