

Einführung in die Theoretische Informatik

Christian Dalvit Manuel Eberl
Samuel Frontull **Cezary Kaliszyk** Daniel Ranalter

Sommersemester 2023

Theoretische Informatik

Sommersemester 2023

Organization: Lecture

- VO2, Recording in OLAT
- Slides available
<http://cl-informatik.uibk.ac.at/teaching/ss23/eti/>
- Lecture notes, Studia, optional
- Office Hours (CK): Wednesday 11:15 in 3M12 (please email to confirm)
- Online registration required
- Evaluation: Written exam. Closed book. Registration required.
- This year exceptionally in English (most materials in German)

Organization: SL

- Fridays, 3-4 groups
- Online registration required
- Exchanges between groups possible
- Attendance **very** highly recommended
- Assignments
 - Do the assignments and present them!
 - Announced Friday the week before
- Evaluation: Written exam. Closed book. In the “last slot”.

Begriffsdefinition

„Die Theoretische Informatik beschäftigt sich mit der Abstraktion, Modellbildung und grundlegenden Fragestellungen, die mit der Struktur, Verarbeitung, Übertragung und Wiedergabe von Informationen in Zusammenhang stehen.“

Begriffsdefinition

„Die Theoretische Informatik beschäftigt sich mit der Abstraktion, Modellbildung und grundlegenden Fragestellungen, die mit der Struktur, Verarbeitung, Übertragung und Wiedergabe von Informationen in Zusammenhang stehen.“

„Ihre Inhalte sind Automatentheorie, Theorie der formalen Sprachen, Berechenbarkeits- und Komplexitätstheorie, aber auch Logik und formale Semantik sowie die Informations-, Algorithmen- und Datenbanktheorie.“

<http://de.wikipedia.org/> 2021

Begriffsdefinition

„Die Theoretische Informatik beschäftigt sich mit der Abstraktion, Modellbildung und grundlegenden Fragestellungen, die mit der Struktur, Verarbeitung, Übertragung und Wiedergabe von Informationen in Zusammenhang stehen.“

*„Ihre Inhalte sind **Automatentheorie**, Theorie der formalen Sprachen, Berechenbarkeits- und Komplexitätstheorie, aber auch Logik und formale Semantik sowie die Informations-, Algorithmen- und Datenbanktheorie.“*

<http://de.wikipedia.org/> 2021

1 Automatentheorie

Begriffsdefinition

„Die Theoretische Informatik beschäftigt sich mit der Abstraktion, Modellbildung und grundlegenden Fragestellungen, die mit der Struktur, Verarbeitung, Übertragung und Wiedergabe von Informationen in Zusammenhang stehen.“

*„Ihre Inhalte sind Automatentheorie, **Theorie der formalen Sprachen**, Berechenbarkeits- und Komplexitätstheorie, aber auch Logik und formale Semantik sowie die Informations-, Algorithmen- und Datenbanktheorie.“*

<http://de.wikipedia.org/> 2021

- 1 Automatentheorie
- 2 Theorie der formalen Sprachen

Begriffsdefinition

„Die Theoretische Informatik beschäftigt sich mit der Abstraktion, Modellbildung und grundlegenden Fragestellungen, die mit der Struktur, Verarbeitung, Übertragung und Wiedergabe von Informationen in Zusammenhang stehen.“

*„Ihre Inhalte sind Automatentheorie, Theorie der formalen Sprachen, **Berechenbarkeits- und Komplexitätstheorie**, aber auch Logik und formale Semantik sowie die Informations-, Algorithmen- und Datenbanktheorie.“*

<http://de.wikipedia.org/> 2021

- 1 Automatentheorie
- 2 Theorie der formalen Sprachen
- 3 Berechenbarkeits- und Komplexitätstheorie

Begriffsdefinition

„Die Theoretische Informatik beschäftigt sich mit der Abstraktion, Modellbildung und grundlegenden Fragestellungen, die mit der Struktur, Verarbeitung, Übertragung und Wiedergabe von Informationen in Zusammenhang stehen.“

*„Ihre Inhalte sind Automatentheorie, Theorie der formalen Sprachen, Berechenbarkeits- und Komplexitätstheorie, aber auch **Logik und formale Semantik** sowie die Informations-, Algorithmen- und Datenbanktheorie.“*

<http://de.wikipedia.org/> 2021

- 1 Automatentheorie
- 2 Theorie der formalen Sprachen
- 3 Berechenbarkeits- und Komplexitätstheorie
- 4 Logik und formale Semantik

Begriffsdefinition

„Die Theoretische Informatik beschäftigt sich mit der Abstraktion, Modellbildung und grundlegenden Fragestellungen, die mit der Struktur, Verarbeitung, Übertragung und Wiedergabe von Informationen in Zusammenhang stehen.“

*„Ihre Inhalte sind Automatentheorie, Theorie der formalen Sprachen, Berechenbarkeits- und Komplexitätstheorie, aber auch Logik und formale Semantik sowie die **Informations-, Algorithmen- und Datenbanktheorie.**“*

<http://de.wikipedia.org/> 2021

- 1 Automatentheorie
- 2 Theorie der formalen Sprachen
- 3 Berechenbarkeits- und Komplexitätstheorie
- 4 Logik und formale Semantik
- 5 Informations-, Algorithmen- und Datenbanktheorie

Handbook of Theoretical Computer Science



+



= 2293 Seiten, 4 Kilogramm

Handbook of Theoretical Computer Science



+



= 2293 Seiten, 4 Kilogramm

Inhaltsverzeichnis Band „Algorithms and Complexity“

Machine models and simulations, A catalog of complexity classes, Machine-independent complexity theory, Kolmogorov complexity and its applications, Algorithms for finding patterns in strings, Data structures, Computational geometry, Algorithmic motion planning in robotics, Average-case analysis of algorithms and data structures, Graph algorithms, Cryptography, Algebraic complexity theory, Algorithms in number theory, The complexity of finite functions, Communication networks, VLSI theory, Parallel algorithms for shared-memory machines, General purpose parallel architectures

Handbook of Theoretical Computer Science



+



= 2293 Seiten, 4 Kilogramm

Inhaltsverzeichnis Band „Algorithms and Complexity“

Machine models and simulations, A catalog of complexity classes, Machine-independent complexity theory, Kolmogorov complexity and its applications, Algorithms for finding patterns in strings, Data structures, Computational geometry, Algorithmic motion planning in robotics, Average-case analysis of algorithms and data structures, Graph algorithms, Cryptography, Algebraic complexity theory, Algorithms in number theory, The complexity of finite functions, Communication networks, VLSI theory, Parallel algorithms for shared-memory machines, General purpose parallel architectures

Inhaltsverzeichnis Band „Formal Models and Semantics“

Finite automata, Context-free languages, Formal languages and power series, Automata on infinite objects, Graph rewriting: an algebraic and logic approach, Rewrite systems, Functional programming and lambda calculus, Type systems for programming languages, Recursive applicative program schemes, Logic programming, Denotational semantics, Semantic domains, Algebraic specification, Logics of programs, Methods and logics for proving programs, Temporal and modal logic, Elements of relational database theory, Distributed computing: models and methods, Operational and algebraic semantics of concurrent processes

Handbook of Theoretical Computer Science



+



= 2293 Seiten, 4 Kilogramm

Inhaltsverzeichnis Band „Algorithms and Complexity“

Machine models and simulations, A catalog of complexity classes, Machine-independent complexity theory, Kolmogorov complexity and its applications, Algorithms for finding patterns in strings, Data structures, Computational geometry, Algorithmic motion planning in robotics, Average-case analysis of algorithms and data structures, Graph algorithms, Cryptography, Algebraic complexity theory, Algorithms in number theory, The complexity of finite functions, Communication networks, VLSI theory, Parallel algorithms for shared-memory machines, General purpose parallel architectures

Inhaltsverzeichnis Band „Formal Models and Semantics“

Finite automata, Context-free languages, Formal languages and power series, Automata on infinite objects, Graph rewriting: an algebraic and logic approach, Rewrite systems, Functional programming and lambda calculus, Type systems for programming languages, Recursive applicative program schemes, Logic programming, Denotational semantics, Semantic domains, Algebraic specification, Logics of programs, Methods and logics for proving programs, Temporal and modal logic, Elements of relational database theory, Distributed computing: models and methods, Operational and algebraic semantics of concurrent processes

Handbook of Theoretical Computer Science



+



= 2293 Seiten, 4 Kilogramm

Inhaltsverzeichnis Band „Algorithms and Complexity“

Machine models and simulations, A catalog of complexity classes, Machine-independent complexity theory, Kolmogorov complexity and its applications, Algorithms for finding patterns in strings, Data structures, Computational geometry, Algorithmic motion planning in robotics, Average-case analysis of algorithms and data structures, Graph algorithms, Cryptography, Algebraic complexity theory, Algorithms in number theory, The complexity of finite functions, Communication networks, VLSI theory, Parallel algorithms for shared-memory machines, General purpose parallel architectures

Inhaltsverzeichnis Band „Formal Models and Semantics“

Finite automata, **Context-free languages**, **Formal languages** and power series, Automata on infinite objects, Graph rewriting: an algebraic and logic approach, Rewrite systems, Functional programming and lambda calculus, Type systems for programming languages, Recursive applicative program schemes, Logic programming, Denotational semantics, Semantic domains, Algebraic specification, Logics of programs, Methods and logics for proving programs, Temporal and modal logic, Elements of relational database theory, Distributed computing: models and methods, Operational and algebraic semantics of concurrent processes

Handbook of Theoretical Computer Science



+



= 2293 Seiten, 4 Kilogramm

Inhaltsverzeichnis Band „Algorithms and Complexity“

Machine models and simulations, A catalog of complexity classes, Machine-independent complexity theory, Kolmogorov complexity and its applications, Algorithms for finding patterns in strings, Data structures, Computational geometry, Algorithmic motion planning in robotics, Average-case analysis of algorithms and data structures, Graph algorithms, Cryptography, Algebraic complexity theory, Algorithms in number theory, The complexity of finite functions, Communication networks, VLSI theory, Parallel algorithms for shared-memory machines, General purpose parallel architectures

Inhaltsverzeichnis Band „Formal Models and Semantics“

Finite automata, Context-free languages, Formal languages and power series, Automata on infinite objects, Graph rewriting: an algebraic and logic approach, Rewrite systems, Functional programming and lambda calculus, Type systems for programming languages, Recursive applicative program schemes, Logic programming, Denotational semantics, Semantic domains, Algebraic specification, Logics of programs, Methods and logics for proving programs, Temporal and modal logic, Elements of relational database theory, Distributed computing: models and methods, Operational and algebraic semantics of concurrent processes

Handbook of Theoretical Computer Science



+



= 2293 Seiten, 4 Kilogramm

Inhaltsverzeichnis Band „Algorithms and Complexity“

Machine models and simulations, A catalog of complexity classes, Machine-independent complexity theory, Kolmogorov complexity and its applications, Algorithms for finding patterns in strings, Data structures, Computational geometry, Algorithmic motion planning in robotics, Average-case analysis of algorithms and data structures, Graph algorithms, Cryptography, Algebraic complexity theory, Algorithms in number theory, The complexity of finite functions, Communication networks, VLSI theory, Parallel algorithms for shared-memory machines, General purpose parallel architectures

Inhaltsverzeichnis Band „Formal Models and Semantics“

Finite automata, Context-free languages, Formal languages and power series, Automata on infinite objects, Graph rewriting: an algebraic and logic approach, Rewrite systems, Functional programming and lambda calculus, Type systems for programming languages, Recursive applicative program schemes, **Logic programming**, **Denotational semantics**, **Semantic domains**, **Algebraic specification**, **Logics of programs**, **Methods and logics for proving programs**, **Temporal and modal logic**, Elements of relational database theory, Distributed computing: models and methods, **Operational and algebraic semantics of concurrent processes**

Handbook of Theoretical Computer Science



+



= 2293 Seiten, 4 Kilogramm

Inhaltsverzeichnis Band „Algorithms and Complexity“

Machine models and simulations, A catalog of complexity classes, Machine-independent complexity theory, Kolmogorov complexity and its applications, **Algorithms for finding patterns in strings**, **Data structures**, Computational geometry, Algorithmic motion planning in robotics, Average-case analysis of algorithms and data structures, Graph algorithms, Cryptography, Algebraic complexity theory, **Algorithms in number theory**, The complexity of finite functions, Communication networks, VLSI theory, Parallel algorithms for shared-memory machines, General purpose parallel architectures

Inhaltsverzeichnis Band „Formal Models and Semantics“

Finite automata, Context-free languages, Formal languages and power series, Automata on infinite objects, Graph rewriting: an algebraic and logic approach, Rewrite systems, Functional programming and lambda calculus, Type systems for programming languages, Recursive applicative program schemes, Logic programming, Denotational semantics, Semantic domains, Algebraic specification, Logics of programs, Methods and logics for proving programs, Temporal and modal logic, **Elements of relational database theory**, Distributed computing: models and methods, Operational and algebraic semantics of concurrent processes

Handbook of Theoretical Computer Science



+



= 2293 Seiten, 4 Kilogramm

Inhaltsverzeichnis Band „Algorithms and Complexity“

Machine models and simulations, A catalog of complexity classes, Machine-independent complexity theory, Kolmogorov complexity and its applications, Algorithms for finding patterns in strings, Data structures, Computational geometry, Algorithmic motion planning in robotics, Average-case analysis of algorithms and data structures, Graph algorithms, Cryptography, Algebraic complexity theory, Algorithms in number theory, The complexity of finite functions, Communication networks, VLSI theory, Parallel algorithms for shared-memory machines, General purpose parallel architectures

Inhaltsverzeichnis Band „Formal Models and Semantics“

Finite automata, Context-free languages, Formal languages and power series, Automata on infinite objects, Graph rewriting: an algebraic and logic approach, Rewrite systems, Functional programming and lambda calculus, Type systems for programming languages, Recursive applicative program schemes, Logic programming, Denotational semantics, Semantic domains, Algebraic specification, Logics of programs, Methods and logics for proving programs, Temporal and modal logic, Elements of relational database theory, Distributed computing: models and methods, Operational and algebraic semantics of concurrent processes



Turing Maschinen,
Berechenbarkeitstheorie,
Alan Turing

—

1930er



Turing Maschinen,
Berechenbarkeitstheorie,
Alan Turing

—

1930er

Formale Sprachen,
Automatentheorie

Zuse Z3, ENIAC

1940er



Turing Maschinen,
Berechenbarkeitstheorie,
Alan Turing

—

1930er

Formale Sprachen,
Automatentheorie

Zuse Z3, ENIAC

1940er



Grammatiken, Grundlagen
des Compilerbaus,
Noam Chomsky

UNIVAC, Transistoren
statt Röhren

1950er



Turing Maschinen,
Berechenbarkeitstheorie,
Alan Turing

—

1930er

Formale Sprachen,
Automatentheorie

Zuse Z3, ENIAC

1940er



Grammatiken, Grundlagen
des Compilerbaus,
Noam Chomsky

UNIVAC, Transistoren
statt Röhren

1950er



P vs. NP
Komplexitätstheorie,
Stephen Cook

Minicomputer,
integrierte
Schaltkreise

1960er

Inhalte der Lehrveranstaltung

Einführung in die Logik

Syntax & Semantik der Aussagenlogik, Kalkül des natürlichen Schließens, Konjunktive und Disjunktive Normalformen

Inhalte der Lehrveranstaltung

Einführung in die Logik

Syntax & Semantik der Aussagenlogik, Kalkül des natürlichen Schließens, Konjunktive und Disjunktive Normalformen

Einführung in die Algebra

algebraische Strukturen, Boolesche Algebra

Inhalte der Lehrveranstaltung

Einführung in die Logik

Syntax & Semantik der Aussagenlogik, Kalkül des natürlichen Schließens, Konjunktive und Disjunktive Normalformen

Einführung in die Algebra

algebraische Strukturen, Boolesche Algebra

Einführung in die Theorie der Formalen Sprachen

Grammatiken und Formale Sprachen, Reguläre Sprachen, Kontextfreie Sprachen, Chomsky-Hierarchie, Anwendungen von formalen Sprachen

Inhalte der Lehrveranstaltung

Einführung in die Logik

Syntax & Semantik der Aussagenlogik, Kalkül des natürlichen Schließens, Konjunktive und Disjunktive Normalformen

Einführung in die Algebra

algebraische Strukturen, Boolesche Algebra

Einführung in die Theorie der Formalen Sprachen

Grammatiken und Formale Sprachen, Reguläre Sprachen, Kontextfreie Sprachen, Chomsky-Hierarchie, Anwendungen von formalen Sprachen

Einführung in die Berechenbarkeitstheorie und Komplexitätstheorie

Algorithmisch unlösbare Probleme, Turing Maschinen, Registermaschinen, Komplexitätstheorie

Inhalte der Lehrveranstaltung

Einführung in die Logik

Syntax & Semantik der Aussagenlogik, Kalkül des natürlichen Schließens, Konjunktive und Disjunktive Normalformen

Einführung in die Algebra

algebraische Strukturen, Boolesche Algebra

Einführung in die Theorie der Formalen Sprachen

Grammatiken und Formale Sprachen, Reguläre Sprachen, Kontextfreie Sprachen, Chomsky-Hierarchie, Anwendungen von formalen Sprachen

Einführung in die Berechenbarkeitstheorie und Komplexitätstheorie

Algorithmisch unlösbare Probleme, Turing Maschinen, Registermaschinen, Komplexitätstheorie

Einführung in die Programmverifikation

Prinzipien der Analyse von Programmen, Verifikation nach Hoare

Bachelor Informatik: die ersten zwei Jahre

1. Semester	Einführung in die Programmierung		Einf. Theoretische Informatik	Rechnerarchitektur
	Funktionale Programmierung		Lineare Algebra	
2. Semester	Programmiermethodik	Algorithmen und Datenstrukturen	Angewandte Mathematik	Betriebssysteme
3. Semester	Softwarearchitektur	Datenbanksysteme	Diskrete Strukturen	Rechnernetze und Internettechnik
		Daten und Wahrscheinlichkeiten		
4. Semester	Software Engineering	Maschinelles Lernen	Logik	Einführung in das wissenschaft. Arbeiten
	Parallele Programmierung			

Bachelor Informatik: die ersten zwei Jahre

1. Semester

Einführung in die
Programmierung

Funktionale
Programmierung

Einf. Theoretische
Informatik

Lineare Algebra

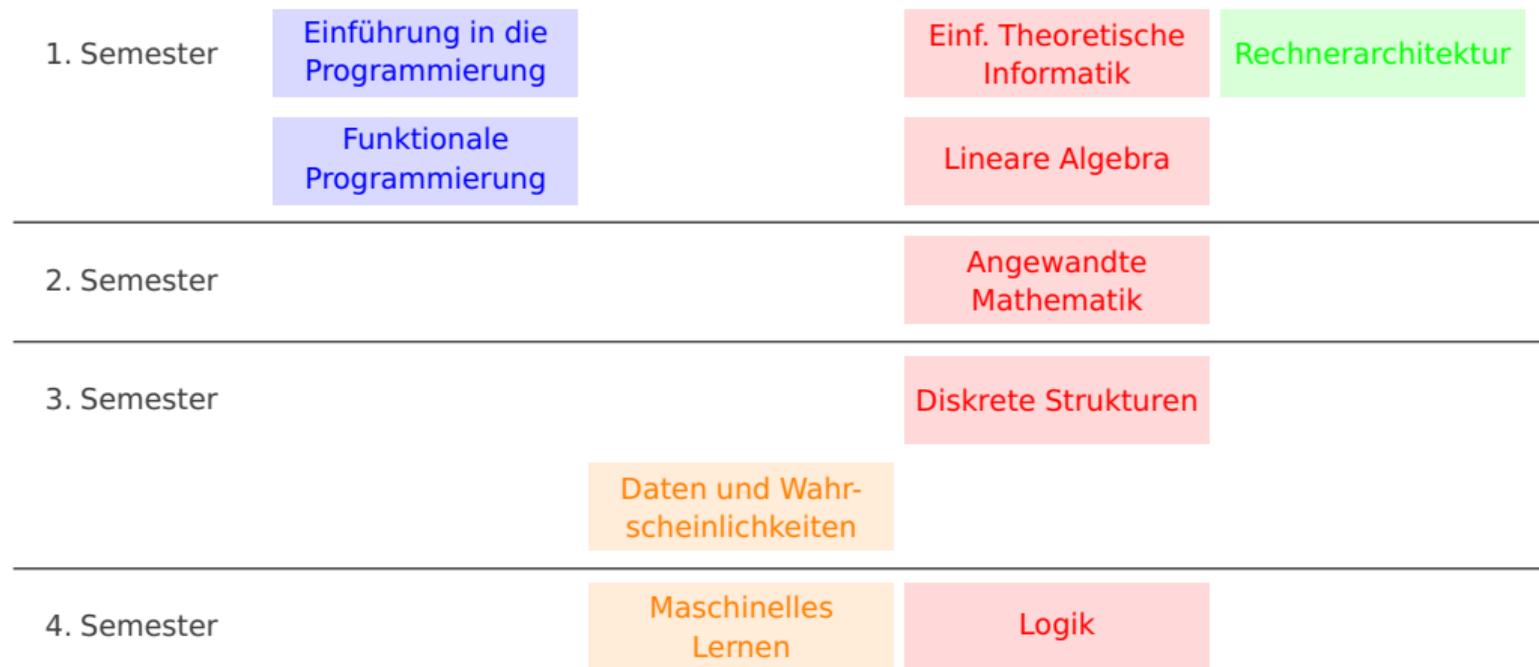
Rechnerarchitektur

2. Semester

3. Semester

4. Semester

Bachelor Informatik: die ersten zwei Jahre



Bachelor Informatik: die ersten zwei Jahre

1. Semester	Einführung in die Programmierung		Einf. Theoretische Informatik	Rechnerarchitektur
	Funktionale Programmierung		Lineare Algebra	
2. Semester			Angewandte Mathematik	
3. Semester	Softwarearchitektur	Datenbanksysteme	Diskrete Strukturen	
		Daten und Wahrscheinlichkeiten		
4. Semester	Software Engineering	Maschinelles Lernen	Logik	

Bachelor Informatik: die ersten zwei Jahre

1. Semester	Einführung in die Programmierung		Einf. Theoretische Informatik	Rechnerarchitektur
	Funktionale Programmierung		Lineare Algebra	
2. Semester		Algorithmen und Datenstrukturen	Angewandte Mathematik	
3. Semester	Softwarearchitektur	Datenbanksysteme	Diskrete Strukturen	Rechnernetze und Internettechnik
		Daten und Wahrscheinlichkeiten		
4. Semester	Software Engineering	Maschinelles Lernen	Logik	

Bachelor Informatik: die ersten zwei Jahre

1. Semester	Einführung in die Programmierung		Einf. Theoretische Informatik	Rechnerarchitektur
	Funktionale Programmierung		Lineare Algebra	
2. Semester		Algorithmen und Datenstrukturen	Angewandte Mathematik	
3. Semester	Softwarearchitektur	Datenbanksysteme	Diskrete Strukturen	Rechnernetze und Internettechnik
		Daten und Wahrscheinlichkeiten		
4. Semester	Software Engineering	Maschinelles Lernen	Logik	

...

Einführung in die Logik

Sommersemester 2023

Logisches Schließen im Allgemeinen

Beispiel

- We arrive at the following paradox in a globalised world: when nationalists pursue more formal sovereignty they achieve less real sovereignty of the people. They want to take back control and they end up with less control.
- That's what the UK will end up with. [...]

¹Siehe <https://blogs.lse.ac.uk/brexit/2017/10/06/the-catalan-crisis-and-brexit-stem-from-the-same-kind-of-nationalism/>.

Logisches Schließen im Allgemeinen

Beispiel

- We arrive at the following paradox in a globalised world: when nationalists pursue more formal sovereignty they achieve less real sovereignty of the people. They want to take back control and they end up with less control.
- That's what the UK will end up with. [...]
- Yet this paradox also has a corollary: when countries in Europe renounce formal sovereignty this leads to more real sovereignty for the people of Europe.

Paul De Grauwe, 2017¹

¹Siehe <https://blogs.lse.ac.uk/brexit/2017/10/06/the-catalan-crisis-and-brexit-stem-from-the-same-kind-of-nationalism/>.

Logisches Schließen im Allgemeinen

Beispiel

- We arrive at the following paradox in a globalised world: when nationalists pursue more formal sovereignty they achieve less real sovereignty of the people. They want to take back control and they end up with less control.
- That's what the UK will end up with. [...]
- Yet this paradox also has a corollary: when countries in Europe renounce formal sovereignty this leads to more real sovereignty for the people of Europe.

Paul De Grauwe, 2017¹

¹Siehe <https://blogs.lse.ac.uk/brexit/2017/10/06/the-catalan-crisis-and-brexit-stem-from-the-same-kind-of-nationalism/>.

Results of Brexit



Beispiel

*Yet this paradox also has a **corollary**: when countries in Europe renounce formal sovereignty this leads to more real sovereignty for the people of Europe.*

Beispiel

*Yet this paradox also has a **corollary**: when countries in Europe renounce formal sovereignty this leads to more real sovereignty for the people of Europe.*

Bemerkung

- In der Informatik und im Allgemeinen in den Natur- und Ingenieurwissenschaften muss ein „Korollar“ logisch folgen.
- Dazu haben wir in der Logik eine eigene **formale Sprache** für Folgerungen und allgemeiner **gültige** Schlüsse eingeführt.

Beispiel

*Yet this paradox also has a **corollary**: when countries in Europe renounce formal sovereignty this leads to more real sovereignty for the people of Europe.*

Bemerkung

- In der Informatik und im Allgemeinen in den Natur- und Ingenieurwissenschaften muss ein „Korollar“ logisch folgen.
- Dazu haben wir in der Logik eine eigene **formale Sprache** für Folgerungen und allgemeiner **gültige** Schlüsse eingeführt.

Beispiel (Fortsetzung)

In dieser Sprache, stellt sich das sogenannte „Korollar“, wie folgt dar :

$$\begin{aligned} & \text{„more formal sovereignty“} \rightarrow \text{„less real sovereignty“} \models \\ & \models \neg \text{„more formal sovereignty“} \rightarrow \neg \text{„less real sovereignty“} \end{aligned}$$

Beispiel

*Yet this paradox also has a **corollary**: when countries in Europe renounce formal sovereignty this leads to more real sovereignty for the people of Europe.*

Bemerkung

- In der Informatik und im Allgemeinen in den Natur- und Ingenieurwissenschaften muss ein „Korollar“ logisch folgen.
- Dazu haben wir in der Logik eine eigene **formale Sprache** für Folgerungen und allgemeiner **gültige** Schlüsse eingeführt.

Beispiel (Fortsetzung)

In dieser Sprache, stellt sich das sogenannte „Korollar“, wie folgt dar (und ist logisch falsch):

$$p \rightarrow q \not\equiv \neg p \rightarrow \neg q$$

Grundlagen der Logik

Sommersemester 2023

Aristoteles sagte

Topik I 1, 100a25-27

Eine Deduktion (syllogismos) ist also ein Argument, in welchem sich, wenn etwas gesetzt wurde, etwas anderes als das Gesetzte mit Notwendigkeit durch das Gesetzte ergibt



Beispiel

Sokrates ist ein Mensch

Alle Menschen sind sterblich

Somit ist Sokrates sterblich

Beispiel

Sokrates ist ein Mensch	}	Prämisse ①
Alle Menschen sind sterblich	}	Prämisse ②
Somit ist Sokrates sterblich	}	Konklusion

Beispiel

Sokrates ist ein Mensch	}	Prämisse ①
Alle Menschen sind sterblich	}	Prämisse ②
Somit ist Sokrates sterblich	}	Konklusion

Definition

- Schlussfiguren dieser Art heißen **Syllogismen**
- Syllogismen wurden bereits im antiken Griechenland untersucht, Grundlage der modernen Logik

Beispiel

Sokrates ist ein Mensch	}	Prämisse ①
Alle Menschen sind sterblich	}	Prämisse ②
Somit ist Sokrates sterblich	}	Konklusion

Definition

- Schlussfiguren dieser Art heißen **Syllogismen**
- Syllogismen wurden bereits im antiken Griechenland untersucht, Grundlage der modernen Logik

Fakt

*Nicht die Wahrheit der Prämissen, oder der Konklusion, sondern die Wahrheit der **Schlussfigur** ist entscheidend*

Beispiele für Arten von Syllogismen

Alle Griechen sind Menschen
Alle Menschen sind sterblich
Somit sind alle Griechen sterblich

Alle Professoren sind ernst
Einige Dozenten sind nicht ernst
Somit sind einige Dozenten keine Professoren

AOO - modus baroco

Beispiele für Arten von Syllogismen

Alle Griechen sind Menschen

AAA - modus barbara

Alle Menschen sind sterblich

Somit sind alle Griechen sterblich

Alle Professoren sind ernst

AOO - modus baroco

Einige Dozenten sind nicht ernst

Somit sind einige Dozenten keine Professoren

Typisierung der Relationen

A Alle S sind P E Keine S sind P

I Einige S sind P O Einige S sind nicht P

Modus Ponens

Beispiel

Wenn das Kind schreit, hat es Hunger

Das Kind schreit

Also, hat das Kind Hunger

Modus Ponens

Beispiel

Wenn das Kind schreit, hat es Hunger

Das Kind schreit

Also, hat das Kind Hunger

Fakt

Korrektheit dieser Schlussfigur ist unabhängig von den konkreten Aussagen

Modus Ponens

Beispiel

Wenn das Kind schreit, hat es Hunger

Das Kind schreit

Also, hat das Kind Hunger

Fakt

Korrektheit dieser Schlussfigur ist unabhängig von den konkreten Aussagen

Definition (Modus Ponens)

Wenn A , dann B

A gilt

Also, gilt B

Logik in der Informatik



Logik in der Informatik



Logik in der Informatik



