Einführung in die Theoretische Informatik

Tutorium – Woche 11

Esther Ney

Sommersemester 2025

Organisatorisches

Zulip

Für die diesjährige THEO-Vorlesung wurden **inoffizielle** Streams auf der ZULIP-Instanz der TUM INFORMATIK erstellt. Dieser wird von den meisten Tutor*innen, aber nicht von PROF. ALBERS gelesen; wir versuchen natürlich trotzdem, organisatorische Fragen bei Bedarf weiterzuleiten. Man kann mich dort auch in Direktnachrichten anschreiben.

- Tutorin: ESTHER NEY
- Tutorien: Dienstag 16-18 (Di-16-2) und Donnerstag 16-18 (Do-16-1)
- E-Mail: esther.ney@tum.de
- Aufbau einer Tutorstunde: kurze Wiederholung wichtiger Definitionen und Lemmata, dann selbständiges Bearbeiten der Aufgaben
- Folien: auf ZULIP oder https://reflect.moe/uni/theo25/

Disclaimer

Hinweis

Diese Folien sind keine offizielle Musterlösung und dienen lediglich als zusätzliches Material für die Nachbereitung der Tutorien.

Tutorienlösungen können fehlerhaft sein.

Im Zweifel gilt immer das, was in den offiziellen Vorlesungsfolien und in der Musterlösung steht.

Urheberrechtlicher Hinweis

Decidedly indecisive

Definition 5.28 – Entscheidbarkeit

Wir sagen, dass eine Menge $A \subseteq \Sigma^*$ *entscheidbar* ist, falls ihre charakteristische Funktion $\chi_A(a) := \begin{cases} 1 & \texttt{wenn} \quad a \in A \\ 0 & \texttt{wenn} \quad a \notin A \end{cases}$ berechenbar ist.

Wir sagen, dass ein Problem P entscheidbar ist, wenn $\{x \mid P(x)\}$ entscheidbar ist.

Definition 5.30 – Gödelisierung

Nicht alle Fragestellungen sind “offensichtlich” eine Teilmenge von Σ^*.

Implizit betrachten wir daher immer eine Kodierung über der Menge Σ^* – diesen Vorgang nennt man *Gödelisierung*. Siehe Folie 264.

Reduktion

Definition 5.36 – Reduktion

Wir sagen, dass eine Menge $A \subseteq \Sigma^*$ auf eine Menge $B \subseteq \Gamma^*$ *reduzierbar* ist, falls es eine totale (überall definierte) und berechenbare Funktion $f : \Sigma^* \to \Gamma^*$ gibt, sodass

$$w \in A \iff f(w) \in B \quad \texttt{für alle} \quad w \in \Sigma^*.$$

In diesem Fall schreiben wir auch $A \leq B$.

Reduktion ist transitiv – also $A \leq B$ und $B \leq C$ impliziert $A \leq C$ – und, wenn wir Äquivalenzklassen bilden, eine Halbordnung.

while(true) { goto conversion; }

Definition 5.17 – WHILE-Programme

Ein WHILE-Programm hat Variablen $x_1, \ldots, x_j$ und die Semantik $X := A + B$, $X := A - B$, $P_1; P_2$, `if` $X = 0$ `then` P_1 `else` P_2 `end`, `while` $X = 0$ `do` P `end`.

Wir können ein WHILE-Programm mit einer j-Band-Turingmaschine simulieren: jede Variable bekommt ein eigenes Band. Siehe auch Folien 247-249.

Definition 5.23 – GOTO-Programme

Ein GOTO-Programm hat Variablen $x_1, \ldots, x_j$ und markierte Anweisungen $(M_1 : A_1), \ldots, (M_k : A_k)$ mit Semantik $X := A + B$, $X := A - B$, `goto` M_i, `if` $X = n$ `then goto` M_i, `halt`.

Wir können ein GOTO-Programm mit einem WHILE-Programm simulieren, indem wir einen “program counter” einführen, der die neueste Marke darstellt. Siehe Satz 5.24.

Wir können ein WHILE-Programm mit einem GOTO-Programm simulieren, aber eine Konstruktion wurde nicht angegeben (ist aber leicht). Siehe Fakt 5.23.

Aufgabe T 11.1

Diese Aufgabe studiert die Konvertierung von WHILE-Programmen zu deterministischen Turingmaschinen.

(a) Sei $\Sigma := \{0, 1\}$. Geben Sie eine TM M_0 an, die die Funktion $f : \Sigma^* \to \Sigma^*$ berechnet, mit $f(w) = 0$ für alle w.

(b) Geben Sie eine TM M^- an, die eine Funktion $f : \Sigma^* \to \Sigma^*$ berechnet, wobei $f(\texttt{bin}(i)) := \texttt{bin}(i-1)$ für alle $i \in \mathbb{N}_{>0}$ gilt, und $f(0) := 0$. (Wie aus der Vorlesung bekannt, bezeichnet $\texttt{bin}(i)$ die Binärdarstellung von i, also das kürzeste $w \in \{0, 1\}^+$ mit $(w)_2 = i$.)

Für die folgenden Aufgaben dürfen Sie die Makros $\texttt{setzero}(i)$ (M_0) und $\texttt{dec}(i)$ (M^-) sowie $\texttt{inc}(i)$ (Beispiel 5.13) und $\texttt{iszero}(i)$ (Folie 248) verwenden. Hierbei beschreibt das Argument i, auf welchem Band einer k-bändigen TM das Makro operieren soll.

Die Notation aus Folie 247 ist ab hier wärmstens empfohlen.

(c) Seien $n, i \in \mathbb{N}$ fix. Konstruieren Sie nun eine k-Band-TM $\texttt{add}(i, n)$, die die Binärzahl auf Band i um n erhöht, und eine k-Band-TM $\texttt{sub}(i, n)$, die die Binärzahl auf Band i um n senkt (statt negativen Ergebnissen ergibt sich 0).

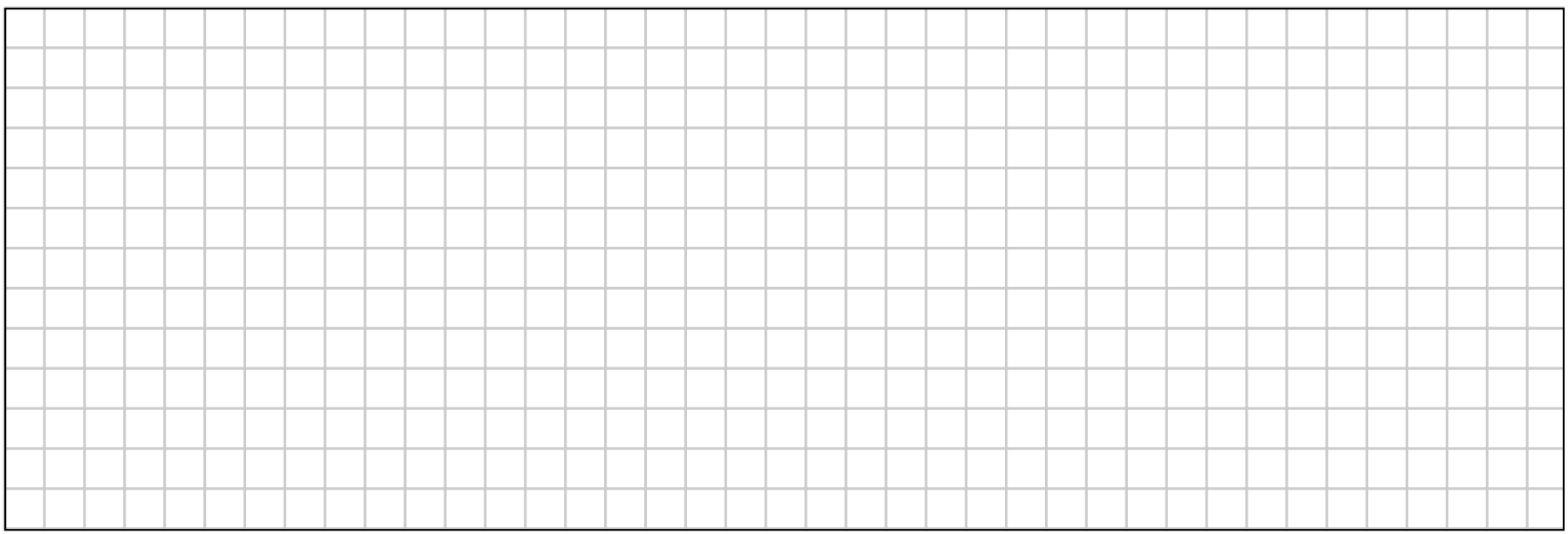
Hinweis: Aufgaben (d) und (e) sind auf der folgenden Folie.

Aufgabe T 11.1

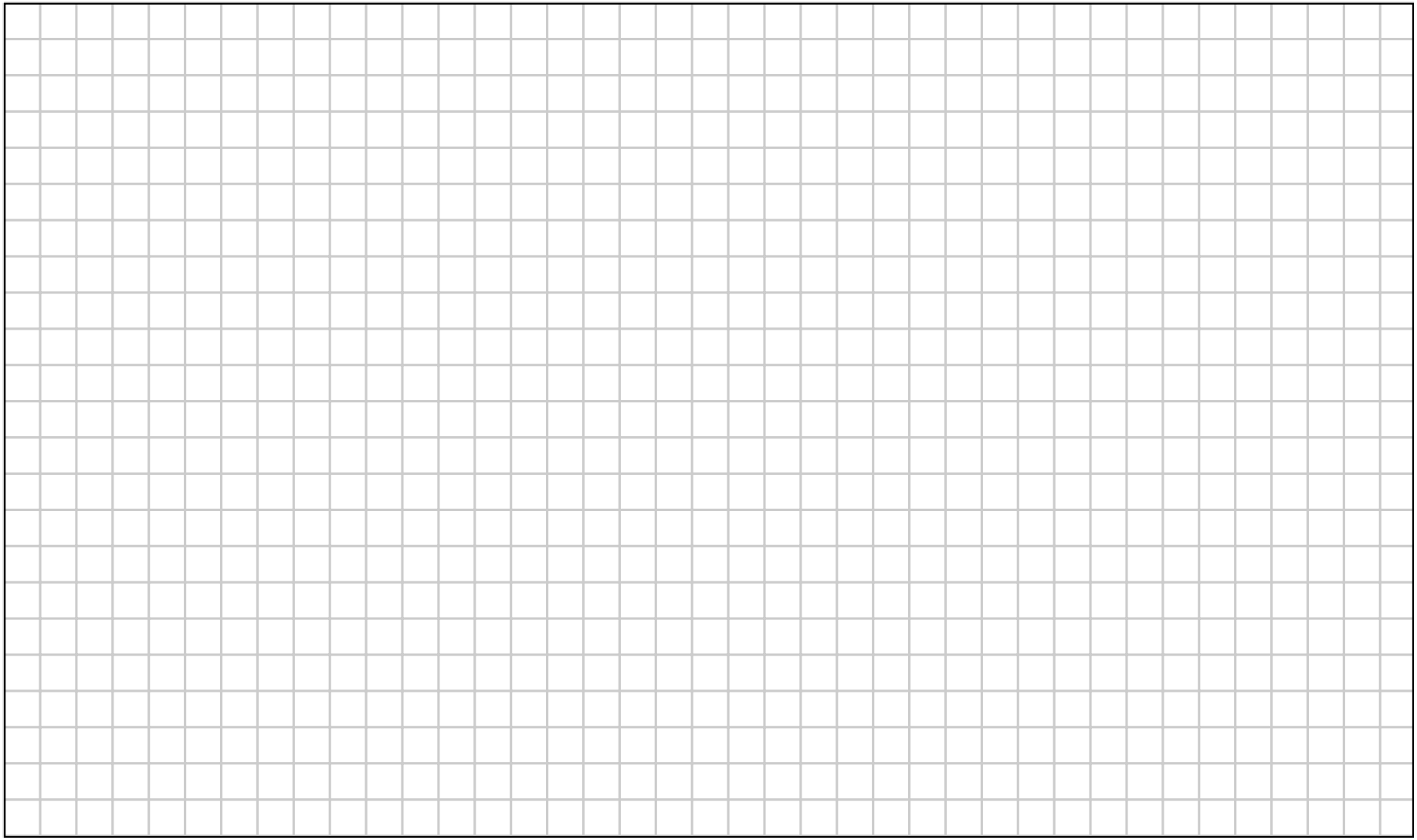
(d) Konstruieren Sie eine k-Band-TM $\texttt{copy}(i, j)$, die das WHILE-Programm $x_i := x_j$ simuliert.

(e) Konvertieren Sie folgendes WHILE-Programm, das eine Funktion $f : \mathbb{N} \to \mathbb{N}$ berechnet, zu einer k-Band-TM:

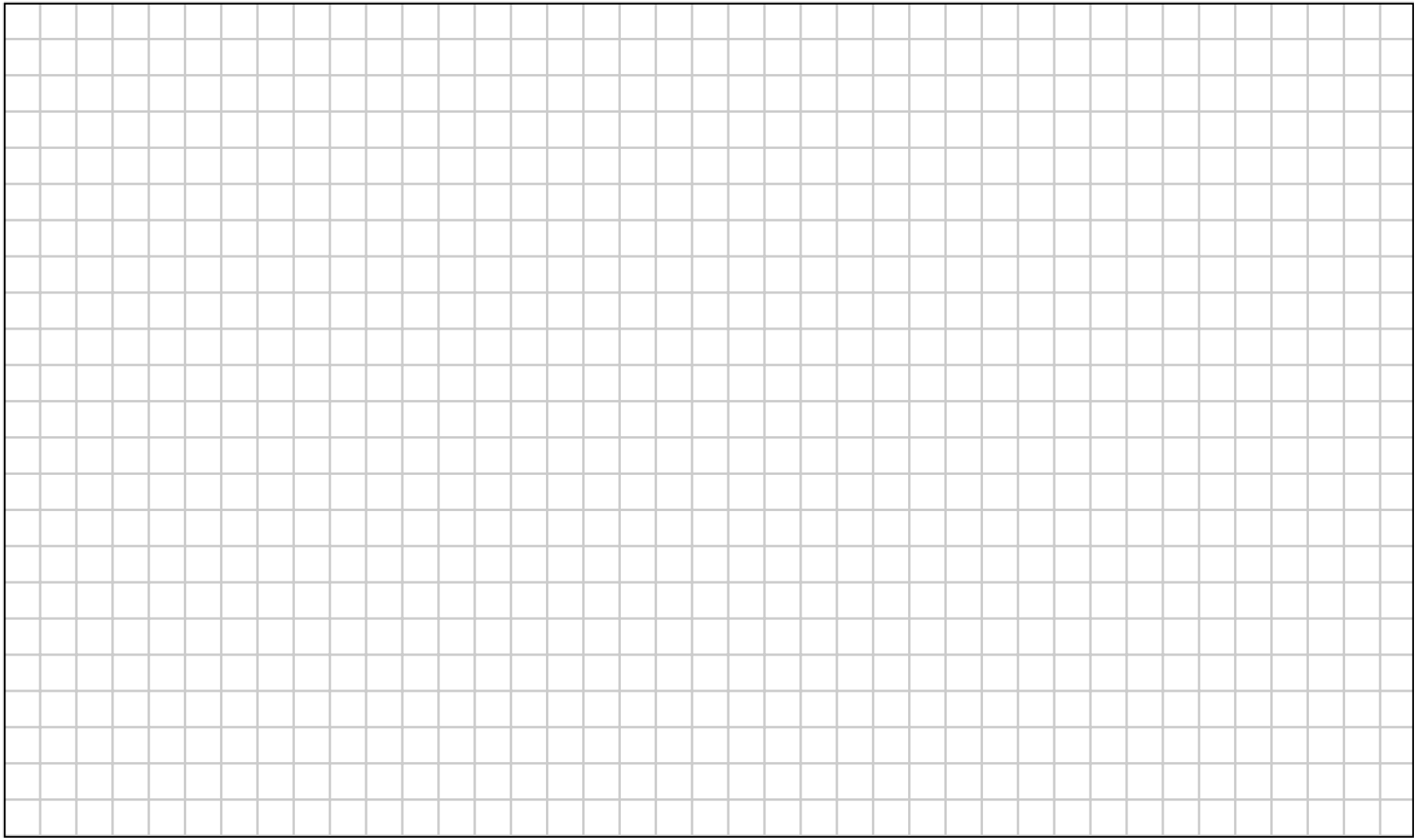
$\texttt{while } x_1 \neq 0 \texttt{ do } \{x_2 := 0; \texttt{while } x_1 \neq 0 \texttt{ do } \{x_1 := x_1 2; x_2 := x_2 + 1\} \texttt{ end while}; x_1 := x_2; x_0 := x_0 + 1\} \texttt{ end while}$



Aufgabe T 11.1



Aufgabe T 11.1

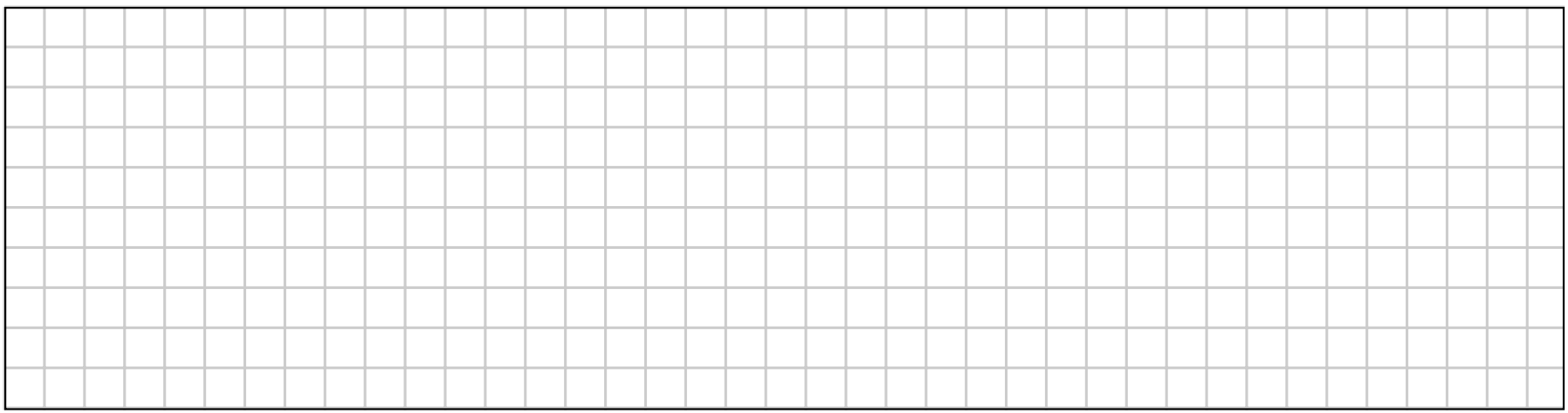


Aufgabe T 11.2

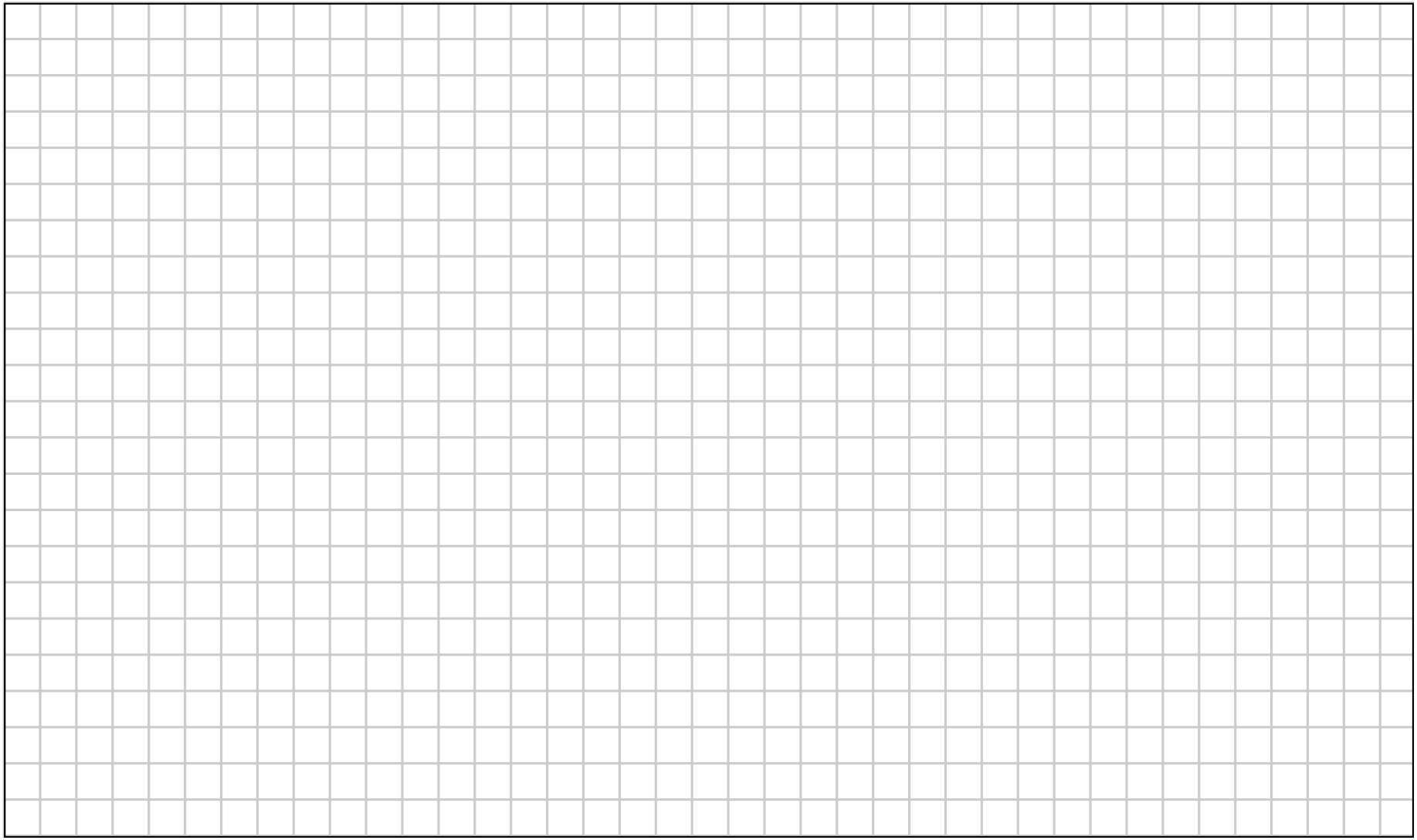
Geben Sie für die Funktion $f(x_1, x_2) = x_1 \mod x_2$, wobei $x_2 > 0$, sowohl ein WHILE-Programm als auch ein GOTO-Programm an, das die Funktion implementiert.

Verwenden Sie nur das Grundschema für WHILE- und GOTO-Programme und folgende Abkürzungen: $x_i := x_j$, $x_i := n$, $x_i := x_j + x_k$, $x_i := x_j - x_k$.

Halten Sie sich an die Konventionen aus der Vorlesung: Soll ein WHILE-Programm eine Funktion $f : \mathbb{N}_0^k \to \mathbb{N}_0$ berechnen, so werden die Variablen $x_1, \ldots, x_k$ entsprechend mit den Eingabewerten $n_1, \ldots, n_k \in \mathbb{N}_0$ initialisiert, während alle anderen Variablen zu Beginn auf 0 initialisiert werden. Nach Terminierung speichert die Variable x_0 den Funktionswert $f(n_1, \ldots, n_k)$.



Aufgabe T 11.2



Aufgabe T 11.3

Ordnen Sie die folgenden Satzanfänge allen Satzenden zu, sodass richtige Aussagen entstehen. Sei dazu $A, B \subseteq \{0,1\}^*$:

(a) Die Funktion χ_A ist berechenbar, wenn ...

(b) A ist entscheidbar, wenn ...

(c) B ist nicht entscheidbar, wenn ...

(i) ... $A \leq B$ gilt und B entscheidbar ist.

(ii) ... A entscheidbar ist.

(iii) ... $A \leq B$ gilt und A nicht entscheidbar ist.

Aufgabe T 11.4

Entscheiden Sie, ob die folgenden Aussagen wahr oder falsch sind. Falls die Aussage wahr ist, geben Sie einen Algorithmus an, der die charakteristische Funktion χ_L berechnet. Falls die Aussage falsch ist, leiten Sie einen Widerspruch zu einem Ergebnis der Vorlesung ab.

(a) Wenn A und B entscheidbare Sprachen sind, dann ist $A \cap B$ entscheidbar.

(b) Wenn A und $A \cup B$ entscheidbare Sprachen sind, dann ist B entscheidbar.

Altklausuraufgabe

Heute keine Altklausuraufgabe!

Wegen des Stoffumfangs **entfällt** heute die Altklausuraufgabe.