

Seminar: Selbstbezüglichkeit

1. Organisation, Zeitplan & Themen

Malte Helmert und Florian Pommerening

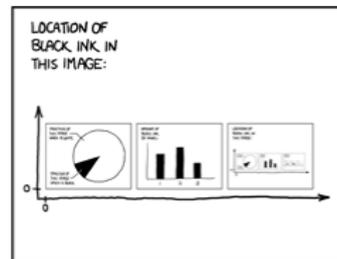
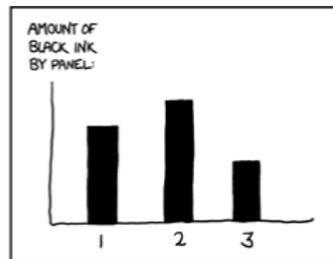
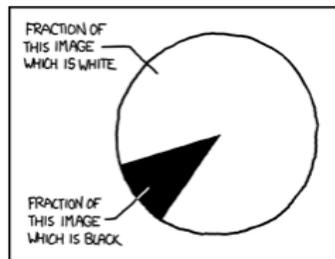
Universität Basel

19. September 2019

Selbstbezüglichkeit

Thema des Seminars

Selbstbezüglichkeit



[www.xkcd.com]

- kommt in der Informatik, Mathematik und Logik vor
- wichtige Anwendungen
 - Halteproblem
 - Russellsche Antinomie
 - Gödels Unvollständigkeitssatz

Organisation

Zielgruppe und Voraussetzungen

Zielgruppe

- BSc Studierende Computer Science und verwandte Fächer

Voraussetzungen

- Grundlegendes Wissen aus der theoretischen Informatik
- Fähigkeit selbstständig zu arbeiten
- Programmierkenntnisse (für das Softwareprojekt)

... oder die Bereitschaft sich die Grundlagen selbst anzueignen

Format

Format des Seminars

- Theoretischer Teil + Programmierprojekt
- 6 ECTS Punkte
- Bewertung: benotet

Lernziele

Lernziele

Seminar: mit wissenschaftlicher Literatur arbeiten

- lesen und verstehen
- erklären und präsentieren
- vergleichen und diskutieren

Projekt: implementieren von effizienten Programmen

- Programmierübung
- sauberer und effizienter Code (↔ Code-Reviews)
- Auswertung von Algorithmen
(↔ wissenschaftliche Experimente)

Treffen

19. Sep	16:15–18:00	Organisation & Themen (heute)
26. Sep	16:15–18:00	Tipps zu Seminararbeiten und Präsentationen
9. Nov	9:00–18:00	Seminarpräsentationen
10. Nov	9:00–18:00	Seminarpräsentationen
14. Dez	9:00–18:00	Projektpräsentationen

Raum 00.003, Spiegelgasse 1

Anforderungen I

Anforderungen zum Bestehen

- Schreiben einer Seminararbeit
 - 10–12 Seiten, \LaTeX
 - Abgabe: erste Version am 27. Okt, zweite Version am 17. Nov

Anforderungen I

Anforderungen zum Bestehen

- Schreiben einer Seminararbeit
 - 10–12 Seiten, \LaTeX
 - Abgabe: erste Version am 27. Okt, zweite Version am 17. Nov
- Schreiben eines Peer-Reviews
 - 1–2 Seiten
 - Abgabe am 1. November

Anforderungen I

Anforderungen zum Bestehen

- Schreiben einer Seminararbeit
 - 10–12 Seiten, \LaTeX
 - Abgabe: erste Version am 27. Okt, zweite Version am 17. Nov
- Schreiben eines Peer-Reviews
 - 1–2 Seiten
 - Abgabe am 1. November
- Halten einer Seminarpräsentation
 - 30 Minuten mit anschließender Diskussion
 - Abgabe der Folien am 6. November
 - Abgabe der endgültigen Version der Folien am 8. November

Anforderungen I

Anforderungen zum Bestehen

- Schreiben einer Seminararbeit
 - 10–12 Seiten, \LaTeX
 - Abgabe: erste Version am 27. Okt, zweite Version am 17. Nov
- Schreiben eines Peer-Reviews
 - 1–2 Seiten
 - Abgabe am 1. November
- Halten einer Seminarpräsentation
 - 30 Minuten mit anschließender Diskussion
 - Abgabe der Folien am 6. November
 - Abgabe der endgültigen Version der Folien am 8. November
- Aktive Beteiligung an den Diskussionen

Anforderungen II

Anforderungen zum Bestehen (Fortsetzung)

- Abgabe einer Implementierung für das Programmierprojekt
 - Abgabe am 8. Dezember

Anforderungen II

Anforderungen zum Bestehen (Fortsetzung)

- Abgabe einer Implementierung für das Programmierprojekt
 - Abgabe am 8. Dezember
- Halten einer Projektpräsentation
 - 15-18 Minuten mit anschließender Diskussion

Anforderungen II

Anforderungen zum Bestehen (Fortsetzung)

- Abgabe einer Implementierung für das Programmierprojekt
 - Abgabe am 8. Dezember
- Halten einer Projektpräsentation
 - 15-18 Minuten mit anschließender Diskussion
- Teilnahme an allen Wochenendtreffen

Benotung

Benotung

- Seminausarbeitung (15+15%)
- Peer-Review (10%)
- Seminarpräsentation (15%)
- Teilnahme an Diskussionen (10%)
- Implementierung für das Programmierprojekt (20%)
- Projektpräsentation (15%)

Die Komponenten werden einzeln auf einer Skala von 1.0-6.0 bewertet. Die endgültige Note ist dann der gewichtete Durchschnitt aller Teilnoten.

Peer-Review

- Sie bekommen eine Seminararbeit von jemand anderem
- und geben schriftlich Feedback.
- Feedback ist anonym.
- Ziel: Lernen durch neue Perspektive

Programmierprojekt

- Programmierprojekt hat mit dem Seminarthema zu tun
- Besprechen Sie die folgenden Punkte mit Ihrer Betreuerin:
 - Programmiersprache
 - Projektbeschreibung
- Wann?
 - Sobald Sie einen ausreichenden Überblick über das Thema haben
 - spätestens am 12. November

Betreuende



Malte Helmert



Florian Pommerening



Silvan Sievers



Jendrik Seipp



Thomas Keller



Salomé Eriksson



Cedric Geissmann

Organisatoren

Organisatoren

Malte Helmert

- **email:** malte.helmert@unibas.ch
- **office:** Spiegelgasse 1, room 06.004

Florian Pommerening

- **E-Mail:** florian.pommerening@unibas.ch
- **Büro:** Spiegelgasse 1, Raum 04.005

Weitere Betreuende

- Spiegelgasse 1, Raum 04.005
 - Thomas Keller
tho.keller@unibas.ch
- Spiegelgasse 1, Raum 04.002
 - Silvan Sievers
silvan.sievers@unibas.ch
- Spiegelgasse 5, Raum 04.001
 - Jendrik Seipp
jendrik.seipp@unibas.ch
 - Salomé Eriksson
salome.eriksson@unibas.ch
 - Cedric Geissmann
cedric.geissmann@unibas.ch

Material & Anmeldung

Seminar Homepage

<https://dmi.unibas.ch/de/studium/computer-science-informatik/lehrangebot-hs19/seminar-selbstbeueglichkeit/>

- Beschreibung des Seminars
- Folien

Adam

<https://adam.unibas.ch/>

- Zusätzliches Material

Anmeldung:

- <https://services.unibas.ch/>

Plagiate

Plagiate

- **Plagiat:** Darstellen fremder Arbeit oder Ideen als die eigenen
- Konsequenz: Seminar nicht bestanden
- Im Zweifelsfall **vorher fragen!**

Bei wiederholtem Vorkommen ist ein Ausschluss aus dem Studiengang möglich.

Sprache

- Seminararbeit auf Englisch (empfohlen) oder Deutsch
- Peer-Review auf Englisch oder Deutsch
(Sprache muss für den Empfänger in Ordnung sein)
- Präsentationen auf Englisch oder Deutsch
- Programmiersprache nach Absprache mit Betreuer/-in

Fragen zur Organisation

Fragen?

Seminarthemen

Thema #1: Theorie und Anwendungen von Selbstbezüglichkeit



The Ultimate Self-Referential Comic.

- Was ist Selbstbezüglichkeit?
- Wofür ist Selbstbezüglichkeit wichtig?

Thema #2: Paradoxien durch Selbstbezüglichkeit



[Quinn Dombrowski, CC BY-SA 4.0, Flickr]

- Was ist ein Paradox?
- Paradoxien aus verschiedenen Bereichen
- gemeinsame Eigenschaften

Thema #3: Das **MU**-Rätsel

MU-Rätsel: Ersetzungsregeln

Regel 1: $xI \rightarrow xIU$

Regel 2: $Mx \rightarrow Mxx$

Regel 3: $xIIIy \rightarrow xUy$

Regel 4: $xUUy \rightarrow xy$

- Rätsel aus „Gödel, Escher, Bach“
 - Erzeuge **MU** aus **MI** durch Anwendung der Regeln.
- Aussagen im System \Leftrightarrow Aussagen über das System

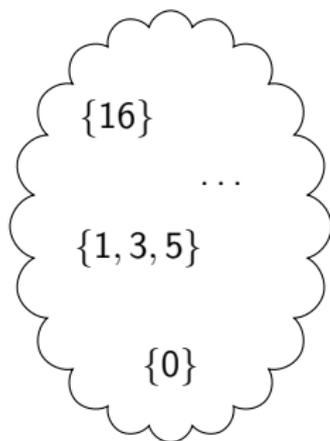
Thema #4: Gödels Unvollständigkeitssatz



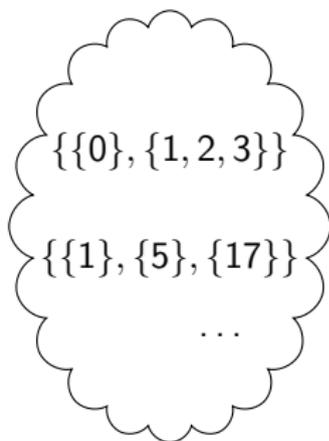
- Beweissysteme für natürliche Zahlen sind alle unvollständig oder inkonsistent
 - in jedem gibt es unbeweisbare wahre Sätze
- Diese Systeme können ihre eigene Konsistenz nicht beweisen

Thema #5: Auflösen von Paradoxien durch Hierarchien

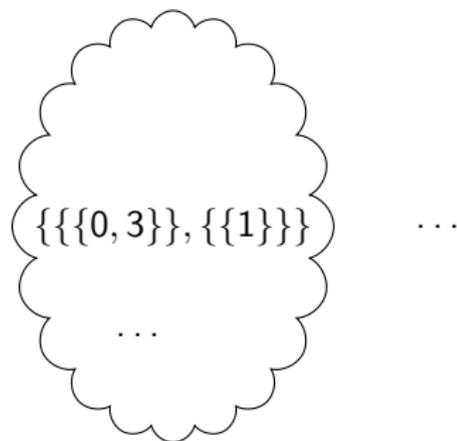
Type 1



Type 2



Type 3



- Russells Typenhierarchie
- Tarskis Sprachhierarchie

Thema #6: Auflösen von Paradoxien in der Logik

Lügner-Paradox

Ich lüge immer.

- Wie können Paradoxien in der Logik gelöst werden?
- Mehrwertige Logik
- Fuzzy Logic

Thema #7: Selbstbezüglichkeit in Modallogik



[Ggobeil22, CC BY-SA 4.0, creativecommons.org]

- Paradox der unerwarteten Prüfung
- Kripke-Strukturen

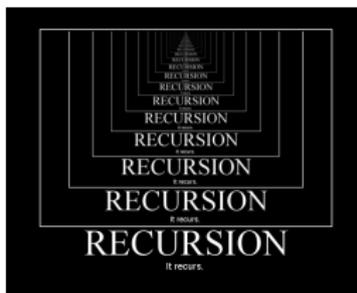
Thema #8: Auflösen von „Paradoxes of Denotation“

Berry's paradox

the least number that cannot be referred to by a description containing less than 100 symbols.

- Paradox entsteht durch Beschreibung einer Eigenschaft
- Funktionieren „Standard“-Lösungen hier auch?

Thema #9: Rekursive Strukturen und Prozesse



[<https://www.flickr.com/photos/torley/2361164281> (CC BY-SA)]

Rekursion als Beispiel für Selbstbezüglichkeit in verschiedenen Zusammenhängen:

- Mathematische Funktionen
- Computerprogramme
- Physikalische Theorien
- Sprache, Musik, Geometrie, ...

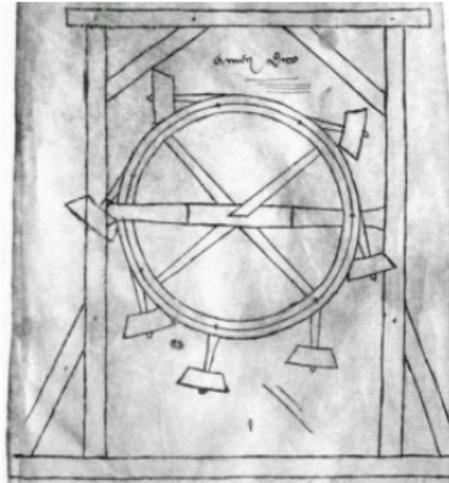
Thema #10: BlooP und FlooP und GlooP



[abstrusegoose.com, CC-BY-NC 3.0]

- Welche dieser Programmiersprachen ist die mächtigste?
- In BlooP ist kein Programm möglich, das sich selbst beschreibt.
- Zusammenhang zu primitiv rekursiven Funktionen

Thema #11: Das Halteproblem



- Spezielles Halteproblem: Wird eine Turingmaschine auf ihrer eigenen Kodierung halten?
- Beweis verwendet Selbstbezüglichkeit

Thema #12: Kombinatorische Logik



- „To Mock a Mockingbird“
 - Jeder Vogel reagiert auf Namen mit bestimmter Antwort
 - Wenn Mockingbird den Vogelnamen x hört, ruft er den Vogelnamen, den x auf sich selbst antwortet.
 - Wie reagiert er auf „Mockingbird“?
- Einführung in die Kombinatorische Logik

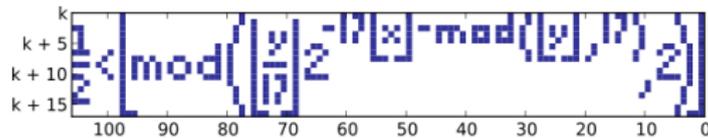
Thema #13: Quines

Quine

Schreibe den folgenden Satz zweimal auf; beim zweiten Mal in Anführungszeichen: „Schreibe den folgenden Satz zweimal auf; beim zweiten Mal in Anführungszeichen: “

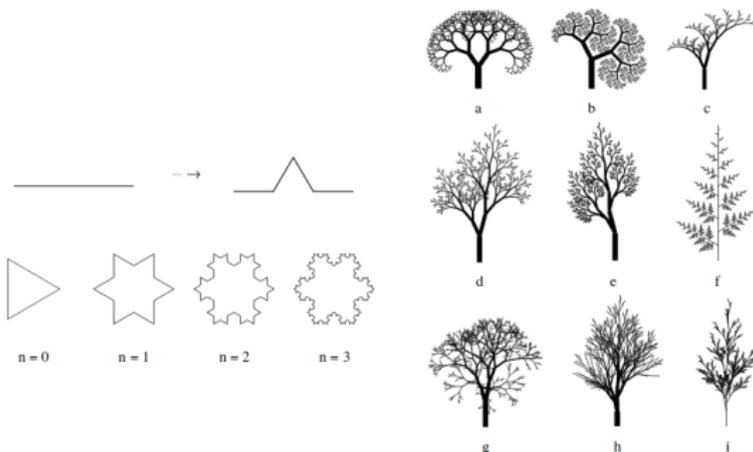
- Quines sind Programme, die ihren Quelltext ausgeben
- Formaler Hintergrund: Kleenes Fixpunktsatz

Thema #14: Tuppers selbstbezügliche Formel



- Formel plottet ihre eigene Beschreibung
- Kritik und Erweiterung durch Trávník

Thema #15: Kontextfreie Grammatiken und Fraktale



[Prusinkiewicz et al. 1996]

- Lindenmayer-Systeme
- Anwendung zur Simulation von Fraktalen und Pflanzen

Nächste Schritte

Zuteilung der Themen

- Wir verschicken einen Link zu einem Doodle
- Nummer im Doodle = Nummer des Themas in diesen Folien
- Markieren Sie **mindestens 3** Themen mit **Yes**
- Markieren Sie **mindestens 5** Themen positiv: **Yes** oder **(Yes)**
- Bis **25. September (kommender Mittwoch)**

Danach:

- Zuteilung der Themen und Betreuenden wird am 30. September bekannt gegeben.
- Fangen Sie **so früh wie möglich** damit an, das Material zu lesen und Kontakt mit Ihren Betreuenden aufzunehmen.

Wichtige Daten

- 25. Sep Lieblingsthemen im Doodle markiert
- 26. Sep Zweites Treffen & Themenzuweisung
- 27. Okt Abgabe: Seminarausarbeitung (erste Version)
- 1. Nov Abgabe: Peer-Review
- 6. Nov Abgabe: Folien (erste Version)
- 7/8. Nov Feedback zu Folien von Betreuer/-in
- 8. Nov Abgabe: Folien (endgültige Version)
- 9/10. Nov Seminarpräsentationen
- 12. Nov Programmierprojekt abgesprochen
- 17. Nov Abgabe: Seminarausarbeitung (endgültige Version)
- 8. Dez Abgabe: Implementierung
- 14. Dez Projektpräsentation