Arbeitstagebuch

Bachelorarbeit 2017

Timo Bergerbusch

Die ist ein Arbeitstagebuch um den Überblick über bereits geleistete Arbeit zu behalten und Probleme und Änderungen zu protokollieren. Dabei werden die verschiedenen Tage unterteilt in die Bereiche *Allgemein* und *Probleme/Offene Fragen*. *Allgemein* beschreit was ich an dem Tag getan habe und womit ich mich beschäftigt habe und *Probleme/Offene Fragen* beschreibt alle Probleme welche in Folge der Arbeit auftraten. Fragen, welche beantwortet werden sollen dann als Frage mit zugehöriger Erklärung im *Allgemein*-Teil aufgegriffen werden.

20.04.2017

Allgemein

- Einlesen in die Paper *main* und das Paper *non-term*
- Installieren der Software auf dem Laptop und in der VM

Probleme/Offene Fragen

Zu *main*:

1. Seite 2, Preliminaries
   Die Summe startet bei 0, jedoch passt dies nicht. Wenn man $k = 0$ setzt sollte $\begin{pmatrix} 3 & 1 \end{pmatrix}^T$ raus kommen, da keine Schleifeniteration durchgeführt wird also nur der **STEM**-Teil relevant ist. allerdings kommt dann $\begin{pmatrix} 10 & 2 \end{pmatrix}^T$ raus, was der Wert nach der 1. Iteration ist.

2. Seite 3, Definition 2.2
   $Gx < g \wedge MX + m = x'$, was sind G, g, M und m?

zu *non-term*:

1. Seite 4, Definition 1:
   $x, x' \in \mathbb{R}^n$, also $x = \begin{pmatrix} x_1, & \dots x_n \end{pmatrix}^T$
   Welche Dimension hat dann $\begin{pmatrix} x \\ x' \end{pmatrix}$?
   $\begin{pmatrix} \begin{pmatrix} x_1, & \dots x_n \end{pmatrix} \\ \begin{pmatrix} x'_1, & \dots x'_n \end{pmatrix} \end{pmatrix} \in R^{2 \times n}$?

26.04.2017

Allgemein

- Script für das Tagebuch erstellt
- Versucht das Git-Repository in der VM zu installieren. Vergeblich

Probleme/Offene Fragen

1. Vm - Git
   Erstellen der Projekte wirft sofort Fehler. Wieso ist einfaches Clonen nicht ausreichend? Antwort: sollte eigentlich ausreichen

27.04.2017

Allgemein

- Script abgeändert
- Repositorys in VM geklont und danach die Projekte angelegt. Jedoch nun 4000+ Errors. Code bis auf weiteres verschoben.
- weiteres einlesen in *main* und *non-term*

Probleme/Offene Fragen

Zum Ecplise-projekt:

1. *ant grammars* hat kein passendes *build.xml*. Selbst mit *build-aprove.xml* ist *grammars* nicht definiert

Zu *main*:

1. Seite3, Definition 2.2: Was sind G und M? M ist die „*actual update matrix* “ aber was soll das sein? $direction \times speed^i$ von der Introduction?
2. Seite 4, Definition 2.6: defekt einer Matrix nur noch schleierhaft.
   $def(A) = dim(ker(A))$ und $ker(A) = \{v \in \mathbb{R} | Av = 0\}$

16.05.2017

Allgemein

- David eine E-Mail geschrieben Antwort: Jera kommt Mittwoch (17.05.2017) wieder. David könnte mit bei der Installation am Freitag (19.05.2017) helfen
- erste Gedanken über den Ablauf:
  1. Syntaxcheck:
     a) Teste auf erlaubte Elemente
        - keine *for*-Schleifen
        - keine *GOTO*'s oder ähnliches

     b) Teste auf Unterteilung in *STEM*
        i. Anfangswerte für Variablen
        ii. nicht aufgeführte Variablen werden mitgeschrieben

     c) Teste auf Unterteilung von *LOOP*
        - *Guard* identifizieren
        - ausschließlich lineare Updates

  2. Simple Fälle abfangen
     - eine Variable wird immer auf `nondet()` gesetzt

Probleme/Offene Fragen

Zu non-term:

1. non-term, Seite 2: Die Ausführung von *Figure 1a*: Wieso $(2,0)^T$ und $(2,1)^T$? Angenommen die Reihenfolge ist $(a,b)^T$ dann müsste es doch mit so etwas wie $(undef/0,1)^T$ starten. b wird immer wieder auf `nondet` gesetzt was jede Ausführung sein kann. Sind also 2 und (immer) 1 zufällig gewählt? Überlegung: Die ersten Einträge sind vor dem *STEM* und somit beide `nondet`. Dann kommt der *STEM* und dann die *LOOP*. Zudem muss a so gesetzt sein, dass die *Guard* "passt"
2. die Geometrische Reihe passt für *Figure 1a* und *Figure 1b* nicht
3. zu *Figure 1c*: μ Faktor von b?

Allgemein:

1. Wo wird das Programm angesetzt? Als einzelner Thread nebenher oder an einer bestimmten Stelle?
2. Gibt es dann Elemente auf die ich bereits zurückgreifen kann?
3. Nur für Java?

17.05.2017

Allgemein

- Weiteres Einlesen in *non-term* und *main*
- Vielleicht erst eine Methode um für geg. *GNA* zu testen ob die 4 Bedingungen(*non-term*, Seite 5) halten
- Verwendung von SMT-Solver:
    1. Berechnen der Eigenwerte als λ's der Updatematrix
    2. Berechnen der Eigenvektoren zu den Eigenwerten

Probleme/Offene Fragen

zu *main*:

1. Seite 5, Definition 3.1: Wenn wir k λ's haben aber nur $k-1$ μ's, wie sind dann Programme wie:

```
b=1;
while a+b >= 4 do
        a = 3*a+b;
        b = 2*b +a;
end while
```

   möglich? Das eine μ wird für die Beziehung von a zu b gebraucht aber dann ex. kein weiteres μ für die Beziehung von b zu a.

28.05.2017

Allgemein

- Ecplise versucht wieder ans laufen zu bekommen
- ITRS eingelesen

Probleme/Offene Fragen

Generell:

1. Bis wann muss die Bachelorarbeit angemeldet sein?
2. Kann das Projekt mit Eclipse nicht mehr öffnen
3. Toolbar verschwindet ständig: Antwort: *workbench.xmi* löschen und Neustarten
4. Wie soll man aus einem *ITRS STEM* und *LOOP* ablesen können?

08.06.2017

Allgemein

- Compilieren von *.c*-Dateien in *.llvm*-Dateien
- Reproduzieren von verschiedenen Beweisen um mehr Verständnis zu erhalten wie ein ITRS abgelesen werden könnte
- Eingearbeitet in *sat4j* und Erstellung einer Basis, welche eine Datein in CNF auf Erfüllbarkeit prüft:

```
import org.sat4j.minisat.SolverFactory;
import org.sat4j.reader.DimacsReader;
import org.sat4j.reader.Reader;
import org.sat4j.specs.IProblem;
import org.sat4j.specs.ISolver;

//eine Testklasse fuer SAT4J
public class main {

        public static void main(final String[] args) {
                //Anlegen eines neuen Solvers
                ISolver solver =
                    SolverFactory.newDefault();

            solver.setTimeout(300);
            //Anlegen eines Readers
            Reader reader = new DimacsReader(solver);

            try{
                //Auslesen des Programms aus der Datei
                IProblem problem =
                    reader.parseInstance("src/input.txt");

                //Pruefen der Erfuellbarkeit
                if (problem.isSatisfiable()) {
                        //Wenn erfuellbar ein Model
                            angeben
                System.out.println("Satisfiable !");
                System.out.println(reader.decode(problem.model()));
            } else {
```

```
                    //Ausgeben dass es nicht Erfuellbar ist
                    System.out.println("Unsatisfiable !");
                }
                } catch (Exception e){
                    //Fangen aller Exceptions
                    e.printStackTrace();
                }
        }

    }
```

Das Problem als Input

```
p cnf 5 3
1 -5 4 0
-1 5 3 4 0
-3 -4 0
```

Probleme/Offene Fragen

1. Wenn ich

   ```
   timo@Ubuntu:~/Downloads$ clang -S -emit-llvm ctest.c
   ```

   für meine Testdatei *ctest.c* eingebe erstellt mir *Clang* eine *.ll*-Datei, welche in Eclipse zu dem (bekannten) *computePointerSizeFromDataLayout*-NullPointer führt.
   Antwort: Benutze den Befehl:

   ```
   timo@Ubuntu:~/Downloads$ clang ctest.c -S -emit-llvm -o ctest.llvm
   ```

2. die *clang* -Kompilierung verwirft die *query*-Angabe und schreibt jedes mal *source_filename = "ctest.c"* hinzu, was Eclipse Probleme bereitet

3. wenn ich einen *.llvm*-Code erstellen lasse und die Anpassungen für *query* und *source_filename* mache erhalte ich immer eine *InconsistentStateException* im Graphen

4. Auch beim Web Interface bekomme ich die selben Probleme (nicht ausführen des Beweises: *Analyze Termination of all function calls matching the pattern: main()*) . Z.B. für mein Test-File

```
/* query: main(Int) */

int main(){
    int a=2, b =1;

    while(a+b<10){
        b=b-1;
    }
}
```

5. Wo genau kann ich das ITRS herbekommen um darauf zu arbeiten? An welcher Stelle/Klasse o.ä. Antwort: Bekomme das Problem als IRSwTProblem-Instanz. Siehe 14.06.2017

09.06.2017

Allgemein

- Änderung von den "Bounded"Variablen von *True* in *False* in:
  - src/aprove/Testing/Manual/LLVMTermGraphTester.java
  - src/aprove/PredefinedStrategies/Auto/current.strategy
  - src/aprove/PredefinedStrategies/Auto/noT2.strategy
- Git-Repo in Ubuntu geklont und verbunden
- Einlesen in die *IRSwTPRoblem*-Art (src/aprove/Framework/IntTRS/IRSwTProblem.java)

Probleme/Offene Fragen

1. Zum *IRSwTProblem*: im Beispiel des Javadoc's ist die 17 eine zufällige Zahl? Zitat: "Variables not occurring in ϕ or in arithmetical operations may be instantiated by terms consisting of constructor symbols."

14.06.2017

Allgemein

- Erstellung der *testingGNT.strategy* in *aprove/PredefinedStartegies/Auto* als Kopie der *Current*-Strategie und abändern der *LLVM*-Prozessoren die ausgeführt werden sollen. Hinweis: dort ist noch immer ein T2 Prozessor und nicht "meinÄnsatz vermerkt.

- Hinzufügen eines

  `Zeile 756: declare`

- Ausgeben lassen des IRSwTProblems durch folgenden Code:

```
try {
        FileWriter fw = new
           FileWriter("/home/timo/Downloads/ausgabe.txt");
        BufferedWriter bw = new BufferedWriter(fw);

        bw.write(result.toString());

        bw.close();
        fw.close();
} catch (IOException e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
}
```

  Dieser wird im IRSwTProblem in die `toString()`-Methode unmittelbar vor `return` eingefügt.

- wenn ich eine Instanz der Klasse `IRSwTProblem irswt` bekomme, bekomme ich

- das Startsymbol per `irswt.getStartTerm()`
- die Regeln per `irswt.getRules()`

Integer Rewrite System with Terms (IRSwT)
Consists of rules of the form $f(t_1, ..., t_n) -> g(s_1, ..., s_m)|\phi$ where the arguments $t_1, ..., t_n, s_1, ..., s_m$ are terms that only use non-defined symbols (i.e., symbols h for which no rule $h(...) -> ...$) exists. The right arguments $s_1, ..., s_m$ might additionally use basic arithmetical operations (+,-,* and constants).

To apply such a rule the instantiation of the variables has to satisfy the condition ϕ, which is a conjunction of atomic formulae over the signature +, -, *, > (,=) $\cup\mathbb{Z}$. Variables not occurring in ϕ or in arithmetical operations may be instantiated by terms consisting of constructor symbols.

Example: $f(x, y) -> g(x, y)|TRUE$
$g(s(x), y) -> g(x, y)|TRUE$
$g(0, y) -> f(z, y - 1)|y > 0$

Here we have the following decreasing ->-chain:
$f(s(0), 1) -> g(s(0), 1) -> g(0, 1) -> f(s(s(17)), 0) -> ...$

Please note that every variable occurring in an arithmetical operation or in ϕ has to be instantiated by an integer.

Probleme/Offene Fragen

1. Git Submodules/SymLinks
2. Zitat von Jera: "*Dein Arbeitsschwerpunkt sollte eher auf den Ideen des Papiers liegen und nicht auf technischen Details von AProVE.* Heißt das ich soll es gar nicht programmieren sondern nur überlegen wie man das theoretisch auslesen kann?

15.06.2017

Allgemein

- Einlesen in die Klassen *IGeneralizedRule* (I für Integer) und *GeneralizedRule* um zu finden, wie ich die Ersetzungsterme erreichen kann um sie als Lineares Programm aufzufassen

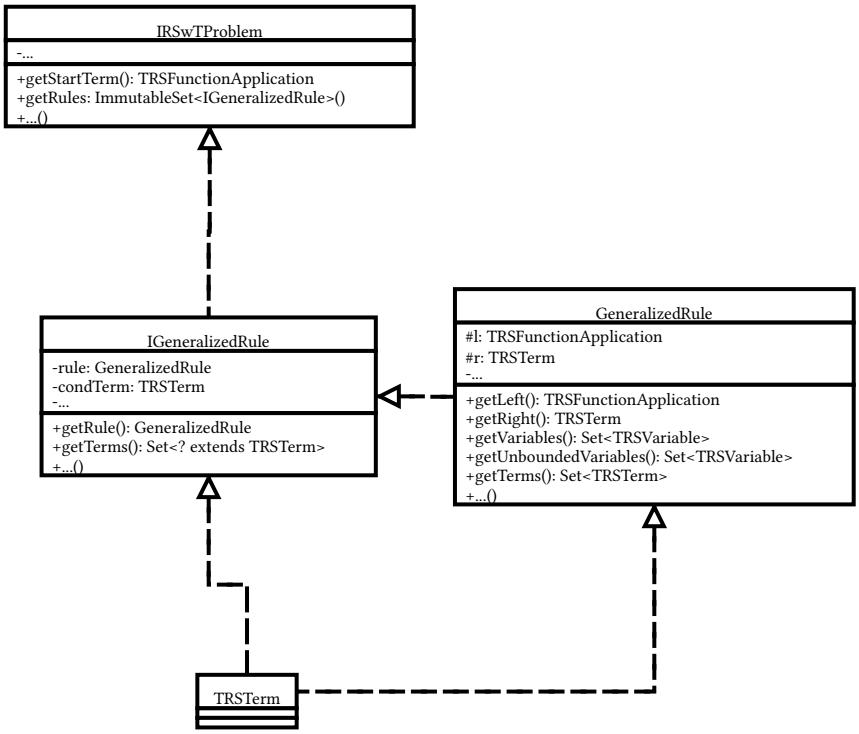

Abbildung 1: Das Klassendiagram des IRSwTProblem und seinen Komponenten zum besseren Verständnis der notwendigen Programmteile. Stand: 15.06.2017

21.05.2017

Allgemein

- Erstellen der 3 Testprogramme aus dem Paper (Programm 1 wird nicht laufen können wegen `undef()`)

- rumschlagen mit Ecplise was schlussendlich zum neuen Aufsetzen der Vm führte, incl. neues Aufsetzen von Eclipse Neon.
- einfügen der *onlyT2.strategy*
- nicht erneutes Erstellen der Grammars per *ant*, da sonst alles doppelt vorkommt und/oder Fehler schmeißt.

Probleme/Offene Fragen

1. wie sollen die "neuen"Variablen aufgefasst werden? Bekommen die einen random Wert? Beispiel:

$$f_{79}(x_3, x_8) \to f_{79}(x_{12}, 2 * x_8) \quad :| : 3 < 2 * x_8 \quad \&\& \quad 3 * x_3 = 2 + x_{12} \quad \&\& \quad x_3 + x_8 > 5 \quad \&\& \quad x_{12} > 9$$

   Woher kommt x_{12} und was weiß man darüber? Antwort: die zweite Regel besagt: $3 * x_3 = 2 + x_{12}$ somit kann man sich den Wert von x_{12} herleiten wenn man x_3 kennt.
   Wo ist der STEM-Teil des Programms hin? Das Programm (Testprogramme/non-term-paper-example2.c) beginnt mit

```
int a=2,b=1;
```

   Zudem wird bewiesen, dass es nicht terminiert durch das AProVE-Tool, jedoch die das ITRS hält bereits für die erste Iteration mit $a = 2$ und $b = 1$ nicht, weil $x_8 = b = 1$ gilt und dann $3 < 2 * x_8$ nicht mehr gilt. Antwort: b ist nicht x_8

2. Startterm wird beim IRSwTProblem nicht mit angegeben, wenn *noT2.strategy* verwendet wird. Antwort: verwende die *onlyT2.strategy*
3. kann trotz neuem *aprove-build.xml* keine neuen Grammars erzeugen und somit nicht automatisiert C-Programme testen

22.06.2017

Allgemein

- Weiteres einlesen in den Vorgang des Papers
- erste Programmierung von versch. Klassen
  - GeoNonTermAnalysis
  - STEM
  - Logger
    * MÖGLICHE ERWEITERUNG: weiteren Konstruktor, Modi zum überschreiben etc

  Dargestellt in einem Klassendiagramm :

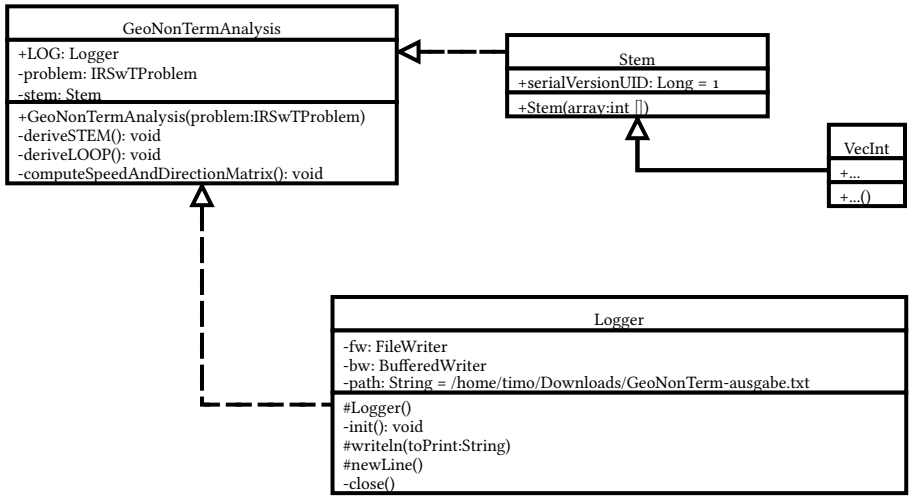

Abbildung 2: Das Klassendiagram des Geometrischen Nicht-Terminierungs Ansatz und seinen Komponenten zum besseren Verständnis der notwendigen Programmteile. Stand: 22.06.2017

- Erweiterung des Klassendiagramms vom 15.06.2017

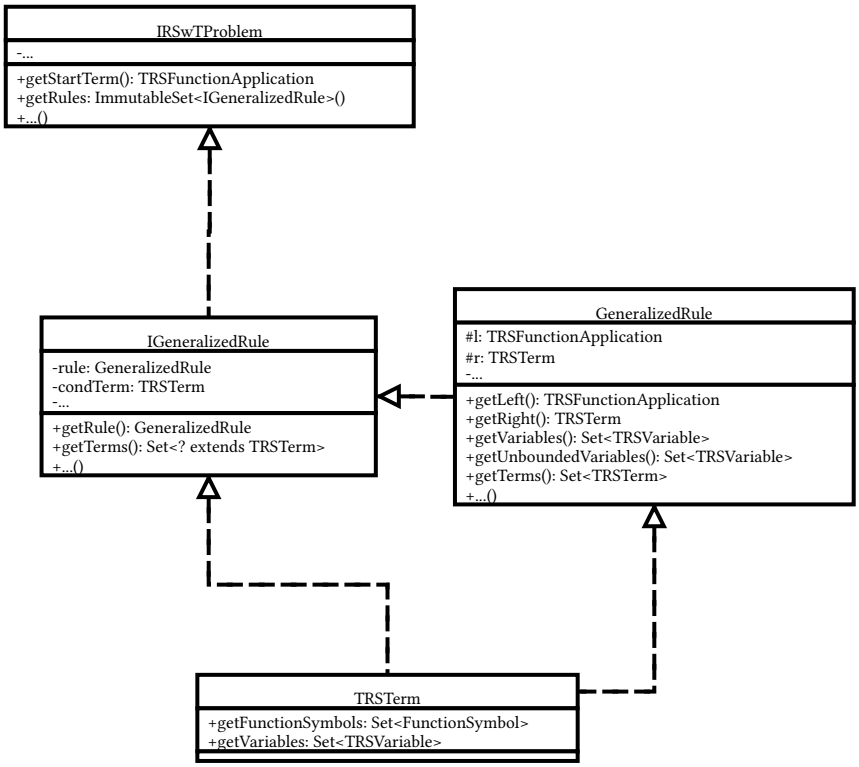

Abbildung 3: Das Klassendiagram des IRSwTProblem und seinen Komponenten zum besseren Verständnis der notwendigen Programmteile. Stand: 22.06.2017

Probleme/Offene Fragen

1. Probleme mit Dia und LaTeX. Antwort: als LaTeX PGF Makros exportieren und ”[scale=0.6, every node/.style=scale=0.6]” hinter ” begin{texpicture}” einfügen
2. Die *Conditions* der Regeln ist in UPN (umgekehrt polnische Notation). Hilft mir das oder ist es hinderlich?

28.06.2017

Allgemein

- Änderung des *STEM*'s von einer Erweiterung der Klasse *VecInt* zu einer normalen Klasse mit einem Element von *VecInt*
- Einlesen des *STEM*'s aus dem Programm heraus fertig.
- *TRSFunctionApplication* zum Diagramm vom *IRSwTProblem* hinzufügen

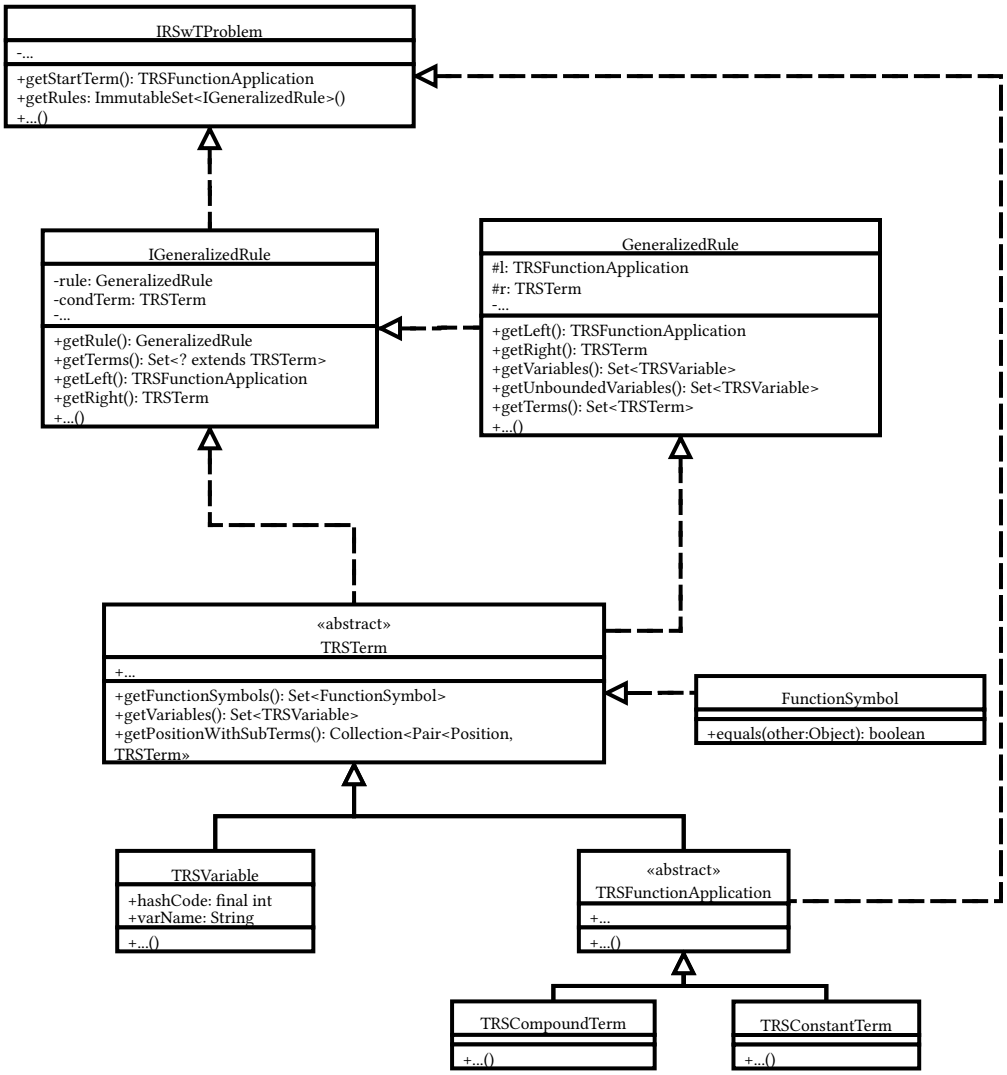

Abbildung 4: Das Klassendiagram des IRSwTProblem und seinen Komponenten zum besseren Verständnis der notwendigen Programmteile. Stand: 28.06.2017

- Änderungen der Methoden und Attribute in dem *GeoNonTerm*-Package

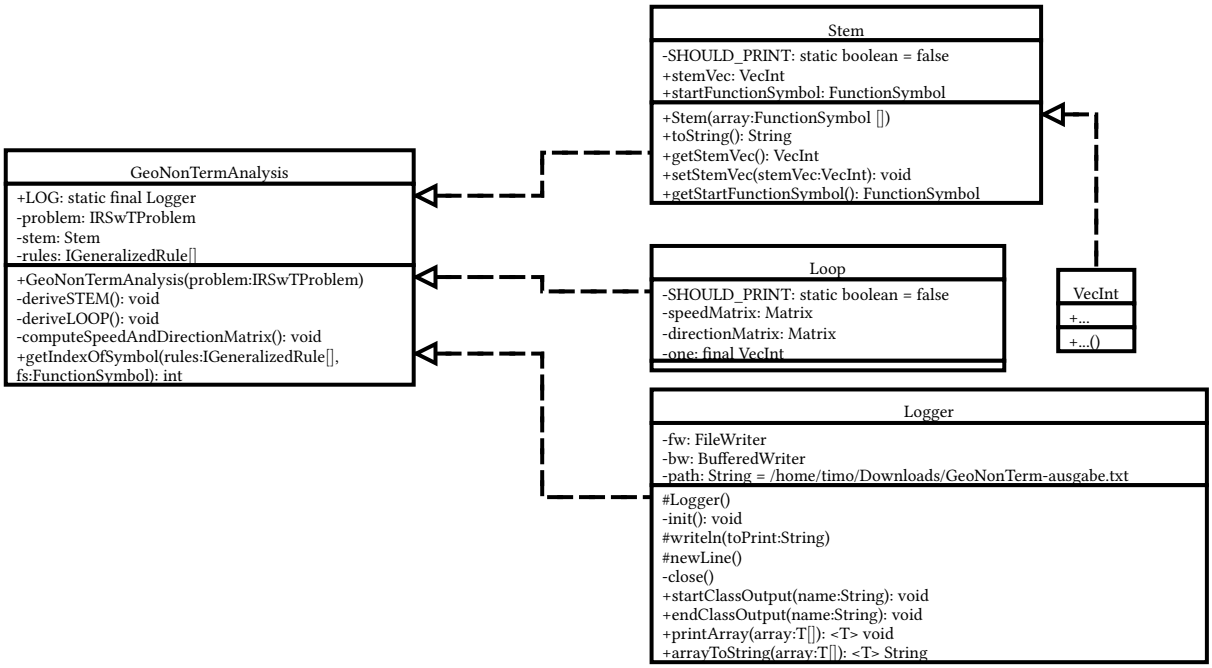

Abbildung 5: Das Klassendiagram des Geometrischen Nicht-Terminierungs Ansatz und seinen Komponenten zum besseren Verständnis der notwendigen Programmteile. Stand: 28.06.2017

- Definieren der verschiedenen Ausgaben:
  - `##########`: dies sind Ausgaben für die Verständlichkeit/Nachvollziehbarkeit
  - `++++++++++`: dies sind Ausgaben rein zum Testen
- mit `getPositionsWithSubTerms` die Collection als Baum aufbauen und so die Postfix-Notation bzw. Matrixeinträge herleiten
- kommentieren aller Klassen und Methoden

TODO am Code

1. Verbesserung des *Loggers* hinsichtlich der Ausgabe verschiedener Klassen in vielleicht verschiedene Dokumente

Probleme/Offene Fragen

1. Wieso kommt bei z.B.

$$f_{99}(x_3, x_8) \to f_{99}(3 * x_3 + x_8, 2 * x_8) : | : \ldots$$
$$f_2 \to f_{99}(10, 2)$$

   Als Symbol der linken Seite der ersten Regel `f_99` und als Symbol bei der rechten Seite der zweiten Regel `f_99_2` raus? Eine Prüfung auf *equals* ergibt aber auch Gleichheit. Überlegung: Beim zweiten wird die Anzahl der Argumente mit angehängt, welche beim ersten die selbe ist aber nicht mit angegeben wird. Antwort: die *aprove/Framework/BasicStructures/FunctionSymbol.java* unter der Methode *equals* macht genau den Namens, hashCode und Größencheck, sodass die verschiedenen `toString`-Werte keinen Ausschlag geben.

2. Wo packt man das `LOG.close();` am besten hin?

3. Wie soll ich die *Umgekehrt Polnische Notation* behandeln um z.B. die Vorfaktoren einer Regel abzuleiten Antwort: Umschreiben in Postfix und stupides Iterieren

4. geht als IRSwTProblem auch:

$$f_x(a + 1, b) \to f_x(a, b) : | : cond$$

   Antwort: Ja geht soll aber vorerst mit *assert* abgefangen werden

29.06.2017

Allgemein

- Viel rumprobiererei wegen der Umgekehrt Polnischen Notation

- die *TRSCompound*-s sind die Funktionsterme und im *RPNTree* die *RPNFunctionSymbols*. Sie haben den Aufbau:

$$\underbrace{+}_{var:f}\underbrace{(\_,\ldots,\_)}_{var:args}$$

  also wäre für $+(3, x3)$: $f = +$ und $args = \{3, x3\}$

- Die rechte Seite in den eigenen *RPNTree* geparst. Beschränkt derzeit auf 2 argumentige Funktionssymbole.

- Die *UpdateMatric geschrieben*

- auslesen der Vorfaktoren der Variablen in den Iterationsschritten

- Änderungen der Methoden und Attribute in dem *GeoNonTerm*-Package

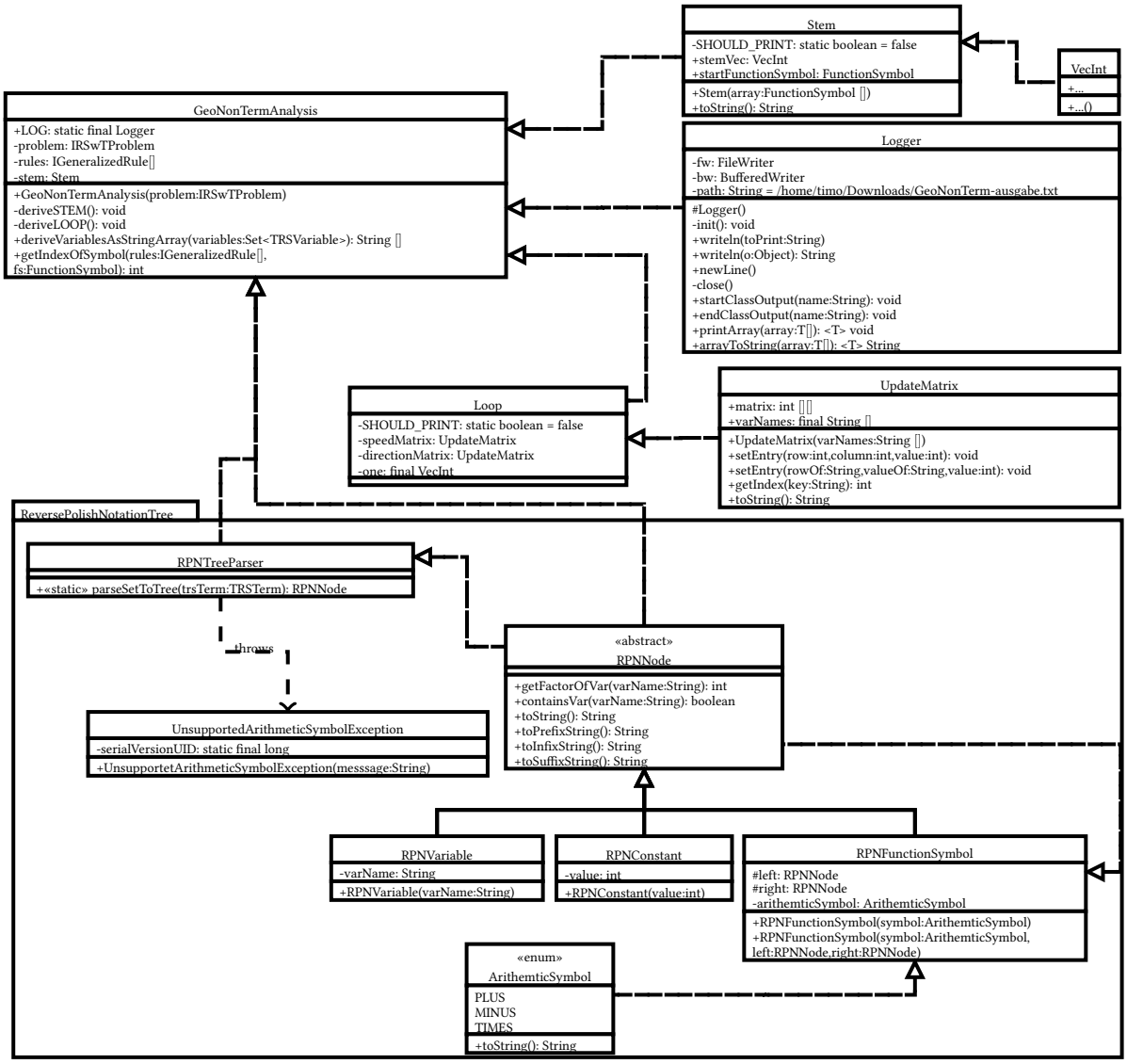

Abbildung 6: Das Klassendiagram des Geometrischen Nicht-Terminierungs Ansatz und seinen Komponenten zum besseren Verständnis der notwendigen Programmteile. Stand: 29.06.2017

Probleme/Offene Fragen

1. Bekomme es nicht in den Baum geparst. Antwort: doch.
2. Wie bringt man LP's der Form

$f_a(x_1, x_2, ..., x_n) \rightarrow f_a(x_1 + 1, x_2, ..., x_n)$ oder $f_a(x_1, x_2, ..., x_n) \rightarrow f_a(x_1 * x_2, x_2, ..., x_n)$

   in eine Matrixschreibweise

3. bei ArithmeticalSymbol.MINUS wirklich flippen?

04.07.2017

Allgemein

- Mir ist aufgefallen, dass die *LOOP* in der angedachten Form wenig bringt. Was ich brauche ist nachher $A * x < b$ um darauf die non-term Bedingungen zu prüfen. Aus *main* wird deutlich wie man A und b herleitet. Als:

$$G * x < g \wedge M * x + m = x'$$
$$A = \begin{pmatrix} G & 0 \\ M & -I \\ -M & I \end{pmatrix}, b = \begin{pmatrix} g \\ -m \\ m \end{pmatrix}$$

  Die bisher hergeleitete *Speed*-Matrix kann als M übernommen werden, da es die linearen Updates schreibt. Im folgenden werde ich die Matrizen wie folgt nennen:

  G: GuardUpdates — m: UpdateConstants

  g: GuardConstants — A: IterationMatrix

  M: UpdateMatrix — b: IterationConstants

- Muss den RPNTree anpassen, damit ich die conditions parsen kann. Vielleicht kann ich die &&'s splitten, sodass n-conds bleiben. Antwort: und hat auch genau so funktioniert.

- Beobachtung:

  Sei $n = \#vars$ dann sind sind $x, x' \in \mathbb{Z}^n$

  Demnach muss die Updatematrix $M \in \mathbb{Z}^{n \times n}$, da es die Updates für die Variablen ist und $m \in \mathbb{Z}^n$

  Sei $m = \#conditions$ Somit ist $G \in \mathbb{Z}^{m \times n}$ und $g \in \mathbb{Z}^m$

  Somit folgt durch $b = \begin{pmatrix} g \\ -m \\ m \end{pmatrix} = \begin{pmatrix} \mathbb{Z}^m \\ \mathbb{Z}^n \\ \mathbb{Z}^n \end{pmatrix} \in \mathbb{Z}^{2*n+m}$, dass

  $$A = \begin{pmatrix} G & 0 \\ M & -I \\ -M & I \end{pmatrix} = \begin{pmatrix} \mathbb{Z}^{m \times n} & \{0\}^{n \times n} \\ \mathbb{Z}^{n \times n} & I^{n \times n} \\ \mathbb{Z}^{n \times n} & I^{n \times n} \end{pmatrix} \in \mathbb{Z}^{2*n+m \times 2*n}.$$

- Habe die GuardMatrix und die GuardConstants auslesen lassen. Jedoch siehe Problem 3.

- Änderungen der Methoden und Attribute in dem *ReversePolishNotationTree*-Package

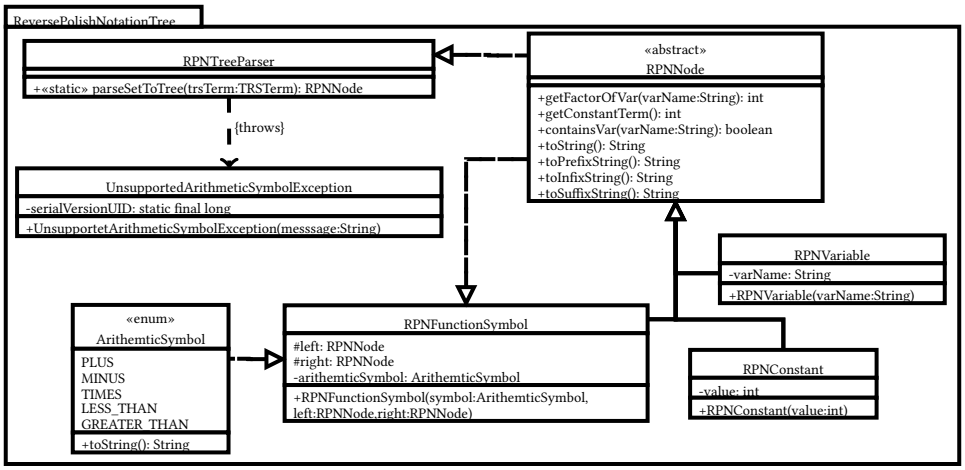

Abbildung 7: Das Klassendiagram des Reverse Polish Notation Tree und seinen Komponenten zum besseren Verständnis der notwendigen Programmteile. Stand: 04.07.2017

- Änderungen der Methoden und Attribute in dem *GeoNonTerm*-Package

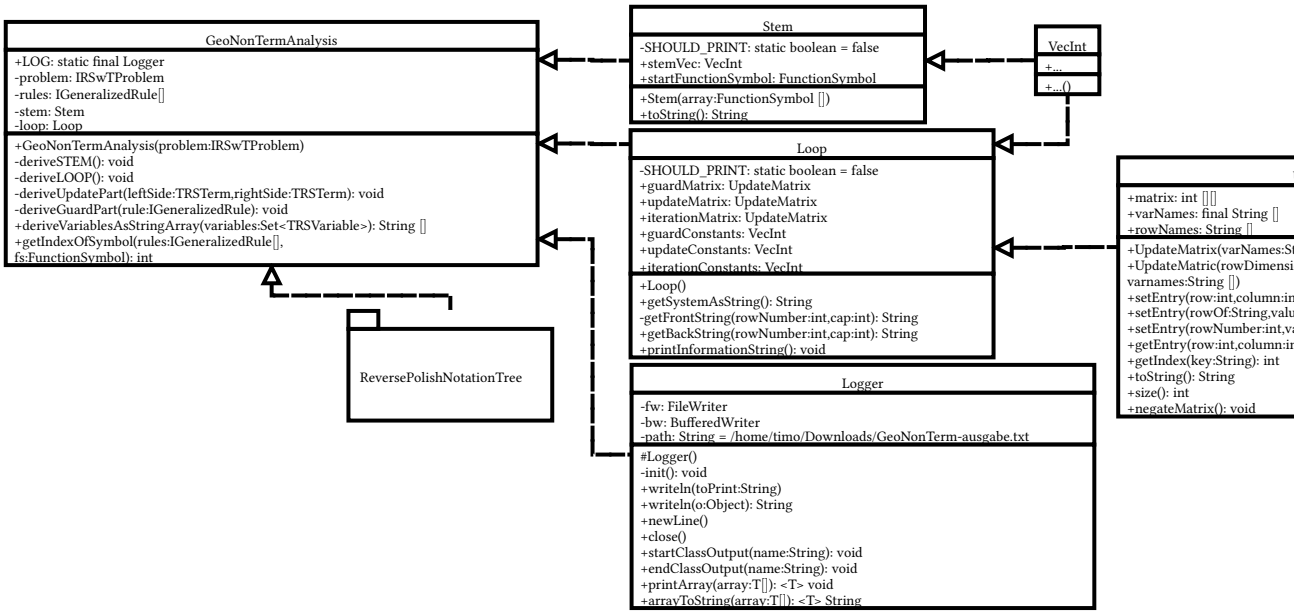

Abbildung 8: Das Klassendiagram des Geometrischen Nicht-Terminierungs Ansatz und seinen Komponenten zum besseren Verständnis der notwendigen Programmteile. Stand: 04.07.2017

Probleme/Offene Fragen

1. brauche ich dann den *STEM* noch? Antwort: ja für die zweite Bedingung an ein GNA
2. woher kommt $Y = \begin{pmatrix} 4 & 3 \\ 0 & 1 \end{pmatrix}$?
3. Muss in der Guard noch alle $>$ zu $<$ umdrehen damit diese einheitlich sind zu der Literatur Antwort: Gemacht aber ist das richtig?

06.07.2017

Allgemein

- Berechnung der Iterationmatrix A und der Iterationskonstanten b wie am 05.07.2017 beschrieben.
- kommentieren neuer Methoden und Änderung mancher Sichtbarkeiten
- Änderungen der Methoden und Attribute in dem *GeoNonTerm*-Package. Änderungen:
  - UpdateMatrix
    - IdentityMatrix-Methode, welche mir statisch eine Einheitsmatrix von angebbarer Größe liefert
    - negateMatrix-Methode: welche eine Matrix mit -1 multipliziert
    - Mmbenennung von varNames in columnNames
    - neuen Konstruktor mit nur den Dimensionen
    - insert-Methode: fügt eine Matrix in die aktuelle ein
  - Loop
    - computeIterationsMatrixAndConstants-Methode
    - negateVec-Methode: multipliziert einen Vektor mit -1