Vorlesung

Grundlagen der Theoretischen Informatik / Einführung in die Theoretische Informatik I

Bernhard Beckert

Institut für Informatik

UNIVERSITÄT KOBLENZ · LANDAU

Sommersemester 2007

Dank

Diese Vorlesungsmaterialien basieren ganz wesentlich auf den Folien zu den Vorlesungen von

Katrin Erk (gehalten an der Universität Koblenz-Landau)

Jürgen Dix (gehalten an der TU Clausthal)

Ihnen beiden gilt mein herzlicher Dank.

– Bernhard Beckert, April 2007

Teil I

Einführung

Teil I

Einführung

Kontakt

Bernhard Beckert

Email: beckert@uni-koblenz.de

Webseite: www.uni-koblenz.de/~beckert

Raum: B 218

Sprechstunde:

Donnerstags, 16 Uhr

(sonst auch immer, wenn die Tür offen steht)

Ulrich Koch

Email: koch@uni-koblenz.de

Webseite: www.uni-koblenz.de/~koch

Raum: A 219

Kontakt

Bernhard Beckert

Email: beckert@uni-koblenz.de

Webseite: www.uni-koblenz.de/~beckert

Raum: B 218

Sprechstunde:

Donnerstags, 16 Uhr

(sonst auch immer, wenn die Tür offen steht)

Ulrich Koch

Email: koch@uni-koblenz.de

Webseite: www.uni-koblenz.de/~koch

Raum: A 219

Webseite

http://www.uni-koblenz.de/~beckert/Lehre/TheoretischeInformatik/

Alle relevante Information auf der Webseite

- Folien
- Weitere Materialien
- Termine usw.

Webseite

http://www.uni-koblenz.de/~beckert/Lehre/TheoretischeInformatik/

Alle relevante Information auf der Webseite

- Folien
- Weitere Materialien
- Termine usw.

Übungen

Termine und Einteilung

Stehen noch nicht endgültig fest

Webseite beachten:

MeToo-Registrierung sobald Termine feststehen!

Übungsblätter

- Wöchentlich, jeden Montag
- Dürfen – müssen aber nicht – abgegeben werden
- Werden in den Übungen der darauffolgenden Woche besprochen
- Kein Einfluss auf Scheinvergabe

Übungen

Termine und Einteilung

Stehen noch nicht endgültig fest

Webseite beachten:

MeToo-Registrierung sobald Termine feststehen!

Übungsblätter

- Wöchentlich, jeden Montag
- Dürfen – müssen aber nicht – abgegeben werden
- Werden in den Übungen der darauffolgenden Woche besprochen
- Kein Einfluss auf Scheinvergabe

Prüfungen und Scheinvergabe

Teilklausuren während des Semesters

- Während des Semesters
- drei Teilklausuren (je ca. 30-40 Minuten)
- jeweils Mitte Mai, Mitte Juni, Mitte Juli
- Schein bei 50% der insgesamt in den drei Teilklausuren zu erzielenden Punkte
- Freiversuch gilt für alle Teilklausuren zusammen
- Wiederholung einzelner Teilklausuren nicht möglich

Nachklausur zum Ende der Semesters

- Gegen Ende des Semesters (Ende September)
- Nachklausur hat gleichen „Wert“ wie alle Teilklausuren zusammen
- 90 Minuten Dauer
- Schein bei 50% der Punkte in der Nachklausur

Prüfungen und Scheinvergabe

Teilklausuren während des Semesters

- Während des Semesters
- drei Teilklausuren (je ca. 30-40 Minuten)
- jeweils Mitte Mai, Mitte Juni, Mitte Juli
- Schein bei 50% der insgesamt in den drei Teilklausuren zu erzielenden Punkte
- Freiversuch gilt für alle Teilklausuren zusammen
- Wiederholung einzelner Teilklausuren nicht möglich

Nachklausur zum Ende der Semesters

- Gegen Ende des Semesters (Ende September)
- Nachklausur hat gleichen „Wert“ wie alle Teilklausuren zusammen
- 90 Minuten Dauer
- Schein bei 50% der Punkte in der Nachklausur

Prüfungen und Scheinvergabe

Anmeldung zu und Teilnahme an der ersten Prüfung (also den Teilklausuren während des Semesters) ist Voraussetzung für die Teilnahme an der Nachklausur.

Zudem darf an der Nachklausur (wie auch an den Teilklausuren) teilnehmen, wer an einer Prüfung zu „Einführung in die Theoretische Informatik I“ in frühren Semestern teilgenommen und diese nicht bestanden oder einen Freiversuch gesetzt hat.

Prüfungen und Scheinvergabe

Anmeldung zu und Teilnahme an der ersten Prüfung (also den Teilklausuren während des Semesters) ist Voraussetzung für die Teilnahme an der Nachklausur.

Zudem darf an der Nachklausur (wie auch an den Teilklausuren) teilnehmen, wer an einer Prüfung zu „Einführung in die Theoretische Informatik I“ in frühren Semestern teilgenommen und diese nicht bestanden oder einen Freiversuch gesetzt hat.

Literatur

Buch zur Vorlesung

Erk, Katrin and Priese, Lutz (2002).
Theoretische Informatik: Eine umfassende Einführung.
2. Auflage.
Springer-Verlag.

Literatur

Weitere Literatur


J. Hopcroft, R. Motwani, and J. Ullman (2002).
Einführung in die Automatentheorie, Formale Sprachen und Komplexitätstheorie.
Pearson.


G. Vossen and K.-U. Witt (2004).
Grundkurs Theoretische Informatik.
Vieweg.


U. Schöning (1994).
Theoretische Informatik: kurzgefaßt.
Spektrum-Verlag.

Teil I

Einführung

Teil I

Einführung

Theoretische Informatik befasst sich mit . . .

Grundlegende Konzepte der Informatik

- Probleme und ihre Beschreibung
- Systeme/Automaten/Maschinen, die Probleme lösen
- Lösbarkeit von Problemen
  (Entscheidbarkeit/Berechenbarkeit und deren Grenzen)
- Schwierigkeit (Komplexität) der Lösung von Problemen

Theoretische Informatik befasst sich mit . . .

Grundlegende Konzepte der Informatik

- Probleme und ihre Beschreibung
- Systeme/Automaten/Maschinen, die Probleme lösen
- Lösbarkeit von Problemen
  (Entscheidbarkeit/Berechenbarkeit und deren Grenzen)
- Schwierigkeit (Komplexität) der Lösung von Problemen

Theoretische Informatik befasst sich mit . . .

Grundlegende Konzepte der Informatik

- Probleme und ihre Beschreibung
- Systeme/Automaten/Maschinen, die Probleme lösen
- Lösbarkeit von Problemen (Entscheidbarkeit/Berechenbarkeit und deren Grenzen)
- Schwierigkeit (Komplexität) der Lösung von Problemen

Theoretische Informatik befasst sich mit . . .

Grundlegende Konzepte der Informatik

- Probleme und ihre Beschreibung
- Systeme/Automaten/Maschinen, die Probleme lösen
- Lösbarkeit von Problemen
  (Entscheidbarkeit/Berechenbarkeit und deren Grenzen)
- Schwierigkeit (Komplexität) der Lösung von Problemen

Teilgebiete der Theoretischen Informatik

Teilgebiete

- Formale Sprachen
- Automatentheorie
- Berechenbarkeitstheorie
- Komplexitätstheorie
- (Logik)

Teilgebiete der Theoretischen Informatik

Teilgebiete

- Formale Sprachen
- Automatentheorie
- Berechenbarkeitstheorie
- Komplexitätstheorie
- (Logik)

Teilgebiete der Theoretischen Informatik

Teilgebiete

- Formale Sprachen
- Automatentheorie
- Berechenbarkeitstheorie
- Komplexitätstheorie
- (Logik)

Teilgebiete der Theoretischen Informatik

Teilgebiete

- Formale Sprachen
- Automatentheorie
- Berechenbarkeitstheorie
- Komplexitätstheorie
- (Logik)

Teilgebiete der Theoretischen Informatik

Teilgebiete

- Formale Sprachen
- Automatentheorie
- Berechenbarkeitstheorie
- Komplexitätstheorie
- (Logik)

Warum Theoretische Informatik?

Theoretische Informatik . . .

- ist die Grundlage
- ist wichtig
  (bspw. für Algorithmentechnik, Software Engineering, Compilerbau)
- hilft, weitere Themen/Vorlesungen der Informatik zu verstehen
- veraltet nicht
- macht Spaß

Warum Theoretische Informatik?

Theoretische Informatik . . .

- ist die Grundlage
- ist wichtig
  (bspw. für Algorithmentechnik, Software Engineering, Compilerbau)
- hilft, weitere Themen/Vorlesungen der Informatik zu verstehen
- veraltet nicht
- macht Spaß

Warum Theoretische Informatik?

Theoretische Informatik . . .

- ist die Grundlage
- ist wichtig
  (bspw. für Algorithmentechnik, Software Engineering, Compilerbau)
- hilft, weitere Themen/Vorlesungen der Informatik zu verstehen
- veraltet nicht
- macht Spaß

Warum Theoretische Informatik?

Theoretische Informatik . . .

- ist die Grundlage
- ist wichtig
  (bspw. für Algorithmentechnik, Software Engineering, Compilerbau)
- hilft, weitere Themen/Vorlesungen der Informatik zu verstehen
- veraltet nicht
- macht Spaß

Warum Theoretische Informatik?

Theoretische Informatik . . .

- ist die Grundlage
- ist wichtig
  (bspw. für Algorithmentechnik, Software Engineering, Compilerbau)
- hilft, weitere Themen/Vorlesungen der Informatik zu verstehen
- veraltet nicht
- macht Spaß

Inhalt der Vorlesung

1. Terminologie
2. Endliche Automaten und reguläre Sprachen
3. Kellerautomaten und kontextfreie Sprachen
4. Turingmaschinen und rekursiv aufzählbare Sprachen
5. Berechenbarkeit, (Un-)Entscheidbarkeit
6. Komplexitätsklassen P und NP

Inhalt der Vorlesung

1. Terminologie
2. Endliche Automaten und reguläre Sprachen
3. Kellerautomaten und kontextfreie Sprachen
4. Turingmaschinen und rekursiv aufzählbare Sprachen
5. Berechenbarkeit, (Un-)Entscheidbarkeit
6. Komplexitätsklassen P und NP

Inhalt der Vorlesung

1. Terminologie
2. Endliche Automaten und reguläre Sprachen
3. Kellerautomaten und kontextfreie Sprachen
4. Turingmaschinen und rekursiv aufzählbare Sprachen
5. Berechenbarkeit, (Un-)Entscheidbarkeit
6. Komplexitätsklassen P und NP

Inhalt der Vorlesung

1. Terminologie
2. Endliche Automaten und reguläre Sprachen
3. Kellerautomaten und kontextfreie Sprachen
4. Turingmaschinen und rekursiv aufzählbare Sprachen
5. Berechenbarkeit, (Un-)Entscheidbarkeit
6. Komplexitätsklassen P und NP

Inhalt der Vorlesung

1. Terminologie
2. Endliche Automaten und reguläre Sprachen
3. Kellerautomaten und kontextfreie Sprachen
4. Turingmaschinen und rekursiv aufzählbare Sprachen
5. Berechenbarkeit, (Un-)Entscheidbarkeit
6. Komplexitätsklassen P und NP

Inhalt der Vorlesung

1. Terminologie
2. Endliche Automaten und reguläre Sprachen
3. Kellerautomaten und kontextfreie Sprachen
4. Turingmaschinen und rekursiv aufzählbare Sprachen
5. Berechenbarkeit, (Un-)Entscheidbarkeit
6. Komplexitätsklassen P und NP

Teil I

Einführung

Teil I

Einführung

Logische Formeln

Aussagenlogische Operatoren

- $\neg$ Negationssymbol („nicht“)
- $\wedge$ Konjunktionssymbol („und“)
- $\vee$ Disjunktionssymbol („oder“)
- $\Rightarrow$ Implikationssymbol („wenn . . . dann“)
- $\Leftrightarrow$ Symbol für Äquivalenz („genau dann, wenn“)

Zusätzliche prädikatenlogische Operatoren

- $\forall$ Allquantor („für alle“)
- $\exists$ Existenzquantor („es gibt“)

Logische Formeln

Aussagenlogische Operatoren

- $\neg$ Negationssymbol („nicht“)
- $\wedge$ Konjunktionssymbol („und“)
- $\vee$ Disjunktionssymbol („oder“)
- $\Rightarrow$ Implikationssymbol („wenn . . . dann“)
- $\Leftrightarrow$ Symbol für Äquivalenz („genau dann, wenn“)

Zusätzliche prädikatenlogische Operatoren

- $\forall$ Allquantor („für alle“)
- $\exists$ Existenzquantor („es gibt“)

Logische Formeln

Formeln

- Atomare Aussagen sind Formeln
- Seien A, B Formeln, x eine Variable, dann sind

$$\neg A, \quad (A \wedge B), \quad (A \vee B), \quad (A \Rightarrow B), \quad (A \Leftrightarrow B), \quad \forall x A, \quad \exists x A$$

Formeln

Logische Formeln

Beispiel 3.1

$$\neg \underbrace{(y \leq x)}_{\textit{Atom}} \Rightarrow \exists z \underbrace{(\neg(z \leq x) \wedge \neg(y \leq z))}_{\textit{Skopus von } \exists z}$$

Logische Formeln

Beispiel 3.2

„Alle, die in Koblenz studieren, sind schlau“

$$\underbrace{\forall \underbrace{x}_{\text{Variable}} \underbrace{(\mathit{studiertIn}(x, \mathit{koblenz}) \Rightarrow \mathit{schlau}(x))}_{\text{Formel (Skopus)}}}_{\text{Formel}}$$

„Jemand, der in Landau studiert ist schlau“

$$\underbrace{\exists \underbrace{x}_{\text{Variable}} \underbrace{(\mathit{studiertIn}(x, \mathit{landau}) \wedge \mathit{schlau}(x))}_{\text{Formel (Skopus)}}}_{\text{Formel}}$$

Logische Formeln

Beispiel 3.2

„Alle, die in Koblenz studieren, sind schlau“

$$\underbrace{\forall \underbrace{x}_{\text{Variable}} \underbrace{(studiertIn(x, koblenz) \Rightarrow schlau(x))}_{\text{Formel (Skopus)}}}_{\text{Formel}}$$

„Jemand, der in Landau studiert ist schlau“

$$\underbrace{\exists \underbrace{x}_{\text{Variable}} \underbrace{(studiertIn(x, landau) \wedge schlau(x))}_{\text{Formel (Skopus)}}}_{\text{Formel}}$$

Eigenschaften von Quantoren

Quantoren gleicher Art kommutieren

$\forall x \forall y$	ist das gleiche wie	$\forall y \forall x$
$\exists x \exists y$	ist das gleiche wie	$\exists y \exists x$

Eigenschaften von Quantoren

Verschiedene Quantoren kommutieren NICHT

$\exists x \forall y$ ist nicht das gleiche wie $\forall y \exists x$

Beispiel 3.3

$\exists x \forall y \; loves(x, y)$

Es gibt eine Person, die jeden Menschen in der Welt liebt (einschließlich sich selbst)

$\forall y \exists x \; loves(x, y)$

Jeder Mensch wird von mindestens einer Person geliebt

(Beides hoffentliche wahr aber verschieden: das erste impliziert das zweite aber nicht umgekehrt)

Eigenschaften von Quantoren

Verschiedene Quantoren kommutieren NICHT

$\exists x \forall y$ ist nicht das gleiche wie $\forall y \exists x$

Beispiel 3.3

$\exists x \forall y \; loves(x, y)$

Es gibt eine Person, die jeden Menschen in der Welt liebt (einschließlich sich selbst)

$\forall y \exists x \; loves(x, y)$

Jeder Mensch wird von mindestens einer Person geliebt

(Beides hoffentliche wahr aber verschieden: das erste impliziert das zweite aber nicht umgekehrt)

Eigenschaften von Quantoren

Verschiedene Quantoren kommutieren NICHT

$\exists x \forall y$ ist nicht das gleiche wie $\forall y \exists x$

Beispiel 3.3

$\exists x \forall y \; loves(x, y)$

Es gibt eine Person, die jeden Menschen in der Welt liebt (einschließlich sich selbst)

$\forall y \exists x \; loves(x, y)$

Jeder Mensch wird von mindestens einer Person geliebt

(Beides hoffentliche wahr aber verschieden: das erste impliziert das zweite aber nicht umgekehrt)

Eigenschaften von Quantoren

Verschiedene Quantoren kommutieren NICHT

$\exists x \forall y$ ist nicht das gleiche wie $\forall y \exists x$

Beispiel 3.4

$\forall x \exists y \; mutter(y, x)$

Jeder hat eine Mutter ***(richtig)***

$\exists y \forall x \; mutter(y, x)$

Es gibt eine Person, die die Mutter von jedem ist ***(falsch)***

Eigenschaften von Quantoren

Verschiedene Quantoren kommutieren NICHT

$\exists x \forall y$ ist nicht das gleiche wie $\forall y \exists x$

Beispiel 3.4

$\forall x \exists y \; mutter(y, x)$

Jeder hat eine Mutter ***(richtig)***

$\exists y \forall x \; mutter(y, x)$

Es gibt eine Person, die die Mutter von jedem ist ***(falsch)***

Eigenschaften von Quantoren

Verschiedene Quantoren kommutieren NICHT

$\exists x \forall y$ ist nicht das gleiche wie $\forall y \exists x$

Beispiel 3.4

$\forall x \exists y \; mutter(y, x)$

Jeder hat eine Mutter ***(richtig)***

$\exists y \forall x \; mutter(y, x)$

Es gibt eine Person, die die Mutter von jedem ist ***(falsch)***

Eigenschaften von Quantoren

Dualität der Quantoren

$\forall x \ldots$	ist das gleiche wie	$\neg\exists x \neg \ldots$
$\exists x \ldots$	ist das gleiche wie	$\neg\forall x \neg \ldots$

Beispiel 3.5

$\forall x\ mag(x, eiskrem)$	*ist das gleiche wie*	$\neg\exists x \neg mag(x, eiskrem)$
$\exists x\ mag(x, broccoli)$	*ist das gleiche wie*	$\neg\forall x \neg mag(x, broccoli)$

Eigenschaften von Quantoren

Dualität der Quantoren

$\forall x \ldots$ ist das gleiche wie $\neg\exists x \neg \ldots$

$\exists x \ldots$ ist das gleiche wie $\neg\forall x \neg \ldots$

Beispiel 3.5

$\forall x\ mag(x, eiskrem)$ *ist das gleiche wie* $\neg\exists x \neg mag(x, eiskrem)$

$\exists x\ mag(x, broccoli)$ *ist das gleiche wie* $\neg\forall x \neg mag(x, broccoli)$

Eigenschaften von Quantoren

Dualität der Quantoren

$\forall x \ldots$	ist das gleiche wie	$\neg\exists x \neg \ldots$
$\exists x \ldots$	ist das gleiche wie	$\neg\forall x \neg \ldots$

Beispiel 3.5

$\forall x\ mag(x, eiskrem)$	*ist das gleiche wie*	$\neg\exists x\ \neg mag(x, eiskrem)$
$\exists x\ mag(x, broccoli)$	*ist das gleiche wie*	$\neg\forall x\ \neg mag(x, broccoli)$

Eigenschaften von Quantoren

$\forall$ distributiert NICHT über $\vee$

$\forall x\,(\ldots \vee \ldots)$ ist NICHT das gleiche wie $(\forall x \ldots) \vee (\forall x \ldots)$

Beispiel 3.6

$\forall x\,(\mathit{eiskrem}(x) \vee \mathit{broccoli}(x))$ *ist NICHT das gleiche wie*
$(\forall x\ \mathit{eiskrem}(x)) \vee (\forall x\ \mathit{broccoli}(x))$

Eigenschaften von Quantoren

$\forall$ distributiert NICHT über $\vee$

$\forall x\,(\ldots \vee \ldots)$ ist NICHT das gleiche wie $(\forall x \ldots) \vee (\forall x \ldots)$

Beispiel 3.6

$\forall x\,(\mathit{eiskrem}(x) \vee \mathit{broccoli}(x))$ *ist NICHT das gleiche wie*
$(\forall x\ \mathit{eiskrem}(x)) \vee (\forall x\ \mathit{broccoli}(x))$

Eigenschaften von Quantoren

$\exists$ distributiert NICHT über $\wedge$

$\exists x\,(\ldots \wedge \ldots)$ ist NICHT das gleiche wie $(\exists x \ldots) \wedge (\exists x \ldots)$

Beispiel 3.7

$\exists x\,(\mathit{gerade}(x) \wedge \mathit{ungerade}(x))$ *ist NICHT das gleiche wie*
$(\exists x\ \mathit{gerade}(x)) \wedge (\exists x\ \mathit{ungerade}(x))$

Eigenschaften von Quantoren

$\exists$ distributiert NICHT über $\wedge$

$\exists x\,(\ldots \wedge \ldots)$ ist NICHT das gleiche wie $(\exists x \ldots) \wedge (\exists x \ldots)$

Beispiel 3.7

$\exists x\,(\mathit{gerade}(x) \wedge \mathit{ungerade}(x))$ *ist NICHT das gleiche wie*
$(\exists x\ \mathit{gerade}(x)) \wedge (\exists x\ \mathit{ungerade}(x))$

Beispiele: Familienverhältnisse

Beispiel 3.8

- *„Brüder sind Geschwister“*
  $\forall x \forall y\ (bruder(x,y) \Rightarrow geschwister(x,y))$
- *„bruder“ ist symmetrisch*
  $\forall x \forall y\ (bruder(x,y) \Leftrightarrow bruder(y,x))$
- *„Mütter sind weibliche Elternteile“*
  $\forall x \forall y\ (mutter(x,y) \Leftrightarrow weiblich(x) \wedge elter(x,y)))$
- *„Ein Cousin ersten Grades ist*
  das Kind eines Geschwisters eines Elternteils“
  $\forall x \forall y\ (cousin1(x,y) \Leftrightarrow$
  $\exists p \exists ps\ (elter(p,x) \wedge geschwister(ps,p) \wedge elter(ps,y)))$

Beispiele: Familienverhältnisse

Beispiel 3.8

- *„Brüder sind Geschwister“*
  $\forall x \forall y\ (bruder(x,y) \Rightarrow geschwister(x,y))$
- *„bruder“ ist symmetrisch*
  $\forall x \forall y\ (bruder(x,y) \Leftrightarrow bruder(y,x))$
- *„Mütter sind weibliche Elternteile“*
  $\forall x \forall y\ (mutter(x,y) \Leftrightarrow weiblich(x) \wedge elter(x,y)))$
- *„Ein Cousin ersten Grades ist das Kind eines Geschwisters eines Elternteils“*
  $\forall x \forall y\ (cousin1(x,y) \Leftrightarrow \exists p \exists ps\ (elter(p,x) \wedge geschwister(ps,p) \wedge elter(ps,y)))$

Beispiele: Familienverhältnisse

Beispiel 3.8

- *„Brüder sind Geschwister“*
  $\forall x \forall y\ (bruder(x,y) \Rightarrow geschwister(x,y))$
- *„bruder“ ist symmetrisch*
  $\forall x \forall y\ (bruder(x,y) \Leftrightarrow bruder(y,x))$
- *„Mütter sind weibliche Elternteile“*
  $\forall x \forall y\ (mutter(x,y) \Leftrightarrow weiblich(x) \wedge elter(x,y)))$
- *„Ein Cousin ersten Grades ist das Kind eines Geschwisters eines Elternteils“*
  $\forall x \forall y\ (cousin1(x,y) \Leftrightarrow \exists p \exists ps\ (elter(p,x) \wedge geschwister(ps,p) \wedge elter(ps,y)))$

Beispiele: Familienverhältnisse

Beispiel 3.8

- *„Brüder sind Geschwister“*
  $\forall x \forall y \, (bruder(x,y) \Rightarrow geschwister(x,y))$
- *„bruder“ ist symmetrisch*
  $\forall x \forall y \, (bruder(x,y) \Leftrightarrow bruder(y,x))$
- *„Mütter sind weibliche Elternteile“*
  $\forall x \forall y \, (mutter(x,y) \Leftrightarrow weiblich(x) \wedge elter(x,y)))$
- *„Ein Cousin ersten Grades ist das Kind eines Geschwisters eines Elternteils“*
  $\forall x \forall y \, (cousin1(x,y) \Leftrightarrow$
  $\exists p \exists ps \, (elter(p,x) \wedge geschwister(ps,p) \wedge elter(ps,y)))$

Beispiele: Familienverhältnisse

Beispiel 3.9

Formalisierung von „Bruder, der nicht nur Halbbruder ist“

$$\begin{aligned}\forall x \forall y\ \mathit{bruder}(x,y) \Leftrightarrow\ & (\neg(x=y) \wedge \\ & \exists m \exists v\ (\neg(m=v) \wedge \\ & \quad \mathit{elter}(m,x) \wedge \mathit{elter}(v,x) \wedge \\ & \quad \mathit{elter}(m,y) \wedge \mathit{elter}(v,y)))\end{aligned}$$