Deduktionstheorem:

T Theorie, F geschl. Formel

$T \cup \{F\} \vdash G \Rightarrow T \vdash F \to G$

Bew.: Sei $G_1, G_2, \dots, G_n = G$ ein formaler Beweis (mit Ax nur aus T) von G aus den nichtlog. Axiomen in T.

Induktiv konstruieren wir aus $G_1, G_2, \dots, G_n$ einen formalen Beweis $F \to G_1, \dots, F \to G_2, \dots, F \to G_{n-1}, \dots, F \to G_n$ (wobei zwischen $F \to G_i$ und $F \to G_{i+1}$ eventuell weitere Formeln stehen).

Von $n-1$ auf n:
Wenn $G_n = F$, dann nach $F \to G_{n-1}$ $F \to F$ anfügen, ist Tautologie ✓

$F \to F$ ist $F \to G_n$ ✓

Wenn G_n ein anderes Axiom ist ($G_n \in T$ oder G_n log. Axiom), dann nach $F \to G_{n-1}$ anfügen G_n, $G_n \to (F \to G_n)$,
Axiom ✓, Tautologie
$F \to G_n$ mit m.p.

Wenn G_n durch m.p. aus einem G_i und $(G_i \to G_n) = G_j$ mit $i, j < n$ hervorgeht, dann nach $F \to G_{n-1}$ anfügen:
$(F \to G_i) \to ((F \to (G_i \to G_n)) \to (F \to G_n))$ Tautologie
$(F \to (G_i \to G_n)) \to (F \to G_n)$ mit m.p. aus voriger Formel und $(F \to G_i)$ (schon vorgekommen nach IV.)
$F \to G_n$ mit m.p. aus vorhergehender Formel und $(F \to (G_i \to G_n)) = (F \to G_j)$ schon vorgekommen.

Zusammenfassend

G_i	$F \to G_i$
$(G_i \to G_n)$ ist G_j	$F \to G_j$ ist $F \to (G_i \to G_n)$
G_n	$(F \to G_i) \to ((F \to (G_i \to G_n)) \to (F \to G_n))$
	$(F \to (G_i \to G_n)) \to (F \to G_n)$
	$F \to G_n$

Wenn $G_n = \forall v\, G_j$ für ein $j < n$
Aus $F \to G_j$ (schon vorgekommen nach I.V.)
mit Generalisierungsregel: $\forall v (F \to G_j)$

Ax: $\forall v\, (F \to G_j) \to (F \to \forall v\, G_j)$
F geschl., insbes. v nicht frei in F mit m.p.
$F \to \forall v\, G_j$

G_j	$F \to G_j$	
⋮	⋮	
G_{n-1}	$F \to G_{n-1}$	
G_n	$\forall v (F \to G_j) \to (F \to \forall v\, G_j)$	axiom
	$\forall v\, (F \to G_j)$	Generalisierung
	$F \to \forall v\, G_j$	m.p.

Kor.: T Theorie, F geschl. Formel
$T \vdash F \iff T \cup \{\neg F\}$ inkonsistent
$(T \cup \{\neg F\} \vdash G \wedge \neg G)$

Bew:
$\Rightarrow$ klar ✓

$\Leftarrow$ $T \cup \{\neg F\} \vdash G \wedge \neg G$
dann (DT) $T \vdash \neg F \to (G \wedge \neg G)$
$(\neg F \to (G \wedge \neg G)) \to F$ Tautologie
also $T \vdash F$

Mit diesem Korollar andere Version des Gödelschen Vollständigk.-satzes beweisen.

Satz: T Theorie, F geschl. Formel
$T \vdash F \iff T \models F$
syntaktische Folgerung — semantische Folgerung

$\forall \mathcal{M}$ Struktur von L mit $\mathcal{M} \models T$ gilt $\mathcal{M} \models F$.

Bew:
($\Rightarrow$) Überprüfen, dass der Kalkül des formalen Beweisens „gültig" ist, d.h. log. Axiome sind allgem. gültig und für die Regeln gilt, jedes Modell der Voraussetzungen erfüllt die Konklusion.

zB. modus ponens: $\mathcal{M} \models A$ und $\mathcal{M} \models A \to B$ dann folgt $\mathcal{M} \models B$

($\Leftarrow$) Wenn $T \models F$, d.h. jedes Modell $\mathcal{M}$ von T erfüllt $\mathcal{M} \models F$, dann hat $T \cup \{\neg F\}$ kein Modell. Nach Gödels Vollständigkeitssatz ist $T \cup \{\neg F\}$ inkons. mit Lemma folgt $T \vdash F$.

Berechenbarkeit

Alle Versuche, den Begriff der Berechenbarkeit zu formalisieren, haben bisher die selbe Klasse von Funktionen ergeben.
"Church's thesis": dieser Begriff der Berechenbarkeit ist "der richtige".

Betrachten partielle Funktionen $\mathbb{N} \to \mathbb{N}$ bzw. $\mathbb{N}^n \to \mathbb{N}$,
partiell: dom $f = S \subseteq \mathbb{N}$ $\quad f: S \to \mathbb{N}$ bzw.
$T \subseteq \mathbb{N}^n$, $\quad g: T \to \mathbb{N}$

Definitionsbereich

Erste Formalisierung des Begriffs der Berechenbarkeit
Turing Maschinen

0	0	1	0	1	1	0	1	1

1 (Lesekopf)
q_i (Regel)

(TM)
T ist partielle Funktion $\quad T: (\text{dom } T) \subseteq \{0,1\} \times \{q_0, q_1, \dots, q_k\} \to \{0,1\} \times \{q_0, q_1, \dots, q_k\} \times \{L, R\}$

Turing-M mit k Regeln.

Interpretation: Wenn die Maschine im Zustand der Regel q_i auf einem Feld des Bandes steht, auf dem $x \in \{0,1\}$ steht, und $T(x, q_i) = (y, q_j, z)$ mit $y \in \{0,1\}$, $z \in \{L, R\}$
dann wird das Feld mit y überschrieben, zur Regel q_j übergegangen und je nach z nach L oder R ein Feld weiter gerückt.

Eine Turing-M. T berechnet die partielle Fkt $f: S \subseteq \mathbb{N} \to \mathbb{N}$ wenn $\forall n \in \mathbb{N}$, T auf einem Band mit einem Block von n "1" sonst "0". auf dem ersten "1" ganz links stehend gestartet, sich folgendermaßen verhält:
Wenn $f(n)\uparrow$ ($n \notin$ dom f), dann bricht die Berechnung nie ab;
wenn $f(n)\downarrow$ ($n \in$ dom f) dann bricht die Berechnung schließlich mit Ausgabe $f(n)$ ab, dh. auf dem Band ein Block von $f(n)$ "1" sonst "0", und Maschine steht auf dem ersten "1" links, im Zustand einer Regel q_i, die bei 1 undef. ist $(1, q_i) \notin \text{dom } T$

Allgem. Eine Berechnung bricht ab genau dann, wenn die Maschine im Zustand q_i auf einem Feld mit $x \in \{0,1\}$ zu stehen kommt wobei $(x, q_i) \notin$ dom T.

Turing-M. berechnet partielle Fkt $f: S \subseteq \mathbb{N}^n \to \mathbb{N}$ wenn sie sich, gestartet auf einem Band mit n Blöcken von "1", getrennt durch jeweils eine "0", der Länge $m_1, m_2, \dots, m_n$ (ganz links am Beginn des 1-ten Blocks stehend) verhält wie oben und $f(m_1, \dots, m_n)$ ausgibt, wenn $f(m_1, \dots, m_n)\uparrow$ und sonst nicht stehen bleiben.

Eine part. Fkt heißt Turing-berechenbar, wenn eine Turing-M, die die Fkt berechnet, existiert.

BSP) $f: \mathbb{N}^2 \to \mathbb{N}$ $\quad f(m,n) = m+n$ $\quad$ Turing-berechenbar.

$f: \mathbb{N} \to \mathbb{N}$ $\quad f(n) = 2n$ $\quad$ Turing-berechenbar.

$f: \mathbb{N}^2 \to \mathbb{N}$ $\quad f(m,n) = m \cdot n$ $\quad$ detto

EX Turing-M für Berechnung von $m+n$ konstruieren.

EX (ÜB 1-2) f, g Turing-berechenbar, dann auch $f \circ g$, und zwar mit einer Maschine mit $\leq i_f + i_g + 3$ Regeln, wenn Maschine für f bzw. g i_f bzw. i_g Regeln hat.

(wichtig: die Zahl der hinzukommenden Regeln hängt nicht von f, g ab).

BSP für eine nicht Turing berechenbare Fkt.

"busy beaver" $\quad b: \mathbb{N} \to \mathbb{N}$ (total)

$b(n)$ ist maximale Zahl von "1" die eine Turing-M mit n Regeln (äquiv. $\leq n$ Regeln) auf ein Band, das zu Beginn nur "0" enthält, schreiben kann (unter der Bedingung, dass die Berechnung irgendwann abbricht).

(Partiell-) rekursive Funktionen $\mathcal{R}$

$f: (\text{dom } f) \subseteq \mathbb{N}_0^n \to \mathbb{N}_0$ induktiv def. als kleinste Menge von Funktionen die folg. Funkt. enthält:

1) die (totalen) konstanten Fkt
2) die Nachfolger fkt $S(x) = x+1$
3) die Projektionen $\pi_i^{(n)}(x_1, \dots, x_n) = x_i$

und bzgl. folgender Operationen abgeschlossen ist.

1) Komposition $f_1, \dots, f_n, g \in \mathcal{R}$ $h(x_1, \dots, x_k) = g(f_1(x_1, \dots, x_n), f_2(x_1, \dots, x_n), \dots f_n(x_1, \dots, x_n))$
dann auch $h \in \mathcal{R}$
$h(x_1, \dots, x_n) \downarrow$, wenn für ein i $f_i(x_1, \dots, x_n) \downarrow$ oder wenn $g(\dots) \downarrow$

2) Rekursion: wenn g, h part. rek. in $n-1$ bzw. $n+1$ Var, dann ist folgendes f part. rek. in n Var:
$f(x_1, \dots, x_{n-1}, 0) = g(x_1, \dots, x_{n-1})$
$f(x_1, \dots, x_{n-1}, x_n + 1) = h(x_1, \dots, x_{n-1}, x_n, f(x_1, \dots, x_n))$

(Im Folgenden: rekursiv $\hat{=}$ part. rek.)

3) unbeschränkte Suche: $g \in \mathcal{R} \Rightarrow$ auch $f \in \mathcal{R}$

$$f(x_1, \dots, x_n) = \begin{cases} \min \{ y \in \mathbb{N}_0 \mid g(x_1, \dots, x_n, y) = 0 \} \\ \text{undef, wenn } \nexists y \in \mathbb{N}_0 \; g(x_1, \dots, x_n, y) = 0 \end{cases}$$

Klasse $\mathcal{R}$ der part. rek. Fkt stimmt mit der Klasse der Turing-berechenb. Funktionen überein.

Eine kleinere Klasse, echt enthalten in der Menge der total-rek. Fkt.; erhält man mit der beschränkten Suche 3) statt der unbeschränkten Suche 3. „Primitiv rekursive Fkt".
beschr. Suche: g primitiv rek, $n \in \mathbb{N}$

$$f(x_1, \dots, x_n, n) = \begin{cases} \min \{ y < n \mid g(x_1, \dots, x_n, y) = 0 \} & \text{falls existiert} \\ n \end{cases}$$

BSP) für total rek., nicht primitiv rek Fkt., Ackermann's generalized exponentials

$f_0(a, b) = a + b$
$f_1(a, b) = a \cdot b$
$f_2(a, b) = a^b$
$f_3(a, b) = (((a^a)^a)^{\dots})^a$ b Mal potentiert

Halteproblem für Turing-Maschinen

Alle Turing-M. sind **aufzählbar** in dem Sinn, man kann sie mit Nummern $\in \mathbb{N}$ durchnummerieren, und es gibt eine Turing-M die aus der Nummer n die **n-te Maschine** berechnet und umgekehrt.

Zuerst muss man die Turing-M als endliche Folge von nat. Zahlen anschreiben.

Dann **universelle** Turing-M konstruieren:
Maschine verhält sich bei Input von m, n so wie die n-te Turing-M bei Eingabe von m:
wenn n-te TM bei Input m nicht stehen bleibt, dann bleibt T bei Input n, m nicht stehen.
wenn n-te TM bei Input m stehen bleibt, dann bleibt T bei Input n, m schließlich stehen und es steht dasselbe auf dem Band wie nach der Berechnung der n-ten TM mit Input m.

T_n Funktion der n-ten TM zuordnen:
- wenn T_n bei Input m schließlich stehen bleibt mit Output K $T_n(m) = K$;
- wenn stehen bleibt mit sinnlosem Output $T_n(m) = 0$;
- $T_n(m)$ undef., wenn T_n nicht stehen bleibt.

Die part. rek. Fkt, die der universellen TM T zugeordnet ist

$$T(n,m) = T_n(m).$$

Das **Halte**-Funktion für TM ist nicht rekursiv.

$$h(n,m) = \begin{cases} 1 & T_n(m) \underset{}{\overset{(\text{def.})}{\downarrow}} \text{ (äquiv: Input von m bleibt stehen} \\ 0 & T_n(m) \underset{(\text{undef.})}{\uparrow} \text{ (äquiv: n-te TM bleibt bei Input von m \underline{nicht} stehen!} \end{cases}$$

diese Fkt nicht rek:

indirekt: Ang. rekursiv, dann auch

$$f(n) = \begin{cases} 0 & \text{, wenn } h(n,n) = 0 \\ \text{undef} & \text{, wenn } h(n,n) = 1 \end{cases}$$

wenn f rek. wäre, dann würde f von einer TM mit Nummer m berechnet
(Es wäre f(m) definiert genau dann wenn f(m) undef.)

busy beaver fcn nicht rekursiv.

$b(n)=$ max. Anzahl von "1" die eine TM mit n Zuständen q_i auf ein leeres Band geschrieben haben kann, wenn sie schließlich stehen bleibt.

Ang. TM mit m Zuständen berechnet b
TM mit k Zuständen berechnet $f(x)=2x$
TM mit 1 Zustand schreibt zusätzlichen "1" aufs Band

$$(0,q_0) \mapsto (1,q_1,R)$$
$$(1,q_0) \mapsto (1,q_0,R)$$

Verknüpfung von f, g die von TM mit n bzw. m Zuständen berechnet werden, kann durch TM mit $n+m+\varepsilon$ (ε unabh. von f und g) Zuständen berechnet werden.

$c(n) = \underset{\text{busy beaver}}{b}(2n)+1$ berechnet durch hintereinander ausführen der Funktionen:

Konstante n, gefolgt von $x \mapsto 2x$, gefolgt von b gefolgt von Maschine, die einen zusätzlichen "1" aufs Band schreibt.

$c(n) = b(2n)+1$ kann mit TM mit $n+k+m+1+3\varepsilon$
für großes n, $n > k+m+1+3\varepsilon$ ist
$n+k+m+1+3\varepsilon < 2n$ und TM, die $c(n)=b(2n)+1$ berechnet, hat $< 2n$ Zustände und schreibt aber $b(2n)+1$ "1" aufs Band. Widerspruch.

<u>Für Mengen $\subseteq \mathbb{N}_0$:</u>
$S \subseteq \mathbb{N}_0$ heißt **rekursiv**, wenn

$$\chi_S(x) = \begin{cases} 1 & x \in S \\ 0 & x \notin S \end{cases}$$

rekursiv ist.

$S \subseteq \mathbb{N}_0$ heißt **rekursiv aufzählbar** (r.e., e.e.), wenn S als Bild einer part. rek. Fkt. $f: (\text{dom } f) \subseteq \mathbb{N}_0 \to \mathbb{N}_0$ auftritt.
(äquivalent wenn S als dom f einer part. rek. Fkt f auftritt)
Es gilt: S rek. $\Leftrightarrow$ S, $\mathbb{N}_0 \setminus S$ $\boxed{\text{beide r.e.}}$