

## 1. Watched Literals (5 Punkte)

Betrachte die Klausel  $\{a, \neg b, c, d, e, \neg f\}$ . Die watched literals werden so initialisiert, dass der erste Zeiger auf  $a$  zeigt und der zweite Zeiger auf  $\neg b$  zeigt. Es soll die Belegung  $a = \perp, b = \top, c = \perp, d = \top, e = \perp, f = \top$  realisiert werden.

Gib nach jeder einzelnen Variablenbelegung (siehe letzte Spalte) die Positionen der Zeiger sowie evtl. den Klauselstatus (SAT, UNIT, EMPTY) an.

	SAT	UNIT	EMPTY													
<table><tr><td>a</td><td>¬b</td><td>c</td><td>d</td><td>e</td><td>¬f</td></tr><tr><td>↑<sub>1</sub></td><td>↑<sub>2</sub></td><td></td><td></td><td></td><td></td></tr></table>	a	¬b	c	d	e	¬f	↑ <sub>1</sub>	↑ <sub>2</sub>								noch keine Belegung
a	¬b	c	d	e	¬f											
↑ <sub>1</sub>	↑ <sub>2</sub>															
<table><tr><td>a</td><td>¬b</td><td>c</td><td>d</td><td>e</td><td>¬f</td></tr><tr><td>1</td><td>2</td><td></td><td></td><td></td><td>F</td></tr></table>	a	¬b	c	d	e	¬f	1	2				F				f = ⊤
a	¬b	c	d	e	¬f											
1	2				F											
<table><tr><td>a</td><td>¬b</td><td>c</td><td>d</td><td>e</td><td>¬f</td></tr><tr><td>F</td><td>2</td><td>1</td><td></td><td></td><td>F</td></tr></table>	a	¬b	c	d	e	¬f	F	2	1			F				a = ⊥
a	¬b	c	d	e	¬f											
F	2	1			F											
<table><tr><td>a</td><td>¬b</td><td>c</td><td>d</td><td>e</td><td>¬f</td></tr><tr><td>F</td><td>2</td><td>F</td><td>1</td><td></td><td>F</td></tr></table>	a	¬b	c	d	e	¬f	F	2	F	1		F				c = ⊥
a	¬b	c	d	e	¬f											
F	2	F	1		F											
<table><tr><td>a</td><td>¬b</td><td>c</td><td>d</td><td>e</td><td>¬f</td></tr><tr><td>F</td><td>F</td><td>F</td><td>1</td><td>2</td><td>F</td></tr></table>	a	¬b	c	d	e	¬f	F	F	F	1	2	F				b = ⊤
a	¬b	c	d	e	¬f											
F	F	F	1	2	F											
<table><tr><td>a</td><td>¬b</td><td>c</td><td>d</td><td>e</td><td>¬f</td></tr><tr><td>F</td><td>F</td><td>F</td><td>1</td><td>F 2</td><td>F</td></tr></table>	a	¬b	c	d	e	¬f	F	F	F	1	F 2	F		1		e = ⊥
a	¬b	c	d	e	¬f											
F	F	F	1	F 2	F											
<table><tr><td>a</td><td>¬b</td><td>c</td><td>d</td><td>e</td><td>¬f</td></tr><tr><td>F</td><td>F</td><td>F</td><td>⊤ 1</td><td>F 2</td><td>F</td></tr></table>	a	¬b	c	d	e	¬f	F	F	F	⊤ 1	F 2	F	1			d = ⊤
a	¬b	c	d	e	¬f											
F	F	F	⊤ 1	F 2	F											

## 2. CDCL-Algorithmus (12 Punkte)

Gegeben ist die folgende Klauselmenge:

$$\{\{\neg x, \neg y\}, \{\neg a, \neg b, c\}, \{\neg a, \neg b, d\}, \{y, \neg x, \neg c, \neg d\}\}$$

Die Auswahlreihenfolge der Variablen ist  $a, b, c, d, x, y$ . Es soll das positive Literal zuerst getestet werden. Führe den CDCL-Algorithmus durch, wobei bei einem Konflikt die 1UIP-Klausel gelernt wird. Gib beim ersten Konflikt den Implikationsgraphen mit den drei Klauseln (first new clause, 1UIP, decision clause) an. Ist die Menge erfüllbar, gib eine erfüllende Variablenbelegung an.

### 3. Theorieaufgaben (3 Punkte)

In welchem Zusammenhang kommen bei einem SAT-Solver Heuristiken vor?

Für die Auswahl von Variablen (VSIDS), Restart oder nicht

Wofür steht VSIDS und in welchem Zusammenhang wird es verwendet?

Dabei wird jedem Literal eine Aktivität zugewiesen, die bestimmt, wie aktiv die Variable ist. Aktive Variablen werden eher vom SAT-Solver mit einem Wert belegt, wenn keine UP gemacht werden kann

variable state independent decaying sum

Ist es sinnvoll, einen SAT-Solver unter bestimmten Bedingungen neuzustarten?

Ja, wenn man sich in einem lokalen Minimum festfrisst

#### 4. Fu & Malik Algorithmus (8 Punkte)

Führe die erste Iteration des Fu & Malik Algorithmus auf  $\varphi = \text{hard} \cup \text{soft}$  aus.

$$\text{hard} = \{\{a, e\}, \{\neg c\}, \{e, c\}\}$$

$$\text{soft} = \{\{\neg a\}, \{\neg e\}, \{e, d\}\}$$

Gib anschließend die neue Formelmenge  $\varphi'$  an.

1. Iteration:  $\text{sat}(\varphi) = \perp$ ,  $\varphi_C = \{\{a, e\}, \{\neg a\}, \{\neg e\}\}$

phi: ae -c ec -ab\_1^1 -eb\_2^1 ed

Gib eine Kodierung für den nach der ersten Iteration in  $\varphi$  auftauchenden Teilterm  $\text{CNF}(\sum \dots)$  an.

## 5. Konvertierung eines C-Programms (8 Punkte)

Programm:

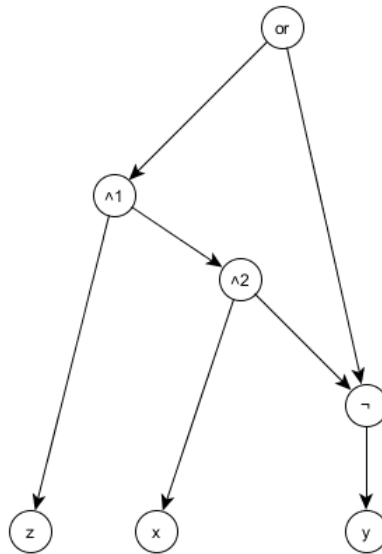
```
int main() {  
    int x = 12;  
    if (x > 17) {  
        x = 25;  
    } else {  
        int i;  
        int y = x;  
        for (i=1; i<8; i=i*2) {  
            x = x * 2;  
            y = y / 2;  
        }  
    }  
    assert(x > 19);           // Bedingung, die überprüft werden soll.  
    return 0;  
}
```

Übersetze das Programm in die Single Static Assignment Form. Führe dazu die folgenden Schritte aus:

1. Ersetzen der for-Schleife durch eine while-Schleife
2. Abrollen der Schleife (zweimal)
3. Übersetzen in Single Static Assignment Form

## 6. Non-CNF SAT Solving (6 Punkte)

Gegeben ist der zur Formel  $F$  zugehörige kompakte gerichteten azyklischen Graph (DAG):



Der Non-CNF SAT Solving Algorithmus wurde bis zum ersten Konflikt ausgeführt. Gib den Implikationsgraphen und die NoGood-Menge und die Klausel, die gelernt wird, an. Es wird die 1UIP-Klausel gelernt.

Level	Variable	Belegung	Grund	Ursache
1	$v$	$\top$	decision	
2	$\neg$	$\perp$	decision	
	$y$	$\top$	parent	$\{\neg = \perp\}$
	$\wedge_1$	$\top$	parent	$\{v = \top, \neg = \perp\}$
	$z$	$\top$	parent	$\{\wedge_1 = \top\}$
	$\wedge_2$	$\top$	parent	$\{\wedge_1 = \top\}$
	$\wedge_2$	$\perp$	child	$\{\neg = \perp\}$

Implikationsgraph: [done](#)

$v=t@1$

$\neg=f@2$

NoGood-Menge und Klausel: [done](#)

## 7. QBF (10 Punkte)

Gib jeweils an, ob folgende Formeln wahr oder falsch sind.

Pro richtige Zeile +0,5 Punkt, pro falsche Zeile -0,5 Punkt, pro leere Zeile 0 Punkte.

Formel	wahr	falsch
$\forall x \forall y \exists a \exists b ((x \Leftrightarrow a) \wedge (y \Leftrightarrow b))$	1	
$\forall y \forall x \exists b \exists a ((x \Leftrightarrow a) \wedge (y \Leftrightarrow b))$	1	
$\exists a \exists b \forall x \forall y ((x \Leftrightarrow a) \wedge (y \Leftrightarrow b))$		1
$\exists b \exists a \forall y \forall x ((x \Leftrightarrow a) \wedge (y \Leftrightarrow b))$		1

Quantifikationslevel: x, y auf Level 1  
a, b, c, d auf Level 2  
z auf Level 3

Belegung:  $x = \perp, a = \top, c = \perp$

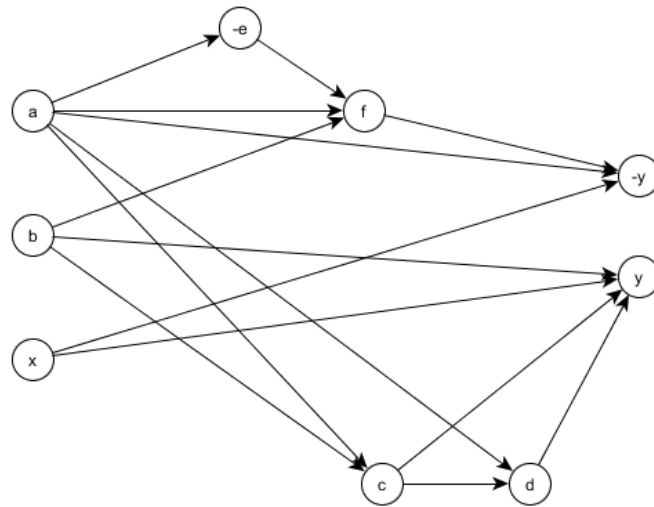
Gib für die folgenden Klauseln an, welchen Status die jeweilige Klausel hat.

Pro richtige Zeile +1 Punkt, pro falsche Zeile -1 Punkt, pro leere Zeile 0 Punkte.

$\forall x \forall y \exists a \exists b \exists c \exists d \forall z$	SAT	UNIT	EMPTY	OPEN
$(\underset{\text{F}}{x} \vee \underset{\text{T}}{z} \vee \underset{\text{T}}{a})$	1			
$(\underset{\text{F}}{x} \vee \underset{\text{F}}{y} \vee \underset{\text{F}}{b} \vee \underset{\text{F}}{d})$				1, weil b und d auf T gesetzt werden können
$(\underset{\text{F}}{x} \vee \underset{\text{F}}{\neg a} \vee \underset{\text{F}}{c} \vee \underset{\text{F}}{z})$			1, da z den Allquantor hat	
$(\underset{\text{F}}{z} \vee \underset{\text{F}}{\neg b} \vee \underset{\text{F}}{d})$				1, weil b und d auf T gesetzt werden können
$(\underset{\text{F}}{x} \vee \underset{\text{F}}{b} \vee \underset{\text{F}}{z})$		1		
$(\neg \underset{\text{F}}{x} \vee \neg \underset{\text{F}}{a} \vee \underset{\text{F}}{c})$			1	
$(\underset{\text{F}}{x} \vee \underset{\text{F}}{y} \vee \neg \underset{\text{F}}{a} \vee \underset{\text{F}}{c})$			1, da y den Allquantor hat	
$(\underset{\text{F}}{x} \vee \underset{\text{F}}{c} \vee \underset{\text{F}}{d})$		1		

## 8. Lokales Minimieren (8 Punkte)

Gegeben ist der folgende Implikationsgraph zum Zeitpunkt eines Konflikts im CDCL-Algorithmus:



Gelernt wird die 1UIP-Klausel  $\{\neg x, \neg c, \neg d, \neg a, \neg b, \neg f\}$ . Minimiere diese Klausel lokal mit Hilfe der folgenden Tabelle:

Variable	Neue Klausel	Reason
d	$\neg x, \neg c, \neg a, \neg b, \neg f$	$\neg a, \neg c, d$
c	$\neg x, \neg a, \neg b, \neg f$	$\neg b, \neg a, c$
f	-	$\neg b, \neg a, e, f$
e	-	$\neg a, \neg e$

Minimierte 1UIP:  $\neg x, \neg a, \neg b, \neg f$