



12.04.2018

Wintersemester 2017/2018  
**Probeklausur zur Vorlesung SAT Solving und Anwendungen**

**Bearbeitungszeit:** 60 Minuten

**Erlaubte Hilfsmittel:** Keine



**Öffnen Sie die Klausur erst, wenn Sie dazu aufgefordert werden! Ein vorzeitiges Öffnen der Klausur führt zum Nichtbestehen.**

Lesen Sie sich die folgenden Informationen durch, **bevor** die Klausur beginnt:

1. Die Klausur besteht aus 6 Aufgaben auf 8 Blättern (incl. Titelblatt). Es gibt insgesamt **60 Punkte** zu erreichen.
2. Bearbeiten Sie jede Aufgabe auf dem jeweiligen Blatt. Sie können auch die Rückseite benutzen. Brauchen Sie mehr Platz für eine Aufgabe, so verlangen Sie nach einem Zusatzblatt — bearbeiten Sie die Aufgabe **nicht** auf dem Blatt einer anderen Aufgabe weiter.
3. Benutzen Sie zur Bearbeitung der Klausur einen **dokumentenechten Stift**.
4. Machen sie **absolut eindeutig**, welches Ihre finale Lösung ist. Streichen Sie falsche Ergebnisse eindeutig durch. Stehen mehrere Lösungen auf dem Blatt, wird diejenige mit der geringsten Punktzahl bewertet.
5. **Schalten Sie Ihr Handy / Laptop / Tablet aus!** Ein eingeschaltetes technisches Gerät wird als Täuschungsversuch betrachtet und die Klausur damit mit der Note 5.0 bewertet!
6. Jeglicher Täuschungsversuch (Abschauen vom Nachbarn, Spickzettel, technische Hilfsmittel, etc.) wird mit sofortigem Entzug der Klausur und Bewertung mit der Note 5.0 geahndet.
7. Das Klausurheft dürfen Sie nicht fotografieren oder mitnehmen.
8. Schreiben Sie auf jedes Blatt oben Ihren Namen und Ihre Matrikelnummer. Blätter ohne Namen und Matrikelnummer werden **nicht korrigiert!**

Mit der folgenden Unterschrift bestätigen Sie, dass Sie in der geistigen und körperlichen Verfassung sind, diese Klausur zu schreiben und dass Sie die obigen Hinweise gelesen und verstanden haben. Ebenso bestätigen Sie, dass Sie sich ordnungsgemäß für diese Klausur angemeldet haben. Nicht unterschriebene Klausuren werden nicht korrigiert und mit der Note 5.0 bewertet.

Studiengang:

Name:

Abschlussart:

Matrikelnummer:

Unterschrift:

Aufgabe	1	2	3	4	5	6	$\Sigma$	Note
Punkte							/ 60	

**Aufgabe 1** (*Watched Literals*)

Wir betrachten die Implementierungstechnik “Two Watched Literals”. Gegeben ist die Klausel  $(\neg a, \neg b, c, d, e)$ . Zeichnen Sie in jedem Schritt ein, auf welche Literale die Watched Literal Pointers zeigen. Sollte es zu einer Ausgabe *UNIT*, *EMPTY* oder *SAT* kommen, geben Sie diese ebenfalls an.

Klausel	Vorgang	Ausgabe
<div>-a</div> <div>-b</div> <div>c</div> <div>d</div> <div>e</div>	Initialisiere Pointer auf die ersten beiden Literale	
<div>-a</div> <div>-b</div> <div>c</div> <div>d</div> <div>e</div>	$e$ wird mit $\perp$ belegt	
<div>-a</div> <div>-b</div> <div>c</div> <div>d</div> <div>e</div>	$a$ wird mit $\top$ belegt	
<div>-a</div> <div>-b</div> <div>c</div> <div>d</div> <div>e</div>	$c$ wird mit $\perp$ belegt	
<div>-a</div> <div>-b</div> <div>c</div> <div>d</div> <div>e</div>	$b$ wird mit $\top$ belegt	
<div>-a</div> <div>-b</div> <div>c</div> <div>d</div> <div>e</div>	$d$ wird mit $\top$ belegt	

**Aufgabe 2 (QBF Solving)**

1. Geben Sie jeweils an, ob folgende Formeln wahr oder falsch sind.

Hinweis: Für jede Frage können Sie einen halben Punkt erreichen. Beantworten Sie die Frage richtig, bekommen Sie +0,5 Punkte, beantworten Sie die Frage falsch, werden 0,5 Punkte abgezogen. Wenn Sie die Frage gar nicht beantworten, wird sie mit Null Punkten bewertet.

Formel	wahr	falsch
$\exists x \exists y ((x \vee y) \wedge (\neg x \vee \neg y))$	<input type="checkbox"/>	<input type="checkbox"/>
$\exists y \exists x ((x \vee y) \wedge (\neg x \vee \neg y))$	<input type="checkbox"/>	<input type="checkbox"/>
$\forall x \forall y ((x \vee y) \wedge (\neg x \vee \neg y))$	<input type="checkbox"/>	<input type="checkbox"/>
$\forall y \forall x ((x \vee y) \wedge (\neg x \vee \neg y))$	<input type="checkbox"/>	<input type="checkbox"/>
$\forall x \exists y ((x \vee y) \wedge (\neg x \vee \neg y))$	<input type="checkbox"/>	<input type="checkbox"/>
$\forall y \exists x ((x \vee y) \wedge (\neg x \vee \neg y))$	<input type="checkbox"/>	<input type="checkbox"/>
$\exists x \forall y ((x \vee y) \wedge (\neg x \vee \neg y))$	<input type="checkbox"/>	<input type="checkbox"/>
$\exists y \forall x ((x \vee y) \wedge (\neg x \vee \neg y))$	<input type="checkbox"/>	<input type="checkbox"/>

2. Gegeben ist das Quantorenpräfix

$$\exists x \forall a \forall b \exists y \exists z \forall c$$

und die aktuelle Belegung ist

$$\{x \mapsto \perp, a \mapsto \top, z \mapsto \top\}.$$

Geben Sie zu folgenden Klauseln jeweils an, ob sie unter der aktuellen Belegung *unit*, *empty* oder *sat* sind.

Hinweis: Für jede Frage können Sie einen halben Punkt erreichen. Beantworten Sie die Frage richtig, bekommen Sie +0,5 Punkte, beantworten Sie die Frage falsch, werden 0,5 Punkte abgezogen. Wenn Sie die Frage gar nicht beantworten, wird sie mit Null Punkten bewertet.

Klausel	unit	empty	sat	nichts davon
$(a, b, c)$	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
$(a, \neg z)$	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
$(c, y, \neg z)$	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
$(b, y)$	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
$(a, b, z)$	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
$(x, c, \neg z)$	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

3. Gegeben ist die Formel

$$\forall x \exists a \exists b \exists c \forall y \forall z (x \vee a \vee \neg y \vee \neg c) \wedge (x \vee y \vee c \vee \neg b) \wedge (b \vee \neg x \vee \neg c) \wedge (c \vee b \vee \neg y) \wedge (\neg x \vee \neg b \vee \neg z).$$

Führen Sie den QBF-Algorithmus bis zum ersten Konflikt aus und geben Sie die UIP-Klausel an. Auf Quantifikationslevel 2 soll die Auswahlreihenfolge der Literale dabei  $a, b, c$  sein, auf Level 3  $y, z$ . Testen Sie immer zuerst die negative Belegung eines Literals.

**Aufgabe 3** (*Software Bounded Model Checking*)

Wir betrachten folgendes kleines C Programm:

```
int qs = 0;
int i;
for (i = 1; i <= 10; i = i*2) {
    qs += i*i;
}
```

1. Welchen Wert hat die Variable `qs` am Ende der Schleife?
2. Wandeln Sie die Schleife in eine äquivalente `while`-Schleife um.
3. Wickeln Sie die Schleife in ein sequentielles Programm ab. Benutzen Sie als Bound  $k$ , wie oft die Schleife abgewickelt wird, den Wert 2.

**Aufgabe 4** (*MaxSAT*)

Wir betrachten die Partial MaxSAT Instanz  $\varphi = \text{Hard} \cup \text{Soft}$  mit folgenden Klauselmengen:

$$\text{Hard} = \{\{\neg x, y\}, \{\neg y, \neg z\}, \{y\}\}$$

$$\text{Soft} = \{\{x\}, \{\neg x, \neg z\}, \{z\}, \{\neg y\}\}$$

1. Bestimmen Sie mit Hilfe des Fu & Malik Algorithmus  $\text{MinUNSAT}(\varphi)$ , d.h. die minimale Anzahl unerfüllbarer soft Klauseln.

Hinweis: Auftretende Cardinality Constraints dürfen mit  $\text{CNF}(\dots)$  beschrieben werden und müssen nicht in Klauselform ausgeschrieben werden.

2. Geben Sie eine Codierung in CNF für den Cardinality Constraint  $\sum_{i=1}^3 x_i = 1$  an.

**Aufgabe 5** (*Non-CNF SAT-Solving*)

Gegeben sei die Formel  $F = (x \wedge y) \vee_0 (y \vee_1 z)$ .

1. Geben Sie den DAG zur Formel  $F$  an.
2. Führen Sie den Non-CNF SAT Solving Algorithmus und geben Sie immer die NoGood Menge, sowie die UIP-Klausel an. Beachten Sie dabei folgende Punkte:
  - Prüfen Sie bei einer Decision immer zuerst die Negation.
  - Belegungsreihenfolge der Knoten bei einer Decision:  $\vee_0, \vee_1, \wedge, x, y, z$ .

**Aufgabe 6 (CDCL SAT Solving)**

Gegeben ist die Klauselmenge  $C$ .

$$C = \{\{\neg a, \neg b, c\}, \{\neg a, \neg d, \neg e, f\}, \{\neg b, \neg c, \neg e \neg f\}\}$$

Bestimmen Sie die Erfüllbarkeit von  $C$  mit Hilfe des CDCL Algorithmus. Die Auswahlreihenfolge für die Literale ist dabei  $a, b, c, d, e, f$ ; es soll immer zuerst das positive Literal getestet werden; als Lernstrategie verwenden Sie 1-UIP.

Gehen Sie dabei wie folgt vor:

1. Führen Sie den CDCL Algorithmus bis zum ersten Konflikt aus.
2. Geben Sie den Implikationsgraph für diesen Konflikt an.
3. Zeichnen Sie im Implikationsgraph jeweils folgende möglichen Schnitte ein und geben Sie die Klauseln explizit an:
  - (a) First New Clause
  - (b) First UIP Clause
  - (c) Decision Clause
4. Geben Sie für die Decision Clause eine Herleitung per Resolution an.
5. Führen Sie dann den Algorithmus bis zum Ende aus und geben an, ob  $C$  erfüllbar ist, oder nicht. Falls ja, geben Sie auch eine vollständige erfüllende Belegung an.