

**Klausur zur Vorlesung
„Berechenbarkeit und Logik“
SS 2025 / July 21, 2025**

Vorname: _____

Nachname: _____

Matrikelnummer: _____

Aufgabe	Punktzahl	Erreicht
1	10	
2	6	
3	3	
4	4	
5	5	
6	5	
7	6	
8	6	
9	5	
10	6	
11	6	
12	8	
Bonus	0	
Σ	70	

Generelle Hinweise:

- Prüfungsdauer: **120 Minuten**
- Wenn Sie in der Klausur **35 Punkte** erreichen, haben Sie mit Sicherheit bestanden.
- Hilfsmittel: Ein beidseitig handbeschriebenes DIN-A4-Blatt.
- Benutzen Sie ein **dokumentenechtes Schreibgerät**.
- Überprüfen Sie die Ihnen ausgehändigte Klausur auf Vollständigkeit (**13 Aufgaben** auf 13 Seiten).
- Tragen Sie **auf jedes Blatt** Ihren **Namen** und Ihre **Matr.-Nr.** in die entsprechenden Felder ein.
- Schreiben Sie Ihre Lösungen in die dafür vorgesehenen Felder. Reicht der Platz in einem Feld nicht aus, so benutzen Sie die Rückseite des entsprechenden Blattes und vermerken Sie dies auf der Vorderseite. Reicht der Platz dennoch nicht aus, können Sie die Aufsicht nach zusätzlichen Blättern fragen.
- Schreiben Sie bitte **deutlich**. Unleserliche Lösungen sind ungültig.
- Ein **Täuschungsversuch** führt umgehend zum Ausschluss und **Nichtbestehen**. Es erfolgt keine Vorwarnung.
- Alle mitgeführten **elektronischen Geräte** sind vor der Klausur bzw. spätestens jetzt auszuschalten.

Inhaltliche Hinweise:

- In den WHILE- und LOOP-Programmen dürfen Sie, zusätzlich zur Definition aus der Vorlesung, auch die Addition, Multiplikation, die in der Vorlesung definierte (abgeschnittene) Subtraktion (sub bzw. $-$) und für eine Variable x die Bedingung `IF  $x = 0$  THEN  $P$  END` als WHILE- bzw. LOOP-berechenbar voraussetzen und in Ihren Programmen verwenden. Geben Sie an, in welcher Variable der Ausgabewert am Ende steht.
- Sie dürfen annehmen, dass die Addition, Multiplikation und die (abgeschnittene) Subtraktion sub zweier natürlicher Zahlen primitiv-rekursiv sind.
- Für die Konstruktion von μ - bzw. primitiv-rekursiven Funktionen und für WHILE- und LOOP-Programme darf die Schreibweise aus den Übungen benutzt werden.

Name:

Matrikelnummer:

Aufgabe 1. Beantworten Sie die folgenden vier Fragen. Für jede Teilaufgabe gibt die angegebene volle Punktzahl, wenn Sie die Aufgabe vollständig und korrekt beantwortet haben.

- (1) (3 Punkte) Sei L eine entscheidbare Sprache. Welche der folgenden Aussagen sind wahr?
- (a) Die charakteristische Funktion χ_L ist Turing-berechenbar.
 - (b) Die charakteristische Funktion χ_L ist total.
 - (c) Die "halbe charakteristische Funktion" χ'_L ist primitiv-rekursiv.
 - (d) L ist semi-entscheidbar und $\bar{L}$ ist entscheidbar.
 - (e) $\bar{L} \leq L$ und $L \leq \bar{L}$.
 - (f) Für das Halteproblem H gilt $H \leq \bar{L}$.

Lösung: a,b,d,e

- (2) (3 Punkte) Sei F eine erfüllbare, aber nicht gültige aussagenlogische Formel. Welche Aussagen sind wahr?
- (a) $\neg F \vee F$ ist unerfüllbar.
 - (b) $\neg F$ ist erfüllbar.
 - (c) $F \in \text{SAT}$.
 - (d) Es gibt eine aussagenlog. Formel G , so dass $G \wedge F$ gültig ist.
 - (e) Es gibt abzählbar unendlich viele Belegungen $\mathcal{B}$ mit $\mathcal{B}(F) = 1$.
 - (f) $\neg(F \leftrightarrow \neg F)$ ist gültig.

Lösung: b,c,f

- (3) (2 Punkte) Hat folgende PCP-Instanz eine Lösung? Begründen Sie!

$$\begin{pmatrix} bb & aaab & b & b \\ bbb & aa & ab & bc \end{pmatrix}.$$

Lösung: Es gibt eine Lösung. Die letzte Spalte können wir nicht verwenden, da c nur in der y -Komponente vorkommt. Aus den anderen drei Spalten bekommen z.B. die Lösung $(2, 3, 1)$.

- (4) (2 Punkte) Nennen Sie drei Algorithmen, um Erfüllbarkeit einer aussagenlogischen Formel zu testen! Die Algorithmen müssen nicht für alle Formeln funktionieren, sondern können auch nur für Spezialfälle anwendbar sein.

Lösung: Markierungsalgorithmus, Resolution, „Ausprobieren“ bzw. Ableiten aus der Wahrheitstafel

Name:

Matrikelnummer:

Aufgabe 2. (6 Punkte)

- (a) Ihre Firma POOL AG hat ein LOOP-Programm für eine Funktion $f : \mathbb{N}^2 \rightarrow \mathbb{N}$ geschrieben:

$$f(x, y) = \begin{cases} 3^{\min(x, y)+1}, & \text{falls } y^2 \leq t(x) + x, \\ t(x) + x, & \text{sonst,} \end{cases}$$

wobei es sich bei $t : \mathbb{N} \rightarrow \mathbb{N}$ um die Fibonacci-Zahlen handelt:

$$\begin{aligned} t(n) &= t(n-1) + t(n-2) \\ t(0) &= 0, \quad t(1) = 1 \end{aligned}$$

Allerdings weist das Programm ein paar Lücken auf. Ihr Kollege ist gerade im Urlaub und Sie müssen das LOOP-Programm vervollständigen (Eingabe x_1, x_2 und Ausgabe in x_1).

```

1  x3 := 0;                \\Berechnung t(x1) + x1
2  x4 := 1;
3  LOOP ... END;
```

Lösung:

```

LOOP x1 DO
  x5 := x3 + x4;
  x3 := x4;
  x4 := x5
END;
```

```

4  x3 := x3 + x1;          \\Ergebnis steht in x3
5  x4 := x2 * x2;
6  x5 := x4 - x3;
7  IF x5 = 0 THEN          \\Fallunterscheidung (wann Fall 1?)
8    x6 := x2 - x1;
9    ...;                  \\Minimum steht in x2
```

Lösung:

```

  LOOP x6 DO x2 := x1 END;
```

```

10  x3 := 3;               \\Berechnung 1. Fall (3 ^ ...)
11  ...
```

Lösung:

```

  LOOP x2 DO x3 := x3 * 3 END
```

```

12 END;
13 x1 := x3
```

Name:**Matrikelnummer:**

- (b) Ihr Chef Dr. Turing verlangt eine weitere Anpassung. Es soll eine Funktion $g : \mathbb{N}^2 \rightarrow \mathbb{N}$ berechnet werden, welche wie folgt definiert ist:

$$g(x, y) = \begin{cases} f(x, y) & \text{falls } x > 42!, \\ \text{undef.}, & \text{sonst.} \end{cases}$$

Die Funktion g soll also genau für $x \leq 42! = 42 \cdot 41 \cdots 2 \cdot 1$ in eine Endlosschleife geraten und ansonsten wie f definiert sein. Wie müssen Sie Ihren Code anpassen?

Hinweis: Das resultierende Programm muss kein LOOP-Programm mehr sein.

Lösung: Eingabe x_1, x_2 , Ausgabe in x_1

Sei P obiges LOOP-Programm zur Berechnung von f . Dann erhalten wir ein WHILE-Programm für g wie folgt:

```
x7 := x1;  
P;  
x8 := 42;  
x9 := 41;  
LOOP x9 DO x8 := x8 * x9; x9 := x9 - 1 END;  
x9 := x7 - x8;  
IF x9 = 0 THEN  
  WHILE x8 ≠ 0 DO x8 := 1 END  
END
```

Bemerkung: x_1 (bzw. x) muss vor Ausführung von P zwischengespeichert werden! Sonst wird der Wert mit $f(x, y)$ überschrieben.

Die letzten Zeilen bedeuten folgendes: Wenn $x - 42! \leq 0$ ist, so laufen wir in eine Endlosschleife, da x_8 zuvor auf $42!$ gesetzt wurde (was > 0 ist).

Name:**Matrikelnummer:**

Aufgabe 3. (3 Punkte) Geben Sie an, welche partielle Funktion $f : \mathbb{N}^2 \rightarrow \mathbb{N}$ das folgende GOTO-Programm berechnet (Eingabevariablen sind x_1 und x_2 , die Ausgabevariable ist x_1):

```

M1 : x3 := x1 + x2;
      x4 := x1 - 5;
      IF x4 = 0 THEN GOTO M3;
      GOTO M2;
M2 : x5 := x1 + x1;
      GOTO M4;
M3 : x1 := x1 + 4;
      HALT;
M4 : x5 := x5 - 12;
      IF x5 = 0 THEN GOTO M2;
      x1 := x3 + 1;
      HALT

```

Lösung:

$$f(x, y) = \begin{cases} x + y + 1, & \text{wenn } x > 6 \\ \text{undefiniert}, & \text{wenn } x = 6 \\ x + 4, & \text{wenn } 0 \leq x < 6 \end{cases}$$

Aufgabe 4. (4 Punkte) Geben Sie an, welche Funktionen von μf und μg berechnet werden, wobei f und g wie folgt definiert sind.

(a) $f : \mathbb{N}^2 \rightarrow \mathbb{N}$ mit $f(n, x) = (n - x)^2 - 4$

Lösung: $(\mu f)(x) = \begin{cases} x - 2 & \text{falls } x \geq 3, \\ 0 & \text{sonst.} \end{cases}$

(b) $g : \mathbb{N}^2 \rightarrow \mathbb{N}$ mit $g(n, x) = (x - 5) - (n \bmod 3)$

Lösung:

$$(\mu g)(x) = \begin{cases} 0 & \text{falls } x \leq 5, \\ x - 5 & \text{falls } x = 6 \text{ oder } x = 7, \\ \text{undefiniert} & \text{sonst.} \end{cases}$$

Name:

Matrikelnummer:

Aufgabe 5. (5 Punkte) Zeigen Sie, dass die folgenden Funktionen primitiv-rekursiv sind. In dieser Aufgabe dürfen Sie nicht die Äquivalenz zur LOOP-Berechenbarkeit nutzen.

(a) $f: \mathbb{N}^2 \rightarrow \mathbb{N}$ mit $f(x, y) = |x - y|$

Lösung: Die Funktion kann als $f(x, y) = (x - y) + (y - x)$ dargestellt werden. Da alle Komponenten primitiv-rekursiv sind, ist auch f primitiv-rekursiv.

(b) $g: \mathbb{N}^2 \rightarrow \mathbb{N}$ mit $g(y, x) = (x + 1) \cdot (x + 2) \cdots (x + y + 1)$

Lösung: Diese Funktion kann rekursiv definiert werden durch:

$$g(0, x) = x + 1 = s(x)$$

$$g(y + 1, x) = g(y, x) \cdot (x + y + 2)$$

Die rekursive Definition verwendet nur Addition, Multiplikation und Primitive Rekursion, daher ist g primitiv-rekursiv.

(c) $h: \mathbb{N}^3 \rightarrow \mathbb{N}$ mit $h(z, x, y) = \begin{cases} x + y & z = 0 \\ \max(x \cdot y, y + z) + (x - z) & z > 0 \end{cases}$

Lösung: Es gilt

$$h(0, x, y) = x + y$$

$$h(z + 1, x, y) = \max(x \cdot y, y + z + 1) + (x - (z + 1))$$

Die Funktion $k(x, y) = \max(x, y)$ wiederum ist auch primitiv-rekursiv (in der Klausur mit Begründung).

Da alle beteiligten Operationen primitiv-rekursiv sind, ist h insgesamt primitiv-rekursiv.

Name:**Matrikelnummer:**

Aufgabe 6. (5 Punkte) Gegeben sei die Funktion $f: \mathbb{N} \rightarrow \mathbb{N}$ mit

$$f(n) = n + (n \bmod 2)$$

Geben Sie eine Turingmaschine an, die f berechnet.

Hinweis: Beachten Sie, dass die Eingabezahl n zu Beginn als Binärzahl auf dem Band steht (zum Beispiel steht bei $n = 5$ zu Beginn 101 auf dem Band).

Achten Sie darauf, dass der Leseschreibkopf am Ende auf dem ersten Zeichen der Ausgabe steht.

Lösung: $M = (\{z_0, z_1, z_2, z_3, z_f\}, \{0, 1\}, \{0, 1, \square\}, \delta, z_0, \square, \{z_f\})$

$$\delta(z_0, x) = \{(z_0, x, R)\}, \quad x \in \{0, 1\}$$

$$\delta(z_0, \square) = \{(z_1, \square, L)\}$$

$$\delta(z_1, 0) = \{(z_3, 0, L)\}$$

$$\delta(z_1, 1) = \{(z_2, 0, L)\}$$

$$\delta(z_2, 1) = \{(z_2, 0, L)\}$$

$$\delta(z_2, 0) = \{(z_3, 1, L)\}$$

$$\delta(z_2, \square) = \{(z_3, 1, L)\}$$

$$\delta(z_3, x) = \{(z_3, x, L)\}$$

$$\delta(z_3, \square) = \{(z_f, \square, R)\}$$

Idee: Die Turingmaschine inkrementiert die Zahl um 1 genau dann, wenn sie ungerade ist. Sonst tut sie nichts.

Name:

Matrikelnummer:

Aufgabe 7. (6 Punkte) Betrachten Sie das folgende Problem:

$$L = \{F \mid F \text{ ist eine erfüllbare aussagenlog. Formel der Form } G \wedge H\}$$

Zeigen Sie, dass L **NP**-vollständig ist.

Hinweis: Dafür dürfen Sie nutzen, dass die Menge der erfüllbaren aussagenlog. Formeln SAT **NP**-vollständig ist (Satz von Cook).

Lösung: Für eine aussagenlogische Formel F definieren wir die Funktion $f(F) = F' = F \wedge (A \vee \neg A)$. Es gilt genau dann $\mathcal{B} \models F'$, wenn $\mathcal{B} \models F$. Also ist $F' \in L$ genau dann, wenn $F \in SAT$ ist, denn F' hat per Definition bereits die passende Form.

Die Reduktion f ist offenbar in Polynomialzeit berechenbar. Somit haben wir $SAT \leq_p L$ und L ist **NP**-hart.

Außerdem können wir $F \in L$ folgendermaßen testen: Scanne zunächst die Formel von links nach rechts und schaue, ob sie die Gestalt $G \wedge H$ hat (geht in Linearzeit). Danach raten wir nichtdeterministisch eine Belegung $\mathcal{B}$ und testen, ob $\mathcal{B} \models F$ gilt. Die Überprüfung geht also in **NP** und es gilt folglich $L \in \mathbf{NP}$.

Alternativ kann man L auf SAT reduzieren mit der Reduktion $f(F) = F$.

Insgesamt haben wir gezeigt, dass L **NP**-vollständig ist.

Name:

Matrikelnummer:

Aufgabe 8. (6 Punkte) Zeigen Sie durch geeignetes Anwenden des Markierungsalgorithmus, dass folgende Aussage wahr ist:

$$(\neg E \vee A), (\neg E \vee \neg C \vee D), (\neg E \vee \neg A \vee C), (\neg C \vee \neg A \vee \neg D \vee B) \models (B \vee \neg E)$$

Geben Sie an, welche atomaren Formeln in jedem Schritt markiert werden, jeweils mit Begründung.

Lösung: Dies gilt genau dann, wenn

$$(E \rightarrow A) \wedge (E \wedge C \rightarrow D) \wedge (E \wedge A \rightarrow C) \wedge (C \wedge A \wedge D \rightarrow B) \wedge (B \rightarrow 0) \wedge (1 \rightarrow E)$$

unerfüllbar ist.

- Markiere E wegen $1 \rightarrow E$.
- Markiere A weil E markiert und $E \rightarrow A$.
- Markiere C , weil E und A markiert und $E \wedge A \rightarrow C$.
- Markiere D , weil E und C markiert und $E \wedge C \rightarrow D$.
- Markiere B , weil C , A und D markiert und $C \wedge A \wedge D \rightarrow B$.
- Gib unerfüllbar aus, weil B markiert und $B \rightarrow 0$.

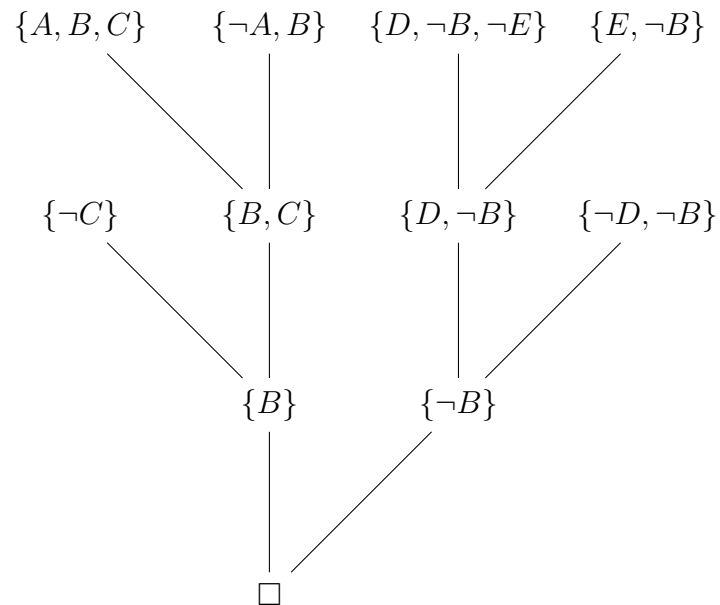
Daher ist die Aussage wahr.

Name:

Matrikelnummer:

Aufgabe 9. (5 Punkte) Verwenden Sie das Resolutionsverfahren der Aussagenlogik, um zu zeigen, dass die folgende Klauselmenge unerfüllbar ist:

$$\{\{\neg C\}, \{A, B, C\}, \{\neg A, B\}, \{D, \neg B, \neg E\}, \{E, \neg B\}, \{\neg D, \neg B\}\}$$

Lösung:

Name:

Matrikelnummer:

Aufgabe 10. Zeigen Sie, dass die folgenden Formeln erfüllbar, aber nicht gültig sind:

- (a) (3 Punkte) $F = \forall x \exists y ((P(f(x, x), y) \vee P(y, y)) \wedge \neg P(x, f(x, y)))$

Lösung: Sei $U_{\mathcal{A}} = \{0, 1\}$, $f^{\mathcal{A}}(x, y) = y$, $P^{\mathcal{A}} = \{(0, 0), (1, 1)\}$, dann gilt $\mathcal{A} \models F$, also ist F erfüllbar.

Sei $U_{\mathcal{A}} = \{0, 1\}$, $f^{\mathcal{A}}(x, y) = x$, $P^{\mathcal{A}} = \{(0, 0), (1, 1)\}$, dann gilt $\mathcal{A} \not\models F$, also ist F nicht gültig.

- (b) (3 Punkte) $G = \exists x \forall y (R(x, g(x, y)) \wedge \neg R(y, y)) \wedge \forall z (\neg R(a, z))$

Lösung: Sei $U_{\mathcal{A}} = \mathbb{Z}$, $R^{\mathcal{A}} = \{(0, 1)\}$, $g^{\mathcal{A}}(x, y) = x + 1$, $a^{\mathcal{A}} = 1$, dann gilt $\mathcal{A} \models G$, also ist G erfüllbar.

Sei $U_{\mathcal{A}} = \{0\}$, $R^{\mathcal{A}} = \{(0, 0)\}$, $g^{\mathcal{A}}(x, y) = x$, $a^{\mathcal{A}} = 0$, dann gilt $\mathcal{A} \not\models G$, also ist G nicht gültig.

Name:**Matrikelnummer:**

Aufgabe 11. (6 Punkte) Gegeben sei die folgende Formel F :

$$\forall x \exists y (P(x, z) \vee Q(f(x, y), f(z, x))) \wedge \forall x (\neg R(g(x, x)))$$

Überführen Sie F in eine erfüllbarkeitsäquivalente Aussage in Klauselform.

Lösung: Bereinigen:

$$\forall x_1 \exists y_1 (P(x_1, z) \vee Q(f(x_1, y_1), f(z, x_1))) \wedge \forall x_2 (\neg R(g(x_2, x_2)))$$

Freie Variablen ersetzen:

$$\forall x_1 \exists y_1 (P(x_1, a_z) \vee Q(f(x_1, y_1), f(a_z, x_1))) \wedge \forall x_2 (\neg R(g(x_2, x_2)))$$

Pränexform:

$$\forall x_1 \exists y_1 \forall x_2 ((P(x_1, a_z) \vee Q(f(x_1, y_1), f(a_z, x_1))) \wedge (\neg R(g(x_2, x_2))))$$

Skolemform:

$$\forall x_1 \forall x_2 ((P(x_1, a_z) \vee Q(f(x_1, f_{y_1}(x_1)), f(a_z, x_1))) \wedge (\neg R(g(x_2, x_2))))$$

Klauselform:

$$\forall x_1 \forall x_2 ((P(x_1, a_z) \vee Q(f(x_1, f_{y_1}(x_1)), f(a_z, x_1))) \wedge (\neg R(g(x_2, x_2))))$$

(Matrix ist bereits in KNF)

Name:

Matrikelnummer:

Aufgabe 12. Gegeben sei die Struktur $\mathcal{A} = (U_{\mathcal{A}}, I_{\mathcal{A}})$, wobei das Universum gegeben ist durch $U_{\mathcal{A}} = \{a, b\}^*$ und die Interpretationsfunktion $I_{\mathcal{A}}$ definiert ist durch:

- $f^{\mathcal{A}}(x, y) = xy$ (die Konkatination von Wörtern),
- $P^{\mathcal{A}} = \{w \in \{a, b\}^* \mid w \text{ enthält genau ein } a\}$
- $Q^{\mathcal{A}} = \{w \in \{a, b\}^* \mid |w| \text{ ist gerade}\}$
- $c^{\mathcal{A}} = aaa$

und „ $=$ “ wie üblich ein 2-stelliges Prädikatsymbol darstellt, das mit der Gleichheit interpretiert wird. Formulieren Sie folgende Aussagen als prädikatenlogische Formeln:

- (a) (2 Punkte) Für jedes Wort x mit gerader Länge gibt es ein Wort y , dessen Konkatination mit x ungerade Länge hat.

Lösung: $\forall x(Q(x) \rightarrow \exists y(\neg Q(f(x, y)))$

- (b) (2 Punkte) Das Wort aaa enthält genau drei a .

Lösung: $\exists x \exists y \exists z (P(x) \wedge P(y) \wedge P(z) \wedge c = f(x, f(y, z)))$

- (c) (2 Punkte) Das Wort aaa enthält kein Teilwort, das genau ein a enthält und gerade Länge hat.

Lösung: $\neg \exists x (P(x) \wedge Q(x) \wedge \exists z_1 \exists z_2 (c = f(z_1, f(x, z_2))))$

- (d) (2 Punkte) Jede Konkatination zweier Wörter ungerader Länge ergibt ein Wort gerader Länge.

Lösung: $\forall x \forall y ((\neg Q(x) \wedge \neg Q(y)) \rightarrow Q(f(x, y)))$

Name:

Matrikelnummer:

Aufgabe 13. (4 Punkte (Bonus))

Gegeben sei folgende Formel:

$$F = \forall x(W(x) \rightarrow J(x))$$

Dabei bedeutet $W(x)$, dass x ein Wanderer ist und $J(x)$, dass x gerne joggen geht. Ein Student behauptet, aus dieser Aussage folge direkt

$$\forall x J(x).$$

Wo liegt der Fehler in der Argumentation des Studenten? Begründen Sie Ihre Antwort.

Kann der Student $\exists x J(x)$ folgern?

Lösung: Die Aussage $\forall x(W(x) \rightarrow J(x))$ bedeutet „Alle Wanderer gehen gerne joggen.“ Der Student schließt jedoch fälschlicherweise, dass alle x (also auch Nicht-Wanderer) gerne joggen gehen. Das ist nicht korrekt.

Formal: Wenn $a \in U_{\mathcal{A}}$ kein Wanderer ist (also $W(a)$ sich zu 0 auswertet), kann $J(a)$ sich auch zu 0 auswerten und es gilt $\mathcal{A}_{[x/a]} \models F$, jedoch $\mathcal{A}_{[x/a]} \not\models J(a)$.

Auch die zweite Folgerung ist falsch. Es kann sein, dass es keine Wanderer *und* keine Menschen gibt, die gerne joggen gehen. Dann wäre F wahr und $\exists x J(x)$ falsch.