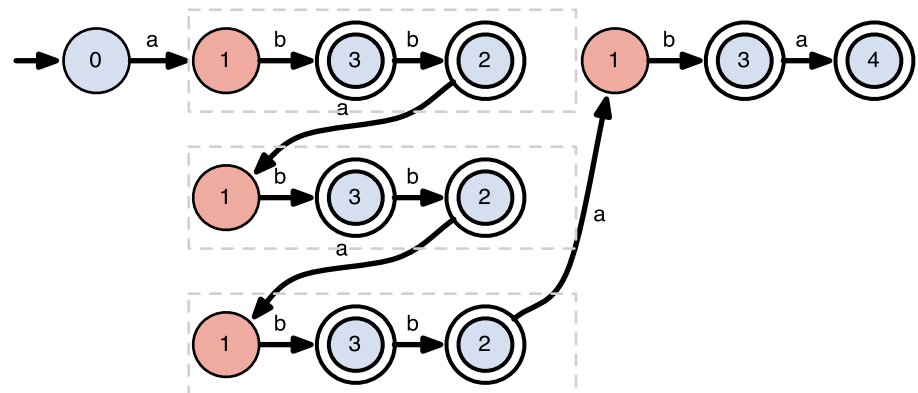


Berechenbarkeitstheorie

19. Vorlesung



Dr. Franziska Jahnke

Institut für Mathematische Logik und Grundlagenforschung

WWU Münster

Erinnerung: Logische Theorien

2

19. Vorlesung

Erinnerung: Logische Theorien

2

- Was ist eine **mathematische Aussage**?

Erinnerung: Logische Theorien

2

- Was ist eine **mathematische Aussage**?

Bsp. $\forall q \exists p \forall x, y [p > q \wedge (x, y > 1 \rightarrow xy \neq p)]$

Erinnerung: Logische Theorien

2

- Was ist eine **mathematische Aussage**?

Bsp. $\forall q \exists p \forall x, y [p > q \wedge (x, y > 1 \rightarrow xy \neq p)]$
(Es gibt unendlich viele Primzahlen)

Erinnerung: Logische Theorien

2

- Was ist eine **mathematische Aussage**?

Bsp. $\forall q \exists p \forall x, y [p > q \wedge (x, y > 1 \rightarrow xy \neq p)]$
(Es gibt unendlich viele Primzahlen)

Bsp. $\forall q \exists p \forall x, y [p > q \wedge (x, y > 1 \rightarrow (xy \neq p \wedge xy \neq p + 2))]$

- Was ist eine **mathematische Aussage**?

Bsp. $\forall q \exists p \forall x, y [p > q \wedge (x, y > 1 \rightarrow xy \neq p)]$
(Es gibt unendlich viele Primzahlen)

Bsp. $\forall q \exists p \forall x, y [p > q \wedge (x, y > 1 \rightarrow (xy \neq p \wedge xy \neq p + 2))]$
(*twin prime conjecture*)

- Was ist eine **mathematische Aussage**?

Bsp. $\forall q \exists p \forall x, y [p > q \wedge (x, y > 1 \rightarrow xy \neq p)]$
(Es gibt unendlich viele Primzahlen)

Bsp. $\forall q \exists p \forall x, y [p > q \wedge (x, y > 1 \rightarrow (xy \neq p \wedge xy \neq p + 2))]$
(*twin prime conjecture*)

- **Aussagen** sind Wörter über folgendem Alphabet

- Was ist eine **mathematische Aussage**?

Bsp. $\forall q \exists p \forall x, y [p > q \wedge (x, y > 1 \rightarrow xy \neq p)]$
(Es gibt unendlich viele Primzahlen)

Bsp. $\forall q \exists p \forall x, y [p > q \wedge (x, y > 1 \rightarrow (xy \neq p \wedge xy \neq p + 2))]$
(*twin prime conjecture*)

- **Aussagen** sind Wörter über folgendem Alphabet

- \wedge, \vee, \neg (boolsche Operatoren)
- $(,), [,]$ (Klammern)
- x (für Variablen), $x_3 = xxx$ etc.
- \exists, \forall (Quantoren)
- R_1, R_2, \dots, R_k (Relationen)
- $,$ (Komma)

- Was ist eine **mathematische Aussage**?

Bsp. $\forall q \exists p \forall x, y [p > q \wedge (x, y > 1 \rightarrow xy \neq p)]$
(Es gibt unendlich viele Primzahlen)

Bsp. $\forall q \exists p \forall x, y [p > q \wedge (x, y > 1 \rightarrow (xy \neq p \wedge xy \neq p + 2))]$
(*twin prime conjecture*)

- **Aussagen** sind Wörter über folgendem Alphabet

- \wedge, \vee, \neg (boolsche Operatoren)
- $(,), [,]$ (Klammern)
- x (für Variablen), $x_3 = xxx$ etc.
- \exists, \forall (Quantoren)
- R_1, R_2, \dots, R_k (Relationen)
- $,$ (Komma)

- **atomare Formeln** sind Wörter der Form $R_i(x_{l_1}, x_{l_2}, \dots, x_{l_j})$, wobei j die **Stelligkeit** des Symbols R_i heißt

- Was ist eine **mathematische Aussage**?

Bsp. $\forall q \exists p \forall x, y [p > q \wedge (x, y > 1 \rightarrow xy \neq p)]$
(Es gibt unendlich viele Primzahlen)

Bsp. $\forall q \exists p \forall x, y [p > q \wedge (x, y > 1 \rightarrow (xy \neq p \wedge xy \neq p + 2))]$
(*twin prime conjecture*)

- **Aussagen** sind Wörter über folgendem Alphabet

- \wedge, \vee, \neg (boolsche Operatoren)
- $(,), [,]$ (Klammern)
- x (für Variablen), $x_3 = xxx$ etc.
- \exists, \forall (Quantoren)
- R_1, R_2, \dots, R_k (Relationen)
- $,$ (Komma)

- **atomare Formeln** sind Wörter der Form $R_i(x_{l_1}, x_{l_2}, \dots, x_{l_j})$, wobei j die **Stelligkeit** des Symbols R_i heißt

- zu jedem Symbol R_i ist eine Stelligkeit angegeben

- **Formeln** sind Wörter über über dem Alphabet $\{\mathbf{x}, (,), [,], \forall, \exists, R_1, \dots, R_k, ,)$ für die gilt:

- **Formeln** sind Wörter über über dem Alphabet $\{\mathbf{x}, (,), [,], \forall, \exists, R_1, \dots, R_k, ,)\}$ für die gilt:
 - 1.) es sind atomare Formeln, oder

■ **Formeln** sind Wörter über über dem Alphabet

$\{\mathbf{x}, (,), [,], \forall, \exists, R_1, \dots, R_k, ,)$ für die gilt:

- 1.) es sind atomare Formeln, oder
- 2.) es sind Wörter der Form $(\phi_1 \wedge \phi_2)$, $(\phi_1 \vee \phi_2)$, (ϕ_1) oder $\neg\phi_1$, wobei ϕ_1, ϕ_2 kürzere Formeln sind, oder

■ **Formeln** sind Wörter über dem Alphabet

$\{\mathbf{x}, (,), [,], \forall, \exists, R_1, \dots, R_k, ,)$ für die gilt:

- 1.) es sind atomare Formeln, oder
- 2.) es sind Wörter der Form $(\phi_1 \wedge \phi_2)$, $(\phi_1 \vee \phi_2)$, (ϕ_1) oder $\neg\phi_1$, wobei ϕ_1, ϕ_2 kürzere Formeln sind, oder
- 3.) es sind Wörter der Form $\exists x_i[\phi]$, oder $\forall x_i[\phi]$, wobei ϕ eine kürzere Formel ist.

- **Formeln** sind Wörter über dem Alphabet $\{\mathbf{x}, (,), [,], \forall, \exists, R_1, \dots, R_k, ,)\}$ für die gilt:
 - 1.) es sind atomare Formeln, oder
 - 2.) es sind Wörter der Form $(\phi_1 \wedge \phi_2)$, $(\phi_1 \vee \phi_2)$, (ϕ_1) oder $\neg\phi_1$, wobei ϕ_1, ϕ_2 kürzere Formeln sind, oder
 - 3.) es sind Wörter der Form $\exists x_i[\phi]$, oder $\forall x_i[\phi]$, wobei ϕ eine kürzere Formel ist.
- wir nehmen an, dass alle Formeln in Pränex-Form gegeben sind, d.h. alle Quantoren stehen vorne

- **Formeln** sind Wörter über dem Alphabet $\{\mathbf{x}, (,), [,], \forall, \exists, R_1, \dots, R_k, ,)\}$ für die gilt:
 - 1.) es sind atomare Formeln, oder
 - 2.) es sind Wörter der Form $(\phi_1 \wedge \phi_2)$, $(\phi_1 \vee \phi_2)$, (ϕ_1) oder $\neg\phi_1$, wobei ϕ_1, ϕ_2 kürzere Formeln sind, oder
 - 3.) es sind Wörter der Form $\exists x_i[\phi]$, oder $\forall x_i[\phi]$, wobei ϕ eine kürzere Formel ist.
- wir nehmen an, dass alle Formeln in Pränex-Form gegeben sind, d.h. alle Quantoren stehen vorne
- eine Variable heißt **freie Variable**, wenn sie nicht quantifiziert ist

- **Formeln** sind Wörter über dem Alphabet $\{\mathbf{x}, (,), [,], \forall, \exists, R_1, \dots, R_k, ,)\}$ für die gilt:
 - 1.) es sind atomare Formeln, oder
 - 2.) es sind Wörter der Form $(\phi_1 \wedge \phi_2)$, $(\phi_1 \vee \phi_2)$, (ϕ_1) oder $\neg\phi_1$, wobei ϕ_1, ϕ_2 kürzere Formeln sind, oder
 - 3.) es sind Wörter der Form $\exists x_i[\phi]$, oder $\forall x_i[\phi]$, wobei ϕ eine kürzere Formel ist.
- wir nehmen an, dass alle Formeln in Pränex-Form gegeben sind, d.h. alle Quantoren stehen vorne
- eine Variable heißt **freie Variable**, wenn sie nicht quantifiziert ist
- eine **Aussage** auch **Satz** ist eine Formel ohne freie Variablen

- **Formeln** sind Wörter über dem Alphabet $\{\mathbf{x}, (,), [,], \forall, \exists, R_1, \dots, R_k, ,)\}$ für die gilt:
 - 1.) es sind atomare Formeln, oder
 - 2.) es sind Wörter der Form $(\phi_1 \wedge \phi_2)$, $(\phi_1 \vee \phi_2)$, (ϕ_1) oder $\neg\phi_1$, wobei ϕ_1, ϕ_2 kürzere Formeln sind, oder
 - 3.) es sind Wörter der Form $\exists x_i[\phi]$, oder $\forall x_i[\phi]$, wobei ϕ eine kürzere Formel ist.
- wir nehmen an, dass alle Formeln in Pränex-Form gegeben sind, d.h. alle Quantoren stehen vorne
- eine Variable heißt **freie Variable**, wenn sie nicht quantifiziert ist
- eine **Aussage** auch **Satz** ist eine Formel ohne freie Variablen

Bsp.1: $\exists x_1[\forall x_2[(R_1(x_2) \wedge R_2(x_2, x_1, x_3))]]$

- **Formeln** sind Wörter über dem Alphabet $\{\mathbf{x}, (,), [,], \forall, \exists, R_1, \dots, R_k, ,)\}$ für die gilt:
 - 1.) es sind atomare Formeln, oder
 - 2.) es sind Wörter der Form $(\phi_1 \wedge \phi_2)$, $(\phi_1 \vee \phi_2)$, (ϕ_1) oder $\neg\phi_1$, wobei ϕ_1, ϕ_2 kürzere Formeln sind, oder
 - 3.) es sind Wörter der Form $\exists x_i[\phi]$, oder $\forall x_i[\phi]$, wobei ϕ eine kürzere Formel ist.
- wir nehmen an, dass alle Formeln in Pränex-Form gegeben sind, d.h. alle Quantoren stehen vorne
- eine Variable heißt **freie Variable**, wenn sie nicht quantifiziert ist
- eine **Aussage** auch **Satz** ist eine Formel ohne freie Variablen

Bsp.1: $\exists x_1[\forall x_2[(R_1(x_2) \wedge R_2(x_2, x_1, x_3))]]$
 (Formel mit einer freien Variable x_3)

- **Formeln** sind Wörter über dem Alphabet $\{\mathbf{x}, (,), [,], \forall, \exists, R_1, \dots, R_k, ,)\}$ für die gilt:
 - 1.) es sind atomare Formeln, oder
 - 2.) es sind Wörter der Form $(\phi_1 \wedge \phi_2)$, $(\phi_1 \vee \phi_2)$, (ϕ_1) oder $\neg\phi_1$, wobei ϕ_1, ϕ_2 kürzere Formeln sind, oder
 - 3.) es sind Wörter der Form $\exists x_i[\phi]$, oder $\forall x_i[\phi]$, wobei ϕ eine kürzere Formel ist.
- wir nehmen an, dass alle Formeln in Pränex-Form gegeben sind, d.h. alle Quantoren stehen vorne
- eine Variable heißt **freie Variable**, wenn sie nicht quantifiziert ist
- eine **Aussage** auch **Satz** ist eine Formel ohne freie Variablen

Bsp.1: $\exists x_1[\forall x_2[(R_1(x_2) \wedge R_2(x_2, x_1, x_3))]]$
 (Formel mit einer freien Variable x_3)

Bsp.2: $\exists x_1[\forall x_2[(\neg R_1(x_1) \wedge (\neg R_2(x_2) \vee R_3(x_1, x_2)))]]$

- **Formeln** sind Wörter über dem Alphabet

$\{\mathbf{x}, (,), [,], \forall, \exists, R_1, \dots, R_k, ,)\}$ für die gilt:

- 1.) es sind atomare Formeln, oder
- 2.) es sind Wörter der Form $(\phi_1 \wedge \phi_2)$, $(\phi_1 \vee \phi_2)$, (ϕ_1) oder $\neg\phi_1$, wobei ϕ_1, ϕ_2 kürzere Formeln sind, oder
- 3.) es sind Wörter der Form $\exists x_i[\phi]$, oder $\forall x_i[\phi]$, wobei ϕ eine kürzere Formel ist.

- wir nehmen an, dass alle Formeln in Pränex-Form gegeben sind, d.h. alle Quantoren stehen vorne
- eine Variable heißt **freie Variable**, wenn sie nicht quantifiziert ist
- eine **Aussage** auch **Satz** ist eine Formel ohne freie Variablen

Bsp.1: $\exists x_1[\forall x_2[(R_1(x_2) \wedge R_2(x_2, x_1, x_3))]]$
(Formel mit einer freien Variable x_3)

Bsp.2: $\exists x_1[\forall x_2[(\neg R_1(x_1) \wedge (\neg R_2(x_2) \vee R_3(x_1, x_2)))]]$
(Formel ohne freie Variable = Aussage)

- Was bedeutet eine Aussage? Dazu muss man spezifizieren: 4

- Was bedeutet eine Aussage? Dazu muss man spezifizieren: 4
 - 1.) Was sagen die Relationen aus?

- Was bedeutet eine Aussage? Dazu muss man spezifizieren: 4
 - 1.) Was sagen die Relationen aus?
 - 2.) Aus welchem Universum werden die Variablen gewählt?

- Was bedeutet eine Aussage? Dazu muss man spezifizieren: 4
 - 1.) Was sagen die Relationen aus?
 - 2.) Aus welchem Universum werden die Variablen gewählt?
 - 3.) Was bedeuten die boolschen Operationen und Quantoren?

- Was bedeutet eine Aussage? Dazu muss man spezifizieren: 4
 - 1.) Was sagen die Relationen aus?
 - 2.) Aus welchem Universum werden die Variablen gewählt?
 - 3.) Was bedeuten die boolschen Operationen und Quantoren?

- Was bedeutet eine Aussage? Dazu muss man spezifizieren: 4
 - 1.) Was sagen die Relationen aus?
 - 2.) Aus welchem Universum werden die Variablen gewählt?
 - 3.) Was bedeuten die boolschen Operationen und Quantoren?

- Das Universum und die Bedeutung der Relationen werden durch das **Modell** beschrieben.

- Was bedeutet eine Aussage? Dazu muss man spezifizieren: 4
 - 1.) Was sagen die Relationen aus?
 - 2.) Aus welchem Universum werden die Variablen gewählt?
 - 3.) Was bedeuten die boolschen Operationen und Quantoren?
- Das Universum und die Bedeutung der Relationen werden durch das **Modell** beschrieben.
 - formal: $\mathcal{M} = (U, P_1, P_2, \dots, P_k)$ heißt Modell

- Was bedeutet eine Aussage? Dazu muss man spezifizieren: 4

- 1.) Was sagen die Relationen aus?
- 2.) Aus welchem Universum werden die Variablen gewählt?
- 3.) Was bedeuten die boolschen Operationen und Quantoren?

- Das Universum und die Bedeutung der Relationen werden durch das **Modell** beschrieben.

formal: $\mathcal{M} = (U, P_1, P_2, \dots, P_k)$ heißt Modell

Menge (Universum)

Bedeutung Relationen (P_i beschreibt R_i)

- Was bedeutet eine Aussage? Dazu muss man spezifizieren: 4

- 1.) Was sagen die Relationen aus?
- 2.) Aus welchem Universum werden die Variablen gewählt?
- 3.) Was bedeuten die boolschen Operationen und Quantoren?

- Das Universum und die Bedeutung der Relationen werden durch das **Modell** beschrieben.

formal: $\mathcal{M} = (U, P_1, P_2, \dots, P_k)$ heißt Modell

Menge (Universum)

Bedeutung Relationen (P_i beschreibt R_i)

- Eine Aussage ϕ ist entweder wahr oder falsch in einem Modell

- Was bedeutet eine Aussage? Dazu muss man spezifizieren: 4
 - 1.) Was sagen die Relationen aus?
 - 2.) Aus welchem Universum werden die Variablen gewählt?
 - 3.) Was bedeuten die boolschen Operationen und Quantoren?

- Das Universum und die Bedeutung der Relationen werden durch das **Modell** beschrieben.

formal: $\mathcal{M} = (U, P_1, P_2, \dots, P_k)$ heißt Modell

Menge (Universum)

Bedeutung Relationen (P_i beschreibt R_i)

- Eine Aussage ϕ ist entweder wahr oder falsch in einem Modell
- Wenn ϕ wahr im Modell \mathcal{M} , sagen wir \mathcal{M} ist ein Modell für ϕ

- Was bedeutet eine Aussage? Dazu muss man spezifizieren: 4
 - 1.) Was sagen die Relationen aus?
 - 2.) Aus welchem Universum werden die Variablen gewählt?
 - 3.) Was bedeuten die boolschen Operationen und Quantoren?

- Das Universum und die Bedeutung der Relationen werden durch das **Modell** beschrieben.

formal: $\mathcal{M} = (U, P_1, P_2, \dots, P_k)$ heißt Modell

Menge (Universum)

Bedeutung Relationen (P_i beschreibt R_i)

- Eine Aussage ϕ ist entweder wahr oder falsch in einem Modell
- Wenn ϕ wahr im Modell \mathcal{M} , sagen wir \mathcal{M} ist ein Modell für ϕ

Bsp.: $\phi = \forall y[\exists x[R_1(x, x, y)]]$

- Was bedeutet eine Aussage? Dazu muss man spezifizieren: 4

- 1.) Was sagen die Relationen aus?
- 2.) Aus welchem Universum werden die Variablen gewählt?
- 3.) Was bedeuten die boolschen Operationen und Quantoren?

- Das Universum und die Bedeutung der Relationen werden durch das **Modell** beschrieben.

formal: $\mathcal{M} = (U, P_1, P_2, \dots, P_k)$ heißt Modell

Menge (Universum)

Bedeutung Relationen (P_i beschreibt R_i)

- Eine Aussage ϕ ist entweder wahr oder falsch in einem Modell
- Wenn ϕ wahr im Modell \mathcal{M} , sagen wir \mathcal{M} ist ein Modell für ϕ

Bsp.: $\phi = \forall y[\exists x[R_1(x, x, y)]]$

- $\mathcal{M}_1 = (\mathbf{R}, \text{PLUS})$ ist Modell für ϕ
 $\text{PLUS} = \{(a, b, c) \mid a + b = c\}$

- Was bedeutet eine Aussage? Dazu muss man spezifizieren: 4
 - 1.) Was sagen die Relationen aus?
 - 2.) Aus welchem Universum werden die Variablen gewählt?
 - 3.) Was bedeuten die boolschen Operationen und Quantoren?

- Das Universum und die Bedeutung der Relationen werden durch das **Modell** beschrieben.

formal: $\mathcal{M} = (U, P_1, P_2, \dots, P_k)$ heißt Modell

Menge (Universum)

Bedeutung Relationen (P_i beschreibt R_i)

- Eine Aussage ϕ ist entweder wahr oder falsch in einem Modell
- Wenn ϕ wahr im Modell \mathcal{M} , sagen wir \mathcal{M} ist ein Modell für ϕ

Bsp.: $\phi = \forall y[\exists x[R_1(x, x, y)]]$

- $\mathcal{M}_1 = (\mathbf{R}, \text{PLUS})$ ist Modell für ϕ
 $\text{PLUS} = \{(a, b, c) \mid a + b = c\}$
- $\mathcal{M}_2 = (\mathbf{N}, \text{PLUS})$ ist kein Modell für ϕ

- Sprache aller wahren Aussagen für ein Modell \mathcal{M} heißt **Theorie⁵ von \mathcal{M}** - Schreibweise $\text{Th}(\mathcal{M})$.

- Sprache aller wahren Aussagen für ein Modell \mathcal{M} heißt **Theorie⁵ von \mathcal{M}** - Schreibweise $\text{Th}(\mathcal{M})$.
- **Erste Theorie:** Betrachte die natürlichen Zahlen \mathbb{N} mit Addition, d.h. dreistellige Relation $\text{PLUS} = \{(a, b, c) \in \mathbb{N}^3 \mid a + b = c\}$

- Sprache aller wahren Aussagen für ein Modell \mathcal{M} heißt **Theorie⁵ von \mathcal{M}** - Schreibweise $\text{Th}(\mathcal{M})$.
- **Erste Theorie:** Betrachte die natürlichen Zahlen \mathbb{N} mit Addition, d.h. dreistellige Relation $\text{PLUS} = \{(a, b, c) \in \mathbb{N}^3 \mid a + b = c\}$

Satz 29

Sei $(\mathbb{N}, \text{PLUS})$ das Modell der natürlichen Zahlen mit der Addition. Dann ist $\text{Th}((\mathbb{N}, \text{PLUS}))$ entscheidbar.

- Sprache aller wahren Aussagen für ein Modell \mathcal{M} heißt **Theorie⁵ von \mathcal{M}** - Schreibweise $\text{Th}(\mathcal{M})$.
- **Erste Theorie:** Betrachte die natürlichen Zahlen \mathbb{N} mit Addition, d.h. dreistellige Relation $\text{PLUS} = \{(a, b, c) \in \mathbb{N}^3 \mid a + b = c\}$

Satz 29

Sei $(\mathbb{N}, \text{PLUS})$ das Modell der natürlichen Zahlen mit der Addition. Dann ist $\text{Th}((\mathbb{N}, \text{PLUS}))$ entscheidbar.

- **Zweite Theorie:** Betrachte die natürlichen Zahlen \mathbb{N} mit Addition und Multiplikation, d.h. zwei dreistellige Relationen $\text{PLUS} = \{(a, b, c) \in \mathbb{N}^3 \mid a + b = c\}$ (schreibe kurz: $+$) und $\text{MAL} = \{(a, b, c) \in \mathbb{N}^3 \mid a \cdot b = c\}$ (schreibe kurz: \cdot)

- Sprache aller wahren Aussagen für ein Modell \mathcal{M} heißt **Theorie⁵ von \mathcal{M}** - Schreibweise $\text{Th}(\mathcal{M})$.
- **Erste Theorie:** Betrachte die natürlichen Zahlen \mathbb{N} mit Addition, d.h. dreistellige Relation $\text{PLUS} = \{(a, b, c) \in \mathbb{N}^3 \mid a + b = c\}$

Satz 29

Sei $(\mathbb{N}, \text{PLUS})$ das Modell der natürlichen Zahlen mit der Addition. Dann ist $\text{Th}((\mathbb{N}, \text{PLUS}))$ entscheidbar.

- **Zweite Theorie:** Betrachte die natürlichen Zahlen \mathbb{N} mit Addition und Multiplikation, d.h. zwei dreistellige Relationen $\text{PLUS} = \{(a, b, c) \in \mathbb{N}^3 \mid a + b = c\}$ (schreibe kurz: $+$) und $\text{MAL} = \{(a, b, c) \in \mathbb{N}^3 \mid a \cdot b = c\}$ (schreibe kurz: \cdot)

Satz 33

Sei $(\mathbb{N}, +, \cdot)$ das Modell der natürlichen Zahlen mit Addition und Multiplikation. Dann ist $\text{Th}((\mathbb{N}, +, \cdot))$ **nicht entscheidbar**.

Beweisbare Aussagen

6

19. Vorlesung

Beweisbare Aussagen

6

- Sei $\pi = (S_1, S_2, \dots, S_n)$ eine Folge von Aussagen für die gilt:

Beweisbare Aussagen

6

- Sei $\pi = (S_1, S_2, \dots, S_n)$ eine Folge von Aussagen für die gilt:
 - S_1 ist axiomatisch definierte wahre Aussage

Beweisbare Aussagen

6

- Sei $\pi = (S_1, S_2, \dots, S_n)$ eine Folge von Aussagen für die gilt:
 - S_1 ist axiomatisch definierte wahre Aussage
 - S_{i+1} folgt aus S_i durch axiomatisch beschriebene Ableitungsregeln

Beweisbare Aussagen

6

- Sei $\pi = (S_1, S_2, \dots, S_n)$ eine Folge von Aussagen für die gilt:
 - S_1 ist axiomatisch definierte wahre Aussage
 - S_{i+1} folgt aus S_i durch axiomatisch beschriebene Ableitungsregeln
- Sei π heißt **Beweis von** S_n .

Beweisbare Aussagen

6

- Sei $\pi = (S_1, S_2, \dots, S_n)$ eine Folge von Aussagen für die gilt:
 - S_1 ist axiomatisch definierte wahre Aussage
 - S_{i+1} folgt aus S_i durch axiomatisch beschriebene Ableitungsregeln
- Sei π heißt **Beweis von** S_n .
- Anforderungen an ein *Beweissystem*:

Beweisbare Aussagen

- Sei $\pi = (S_1, S_2, \dots, S_n)$ eine Folge von Aussagen für die gilt:
 - S_1 ist axiomatisch definierte wahre Aussage
 - S_{i+1} folgt aus S_i durch axiomatisch beschriebene Ableitungsregeln
- Sei π heißt **Beweis von** S_n .
- Anforderungen an ein *Beweissystem*:
 - ① $\{\langle \phi, \pi \rangle \mid \pi \text{ ist Beweis von } \phi\}$ ist entscheidbar

Beweisbare Aussagen

- Sei $\pi = (S_1, S_2, \dots, S_n)$ eine Folge von Aussagen für die gilt:
 - S_1 ist axiomatisch definierte wahre Aussage
 - S_{i+1} folgt aus S_i durch axiomatisch beschriebene Ableitungsregeln
- Sei π heißt **Beweis von** S_n .
- Anforderungen an ein *Beweissystem*:
 - ① $\{\langle \phi, \pi \rangle \mid \pi \text{ ist Beweis von } \phi\}$ ist entscheidbar
 - ② $\forall \phi [\exists \text{ Beweis } \pi \text{ für } \phi] \longrightarrow \phi \in \text{Th}(\cdot)$
(wenn es einen Beweis gibt, ist ϕ in der betrachteten Theorie wahr)

- Sei $\pi = (S_1, S_2, \dots, S_n)$ eine Folge von Aussagen für die gilt:
 - S_1 ist axiomatisch definierte wahre Aussage
 - S_{i+1} folgt aus S_i durch axiomatisch beschriebene Ableitungsregeln
- Sei π heißt **Beweis von** S_n .
- Anforderungen an ein *Beweissystem*:
 - ① $\{\langle \phi, \pi \rangle \mid \pi \text{ ist Beweis von } \phi\}$ ist entscheidbar
 - ② $\forall \phi [\exists \text{ Beweis } \pi \text{ für } \phi] \longrightarrow \phi \in \text{Th}(\cdot)$
(wenn es einen Beweis gibt, ist ϕ in der betrachteten Theorie wahr)
- Ein Beweissystem, welches ② erfüllt heißt **korrekt**.

Beweisbare Aussagen

- Sei $\pi = (S_1, S_2, \dots, S_n)$ eine Folge von Aussagen für die gilt:
 - S_1 ist axiomatisch definierte wahre Aussage
 - S_{i+1} folgt aus S_i durch axiomatisch beschriebene Ableitungsregeln
- Sei π heißt **Beweis von** S_n .
- Anforderungen an ein *Beweissystem*:
 - ① $\{\langle \phi, \pi \rangle \mid \pi \text{ ist Beweis von } \phi\}$ ist entscheidbar
 - ② $\forall \phi [\exists \text{ Beweis } \pi \text{ für } \phi] \longrightarrow \phi \in \text{Th}(\cdot)$
(wenn es einen Beweis gibt, ist ϕ in der betrachteten Theorie wahr)
- Ein Beweissystem, welches ② erfüllt heißt **korrekt**.
- Notation: **Pr(Th(X))** bezeichnet die Menge aller beweisbaren Aussagen von $\text{Th}(X)$ im betrachteten Beweissystem

Sätze über Beweisbare Aussagen

7

Sätze über Beweisbare Aussagen

Satz 34

$$\text{Pr}(\text{Th}(\mathbf{N}, +, \cdot)) \in \mathbb{A}$$

Sätze über Beweisbare Aussagen

Satz 34

$$\text{Pr}(\text{Th}(\mathbf{N}, +, \cdot)) \in \mathbb{A}$$

Satz 35

$$\text{Pr}(\text{Th}(\mathbf{N}, +, \cdot)) \subsetneq \text{Th}(\mathbf{N}, +, \cdot)$$

Sätze über Beweisbare Aussagen

Satz 34

$$\text{Pr}(\text{Th}(\mathbf{N}, +, \cdot)) \in \mathbb{A}$$

Satz 35

$$\text{Pr}(\text{Th}(\mathbf{N}, +, \cdot)) \subsetneq \text{Th}(\mathbf{N}, +, \cdot)$$

Beweisidee

- Beweis durch Widerspruch!

Sätze über Beweisbare Aussagen

Satz 34

$$\text{Pr}(\text{Th}(\mathbf{N}, +, \cdot)) \in \mathbb{A}$$

Satz 35

$$\text{Pr}(\text{Th}(\mathbf{N}, +, \cdot)) \subsetneq \text{Th}(\mathbf{N}, +, \cdot)$$

Beweisidee

- Beweis durch Widerspruch!
- wir nehmen an, dass $\text{Pr}(\text{Th}(\mathbf{N}, +, \cdot)) = \text{Th}(\mathbf{N}, +, \cdot)$

Sätze über Beweisbare Aussagen

Satz 34

$$\text{Pr}(\text{Th}(\mathbf{N}, +, \cdot)) \in \mathbb{A}$$

Satz 35

$$\text{Pr}(\text{Th}(\mathbf{N}, +, \cdot)) \subsetneq \text{Th}(\mathbf{N}, +, \cdot)$$

Beweisidee

- Beweis durch Widerspruch!
- wir nehmen an, dass $\text{Pr}(\text{Th}(\mathbf{N}, +, \cdot)) = \text{Th}(\mathbf{N}, +, \cdot)$
- Konstruiere mittels Aufzählers B für $\text{Pr}(\text{Th}(\mathbf{N}, +, \cdot))$ einen Entscheider D für $\text{Th}(\mathbf{N}, +, \cdot)$

Sätze über Beweisbare Aussagen

Satz 34

$$\text{Pr}(\text{Th}(\mathbf{N}, +, \cdot)) \in \mathbb{A}$$

Satz 35

$$\text{Pr}(\text{Th}(\mathbf{N}, +, \cdot)) \subsetneq \text{Th}(\mathbf{N}, +, \cdot)$$

Beweisidee

- Beweis durch Widerspruch!
- wir nehmen an, dass $\text{Pr}(\text{Th}(\mathbf{N}, +, \cdot)) = \text{Th}(\mathbf{N}, +, \cdot)$
- Konstruiere mittels Aufzählers B für $\text{Pr}(\text{Th}(\mathbf{N}, +, \cdot))$ einen Entscheider D für $\text{Th}(\mathbf{N}, +, \cdot)$
- Nach Satz 33 ist $\text{Th}(\mathbf{N}, +, \cdot)$ aber nicht entscheidbar.

Ein nicht beweisbarer wahrer Satz

Ein nicht beweisbarer wahrer Satz

- Sei die Turingmaschine S wie folgt konstruiert

Ein nicht beweisbarer wahrer Satz

- Sei die Turingmaschine S wie folgt konstruiert

$S(z)$

1. Konstruiere $\langle S \rangle$
2. $\psi = \neg \exists y [\phi_{\langle S, \varepsilon \rangle}]$
3. Simuliere $B(\psi)$ und akzeptiere, gdw, Simulation akzeptiert

- $\phi_{\langle S, \varepsilon \rangle}$ wie im Beweis von Satz 33 definiert
(Formel, die prüft, ob y Berechnungspfad für $S(\varepsilon)$)

Ein nicht beweisbarer wahrer Satz

- Sei die Turingmaschine S wie folgt konstruiert

$S(z)$

1. Konstruiere $\langle S \rangle$
2. $\psi = \neg \exists y [\phi_{\langle S, \varepsilon \rangle}]$
3. Simuliere $B(\psi)$ und akzeptiere, gdw, Simulation akzeptiert

- $\phi_{\langle S, \varepsilon \rangle}$ wie im Beweis von Satz 33 definiert
(Formel, die prüft, ob y Berechnungspfad für $S(\varepsilon)$)
- B ist die TM, die alle beweisbaren Aussagen erkennt

Ein nicht beweisbarer wahrer Satz

- Sei die Turingmaschine S wie folgt konstruiert

$S(z)$

1. Konstruiere $\langle S \rangle$
2. $\psi = \neg \exists y [\phi_{\langle S, \varepsilon \rangle}]$
3. Simuliere $B(\psi)$ und akzeptiere, gdw, Simulation akzeptiert

- $\phi_{\langle S, \varepsilon \rangle}$ wie im Beweis von Satz 33 definiert
(Formel, die prüft, ob y Berechnungspfad für $S(\varepsilon)$)
- B ist die TM, die alle beweisbaren Aussagen erkennt
- Das heißt ψ ist wahr $\iff S(\varepsilon)$ akzeptiert nicht

Ein nicht beweisbarer wahrer Satz

- Sei die Turingmaschine S wie folgt konstruiert

$S(z)$

1. Konstruiere $\langle S \rangle$
2. $\psi = \neg \exists y [\phi_{\langle S, \varepsilon \rangle}]$
3. Simuliere $B(\psi)$ und akzeptiere, gdw, Simulation akzeptiert

- $\phi_{\langle S, \varepsilon \rangle}$ wie im Beweis von Satz 33 definiert
(Formel, die prüft, ob y Berechnungspfad für $S(\varepsilon)$)
- B ist die TM, die alle beweisbaren Aussagen erkennt
- Das heißt ψ ist wahr $\iff S(\varepsilon)$ akzeptiert nicht

Satz 36

Die Aussage $\psi = \neg \exists y [\phi_{\langle S, \varepsilon \rangle}]$ ist wahr aber nicht beweisbar in der Erweiterung von $\text{Th}(\mathbf{N}, +, \cdot)$.

Beweis Satz 36

- Zur Erinnerung:

$\psi = \neg \exists y[\phi_{\langle S, \varepsilon \rangle}]$ ist wahr $\iff S(\varepsilon)$ akzeptiert nicht

$S(z)$

1. Konstruiere $\langle S \rangle$
2. $\psi = \neg \exists y[\phi_{\langle S, \varepsilon \rangle}]$
3. Simuliere $B(\psi)$ und akzeptiere, gdw, Simulation akzeptiert

Beweis Satz 36

- Zur Erinnerung:

$\psi = \neg\exists y[\phi_{\langle S, \varepsilon \rangle}]$ ist wahr $\iff S(\varepsilon)$ akzeptiert nicht

$S(z)$

1. Konstruiere $\langle S \rangle$
2. $\psi = \neg\exists y[\phi_{\langle S, \varepsilon \rangle}]$
3. Simuliere $B(\psi)$ und akzeptiere, gdw, Simulation akzeptiert

- Angenommen B findet Beweis für ψ :
 $\Rightarrow S$ akzeptiert alles, also auch ε

Beweis Satz 36

- Zur Erinnerung:

$\psi = \neg \exists y[\phi_{\langle S, \varepsilon \rangle}]$ ist wahr $\iff S(\varepsilon)$ akzeptiert nicht

$S(z)$

1. Konstruiere $\langle S \rangle$
2. $\psi = \neg \exists y[\phi_{\langle S, \varepsilon \rangle}]$
3. Simuliere $B(\psi)$ und akzeptiere, gdw, Simulation akzeptiert

- Angenommen B findet Beweis für ψ :
 - $\Rightarrow S$ akzeptiert alles, also auch ε
 - $\Rightarrow \psi$ ist nicht wahr

Beweis Satz 36

- Zur Erinnerung:

$\psi = \neg \exists y[\phi_{\langle S, \varepsilon \rangle}]$ ist wahr $\iff S(\varepsilon)$ akzeptiert nicht

$S(z)$

1. Konstruiere $\langle S \rangle$
2. $\psi = \neg \exists y[\phi_{\langle S, \varepsilon \rangle}]$
3. Simuliere $B(\psi)$ und akzeptiere, gdw, Simulation akzeptiert

- Angenommen B findet Beweis für ψ :

$\Rightarrow S$ akzeptiert alles, also auch ε

$\Rightarrow \psi$ ist nicht wahr

\Rightarrow es gibt keinen Beweis für $\psi \rightarrow$ Widerspruch

Beweis Satz 36

- Zur Erinnerung:

$\psi = \neg\exists y[\phi_{\langle S, \varepsilon \rangle}]$ ist wahr $\iff S(\varepsilon)$ akzeptiert nicht

$S(z)$

1. Konstruiere $\langle S \rangle$
2. $\psi = \neg\exists y[\phi_{\langle S, \varepsilon \rangle}]$
3. Simuliere $B(\psi)$ und akzeptiere, gdw, Simulation akzeptiert

- Angenommen B findet Beweis für ψ :
 - $\Rightarrow S$ akzeptiert alles, also auch ε
 - $\Rightarrow \psi$ ist nicht wahr
 - \Rightarrow es gibt keinen Beweis für $\psi \rightarrow$ Widerspruch
- Demnach findet B keinen Beweis:

Beweis Satz 36

- Zur Erinnerung:

$\psi = \neg \exists y[\phi_{\langle S, \varepsilon \rangle}]$ ist wahr $\iff S(\varepsilon)$ akzeptiert nicht

$S(z)$

1. Konstruiere $\langle S \rangle$
2. $\psi = \neg \exists y[\phi_{\langle S, \varepsilon \rangle}]$
3. Simuliere $B(\psi)$ und akzeptiere, gdw, Simulation akzeptiert

- Angenommen B findet Beweis für ψ :

$\Rightarrow S$ akzeptiert alles, also auch ε

$\Rightarrow \psi$ ist nicht wahr

\Rightarrow es gibt keinen Beweis für $\psi \rightarrow$ Widerspruch

- Demnach findet B keinen Beweis:

$\Rightarrow S$ stoppt nie, daraus folgt $S(\varepsilon) \uparrow$

Beweis Satz 36

- Zur Erinnerung:

$\psi = \neg \exists y[\phi_{\langle S, \varepsilon \rangle}]$ ist wahr $\iff S(\varepsilon)$ akzeptiert nicht

$S(z)$

1. Konstruiere $\langle S \rangle$
2. $\psi = \neg \exists y[\phi_{\langle S, \varepsilon \rangle}]$
3. Simuliere $B(\psi)$ und akzeptiere, gdw, Simulation akzeptiert

- Angenommen B findet Beweis für ψ :

$\Rightarrow S$ akzeptiert alles, also auch ε

$\Rightarrow \psi$ ist nicht wahr

\Rightarrow es gibt keinen Beweis für $\psi \rightarrow$ Widerspruch

- Demnach findet B keinen Beweis:

$\Rightarrow S$ stoppt nie, daraus folgt $S(\varepsilon) \uparrow$

$\Rightarrow S(\varepsilon)$ akzeptiert nicht, und demnach ist ψ eine wahre Aussage

□

- Ist es nicht widersprüchlich, das wir beweisen konnten, dass eine Aussage "wahr aber nicht beweisbar" ist?

- Ist es nicht widersprüchlich, das wir beweisen konnten, dass eine Aussage "wahr aber nicht beweisbar" ist?
- Beweis der Nichtbeweisbarkeit innerhalb der Erweiterung von $\text{Th}(\mathbf{N}, +, \cdot)$ wurde innerhalb eines anderen Systems erbracht

- Ist es nicht widersprüchlich, das wir beweisen konnten, dass eine Aussage "wahr aber nicht beweisbar" ist?
- Beweis der Nichtbeweisbarkeit innerhalb der Erweiterung von $\text{Th}(\mathbf{N}, +, \cdot)$ wurde innerhalb eines anderen Systems erbracht

Formulierung als Unvollständigkeitssatz

Nachbetrachtung zum Satz 36

- Ist es nicht widersprüchlich, das wir beweisen konnten, dass eine Aussage "wahr aber nicht beweisbar" ist?
- Beweis der Nichtbeweisbarkeit innerhalb der Erweiterung von $\text{Th}(\mathbf{N}, +, \cdot)$ wurde innerhalb eines anderen Systems erbracht

Formulierung als Unvollständigkeitssatz

- System X heißt **konsistent**, gdw. $\neg \exists \phi [\phi \in X \wedge \neg \phi \in X]$

Nachbetrachtung zum Satz 36

- Ist es nicht widersprüchlich, das wir beweisen konnten, dass eine Aussage "wahr aber nicht beweisbar" ist?

→ Beweis der Nichtbeweisbarkeit innerhalb der Erweiterung von $\text{Th}(\mathbf{N}, +, \cdot)$ wurde innerhalb eines anderen Systems erbracht

Formulierung als Unvollständigkeitssatz

- System X heißt **konsistent**, gdw. $\neg \exists \phi [\phi \in X \wedge \neg \phi \in X]$
- System X heißt **vollständig**, gdw. $\forall \phi [\phi \in X \vee \neg \phi \in X]$

Nachbetrachtung zum Satz 36

- Ist es nicht widersprüchlich, das wir beweisen konnten, dass eine Aussage "wahr aber nicht beweisbar" ist?

→ Beweis der Nichtbeweisbarkeit innerhalb der Erweiterung von $\text{Th}(\mathbf{N}, +, \cdot)$ wurde innerhalb eines anderen Systems erbracht

Formulierung als Unvollständigkeitssatz

- System X heißt **konsistent**, gdw. $\neg \exists \phi [\phi \in X \wedge \neg \phi \in X]$
- System X heißt **vollständig**, gdw. $\forall \phi [\phi \in X \vee \neg \phi \in X]$
- **vollständig und konsistent**:
 $\forall \phi [(\phi \in X \vee \neg \phi \in X) \wedge (\phi \notin X \vee \neg \phi \notin X)]$

Nachbetrachtung zum Satz 36

- Ist es nicht widersprüchlich, das wir beweisen konnten, dass eine Aussage "wahr aber nicht beweisbar" ist?

→ Beweis der Nichtbeweisbarkeit innerhalb der Erweiterung von $\text{Th}(\mathbf{N}, +, \cdot)$ wurde innerhalb eines anderen Systems erbracht

Formulierung als Unvollständigkeitssatz

- System X heißt **konsistent**, gdw. $\neg \exists \phi [\phi \in X \wedge \neg \phi \in X]$
- System X heißt **vollständig**, gdw. $\forall \phi [\phi \in X \vee \neg \phi \in X]$
- **vollständig und konsistent**:
 $\forall \phi [(\phi \in X \vee \neg \phi \in X) \wedge (\phi \notin X \vee \neg \phi \notin X)]$

Gödels 1. Unvollständigkeitssatz

Jedes axiomatische Beweissystem X , in welchem sich der Begriff des Beweis für Aussagen aus $\text{Th}(\mathbf{N}, +, \cdot)$ formalisieren lässt, ist entweder konsistent oder vollständig.

Nachbetrachtung zum Satz 36

- Ist es nicht widersprüchlich, das wir beweisen konnten, dass eine Aussage "wahr aber nicht beweisbar" ist?

→ Beweis der Nichtbeweisbarkeit innerhalb der Erweiterung von $\text{Th}(\mathbf{N}, +, \cdot)$ wurde innerhalb eines anderen Systems erbracht

Formulierung als Unvollständigkeitssatz

- System X heißt **konsistent**, gdw. $\neg \exists \phi [\phi \in X \wedge \neg \phi \in X]$
- System X heißt **vollständig**, gdw. $\forall \phi [\phi \in X \vee \neg \phi \in X]$
- **vollständig und konsistent**:
 $\forall \phi [(\phi \in X \vee \neg \phi \in X) \wedge (\phi \notin X \vee \neg \phi \notin X)]$

Gödels 1. Unvollständigkeitssatz

Jedes axiomatische Beweissystem X , in welchem sich der Begriff des Beweis für Aussagen aus $\text{Th}(\mathbf{N}, +, \cdot)$ formalisieren lässt, ist entweder konsistent oder vollständig.

- Satz 35: Es gibt wahre Aussagen in X , die nicht beweisbar sind.
 Unter Annahme $\forall \phi [\phi \in X \vee \phi \notin X]$ gilt $\exists \phi [\phi \notin X \wedge \neg \phi \notin X]$

Nachbetrachtung zum Satz 36

- Ist es nicht widersprüchlich, das wir beweisen konnten, dass eine Aussage "wahr aber nicht beweisbar" ist?

→ Beweis der Nichtbeweisbarkeit innerhalb der Erweiterung von $\text{Th}(\mathbf{N}, +, \cdot)$ wurde innerhalb eines anderen Systems erbracht

Formulierung als Unvollständigkeitssatz

- System X heißt **konsistent**, gdw. $\neg \exists \phi [\phi \in X \wedge \neg \phi \in X]$
- System X heißt **vollständig**, gdw. $\forall \phi [\phi \in X \vee \neg \phi \in X]$
- **vollständig und konsistent:**

$$\forall \phi [(\phi \in X \vee \neg \phi \in X) \wedge (\phi \notin X \vee \neg \phi \notin X)]$$

widerspricht



Gödels 1. Unvollständigkeitssatz

Jedes axiomatische Beweissystem X , in welchem sich der Begriff des Beweis für Aussagen aus $\text{Th}(\mathbf{N}, +, \cdot)$ formalisieren lässt, ist entweder konsistent oder vollständig.

- Satz 35: Es gibt wahre Aussagen in X , die nicht beweisbar sind.
Unter Annahme $\forall \phi [\phi \in X \vee \phi \notin X]$ gilt $\exists \phi [\phi \notin X \wedge \neg \phi \notin X]$

4. Kapitel

Komplexitätstheorie

- **Bislang:** Was kann ich im Modell X berechnen?

- **Bislang:** Was kann ich im Modell X berechnen?
- **Neue Frage:** Was kann ich im Modell X mit eingeschränkten Ressourcen berechnen?

- **Bislang:** Was kann ich im Modell X berechnen?
- **Neue Frage:** Was kann ich im Modell X mit eingeschränkten Ressourcen berechnen?
- **Ressourcen:** Zeit, Speicher, Energie, ...

- **Bislang:** Was kann ich im Modell X berechnen?
- **Neue Frage:** Was kann ich im Modell X mit eingeschränkten Ressourcen berechnen?
- **Ressourcen:** Zeit, Speicher, Energie, ...

- **Bislang:** Was kann ich im Modell X berechnen?
- **Neue Frage:** Was kann ich im Modell X mit eingeschränkten Ressourcen berechnen?
- **Ressourcen:** Zeit, Speicher, Energie, ...

- **Bislang:** Was kann ich im Modell X berechnen?
- **Neue Frage:** Was kann ich im Modell X mit eingeschränkten Ressourcen berechnen?
- **Ressourcen:** Zeit, Speicher, Energie, ...
- **Motivation:** Eventuell sind Probleme entscheidbar, aber die Berechnung ist nicht effizient durchführbar

- **Bislang:** Was kann ich im Modell X berechnen?
- **Neue Frage:** Was kann ich im Modell X mit eingeschränkten Ressourcen berechnen?
- **Ressourcen:** Zeit, Speicher, Energie, ...
- **Motivation:** Eventuell sind Probleme entscheidbar, aber die Berechnung ist nicht effizient durchführbar

Laufzeit einer Turingmaschine

- **Bislang:** Was kann ich im Modell X berechnen?
- **Neue Frage:** Was kann ich im Modell X mit eingeschränkten Ressourcen berechnen?
- **Ressourcen:** Zeit, Speicher, Energie, ...
- **Motivation:** Eventuell sind Probleme entscheidbar, aber die Berechnung ist nicht effizient durchführbar

Laufzeit einer Turingmaschine

- Laufzeit von M hängt von der Eingabe w ab

- **Bislang:** Was kann ich im Modell X berechnen?
- **Neue Frage:** Was kann ich im Modell X mit eingeschränkten Ressourcen berechnen?
- **Ressourcen:** Zeit, Speicher, Energie, ...
- **Motivation:** Eventuell sind Probleme entscheidbar, aber die Berechnung ist nicht effizient durchführbar

Laufzeit einer Turingmaschine

- Laufzeit von M hängt von der Eingabe w ab

$$T(w \in \Sigma^*) := \# \text{ Schritte für Lauf von } M(w)$$

- **Bislang:** Was kann ich im Modell X berechnen?
- **Neue Frage:** Was kann ich im Modell X mit eingeschränkten Ressourcen berechnen?
- **Ressourcen:** Zeit, Speicher, Energie, ...
- **Motivation:** Eventuell sind Probleme entscheidbar, aber die Berechnung ist nicht effizient durchführbar

Laufzeit einer Turingmaschine

- Laufzeit von M hängt von der Eingabe w ab

$$T(w \in \Sigma^*) := \# \text{ Schritte für Lauf von } M(w)$$

(det. 1-Band TM)

- **Bislang:** Was kann ich im Modell X berechnen?
- **Neue Frage:** Was kann ich im Modell X mit eingeschränkten Ressourcen berechnen?
- **Ressourcen:** Zeit, Speicher, Energie, ...
- **Motivation:** Eventuell sind Probleme entscheidbar, aber die Berechnung ist nicht effizient durchführbar

Laufzeit einer Turingmaschine

- Laufzeit von M hängt von der Eingabe w ab

$$T(w \in \Sigma^*) := \# \text{ Schritte für Lauf von } M(w)$$

(det. 1-Band TM)

- Laufzeit wird meist in Bezug zur Eingabelänge gemessen

- **Bislang:** Was kann ich im Modell X berechnen?
- **Neue Frage:** Was kann ich im Modell X mit eingeschränkten Ressourcen berechnen?
- **Ressourcen:** Zeit, Speicher, Energie, ...
- **Motivation:** Eventuell sind Probleme entscheidbar, aber die Berechnung ist nicht effizient durchführbar

Laufzeit einer Turingmaschine

- Laufzeit von M hängt von der Eingabe w ab

$$T(w \in \Sigma^*) := \# \text{ Schritte für Lauf von } M(w)$$

(det. 1-Band TM)

- Laufzeit wird meist in Bezug zur Eingabelänge gemessen

$$t(n \in \mathbf{N}) := \max_{w \in \Sigma^n} T(w)$$

$M(z)$

1. Prüfe ob z von der Form 0^*1^* ist
2. Streiche für jede 0 eine 1
3. Kontrolliere ob keine 1 übrig bleibt

$M(z)$

1. Prüfe ob z von der Form 0^*1^* ist
2. Streiche für jede 0 eine 1
3. Kontrolliere ob keine 1 übrig bleibt

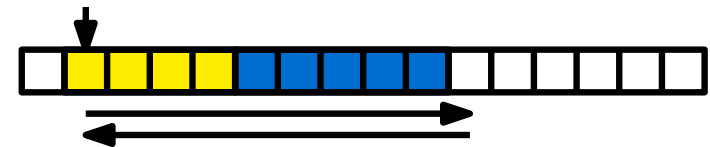
- Laufzeit für $w = 0^k 1^\ell$, $|w| = k + \ell = n$

$M(z)$

1. Prüfe ob z von der Form $0^* 1^*$ ist
2. Streiche für jede 0 eine 1
3. Kontrolliere ob keine 1 übrig bleibt

- Laufzeit für $w = 0^k 1^\ell$, $|w| = k + \ell = n$

1. $2n$ Schritte wenn ich 1. Zeichen markiere



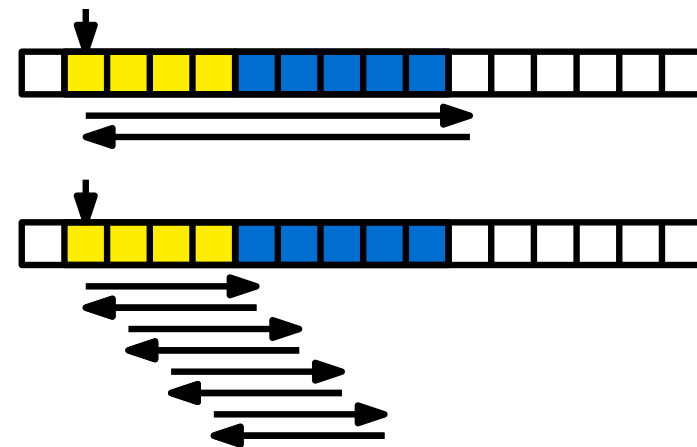
$M(z)$

1. Prüfe ob z von der Form $0^* 1^*$ ist
2. Streiche für jede 0 eine 1
3. Kontrolliere ob keine 1 übrig bleibt

■ Laufzeit für $w = 0^k 1^\ell$, $|w| = k + \ell = n$

1. $2n$ Schritte wenn ich 1. Zeichen markiere

2. $2k \min\{k, \ell\}$ Schritte



$M(z)$

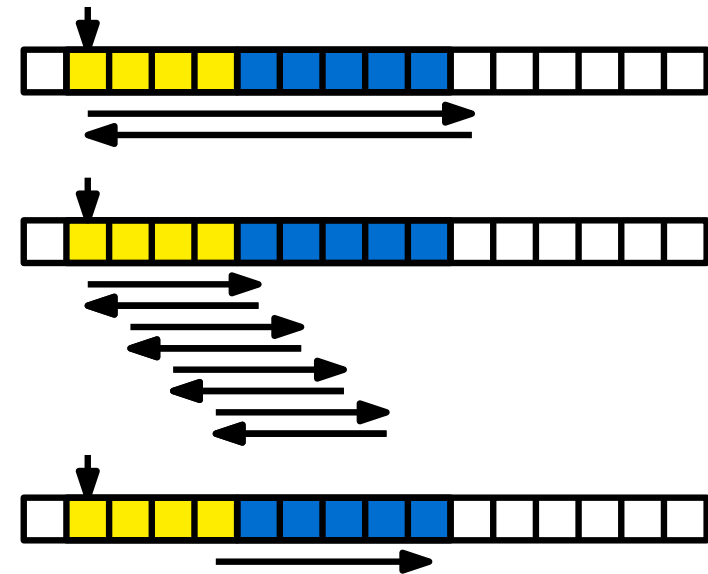
1. Prüfe ob z von der Form $0^* 1^*$ ist
2. Streiche für jede 0 eine 1
3. Kontrolliere ob keine 1 übrig bleibt

- Laufzeit für $w = 0^k 1^\ell$, $|w| = k + \ell = n$

1. $2n$ Schritte wenn ich 1. Zeichen markiere

2. $2k \min\{k, \ell\}$ Schritte

3. $k + 1$ Schritte bei $\ell \geq k$



$M(z)$

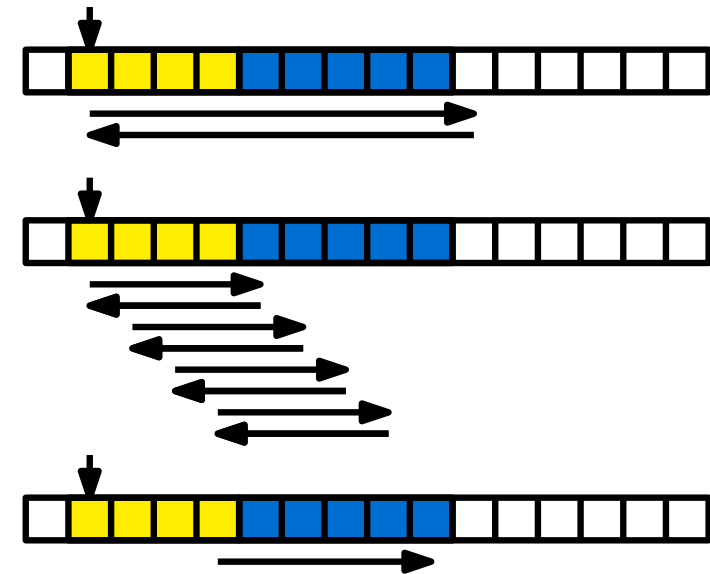
1. Prüfe ob z von der Form $0^* 1^*$ ist
2. Streiche für jede 0 eine 1
3. Kontrolliere ob keine 1 übrig bleibt

- Laufzeit für $w = 0^k 1^\ell$, $|w| = k + \ell = n$

1. $2n$ Schritte wenn ich 1. Zeichen markiere

2. $2k \min\{k, \ell\}$ Schritte

3. $k + 1$ Schritte bei $\ell \geq k$



- $T(w) \leq 2n + 2k \min\{k, l\} + k + 1$

$M(z)$

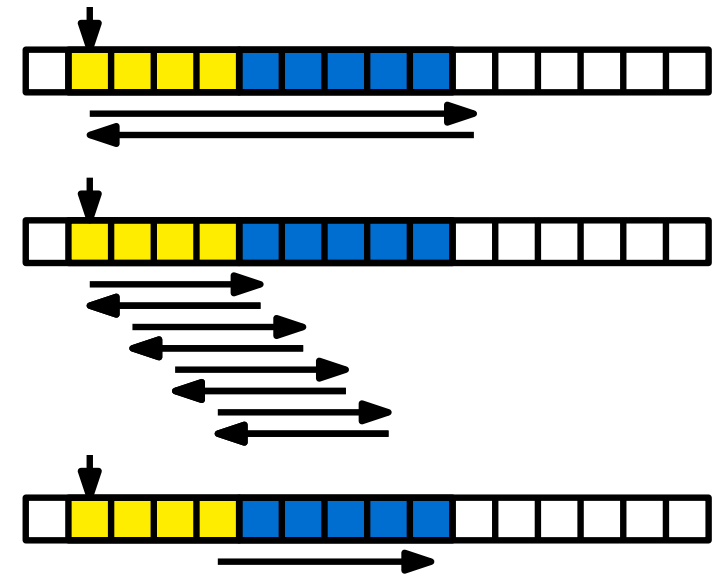
1. Prüfe ob z von der Form $0^* 1^*$ ist
2. Streiche für jede 0 eine 1
3. Kontrolliere ob keine 1 übrig bleibt

- Laufzeit für $w = 0^k 1^\ell$, $|w| = k + \ell = n$

1. $2n$ Schritte wenn ich 1. Zeichen markiere

2. $2k \min\{k, \ell\}$ Schritte

3. $k + 1$ Schritte bei $\ell \geq k$



- $T(w) \leq 2n + 2k \min\{k, \ell\} + k + 1 \leq 2n + n^2 + n/2 + 1$

$M(z)$

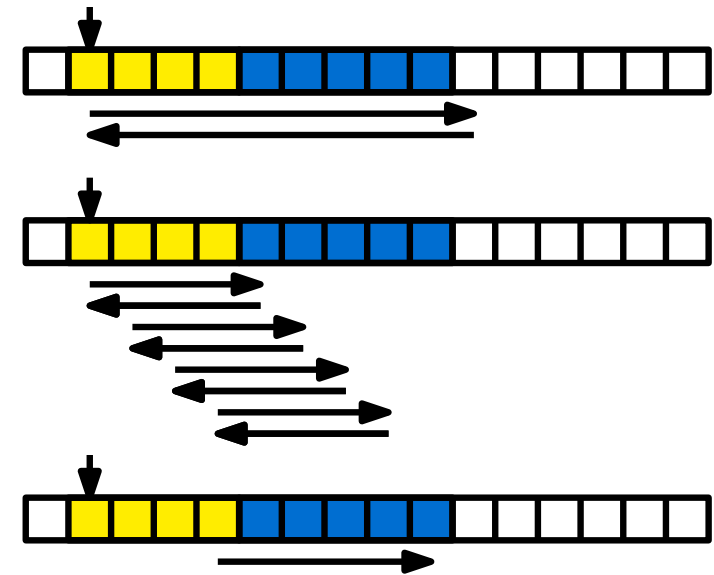
1. Prüfe ob z von der Form $0^* 1^*$ ist
2. Streiche für jede 0 eine 1
3. Kontrolliere ob keine 1 übrig bleibt

- Laufzeit für $w = 0^k 1^\ell$, $|w| = k + \ell = n$

1. $2n$ Schritte wenn ich 1. Zeichen markiere

2. $2k \min\{k, \ell\}$ Schritte

3. $k + 1$ Schritte bei $\ell \geq k$



- $T(w) \leq 2n + 2k \min\{k, \ell\} + k + 1 \leq 2n + n^2 + n/2 + 1$

- $t(n) = 5/2n + n^2/2 + 1$

- genaue Angabe der Laufzeit oft schwierig (und unnötig)
- wir greifen auf asymptotische Schranken zurück
($\rightarrow O, \Omega, \Theta$ -Notation)

- genaue Angabe der Laufzeit oft schwierig (und unnötig)
- wir greifen auf asymptotische Schranken zurück
($\rightarrow O, \Omega, \Theta$ -Notation)
- im Beispiel: $t(n) = \Theta(n^2)$

- genaue Angabe der Laufzeit oft schwierig (und unnötig)
- wir greifen auf asymptotische Schranken zurück
($\rightarrow O, \Omega, \Theta$ -Notation)
- im Beispiel: $t(n) = \Theta(n^2)$

Definition

Sei $f: \mathbb{N} \rightarrow \mathbb{N}$, dann bezeichnet

$$\text{TIME}(f(n)) = \{L \mid \exists \text{TM } M \text{ mit } t_M(n) = O(f(n))\}$$

die **Zeitkomplexitätsklasse** von $f(n)$.

- genaue Angabe der Laufzeit oft schwierig (und unnötig)
- wir greifen auf asymptotische Schranken zurück
($\rightarrow O, \Omega, \Theta$ -Notation)
- im Beispiel: $t(n) = \Theta(n^2)$

Definition

Sei $f: \mathbf{N} \rightarrow \mathbf{N}$, dann bezeichnet

$$\text{TIME}(f(n)) = \{L \mid \exists \text{TM } M \text{ mit } t_M(n) = O(f(n))\}$$

die **Zeitkomplexitätsklasse** von $f(n)$.

- Achtung: TIME ist Menge von Sprachen (Problemen), nicht von Turingmaschinen

- genaue Angabe der Laufzeit oft schwierig (und unnötig)
- wir greifen auf asymptotische Schranken zurück
($\rightarrow O, \Omega, \Theta$ -Notation)
- im Beispiel: $t(n) = \Theta(n^2)$

Definition

Sei $f: \mathbb{N} \rightarrow \mathbb{N}$, dann bezeichnet

$$\text{TIME}(f(n)) = \{L \mid \exists \text{TM } M \text{ mit } t_M(n) = O(f(n))\}$$

die **Zeitkomplexitätsklasse** von $f(n)$.

- Achtung: TIME ist Menge von Sprachen (Problemen), nicht von Turingmaschinen
- Bsp.: $\{0^k 1^k \mid k \geq 0\} \in \text{TIME}(n^2)$

- genaue Angabe der Laufzeit oft schwierig (und unnötig)
- wir greifen auf asymptotische Schranken zurück
($\rightarrow O, \Omega, \Theta$ -Notation)
- im Beispiel: $t(n) = \Theta(n^2)$

Definition

Sei $f: \mathbb{N} \rightarrow \mathbb{N}$, dann bezeichnet

$$\text{TIME}(f(n)) = \{L \mid \exists \text{TM } M \text{ mit } t_M(n) = O(f(n))\}$$

die **Zeitkomplexitätsklasse** von $f(n)$.

- Achtung: TIME ist Menge von Sprachen (Problemen), nicht von Turingmaschinen
- Bsp.: $\{0^k 1^k \mid k \geq 0\} \in \text{TIME}(n^2)$
- Es gilt sogar $\{0^k 1^k \mid k \geq 0\} \in \text{TIME}(n \log n)$

- genaue Angabe der Laufzeit oft schwierig (und unnötig)
- wir greifen auf asymptotische Schranken zurück
($\rightarrow O, \Omega, \Theta$ -Notation)
- im Beispiel: $t(n) = \Theta(n^2)$

Definition

Sei $f: \mathbb{N} \rightarrow \mathbb{N}$, dann bezeichnet

$$\text{TIME}(f(n)) = \{L \mid \exists \text{TM } M \text{ mit } t_M(n) = O(f(n))\}$$

die **Zeitkomplexitätsklasse** von $f(n)$.

- Achtung: TIME ist Menge von Sprachen (Problemen), nicht von Turingmaschinen
- Bsp.: $\{0^k 1^k \mid k \geq 0\} \in \text{TIME}(n^2)$
- Es gilt sogar $\{0^k 1^k \mid k \geq 0\} \in \text{TIME}(n \log n)$
 - Teste ob $\#0 = k$ und $\#1 = \ell$ identisch

- genaue Angabe der Laufzeit oft schwierig (und unnötig)
- wir greifen auf asymptotische Schranken zurück
($\rightarrow O, \Omega, \Theta$ -Notation)
- im Beispiel: $t(n) = \Theta(n^2)$

Definition

Sei $f: \mathbb{N} \rightarrow \mathbb{N}$, dann bezeichnet

$$\text{TIME}(f(n)) = \{L \mid \exists \text{TM } M \text{ mit } t_M(n) = O(f(n))\}$$

die **Zeitkomplexitätsklasse** von $f(n)$.

- Achtung: TIME ist Menge von Sprachen (Problemen), nicht von Turingmaschinen
- Bsp.: $\{0^k 1^k \mid k \geq 0\} \in \text{TIME}(n^2)$
- Es gilt sogar $\{0^k 1^k \mid k \geq 0\} \in \text{TIME}(n \log n)$
 - Teste ob $\#0 = k$ und $\#1 = \ell$ identisch
 - $k = \ell \iff \forall t \leq \log(n): k \equiv \ell \pmod{2^t}$

- genaue Angabe der Laufzeit oft schwierig (und unnötig)
- wir greifen auf asymptotische Schranken zurück
($\rightarrow O, \Omega, \Theta$ -Notation)
- im Beispiel: $t(n) = \Theta(n^2)$

Definition

Sei $f: \mathbb{N} \rightarrow \mathbb{N}$, dann bezeichnet

$$\text{TIME}(f(n)) = \{L \mid \exists \text{TM } M \text{ mit } t_M(n) = O(f(n))\}$$

die **Zeitkomplexitätsklasse** von $f(n)$.

- Achtung: TIME ist Menge von Sprachen (Problemen), nicht von Turingmaschinen
- Bsp.: $\{0^k 1^k \mid k \geq 0\} \in \text{TIME}(n^2)$
- Es gilt sogar $\{0^k 1^k \mid k \geq 0\} \in \text{TIME}(n \log n)$
 - Teste ob $\#0 = k$ und $\#1 = \ell$ identisch
 - $k = \ell \iff \forall t \leq \log(n): k \equiv \ell \pmod{2^t}$
 - bitweises Vergleichen von k und ℓ

- genaue Angabe der Laufzeit oft schwierig (und unnötig)
- wir greifen auf asymptotische Schranken zurück
($\rightarrow O, \Omega, \Theta$ -Notation)
- im Beispiel: $t(n) = \Theta(n^2)$

Definition

Sei $f: \mathbb{N} \rightarrow \mathbb{N}$, dann bezeichnet

$$\text{TIME}(f(n)) = \{L \mid \exists \text{TM } M \text{ mit } t_M(n) = O(f(n))\}$$

die **Zeitkomplexitätsklasse** von $f(n)$.

- Achtung: TIME ist Menge von Sprachen (Problemen), nicht von Turingmaschinen
 - Bsp.: $\{0^k 1^k \mid k \geq 0\} \in \text{TIME}(n^2)$
 - Es gilt sogar $\{0^k 1^k \mid k \geq 0\} \in \text{TIME}(n \log n)$
 - Teste ob $\#0 = k$ und $\#1 = \ell$ identisch
 - $k = \ell \iff \forall t \leq \log(n): k \equiv \ell \pmod{2^t}$
 - bitweises Vergleichen von k und ℓ
- } $\log n$ Bit-Tests
 $O(n)$ Zeit pro Bit

Definition

Die Komplexitätsklasse P bezeichnet alle Probleme, die in Polynomialzeit von einer deterministischen TM entschieden werden können. Das heißt:

$$P := \bigcup_{k=1}^{\infty} \text{TIME}(n^k)$$

Definition

Die Komplexitätsklasse P bezeichnet alle Probleme, die in Polynomialzeit von einer deterministischen TM entschieden werden können. Das heißt:

$$P := \bigcup_{k=1}^{\infty} \text{TIME}(n^k)$$

- Bsp.: $\{0^k 1^k \mid k \geq 0\} \in P$

Definition

Die Komplexitätsklasse P bezeichnet alle Probleme, die in Polynomialzeit von einer deterministischen TM entschieden werden können. Das heißt:

$$P := \bigcup_{k=1}^{\infty} \text{TIME}(n^k)$$

- Bsp.: $\{0^k 1^k \mid k \geq 0\} \in P$
- **Anmerkung** Simulation eines TM-äquivalenten Berechnungsmodells durch ein anderes kostet i.A. einen polynomiellen Faktor

Definition

Die Komplexitätsklasse P bezeichnet alle Probleme, die in Polynomialzeit von einer deterministischen TM entschieden werden können. Das heißt:

$$P := \bigcup_{k=1}^{\infty} \text{TIME}(n^k)$$

- Bsp.: $\{0^k 1^k \mid k \geq 0\} \in P$
- **Anmerkung** Simulation eines TM-äquivalenten Berechnungsmodells durch ein anderes kostet i.A. einen polynomiellen Faktor
→ Beschreibung von P relativ unabhängig vom Berechnungsmodell

Definition

Die Komplexitätsklasse P bezeichnet alle Probleme, die in Polynomialzeit von einer deterministischen TM entschieden werden können. Das heißt:

$$P := \bigcup_{k=1}^{\infty} \text{TIME}(n^k)$$

- Bsp.: $\{0^k 1^k \mid k \geq 0\} \in P$
- **Anmerkung** Simulation eines TM-äquivalenten Berechnungsmodells durch ein anderes kostet i.A. einen polynomiellen Faktor
→ Beschreibung von P relativ unabhängig vom Berechnungsmodell
- **Konvention:** Probleme aus P sind effizient lösbar

Definition

Die Komplexitätsklasse P bezeichnet alle Probleme, die in Polynomialzeit von einer deterministischen TM entschieden werden können. Das heißt:

$$P := \bigcup_{k=1}^{\infty} \text{TIME}(n^k)$$

- Bsp.: $\{0^k 1^k \mid k \geq 0\} \in P$
- **Anmerkung** Simulation eines TM-äquivalenten Berechnungsmodells durch ein anderes kostet i.A. einen polynomiellen Faktor
→ Beschreibung von P relativ unabhängig vom Berechnungsmodell
- **Konvention:** Probleme aus P sind effizient lösbar
- Probleme müssen sinnvoll kodiert sein

Definition

Die Komplexitätsklasse P bezeichnet alle Probleme, die in Polynomialzeit von einer deterministischen TM entschieden werden können. Das heißt:

$$P := \bigcup_{k=1}^{\infty} \text{TIME}(n^k)$$

- Bsp.: $\{0^k 1^k \mid k \geq 0\} \in P$
- **Anmerkung** Simulation eines TM-äquivalenten Berechnungsmodells durch ein anderes kostet i.A. einen polynomiellen Faktor
→ Beschreibung von P relativ unabhängig vom Berechnungsmodell
- **Konvention:** Probleme aus P sind effizient lösbar
- Probleme müssen sinnvoll kodiert sein
 - $\langle k \rangle = \text{bin}(k) \leftrightarrow \langle k \rangle = 1^k$

Definition

Die Komplexitätsklasse P bezeichnet alle Probleme, die in Polynomialzeit von einer deterministischen TM entschieden werden können. Das heißt:

$$P := \bigcup_{k=1}^{\infty} \text{TIME}(n^k)$$

- Bsp.: $\{0^k 1^k \mid k \geq 0\} \in P$
- **Anmerkung** Simulation eines TM-äquivalenten Berechnungsmodells durch ein anderes kostet i.A. einen polynomiellen Faktor
→ Beschreibung von P relativ unabhängig vom Berechnungsmodell
- **Konvention:** Probleme aus P sind effizient lösbar
- Probleme müssen sinnvoll kodiert sein
 - $\langle k \rangle = \text{bin}(k) \leftrightarrow \langle k \rangle = 1^k$
 - exponentieller Unterschied der Länge

Definition

Die Komplexitätsklasse P bezeichnet alle Probleme, die in Polynomialzeit von einer deterministischen TM entschieden werden können. Das heißt:

$$P := \bigcup_{k=1}^{\infty} \text{TIME}(n^k)$$

- Bsp.: $\{0^k 1^k \mid k \geq 0\} \in P$
- **Anmerkung** Simulation eines TM-äquivalenten Berechnungsmodells durch ein anderes kostet i.A. einen polynomiellen Faktor
→ Beschreibung von P relativ unabhängig vom Berechnungsmodell
- **Konvention:** Probleme aus P sind effizient lösbar
- Probleme müssen sinnvoll kodiert sein
 - $\langle k \rangle = \text{bin}(k) \leftrightarrow \langle k \rangle = 1^k$
 - exponentieller Unterschied der Länge
 - Laufzeit wird relativ zur Eingabelänge gemessen!

- NP beinhaltet die Probleme, deren Lösung sich leicht verifizieren lässt

- NP beinhaltet die Probleme, deren Lösung sich leicht verifizieren lässt
- **Def.:** Ein **Verifizierer für eine Sprache L** ist eine TM für die gilt:
 $L = \{w \mid \exists z \in \Sigma^* : \langle w, z \rangle \text{ wird von } V \text{ akzeptiert}\}$


- NP beinhaltet die Probleme, deren Lösung sich leicht verifizieren lässt
- **Def.:** Ein **Verifizierer für eine Sprache L** ist eine TM für die gilt:
 $L = \{w \mid \exists z \in \Sigma^* : \langle w, z \rangle \text{ wird von } V \text{ akzeptiert}\}$

 z heißt **Zeuge** oder **Zertifikat**


- NP beinhaltet die Probleme, deren Lösung sich leicht verifizieren lässt
- **Def.:** Ein **Verifizierer für eine Sprache L** ist eine TM für die gilt:
 $L = \{w \mid \exists z \in \Sigma^* : \langle w, z \rangle \text{ wird von } V \text{ akzeptiert}\}$

z heißt **Zeuge** oder **Zertifikat**

Laufzeit von V wird bzgl. $\langle w, z \rangle$ gemessen

- NP beinhaltet die Probleme, deren Lösung sich leicht verifizieren lässt
- **Def.:** Ein **Verifizierer für eine Sprache L** ist eine TM für die gilt:
 $L = \{w \mid \exists z \in \Sigma^* : \langle w, z \rangle \text{ wird von } V \text{ akzeptiert}\}$



z heißt **Zeuge** oder **Zertifikat** Laufzeit von V wird bzgl. $\langle w, z \rangle$ gemessen
- **Def.:** Ein Verifizierer V für L heißt **polynomiell**, gdw.
 - (1) $\exists k$, sodass $L = \{w \mid \exists z \in \Sigma^* : |z| \leq |w|^k \text{ und } \langle w, z \rangle \in L(V)\}$
 - (2) V hat polynomielle Laufzeit

- NP beinhaltet die Probleme, deren Lösung sich leicht verifizieren lässt
- **Def.:** Ein **Verifizierer für eine Sprache L** ist eine TM für die gilt:
 $L = \{w \mid \exists z \in \Sigma^* : \langle w, z \rangle \text{ wird von } V \text{ akzeptiert}\}$


z heißt **Zeuge** oder **Zertifikat** Laufzeit von V wird bzgl. $\langle w, z \rangle$ gemessen
- **Def.:** Ein Verifizierer V für L heißt **polynomiell**, gdw.
 - (1) $\exists k$, sodass $L = \{w \mid \exists z \in \Sigma^* : |z| \leq |w|^k \text{ und } \langle w, z \rangle \in L(V)\}$
 - (2) V hat polynomielle Laufzeit

Definition

Die Komplexitätsklasse NP umfasst alle Sprachen, für die es einen polynomiellen Verifizierer gibt.


- NP beinhaltet die Probleme, deren Lösung sich leicht verifizieren lässt
- **Def.:** Ein **Verifizierer für eine Sprache L** ist eine TM für die gilt:
 $L = \{w \mid \exists z \in \Sigma^* : \langle w, z \rangle \text{ wird von } V \text{ akzeptiert}\}$


z heißt **Zeuge** oder **Zertifikat** Laufzeit von V wird bzgl. $\langle w, z \rangle$ gemessen
- **Def.:** Ein Verifizierer V für L heißt **polynomiell**, gdw.
 - (1) $\exists k$, sodass $L = \{w \mid \exists z \in \Sigma^* : |z| \leq |w|^k \text{ und } \langle w, z \rangle \in L(V)\}$
 - (2) V hat polynomielle Laufzeit

Definition

Die Komplexitätsklasse NP umfasst alle Sprachen, für die es einen polynomiellen Verifizierer gibt.

- Wenn L aus P dann ist der polynomielle Entscheider für L auch ein polynomieller Verifizierer für L (er ignoriert den Zeugen)

- NP beinhaltet die Probleme, deren Lösung sich leicht verifizieren lässt
- **Def.:** Ein **Verifizierer für eine Sprache L** ist eine TM für die gilt:
 $L = \{w \mid \exists z \in \Sigma^* : \langle w, z \rangle \text{ wird von } V \text{ akzeptiert}\}$


z heißt **Zeuge** oder **Zertifikat** Laufzeit von V wird bzgl. $\langle w, z \rangle$ gemessen
- **Def.:** Ein Verifizierer V für L heißt **polynomiell**, gdw.
 - (1) $\exists k$, sodass $L = \{w \mid \exists z \in \Sigma^* : |z| \leq |w|^k \text{ und } \langle w, z \rangle \in L(V)\}$
 - (2) V hat polynomielle Laufzeit

Definition

Die Komplexitätsklasse NP umfasst alle Sprachen, für die es einen polynomiellen Verifizierer gibt.

- Wenn L aus P dann ist der polynomielle Entscheider für L auch ein polynomieller Verifizierer für L (er ignoriert den Zeugen)
- Es folgt: $P \subseteq NP$