



☐ 4.0 VU Theoretische Informatik 1				
☐ 4.0 VU Theoretische Informatik für Lehramt Informatik I				
1. Teilprüfung		SS 2003		2. Mai 2003
Kennzahl	Matrikelnummer	Familiennamen	Vorname	Gruppe
				B

Kreuzen Sie die zutreffende Lehrveranstaltung an und tragen Sie mit Kugelschreiber Kennzahl, Matrikelnummer, Nach- und Vornamen ein. Legen Sie einen Lichtbildausweis bereit.

Erlaubte Unterlagen: Skriptum, Vorlesungsfolien.

Schreiben Sie alle Lösungen auf diese Blätter und geben Sie die Prüfungsarbeit ohne Zusatzblätter ab. Sie haben 90 Minuten zur Bearbeitung der Aufgaben. Viel Erfolg!

- 1.) Geben Sie einen minimalen deterministischen Automaten an, der die durch den egrep-Ausdruck $(1?21)^*[10]^*1?$ beschriebene Sprache akzeptiert. Überprüfen Sie die Minimalität mit einer Unterscheidbarkeitstabelle, d.h., geben Sie für jedes unterscheidbare Paar von Zuständen ein unterscheidendes Wort an. Stellen Sie den minimalen Automaten graphisch dar. (14 Punkte)
- 2.) Ein Palindrom ist ein Wort, das von vorne nach hinten gelesen dasselbe ergibt wie von hinten nach vorne gelesen. Sei L die Sprache der Palindrome über $\{a, b, c\}$, d.h., $L = \{w \in \{a, b, c\}^* \mid w = w^R\}$. Zeigen Sie: L ist nicht regulär. (12 Punkte)
- 4.) Sei $\mathcal{H}$ folgende Teilmenge der in HTML (Hypertext markup language) verfassten Dokumente. Jedes Dokument in $\mathcal{H}$ beginnt mit `<html>` und endet mit `</html>`. Dazwischen liegt eine beliebige (auch leere) Folge von Texten, geordneten und ungeordneten Listen. Ein Text ist eine nicht-leere Folge von Buchstaben, Ziffern, Leerzeichen, Kommas und Punkten. Eine geordnete Liste beginnt mit `<ol>` und endet mit `</ol>`; dazwischen liegt eine beliebige (auch leere) Folge von Listeneinträgen. Die ungeordnete Liste ist definiert wie die geordnete, nur beginnt sie mit `<ul>` und endet mit `</ul>`. Ein Listeneintrag beginnt mit `<li>` und endet mit `</li>`; dazwischen darf wieder eine beliebige (auch leere) Folge von Texten, geordneten und ungeordneten Listen auftreten. Geben Sie eine kontextfreie Grammatik in EBNF an, die die Sprache $\mathcal{H}$ erzeugt. Verwenden Sie so weit als möglich EBNF-Notationen, um die Grammatik übersichtlich zu halten. (12 Punkte)
- 3.) Sei G die kontextfreie Grammatik $\langle \{S, T\}, \{\underline{s}, \underline{t}, \underline{0}, \underline{1}\}, P, S \rangle$, wobei P aus folgenden Produktionen besteht:

$$S \Rightarrow \underline{s} \mid ST\underline{0}$$

$$T \Rightarrow \underline{t} \mid \underline{1}ST$$

Sei weiters w das Wort $\underline{s}\underline{t}\underline{0}\underline{1}\underline{s}\underline{t}\underline{0}\underline{t}\underline{0}$.

- a) Zeigen Sie $w \in \mathcal{L}(G)$ mittels einer Parallelableitung. (4 Punkte)
- b) Zeigen Sie $w \in \mathcal{L}(G)$ mittels einer Linksableitung. (4 Punkte)
- c) Welche Eigenschaften der Grammatik G ergeben sich aufgrund der Tatsache, dass w diese beiden unterschiedlichen Ableitungen besitzt? (4 Punkte)

☐ 4.0 VU Theoretische Informatik 1 ☐ 4.0 VU Theoretische Informatik für Lehramt Informatik I 1. Teilprüfung SS 2003 2. Mai 2003				
Kennzahl	Matrikelnummer	Familiennamen	Vorname	Gruppe D

Kreuzen Sie die zutreffende Lehrveranstaltung an und tragen Sie mit Kugelschreiber Kennzahl, Matrikelnummer, Nach- und Vornamen ein. Legen Sie einen Lichtbildausweis bereit.

Erlaubte Unterlagen: Skriptum, Vorlesungsfolien.

Schreiben Sie alle Lösungen auf diese Blätter und geben Sie die Prüfungsarbeit ohne Zusatzblätter ab. Sie haben 90 Minuten zur Bearbeitung der Aufgaben. Viel Erfolg!

- 1.) Geben Sie einen minimalen deterministischen Automaten an, der die durch den egrep-Ausdruck $(BA?B)*B?[BC]^*$ beschriebene Sprache akzeptiert. Überprüfen Sie die Minimalität mit einer Unterscheidbarkeitstabelle, d.h., geben Sie für jedes unterscheidbare Paar von Zuständen ein unterscheidendes Wort an. Stellen Sie den minimalen Automaten graphisch dar. (14 Punkte)
- 2.) Zeigen Sie: die Sprache $L = \{w \equiv w \mid w \in \{0, 1\}^*\}$ nicht regulär ist. (12 Punkte)
- 3.) Sei G die kontextfreie Grammatik $\langle \{S, T\}, \{\underline{s}, \underline{t}, 0, 1\}, P, S \rangle$, wobei P aus folgenden Produktionen besteht:

$$\begin{aligned} S &\Rightarrow \underline{s} \mid S T 0 \\ T &\Rightarrow \underline{t} \mid 1 S T \end{aligned}$$

Sei weiters w das Wort $\underline{s} \underline{t} 0 \underline{t} 0 1 \underline{s} \underline{t} 0 \underline{t} 0$.

- a) Zeigen Sie $w \in \mathcal{L}(G)$ mittels einer *Parallelableitung*. (4 Punkte)
 - b) Zeigen Sie $w \in \mathcal{L}(G)$ mittels einer *Linksableitung*. (4 Punkte)
 - c) Welche Eigenschaften der Grammatik G ergeben sich aufgrund der Tatsache, dass w diese beiden unterschiedlichen Ableitungen besitzt? (4 Punkte)
- 4.) Die Syntax der Datenbanksprache DATALOG ist folgendermaßen definiert. Ein *Programm* ist eine möglicherweise leere Folge von Klauseln. Eine *Klausel* ist entweder ein Faktum oder eine Regel, gefolgt von einem Punkt („.“). Ein *Faktum* besteht aus einer Atomformel. Eine *Regel* besteht aus einer Atomformel, gefolgt von den Zeichen „:-“ sowie einer nicht-leeren Liste von Atomformeln, die durch Kommas („.“) getrennt werden. Eine *Atomformel* ist ein Name, dem optional eine in runden Klammern („()“) eingeschlossene Argumentliste folgen kann. Eine *Argumentliste* ist eine nicht-leere Folge von Namen und Variablen, getrennt durch Kommas. Ein *Name* ist eine Folge von Buchstaben und Ziffern, die mit einem Kleinbuchstaben beginnt. Eine *Variable* ist eine Folge von Buchstaben und Ziffern, die mit einem Großbuchstaben beginnt.
- Geben Sie für die Sprache der DATALOG-Programme eine kontextfreie Grammatik in EBNF an. Verwenden Sie so weit als möglich EBNF-Notationen, um die Grammatik übersichtlich zu halten. (12 Punkte)

☐ 4.0 VU Theoretische Informatik 1 ☐ 4.0 VU Theoretische Informatik für Lehramt Informatik I 1. Teilprüfung SS 2003 2. Mai 2003				
Kennzahl	Matrikelnummer	Familiennamen	Vorname	Gruppe C

Kreuzen Sie die zutreffende Lehrveranstaltung an und tragen Sie mit Kugelschreiber Kennzahl, Matrikelnummer, Nach- und Vornamen ein. Legen Sie einen Lichtbildausweis bereit.

Erlaubte Unterlagen: Skriptum, Vorlesungsfolien.

Schreiben Sie alle Lösungen auf diese Blätter und geben Sie die Prüfungsarbeit ohne Zusatzblätter ab. Sie haben 90 Minuten zur Bearbeitung der Aufgaben. Viel Erfolg!

- 1.) Geben Sie einen minimalen deterministischen Automaten an, der die durch den egrep-Ausdruck $x?(x?yx)^*[xz]^*$ beschriebene Sprache akzeptiert. Überprüfen Sie die Minimalität mit einer Unterscheidbarkeitstabelle, d.h., geben Sie für jedes unterscheidbare Paar von Zuständen ein unterscheidendes Wort an. Stellen Sie den minimalen Automaten graphisch dar. (14 Punkte)

- 2.) Sei L die Sprache der wohlgeklammerten Ausdrücke, d.h., L ist die kleinste Menge, für die gilt:
 - $\epsilon \in L$
 - Wenn $w \in L$, dann auch $\underline{w} \in L$.
 - Wenn $w_1, w_2 \in L$, dann auch $w_1 w_2 \in L$.
Zeigen Sie: L ist nicht regulär. (12 Punkte)

- 3.) Sei G die kontextfreie Grammatik $\langle \{A, B\}, \{\underline{a}, \underline{b}, \underline{c}, \underline{d}\}, P, A \rangle$, wobei P aus folgenden Produktionen besteht:
$$A \Rightarrow \underline{a} \mid A B \underline{c}$$

$$B \Rightarrow \underline{b} \mid \underline{d} A B$$
Sei weiters w das Wort adabcbdadabc.
 - a) Zeigen Sie $w \in \mathcal{L}(G)$ mittels einer *Parallelableitung*. (4 Punkte)
 - b) Zeigen Sie $w \in \mathcal{L}(G)$ mittels einer *Linksableitung*. (4 Punkte)
 - c) Welche Eigenschaften der Grammatik G ergeben sich aufgrund der Tatsache, dass w diese beiden unterschiedlichen Ableitungen besitzt? (4 Punkte)

- 4.) Ein $\text{\LaTeX}$ -Dokument beginnt mit dem Kommando `\documentclass Optionen{Art}`, gefolgt von `\begin{document}`, dem eigentlichen Dokumentinhalt und beendet mit dem Kommando `\end{document}` (die geschweiften Klammern sind Teil von $\text{\LaTeX}$). *Art* ist ein einzelner Name. *Optionen* kann entweder fehlen, oder es ist eine in eckigen Klammern eingeschlossene nicht-leere Folge von Namen, die durch Beistriche getrennt werden. Ein *Name* ist eine nicht-leere Folge von Ziffern, Klein- und Großbuchstaben. Der *Dokumentinhalt* ist eine möglicherweise leere Folge von Texten und Listen in beliebiger Reihenfolge. Ein *Text* ist eine nicht-leere Folge von Buchstaben, Ziffern, Leerzeichen, Kommas und Punkten. Eine *Liste* beginnt mit `\begin{enumerate}` und endet mit `\end{enumerate}`; dazwischen liegt eine nicht-leere Folge von Listeneinträgen. Ein *Listeneintrag* besteht aus dem Kommando `\item` gefolgt von Texten und Listen in beliebiger Reihenfolge; ein Listeneintrag kann auch nur aus `\item` bestehen.

Geben Sie für die Sprache der $\text{\LaTeX}$ -Dokumente eine kontextfreie Grammatik in EBNF an. Verwenden Sie so weit als möglich EBNF-Notationen, um die Grammatik übersichtlich zu halten. (12 Punkte)



☐ 4.0 VU Theoretische Informatik 1				
☐ 4.0 VU Theoretische Informatik für Lehramt Informatik I				
21. Jänner 2003				
Kennzahl	Matrikelnummer	Familienname	Vorname	Gruppe
				A

- 1.) Geben Sie einen minimalen deterministischen Automaten an, der die durch den *egrep*-Ausdruck $(aab^*)^*a^*$ beschriebene Sprache akzeptiert. Überprüfen Sie die Minimalität mit einer Unterscheidbarkeitstabelle. Stellen Sie den minimalen Automaten graphisch dar. (10 Punkte)
- 2.) Ein *Ausdruck* in MODULA besteht aus einem einfachen Ausdruck, optional gefolgt von einem Relationssymbol und einem zweiten einfachen Ausdruck. Ein *einfacher Ausdruck* ist eine nicht-leere Folge von Termen, die durch je einen Additionsoperator getrennt sind; der gesamten Folge kann optional ein Plus- oder Minuszeichen vorangehen. Ein *Term* ist eine nicht-leere Folge von Faktoren, die durch je einen Multiplikationsoperator getrennt sind. Ein *Faktor* ist entweder eine Zahl, ein Ausdruck in runden Klammern oder das Wort NOT gefolgt von einem Faktor. Es gibt die *Relationssymbole* $=$, $\neq$, $<$, $\leq$, $>$ und $\geq$, die *Additionsoperatoren* $+$, $-$ und OR sowie die *Multiplikationsoperatoren* $*$, DIV, MOD und AND. Eine *Zahl* ist entweder eine Dezimal-, Oktal- oder Hexadezimalzahl. Eine *Oktalzahl* ist eine nicht-leere Folge der Ziffern 0 bis 7 gefolgt von dem Buchstaben B oder C. Eine *Dezimalzahl* ist eine nicht-leere Folge der Ziffern 0 bis 9. Eine *Hexadezimalzahl* ist eine beliebige (auch leere) Folge der Zeichen 0, ..., 9, A, ..., F, der eine Ziffer vorangeht und der Buchstabe H folgt.
- a) Geben Sie für die Sprache der Ausdrücke eine kontextfreie Grammatik in EBNF an. Verwenden Sie so weit als möglich EBNF-Notationen, um die Grammatik übersichtlich zu halten. (10 Punkte)
- b) Welchen Zweck haben die Buchstaben B, C und H, die Oktal- bzw. Hexadezimalzahlen beenden? Warum sind Hexadezimalzahlen nicht einfacher als „nicht-leere Folge von Hexadezimalziffern (0, ..., 9, A, ..., F)“ definiert? Welche Eigenschaft würde Ihre Grammatik dadurch verlieren? (3 Punkte)
- 3.) Zeigen Sie mit Hilfe des Hoare-Kalküls die Gültigkeit der folgenden partiellen Korrektheitsaussage über dem Datentyp $\mathbb{N}$.
- $$y=21 \{ \begin{array}{l} \text{begin } z \leftarrow 3y; x \leftarrow y+z \text{ end;} \\ \text{while } x \geq 4 \text{ do } \text{begin } x \leftarrow x-4; z \leftarrow z+7 \text{ end end} \end{array} \} z=210$$
- (10 Punkte)
- 4.) Erweitern Sie die imperative Sprache *AL* um eine repeat-Schleife, d.h., definieren Sie die formale Syntax und Semantik des Konstrukts $\text{repeat } \pi \text{ until } B$. Diese Schleife führt das Programmstück π aus und testet anschließend die Bedingung B . Liefert diese *true*, bricht die Schleife ab; andernfalls beginnt ein neuer Schleifendurchlauf. Es genügt, die neu zur Definition von *AL* hinzukommenden Regeln anzugeben. (7 Punkte)
- 5.) Überprüfen Sie mit Hilfe des *Sequentialkalküls*, ob die Konsequenzrelation
- $$A \supset (\neg B \vee C), \neg(C \vee (B \supset C)) \models A \wedge \neg B$$
- gilt. Geben Sie ein Gegenbeispiel an, falls sie nicht gilt. (10 Punkte)

☐ 4.0 VU Theoretische Informatik 1 ☐ 4.0 VU Theoretische Informatik für Lehramt Informatik I 28. November 2002				
Kennzahl	Matrikelnummer	Familienname	Vorname	Gruppe F

Kreuzen Sie die zutreffende Lehrveranstaltung an und tragen Sie **mit Kugelschreiber** Kennzahl, Matrikelnummer, Nach- und Vornamen ein. Legen Sie einen Lichtbildausweis bereit.

Erlaubte Unterlagen: Vorlesungsskriptum sowie alle PDF-Dokumente, die auf der Homepage der Lehrveranstaltung angeboten werden (Vorlesungsfolien, Lösung der letzten Prüfung, diverse Übungsbeispiele). Schreiben Sie alle Lösungen auf diese Blätter und geben Sie die Prüfungsarbeit **ohne Zusatzblätter** ab. Sie haben 120 Minuten zur Bearbeitung der Aufgaben. Viel Erfolg!

- 1.) Geben Sie einen minimalen deterministischen Automaten an, der die durch den egrep-Ausdruck $0+(0?1)*1*$ beschriebene Sprache akzeptiert. Überprüfen Sie die Minimalität mit einer Unterscheidbarkeitstabelle. Stellen Sie den minimalen Automaten graphisch dar. (10 Punkte)

- 2.) Zeigen Sie, dass die Grammatik $\langle \{S\}, \{\underline{a}, \underline{b}, \underline{c}\}, P, S \rangle$ mehrdeutig ist, wobei

$$P = \{S \Rightarrow \underline{a} S \mid \underline{b} \mid \underline{a} S \underline{c} S\}.$$

(10 Punkte)

- 3.) Zeigen Sie mit Hilfe des Hoare-Kalküls die Gültigkeit der folgenden partiellen Korrektheitsaussage über dem Datentyp $\mathbb{N}$.

$$z=3x \wedge y=x \{ \text{begin while } z > 0 \text{ do } z \leftarrow 0; \text{ begin } z \leftarrow z + y; y \leftarrow y + z \text{ end end} \} z=x \wedge y=2x$$

Hinweis: Verwenden Sie als Schleifeninvariante die Formel $y=x$.

(10 Punkte)

- 4.) Berechnen Sie $\mathcal{M}_{AL}(I, \pi)$, wobei $I(z) = 2$, $I(y) = 4$, $I(x) = 6$ und

$$\pi = \text{begin } y \leftarrow z * y; \text{ if } x < 5 \text{ then } z \leftarrow y + y \text{ else } y \leftarrow z \div y \text{ end}.$$

Geben Sie jeden Schritt bei der Auswertung von $\mathcal{M}_{AL}$, $\mathcal{M}_T$ und $\mathcal{M}_{PL}$ an. (10 Punkte)

- 5.) Überprüfen Sie, ob die Konsequenzrelation

$$B \uparrow \neg C, A \equiv \neg B, (B \wedge C) \supset (A \wedge D) \models A \wedge D$$

gilt. Verwenden Sie dazu das DPLL-Verfahren, d.h., schreiben Sie das Problem als Formel, deren Gültigkeit nachzuweisen ist, wandeln Sie die negierte Formel in eine KNF um und wenden Sie DPLL darauf an. Geben Sie ein Gegenbeispiel an, falls die Konsequenzrelation nicht gilt.

(10 Punkte)

☐ 4.0 VU Theoretische Informatik 1 ☐ 4.0 VU Theoretische Informatik für Lehramt Informatik I 2. Teilprüfung				
		SS 2003		27.6.2003
Kennzahl	Matrikelnummer	Familienname	Vorname	Gruppe
		Lösung		A

1.) Kreuzen Sie die jeweils zutreffenden Eigenschaften an. Beachten Sie, dass auch mehrere Eigenschaften gleichzeitig zutreffen können.

- a) Die Formel $(eq?(rest(x), nil) \wedge (\neg atom?(x) \wedge atom?(rest(x))))$ ist
☐ eine Tautologie; ☐ gültig in $\mathbb{L}$; ☐ erfüllbar in $\mathbb{L}$; ☒ unerfüllbar in $\mathbb{L}$.
(5 Punkte)

- b) Die Formel $((ist1?(x) \supset istleer?(pop(x))) \supset ist1?(pop(x)))$ ist
☐ eine Tautologie; ☐ gültig in $\mathbb{S}$; ☒ erfüllbar in $\mathbb{S}$; ☐ unerfüllbar in $\mathbb{S}$.
(5 Punkte)

2.) Finden Sie durch Konstruktion eines geeigneten Ableitungsversuchs im Sequentialkalkül entweder ein Gegenmodell oder den Nachweis, dass die Formel

$$(A \supset B) \supset ((A \vee (G \supset B)) \wedge (\neg G \vee B))$$

gültig ist.

(10 Punkte)

Lösung: Wir bilden folgenden Ableitungsversuch:

$$\begin{array}{c}
 \text{Anti-Axiom} \quad \text{Axiom} \\
 \frac{G \vdash A, B \quad B, G \vdash A, B}{A \supset B, G \vdash A, B} \supset\text{-l} \\
 \frac{A \supset B, G \vdash A, B}{A \supset B \vdash A, G \supset B} \supset\text{-r} \\
 \frac{A \supset B \vdash A, G \supset B}{A \supset B \vdash A \vee (G \supset B)} \vee\text{-r} \quad \vdots \quad \frac{A \supset B \vdash \neg G \vee B}{A \supset B \vdash \neg G \vee B} \\
 \frac{A \supset B \vdash A \vee (G \supset B) \quad A \supset B \vdash \neg G \vee B}{A \supset B \vdash (A \vee (G \supset B)) \wedge (\neg G \vee B)} \wedge\text{-r} \\
 \frac{A \supset B \vdash (A \vee (G \supset B)) \wedge (\neg G \vee B)}{\vdash (A \supset B) \supset ((A \vee (G \supset B)) \wedge (\neg G \vee B))} \supset\text{-r}
 \end{array}$$

Dem bis zum Anti-Axiom entwickelten Ast des gescheiterten Ableitungsversuchs kann man das Modell I mit $I(A) = f, I(B) = f, I(G) = t$ entnehmen.

3.) Verwenden Sie die semantische Methode, um die konjunktive und die disjunktive Normalform von $(D \uparrow A) \not\supset (B \supset \neg A)$ zu bilden.

Geben Sie beide Normalformen in Mengennotation an.

(10 Punkte)

Lösung:

A	B	D	$D \uparrow A$	$B \supset \neg A$	$(D \uparrow A) \not\supset (B \supset \neg A)$	K_I	D_I
f	f	f	t	t	f		$A \vee B \vee D$
f	f	t	t	t	f		$A \vee B \vee \neg D$
f	t	f	t	t	f		$A \vee \neg B \vee D$
f	t	t	t	t	f		$A \vee \neg B \vee \neg D$
t	f	f	t	t	f		$\neg A \vee B \vee D$
t	f	t	f	t	f		$\neg A \vee B \vee \neg D$
t	t	f	t	f	t	$A \wedge B \wedge \neg D$	
t	t	t	f	f	f		$\neg A \vee \neg B \vee \neg D$

Es gilt also:

$$\text{DNF}(F) = \bigvee_{I \in \text{ENV}, \mathcal{M}_T(I, F) = \text{t}} K_I = \{\{A, B, \neg D\}\}$$

$$\text{KNF}(F) = \bigwedge_{I \in \text{ENV}, \mathcal{M}_T(I, F) = \text{f}} D_I = \{\{A, B, D\}, \{A, B, \neg D\}, \{A, \neg B, D\}, \{A, \neg B, \neg D\}, \{\neg A, B, D\}, \{\neg A, B, \neg D\}, \{\neg A, \neg B, \neg D\}\}$$

4.) Verwenden Sie das DPLL-Verfahren, um zu testen, ob die Klauselmenge

$$\mathcal{K} = \{\{E, F\}, \{\neg E, G\}, \{\neg F, G, E\}, \{\neg G, H\}, \{\neg H, \neg G\}\}$$

unerfüllbar ist.

(10 Punkte)

Lösung: Da $\mathcal{K}$ nicht leer ist und weder die leere Klausel noch ein reines Literal noch eine Einerklausel enthält, wählen wir die Variable E und berechnen $\text{DPLL}(\text{Reduce}(\mathcal{K}, E))$.

$$\mathcal{K}' = \text{Reduce}(\mathcal{K}, E) = \{\{G\}, \{\neg G, H\}, \{\neg H, \neg G\}\}.$$

$\mathcal{K}$ enthält die Einerklausel $\{G\}$. Wir berechnen daher $\text{DPLL}(\text{Reduce}(\mathcal{K}', G))$.

$$\mathcal{K}'' = \text{Reduce}(\mathcal{K}', G) = \{\{H\}, \{\neg H\}\}.$$

$\mathcal{K}'$ enthält die Einerklausel $\{H\}$. Wir berechnen daher $\text{DPLL}(\text{Reduce}(\mathcal{K}'', H))$.

$$\mathcal{K}''' = \text{Reduce}(\mathcal{K}'', H) = \{\{\}\},$$

daher liefert $\text{DPLL}(\text{Reduce}(\mathcal{K}'', H)) = \text{DPLL}(\text{Reduce}(\mathcal{K}', G)) = \text{DPLL}(\text{Reduce}(\mathcal{K}, E))$ „unerfüllbar“. Daher muss nun der **else**-Zweig, also $\text{DPLL}(\text{Reduce}(\mathcal{K}, \neg E))$ ausgewertet werden.

Es gilt

$$\mathcal{K}^* = \text{Reduce}(\mathcal{K}, \neg E) = \{\{F\}, \{\neg F, G\}, \{\neg G, H\}, \{\neg H, \neg G\}\}.$$

$\mathcal{K}^*$ enthält die Einerklausel $\{F\}$. Wir berechnen daher $\text{DPLL}(\text{Reduce}(\mathcal{K}^*, F))$.

$$\mathcal{K}^{**} = \text{Reduce}(\mathcal{K}^*, F) = \{\{G\}, \{\neg G, H\}, \{\neg H, \neg G\}\}.$$

$\mathcal{K}^{**}$ enthält die Einerklausel $\{G\}$. Wir berechnen daher $\text{DPLL}(\text{Reduce}(\mathcal{K}^{**}, G))$.

$$\mathcal{K}^{***} = \text{Reduce}(\mathcal{K}^{**}, G) = \{\{H\}, \{\neg H\}\}.$$

$\mathcal{K}^{***}$ enthält die Einerklausel $\{H\}$. Wir berechnen daher $\text{DPLL}(\text{Reduce}(\mathcal{K}^{***}, H))$.

$$\text{Reduce}(\mathcal{K}^{***}, H) = \{\{\}\},$$

daher liefert $\text{DPLL}(\text{Reduce}(\mathcal{K}^{***}, H)) = \text{DPLL}(\text{Reduce}(\mathcal{K}^{**}, G)) = \text{DPLL}(\text{Reduce}(\mathcal{K}^*, F)) = \text{DPLL}(\mathcal{K})$ „unerfüllbar“.

$\mathcal{K}$ ist damit als unerfüllbar nachgewiesen.

5.) Zeigen Sie mit Hilfe des Hoare-Kalküls die Gültigkeit der folgenden partiellen Korrektheitsaussage über dem Datentyp $\mathbb{N}$.

$$z = 9 \wedge y = 1 \{ \text{while } 3 * z \neq 5 * y \text{ do } \underline{\text{begin}} \ z \leftarrow z + 3; y \leftarrow y + 4 \ \underline{\text{end}} \} \ z > 12 \wedge y = 9$$

(10 Punkte)

Lösung: Mit den Abkürzungen $P \equiv (z = 9 \wedge y = 1)$, $B \equiv 3z \neq 5y$ und $Q \equiv (z > 12 \wedge y = 9)$ ergibt sich folgende Ableitung:

$$\begin{array}{c} \text{(1)} \quad P \supset I \quad \text{(H4)} \quad \frac{\text{(H1)} \quad \frac{\text{(2)} \quad (I \wedge B) \supset I \left[\frac{y+4}{y} \right] \left[\frac{z+3}{z} \right]}{(I \wedge B) \{z \leftarrow z + 3\} I \left[\frac{y+4}{y} \right]} \quad \text{(T2)} \quad \frac{(I \wedge B) \{\text{begin } z \leftarrow z + 3; y \leftarrow y + 4 \text{ end}\} I}{P \{\text{while } B \text{ do } \text{begin } z \leftarrow z + 3; y \leftarrow y + 4 \text{ end}\} Q} \quad \text{(3)} \quad (I \wedge \neg B) \supset Q \end{array}$$

Auf Grund der Form der Schleife, kann eine Formel der Form $4z + c = 3y$ oder der Form $4z = 3z + y$, für ein Konstantensymbol c , als Invariante P gewählt werden. Auf Grund der Precondition P ergibt sich die Invariante $I \equiv 4z = 3y + 33$

Die 3 prädikatenlogischen Formeln (1), (2) und (3) lassen sich wie folgt als gültig in $\mathbb{N}$ erkennen:

(1) $(z = 9 \wedge y = 1) \supset 4z = 3y + 33$ ist gültig in $\mathbb{N}$, da in den natürlichen Zahlen $4 * 9 = 3 * 1 + 33$ gilt.

(2) $(4z = 3y + 33 \wedge 3z \neq 5y) \supset 4(z + 3) = 3(y + 4) + 33$.

Die rechte Seite der Implikation lässt sich zu $4z + 12 = 3y + 12 + 33$ vereinfachen. Zieht man nun 12 auf beiden Seiten der Gleichung ab, so erhält man die Gleichung der linken Seite der Implikation. Daher gilt für alle Environments: Wenn $4z = 3y + 33$ zu $\mathbf{t}$ evaluiert, dann evaluiert auch $4(z + 3) = 3(y + 4) + 33$ zu $\mathbf{t}$. Mit anderen Worten, die Implikation ist gültig in $\mathbb{N}$.

Alternative Argumentation:

Es gilt $4(z + 3) = 4z + 12$. Daher ist die rechte Seite der Implikation äquivalent zu $4z + 12 = 3(y + 4) + 33$. Setzt man nun die Gleichung $4z = 3y + 33$ aus der linken Seite der Implikation darin ein, so erhält man $(3y + 33) + 12 = 3(y + 4) + 33$. Da diese Gleichung in $\mathbb{N}$ offensichtlich gültig ist, ist die gesamte Implikation gültig in $\mathbb{N}$.

(3) $(4z = 3y + 33 \wedge 3z = 5y) \supset (z > 12 \wedge y = 9)$.

Durch Lösung des Gleichungssystems $4z = 3y + 33 \wedge 3z = 5y$ ergibt sich, dass die linke Seite der Implikation nur in einem Environment I mit $I(z) = 15$ und $I(y) = 9$ zu $\mathbf{t}$ evaluiert. In diesem Environment ist $(z > 12 \wedge y = 9)$ offensichtlich gültig. Daher ist die Implikation gültig in $\mathbb{N}$.

	AND, Konjunk	OR, Disjunk	Implikation	NAND	NOR	NEXOR, gdw.	XOR, exkl. Oder
t	t	t	t	f	f	t	f
t	f	t	f	f	f	f	t
f	t	t	t	f	f	t	t
f	f	f	f	t	t	f	t

Negation	¬
t	f
f	t

Ersetze alle Operatoren durch $\wedge$, $\vee$ und $\neg$.

$$\begin{aligned}
 A \supset B &= \neg A \vee B \\
 A \subset B &= A \vee \neg B \\
 A \uparrow B &= \neg A \vee \neg B \\
 (A \equiv B) &= (A \wedge B) \vee (\neg A \wedge \neg B) \quad (A \neq B) = (\neg A \wedge B) \vee (A \wedge \neg B)
 \end{aligned}$$

-A. Seien im Folgenden $L_{i,j}$ Literale für $i = 1, \dots, m$, $j = 1, \dots, n_i$.
Eine Formel ist in disjunktiver Normalform (DNF), wenn sie die Form

$$(L_{1,1} \wedge \dots \wedge L_{1,n_1}) \vee \dots \vee (L_{m,1} \wedge \dots \wedge L_{m,n_m})$$

besitzt, d.h., wenn sie eine Disjunktion von Konjunktionen von Literalen ist.
Eine Formel ist in konjunktiver Normalform (KNF, CNF), wenn sie die Form

$$(L_{1,1} \vee \dots \vee L_{1,n_1}) \wedge \dots \wedge (L_{m,1} \vee \dots \vee L_{m,n_m})$$

besitzt, d.h., wenn sie eine Konjunktion von Disjunktionen von Literalen ist. Die einzelnen Disjunktionen einer KNF werden auch Klauseln genannt, die KNF entsprechend auch Klauselnormalform.¹⁰

$$(H1) \quad \frac{(P \supset Q[t_i])}{P \{v \leftarrow t\} Q}$$

(Die Variablen aus $v \leftarrow t$ dürfen nicht quantifiziert in P und Q auftreten)

$$(H2) \quad \frac{P \{\alpha\} Q \quad Q \{\beta\} R}{P \{\text{begin } \alpha; \beta \text{ end}\} R}$$

$$(H3) \quad \frac{(P \wedge B) \{\alpha\} Q \quad (P \wedge \neg B) \{\beta\} Q}{P \{\text{if } B \text{ then } \alpha \text{ else } \beta\} Q}$$

$$(H4) \quad \frac{(P \supset INV) \quad (INV \wedge B) \{\alpha\} INV \quad ((INV \wedge \neg B) \supset Q)}{P \{\text{while } B \text{ do } \alpha\} Q}$$

$$(T1) \quad \frac{(P \wedge v = t) \{\beta\} R}{P \{\text{begin } v \leftarrow t; \beta \text{ end}\} R}$$

(v darf weder in P noch in t vorkommen.)

$$(T2) \quad \frac{P \{\alpha\} R[t_i]}{P \{\text{begin } \alpha; v \leftarrow t \text{ end}\} R}$$

$$(T3) \quad \frac{P \{\alpha\} INV \quad (INV \wedge B) \{\beta\} INV \quad ((INV \wedge \neg B) \supset R)}{P \{\text{begin } \alpha; \text{while } B \text{ do } \beta \text{ end}\} R}$$

$f : A \vee B$	$t : A \vee B$	$t : A \wedge B$	$t : A \wedge B$
$f : A$	$t : A \mid t : B$	$f : A \mid f : B$	$t : A$
$f : B$			$t : B$
$f : A \supset B$	$t : A \supset B$	$f : \neg A$	$t : \neg A$
$t : A$	$f : A \mid t : B$	$t : A$	$f : A$
$f : B$			

ng 4.2.: Die Regeln des Sequentialkalküls für die Aussagen

$$\begin{aligned}
 & \frac{f, A \vdash g, A}{f, A \vdash g, A} \quad \frac{f, A \vdash g, A \quad f, B \vdash g}{f, A \supset B \vdash g} \quad \frac{f, A \supset B \vdash g \quad f, B \vdash g}{f, A \vdash g} \\
 & \frac{f, A \vdash g, A \quad f, B \vdash g}{f, A \wedge B \vdash g} \quad \frac{f, A \wedge B \vdash g}{f, A \vdash g \quad f, B \vdash g} \quad \frac{f, A \vdash g, A \quad f, B \vdash g}{f, A \vee B \vdash g} \\
 & \frac{f, A \vdash g, A \quad f, B \vdash g}{f, A \supset B \vdash g} \quad \frac{f, A \supset B \vdash g \quad f, B \vdash g}{f, A \vdash g} \quad \frac{f, A \vdash g, A \quad f, B \vdash g}{f, A \wedge B \vdash g} \\
 & \frac{f, A \vdash g, A \quad f, B \vdash g}{f, A \vee B \vdash g} \quad \frac{f, A \vee B \vdash g}{f, A \vdash g \quad f, B \vdash g} \quad \frac{f, A \vdash g, A \quad f, B \vdash g}{f, A \supset B \vdash g}
 \end{aligned}$$

```

function DPLL (K: Klauselmenge): { „erfüllbar“, „unerfüllbar“ };
if K = {} then return „erfüllbar“;
if {} ∈ K then return „unerfüllbar“;
if K enthält ein reines Literal L then return DPLL(Reduce(K, L));
if K enthält eine Einerklausel {L} then return DPLL(Reduce(K, L));
Wähle eine Variable A in K;
if DPLL(Reduce(K, A)) = „erfüllbar“ then return „erfüllbar“
else return DPLL(Reduce(K, ¬A));

function Reduce (K: Klauselmenge, L: Literal): Klauselmenge;
K' := ∅;
forall K ∈ K do
    if L ∈ K then K' := K' ∪ {}
    else if ¬L ∈ K then K' := K' ∪ {K - {L}};
    else K' := K' ∪ {K}
return K';
  
```