

**Team:** 11, Mesut und Anton

**Team:** 11, *Mesut Koc* und *Anton Kirakozov*

**Aufgabenaufteilung:** Alle Aufgaben gemeinsam bearbeitet (*Aufgabe\_4.pl und Dokumentation*)

**Quellenangaben:**

**Bearbeitungszeitraum:** Alle Aufgaben wurden gemeinsam bearbeitet (*siehe gemeinsame Bearbeitungszeit*)

**Gemeinsame Bearbeitungszeit:** Vom 28.05.15 bis 17.06.15 ca. 16h

**Aktueller Stand:** Alle Prädikate sind fertig implementiert.

*Konzepte unserer fertigen Aufgaben:*

#### **Aufgabe 4 Teil 1**

**4.1.1)** In dieser Aufgabe präzisieren wir das Prädikat **is\_sbin\_tree(..)**, welches überprüft, ob der gegebene Binärbaum auch in der vorgegebenen syntaktischen Form ist.

**4.1.2)** Das Prädikat **is\_sbin\_tree(..)** fügt in einem syntaktisch korrekten Binärbaum ein neues Element hinzu. Dabei erhalten wir ein neu resultierendes Baum, der mit den eingefügten Elementen sich ergibt. Wir achten darauf, dass das Element in der richtigen Stelle des Baumes hinzugefügt wird.

**4.1.3)** Das Prädikat **inorder\_sbin2tree(..)** traversiert ein gegebenes Baum in die Inorder-Reihenfolge und gibt die enthaltenen Elemente zurück.

**4.1.4)** In dieser Teilaufgabe geht es um die Rückführung von einer Liste in einen sortierten Binärbaum. Wir bekommen eine Liste mit einer beliebigen Anzahl an Elementen. Dabei soll der Binärbaum balanciert aufgebaut werden. Die Liste teilen wir in zwei Teile auf und bestimmen das erste Element aus dem zweiten Teil der Liste als unsere Wurzel. Anhand dieser Wurzel sortieren wir die restlichen Elemente auf kleiner und größer und geben im Endeffekt die sortierte Liste als einen Binärbaum zurück.

#### **Aufgabe 4 Teil 2**

**4.2.1)** Mit unserem Prädikat **is\_atomic\_expr(..)** prüfen wir, ob ein Term eine atomare logische Aussage ist. Ein Atom ist eine Zeichenkette, die mit einem kleinem Buchstaben beginnen muss oder auch eine Zeichenkette die zwischen zwei Anführungszeichen steht wie zum Beispiel: baum, „baum“.

**4.2.2)** In dieser Aufgabe unterscheiden wir, ob ein Term ein Literal ist. Literale sind Zeichenketten, die eine syntaktische Form von Fakten haben, wie zum Beispiel:

**Beispiel:** baum(braun)

Ein Literal ist die negierte Form eines Atoms.

**4.2.3)** Ziel dieser Teilaufgabe ist es, einen aussagenlogischen Term in einen neuen aussagenlogischen Term umzuformen, sodass dieser sich in einer Negationsnormalform befindet. Diese Form gilt dann, wenn die Negation direkt vor den atomaren Aussagen vorkommt. Anhand eines Beispiels der Implikation möchten wir diese Aussage verdeutlichen.  $a \rightarrow b$  hier findet unsere Implikation statt. Jetzt versuchen wir diese umzuformen um die Aussage in die Negationsnormalform zu bringen. Aus  $a$  impliziert  $b$  wird  $\neg a \vee b$ . Damit haben wir die Negation genau vor einem atomaren Ausdruck in diesem Fall  $a$  und haben somit die Darstellung der Negationsnormalform.

**4.2.4)** Mit Hilfe unseres Prädikats `is_clause(..)` überprüfen wir, ob unser gegebener Term eine Klausel ist. Eine Klausel ist in Form einer Funktor / Argumentstruktur notiert, dies bedeutet, dass die Klausel aus einem Funktor und einer Reihe von Argumenten besteht. Die Form einer Klausel definiert sich wie folgt: *funktor(argument1, argument2, ... , argument n)*. Auf der linken Seite befindet sich der Funktor, in den Klammern kann sich eine Menge an Argumenten befinden. Die Menge aller Klauseln mit den selben Prädikatsnamen und der gleichen Anzahl der Argumente, definieren ein Prädikat. Klauseln sind dabei die Disjunktionen von Literalen.

**Beispiel:** is\_clause(a).

4.2.5) Mit unserem Prädikat `is_horn_clause(..)` überprüfen wir, ob ein Term eine Horn-Klausel ist. Horn- Klauseln sind Einschränkungen normaler Klauseln. Die disjunkte Vereinigung bleibt bestehen, allerdings darf in der Klausel nur ein einziges positives Literal vorkommen, die restlichen Literale müssen negiert sein. Anhand eines Beispiels verdeutlichen wir unsere Aussage:

$$K = \{ \neg A \vee \neg B \vee \neg C \vee D \}$$

Wie man an dem Beispiel sehen kann, ist **K** genau dann eine Hornklausel, wenn **K** höchstens ein positives Literal beinhaltet. Dabei unterscheiden wir zwischen drei verschiedenen Typen einer Hornklausel:

- ☑ **K** enthält negative und ein positives Literal (*siehe Beispiel*).
- ☑ **K** enthält nur negative Literale  $K = \{ \neg A \vee \neg B \vee \neg C \vee \neg D \}$
- ☑ **K** enthält nur ein positives Literal  $K = D$

Mit Hilfe der Hornformeln kann man eine effizientere Berechnung von Erfüllbarkeit und Unerfüllbarkeit durchführen.