Script generated by TTT

Title: Esparza: Diskrete Strukturen (21.11.2013)

Date: Thu Nov 21 10:16:21 CET 2013

Duration: 85:10 min

Pages: 51

Kapitel II – Grundlagen; Logik

- Tautologie, Widerspruch, Erfüllbarkeit, ...
  - Eine Formel F ist allgemeingültig wenn für jede Struktur $\mathcal{S}$, die zu F passt, gilt: $[F](\mathcal{S}) = 1$.
  - Eine Formel F ist ein Widerspruch wenn für jede Struktur $\mathcal{S}$, die zu F passt, gilt: $[F](\mathcal{S}) = 0$.
  - Eine Formel F ist erfüllbar wenn es eine Struktur $\mathcal{S}$ gibt, die zu F passt, und $[F](\mathcal{S}) = 1$ erfüllt.
  - Zwei Formeln F und G sind logisch äquivalent (symbolisch: $F \equiv G$) genau dann, wenn für jede Struktur $\mathcal{S}$, die zu F und zu G passt, gilt: $[F](\mathcal{S}) = [G](\mathcal{S})$
  - F folgt aus G (symbolisch: $F \vDash G$) genau dann, wenn $F \to G$ gültig ist.

Kapitel II – Grundlagen; Logik

- $\mathcal{S}$-Tautologien, $\mathcal{S}$-Widersprüche, ...
  - Sei $\boldsymbol{S}$ eine Menge von Strukturen. Eine Formel ist $\boldsymbol{S}$-gültig, wenn für alle Strukturen $\mathcal{S} \in \boldsymbol{S}$ gilt : $[F](\mathcal{S}) = 1$.
  - Sei $null$ eine Konstante und sei $nach$ ein einstelliges Funktionssymbol.
  - Sei N die Menge aller Strukturen $\mathcal{S} = (U_\mathcal{S}, I_\mathcal{S})$ mit
    - $U_\mathcal{S} = \mathbb{N}_0$
    - $null_\mathcal{S} = 0$
    - $nach(n) = n + 1$
    - ($I_\mathcal{S}$ kann für andere Konstanten und Prädikatensymbolen beliebig definiert sein)

Kapitel II – Grundlagen; Logik

- Tautologie, Widerspruch, Erfüllbarkeit, ...
  - Das Induktionsprinzip ist die Formel:

    $$\big(P(null) \wedge \forall y\, (P(y) \to P(nach(y))\big) \to \forall x\, P(x)$$

  - In allen Strukturen aus N „sagt“ diese Formel:

    Wenn 0 die Eigenschaft P hat, und
    für alle Zahlen n gilt: wenn n die Eigenschaft P hat, dann hat auch $n + 1$ die Eigenschaft P,
    dann haben alle Zahlen die Eigenschaft P

  - Das Induktionsprinzip ist nicht gültig, aber N-gültig.

Kapitel II – Grundlagen; Logik

- Äquivalenzregeln für Quantoren
  - De Morgan's: $\neg\forall x\, F \equiv \exists x \neg F$
    $\neg\exists x\, F \equiv \forall x \neg F$
  - Kommutativität: $\forall x \forall y\, F \equiv \forall y \forall x\, F$
    $\exists x \exists y\, F \equiv \exists y \exists x\, F$
  - Distributivität: $\forall x\, (F \wedge G) \equiv \forall x\, F \wedge \forall x\, G$
    $\exists x\, (F \vee G) \equiv \exists x\, F \vee \exists x\, G$
  - Falls x in G nicht frei vorkommt: $\exists x\, (F \wedge G) \equiv \exists x\, F \wedge G$
    $\exists x\, (F \vee G) \equiv \exists x\, F \vee G$
    $\forall x\, (F \wedge G) \equiv \forall x\, F \wedge G$
    $\forall x\, (F \vee G) \equiv \forall x\, F \vee G$

41

Kapitel II – Grundlagen; Logik

- Äquivalenz
  - De Mor
  - Kommu
  - Distribu
  - Falls x i
    frei vor

New
Size
Current page size
Style
5mm
7mm
Orientation
Pages
1
OK
Cancel

$F \wedge G \equiv G \wedge F$
$p \wedge q \equiv q \wedge p$
$\forall x,y,z \in \mathbb{N}$ $\forall x \forall y \forall z$
$(x \in \mathbb{N}\ y \in \mathbb{N})$
$\forall x\, G$
$\exists x\, G$
$\forall x\, F$
$[\forall x F](S) = 1$
gdw
für alle $d \in U_S$
$[\forall x F](S_{x:=d}) = 1$
$\neg \forall x\, F$

41

Kapitel II – Grundlagen; Logik

- Äquivalenzregeln für Quantoren
  - De Morgan's: $\neg\forall x\, F \equiv \exists x \neg F$
    $\neg\exists x\, F \equiv \forall x \neg F$
  - Kommutativität: $\forall x \forall y\, F \equiv \forall y \forall x\, F$
    $\exists x \exists y\, F \equiv \exists y \exists x\, F$
  - Distributivität: $\forall x\, (F \wedge G) \equiv \forall x\, F \wedge \forall x\, G$
    $\exists x\, (F \vee G) \equiv \exists x\, F \vee \exists x\, G$
  - Falls x in G nicht frei vorkommt: $\exists x\, (F \wedge G) \equiv \exists x\, F \wedge G$
    $\exists x\, (F \vee G) \equiv \exists x\, F \vee G$
    $\forall x\, (F \wedge G) \equiv \forall x\, F \wedge G$
    $\forall x\, (F \vee G) \equiv \forall x\, F \vee G$

$F \wedge G \equiv G \wedge F$
$p \wedge q \equiv q \wedge p$
$\forall x,y,z \in \mathbb{N}$ $\forall x \forall y \forall z$
$(x \in \mathbb{N}\ y \in \mathbb{N})$
$\forall x\, F$
$[\forall x F](S) = 1$
gdw
für alle $d \in U_S$
$[\forall x F](S_{x:=d}) = 1$
$\neg \forall x\, F$

41

Kapitel II – Grundlagen; Logik

- Äquivalenz
  - De Mor
  - Kommu
  - Distribu
  - Falls x i
    frei vor

New

Size

Current page size

Style

5mm

7mm

Orientation

Pages

1

OK Cancel

$\forall x\, G$

$\exists x\, G$

G

G

G

G

$F \wedge G \equiv G \wedge F$

$p \wedge q \equiv q \wedge p$

$\forall x, y, z \in \mathbb{N}$ $\forall x \forall y \forall z$

$(x \in \mathbb{N} \wedge y \in \mathbb{N}$

$\forall x\, F$

$\forall x\, F$

Kapitel II – Grundlagen; Logik

- Äquivalenzregeln für Quantoren
  - De Morgan's: $\neg\forall x\, F \equiv \exists x\, \neg F$
    $\neg\exists x\, F \equiv \forall x\, \neg F$
  - Kommutativität: $\forall x \forall y\, F \equiv \forall y \forall x\, F$
    $\exists x \exists y\, F \equiv \exists y \exists x\, F$
  - Distributivität: $\forall x\, (F \wedge G) \equiv \forall x\, F \wedge \forall x\, G$
    $\exists x\, (F \vee G) \equiv \exists x\, F \vee \exists x\, G$
  - Falls x in G nicht frei vorkommt: $\exists x\, (F \wedge G) \equiv \exists x\, F \wedge G$
    $\exists x\, (F \vee G) \equiv \exists x\, F \vee G$
    $\forall x\, (F \wedge G) \equiv \forall x\, F \wedge G$
    $\forall x\, (F \vee G) \equiv \forall x\, F \vee G$

$F \wedge G \equiv G \wedge F$

$p \wedge q \equiv q \wedge p$

$\forall x, y, z \in \mathbb{N}$ $\forall x \forall y \forall z$

$(x \in \mathbb{N} \wedge y \in \mathbb{N}$

$\forall x\, F$

$\forall x\, F$

$\exists x\, \left(P(x) \rightarrow \forall y\, P(y)\right)$

Kapitel II – Grundlagen; Logik

- Prädikatenlogik in der Mathematik

  Mathematiker mischen oft Notationen aus der Logik, der Arithmetik und de Mengenlehre. Z.B. schreiben sie

  $$\forall x \in \mathbb{R} : \exists x \in \mathbb{R} : x \cdot y = 1$$

  - $x \in \mathbb{R}$ ist ein enstelliges Prädikat. Wir schreiben z.B. $Reel(x)$
  - $x \cdot y$ ist ein zweistelliges Funktionsymbol. Wir schreiben z.B. $prod(x, y)$
  - $\exists x \in \mathbb{R} : F$ ist eine Abkürzung für $\exists x (Reel(x) \wedge F)$
  - $\forall x \in \mathbb{R} : F$ ist eine Abkürzung für $\forall x (Reel(x) \rightarrow F)$

15

Kapitel II – Grundlagen; Logik

- Formalisierung von Aussagen
  - Aussagen werden durch eine Formel und eine Basisstruktur formalisiert.
  - Die Struktur legt die Bedeutung der Prädikate fest, die man für allgemein bekannt hält.
  - Die Namen der Prädikate werden so gewählt, dass sie ihre Bedeutung in der Basisstruktur suggerieren. Oft wird dann die Basisstruktur nicht explizit angegeben.
  - Wir betrachten folgendes Beispiel:

„Für jede Zahl gibt es eine größere Primzahl“

(es gibt unendlich viele Primzahlen)

Kapitel II – Grundlagen; Logik

- Formalisierung von Aussagen
  - Aussagen werden durch eine Formel und eine Basisstruktur formalisiert.
  - Die Struktur legt die Bedeutung der Prädikate fest, die man für allgemein bekannt hält.
  - Die Namen der Prädikate werden so gewählt, dass sie ihre Bedeutung in der Basisstruktur suggerieren. Oft wird dann die Basisstruktur nicht explizit angegeben.
  - Wir betrachten folgendes Beispiel:

„Für jede Zahl gibt es eine größere Primzahl“

(es gibt unendlich viele Primzahlen)

Kapitel II – Grundlagen; Logik

- Formalisierung von Aussagen
  - Wenn die Prädikate „Primzahl“ und „größer“ bekannt sind, dann wird die Aussage formalisiert durch:

Formel: $\forall x\, \exists y\, (Prim(y) \wedge Gr(y, x))$

Basisstruktur: $U_S = \mathbb{N}$,

$Prim_S = \{\, n \in \mathbb{N} \mid n \text{ ist Primzahl}\}$

$Gr_S = \{\, (n, m) \in \mathbb{N} \times \mathbb{N} \mid n > m\}$

Kapitel II – Grundlagen; Logik

- Formalisierung von Aussagen
  - Wenn die Bedeutung von „teilt“ bekannt ist, dann kann das Prädikat Primzahl durch eine Formel definiert werden:

$$\forall x\, (Prim(x) \leftrightarrow \forall y\, (Teilt(y, x) \rightarrow (y = x \vee y = eins))$$

  - Die Basisstruktur fixiert nun die Bedeutung des Prädikaten $Teilt$, und der Konstante $eins$.

$$Teilt_S \;=\; \{\, (n, m) \in \mathbb{N} \mid n \; teilt \; m\}$$

$$eins_S \;=\; 1$$

Kapitel II – Grundlagen; Logik

- Formalisierung von Aussagen
  - Wenn die Bedeutung von „Summe“ und „Nachfolger“ bekannt ist, dann kann das Produkt so definiert werden:

$$x \cdot 1 = x$$

$$x \cdot nach(u) = x \cdot u + x$$

  - Diese Definition kann mit der folgenden Formeln formalisiert werden:

$$\forall x \forall y \forall z \forall u \; (z = prod\,(x,y) \leftrightarrow \; (\; (y = eins \land \; z = x) \lor$$
$$(y = nach(u) \land sum(prod(x,u),x)) \;))$$

Kapitel II – Grundlagen; Logik

- Formalisierung von Aussagen
  - Die Basistruktur fixiert nun die Bedeutung von sum, $nach$, und $eins$.

$$sum_S(n,m) = \; n + m$$

$$nach_S(n) = n + 1$$

$$eins_S \; = \; 1$$

Kapitel II – Grundlagen; Logik

- Ein Kalkül für logische Inferenzen
  - Das Kalkül enthält alle Regeln des Kalküls für die Aussagenlogik plus vier Regeln für die Einführung und Beseitigung von Quantoren.
  - Sei F eine Formel und a eine Konstante. Mit $F[x/a]$ bezeichnen wir die Formel, die man erhält, in dem alle FREIEN Vorkommnisse von x in F durch a ersetzt werden
  - Beispiele:

$$F_1 = \forall y\; Q(x,y) \qquad F_1[x/a] \;=\; \forall y\; Q(a,y)$$
$$F_2 = P(x) \land \forall x\; Q(x) \qquad F_2[x/a] = \; P(a) \land \; \forall x\; Q(x)$$
$$F_3 = \forall x\; P(x) \qquad F_3[x/a] \;=\; \forall x\; P(x)$$

Kapitel II – Grundlagen; Logik

- Formalisierung von Aussagen
  - Die „Summe“ kann mit Hilfe von „Nachfolger“ so definiert werden:

4+2 = nach(4) + 1
= 5+1 = 6

$$x + 1 = nach(x)$$
$$x + nach(u) = nach(x) + u$$

Diese Definition wird durch die folgende Formeln formalisiert:

$$\forall x\; \forall y\; \forall z\; (\; z = sum(x,y) \leftrightarrow ((y = eins \land z = nach(x))) \lor$$
$$(y = nach(u) \land z = sum(nach(x),u)))$$

Kapitel II – Grundlagen; Logik

- Formalisierung von Aussagen
  - Die „Summe" kann mit Hilfe von „Nachfolger" so definiert werden:

$4+2 = nach(4)+1 = 5+1 = 6$

$$x + 1 = nach(x)$$
$$x + nach(u) = nach(x) + u$$

Diese Definition wird durch die folgende Formeln formalisiert:

$$\forall x\, \forall y\, \forall z\, (z = sum(x,y) \leftrightarrow ((y = eins \wedge z = nach(x)) \vee (y = nach(u) \wedge z = sum(nach(x), u))))$$

Kapitel II – Grundlagen; Logik

- Ein Kalkül für logische Inferenzen
  - Das Kalkül enthält alle Regeln des Kalküls für die Aussagenlogik plus vier Regeln für die Einführung und Beseitigung von Quantoren.
  - Sei F eine Formel und a eine Konstante. Mit $F[x/a]$ bezeichnen wir die Formel, die man erhält, in dem alle FREIEN Vorkommnisse von x in F durch a ersetzt werden
  - Beispiele:

$F_1 = \forall y\, Q(x,y)$ $\quad F_1[x/a] = \forall y\, Q(a,y)$

$F_2 = P(x) \wedge \forall x\, Q(x)$ $\quad F_2[x/a] = P(a) \wedge \forall x\, Q(x)$

$F_3 = \forall x\, P(x)$ $\quad F_3[x/a] = \forall x\, P(x)$

Kapitel II – Grundlagen; Logik

- Ein Kalkül für logische Inferenzen
  - Das Kalkül enthält alle Regeln des Kalküls für die Aussagenlogik plus vier Regeln für die Einführung und Beseitigung von Quantoren.
  - Sei F eine Formel und a eine Konstante. Mit $F[x/a]$ bezeichnen wir die Formel, die man erhält, in dem alle FREIEN Vorkommnisse von x in F durch a ersetzt werden
  - Beispiele:

$F_1 = \forall y\, Q(x,y)$ $\quad F_1[x/a] = \forall y\, Q(a,y)$

$F_2 = P(x) \wedge \forall x\, Q(x)$ $\quad F_2[x/a] = P(a) \wedge \forall x\, Q(x)$

$F_3 = \forall x\, P(x)$ $\quad F_3[x/a] = \forall x\, P(x)$

45

Vorlesung Diskrete Strukturen WS 13/14
Prof. Dr. J. Esparza – Institut für Informatik, TU München

New
Size
Current page size
Style
5mm
7mm
Orientation
Pages
1
OK
Cancel

Kapitel II – Grundlagen; Logik

- Ein Kalkül für logische Inferenzen
  - Allquantoreinführung.

    Für jede Sequenz A, Variable x, Formel F und für jede Konstante a, die weder in A noch in F vorkommt:

$$\frac{A \vdash F[x/a]}{A \vdash \forall x\, F}$$

  - Intuition: um $\forall x\, F$ zu zeigen, zeige, dass $F[x/a]$ für ein beliebiges a gilt.

15

Kapitel II – Grundlagen; Logik

- Ein Kalkül für logische Inferenzen
  - Das Kalkül enthält alle Regeln des Kalküls für die Aussagenlogik plus vier Regeln für die Einführung und Beseitigung von Quantoren.
  - Sei F eine Formel und a eine Konstante. Mit $F[x/a]$ bezeichnen wir die Formel, die man erhält, in dem alle FREIEN Vorkommnisse von x in F durch a ersetzt werden
  - Beispiele:

$$F_1 = \forall y\, Q(x,y) \qquad F_1[x/a] = \forall y\, Q(a,y)$$
$$F_2 = P(x) \land \forall x\, Q(x) \qquad F_2[x/a] = P(a) \land \forall x\, Q(x)$$
$$F_3 = \forall x\, P(x) \qquad F_3[x/a] = \forall x\, P(x)$$

Kapitel II – Grundlagen; Logik

- Ein Kalkül für logische Inferenzen
  - Allquantoreinführung.

    Für jede Sequenz A, Variable x, Formel F und für jede Konstante a, die weder in A noch in F vorkommt:

$$\frac{A \vdash F[x/a]}{A \vdash \forall x\, F}$$

  - Intuition: um $\forall x\, F$ zu zeigen, zeige, dass $F[x/a]$ für ein beliebiges a gilt.

Kapitel II – Grundlagen; Logik

- Ein Kalkül für logische Inferenzen
  - Existenzquantoreinführung.

    Für jede Sequenz A, Variable x, Formel F und für jede Konstante a:

$$\frac{A \vdash F[x/a]}{A \vdash \exists x\, F} \qquad \frac{A \vdash F[x/a]}{A \vdash \forall x\, F}$$

  - Intuition: um $\exists x\, F$ zu beweisen, finde einen a, für den $F[x/a]$ gilt.

Kapitel II – Grundlagen; Beweise

- Inferenzregeln für Quantoren
  - Sei $S = (U_S, I_S)$ die Struktur mit

    U_S = alle Menschen,
    V_S = Studenten dieser Vorlesung,
    H_S = Menschen, die Harry Potter kennen,
    P_S = Menschen, die die Prüfung bestanden haben.
  - In dieser Struktur sind die Annahmen A:

    (a) $\exists x\, (V(x) \land \neg B(x))$ und (b) $\forall x\, (V(x) \rightarrow P(x))$
  - Die Konklusion ist $\exists x\, (P(x) \land \neg\, B(x))$
  - Wir zeigen: $A \vdash \exists x\, (P(x) \land \neg B(x))$

Kapitel II – Grundlagen; Beweise

- Ein Kalkül für logische Inferenzen
  - Beispiel:

    Zeige, dass aus den zwei Annahmen
    - "Jemand in dieser Vorlesung weiss nicht, wer Harry Potter ist"
    - "Alle in dieser Vorlesung haben die Prüfung bestanden"

  folgendes folgt:
    - "Es gibt jemanden, der die Prüfung bestanden hat und nicht weiss, wer Harry Potter ist".

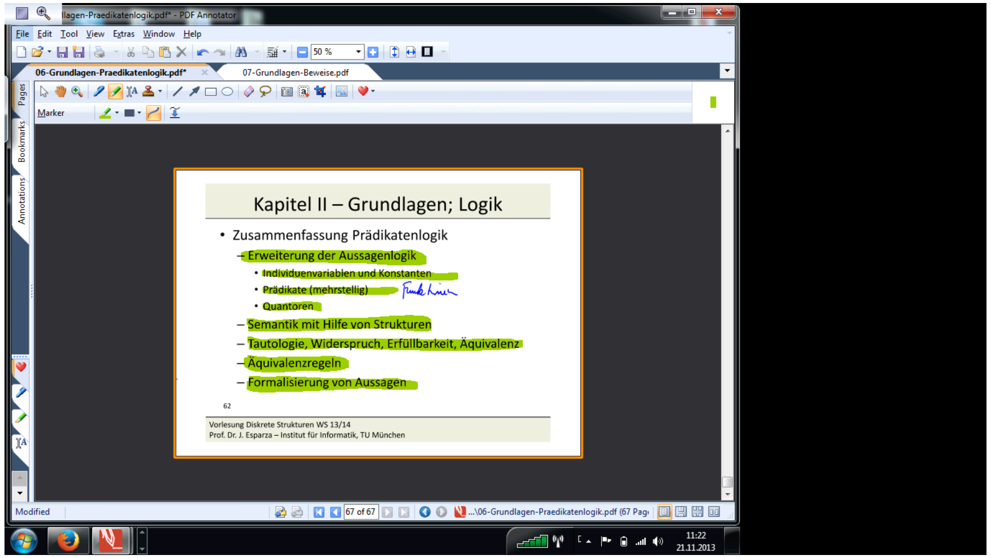
Kapitel II – Grundlagen; Beweise

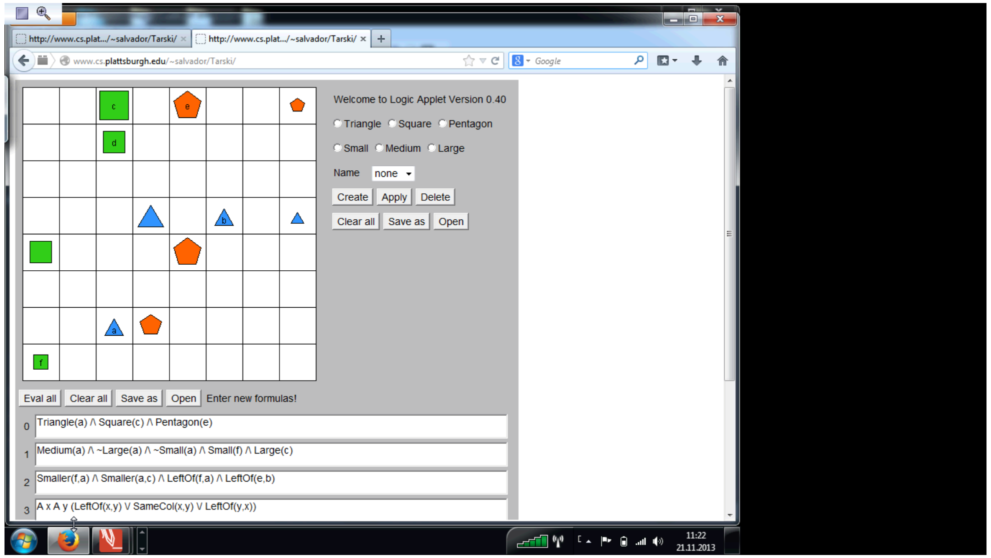
- Inferenzregeln für Quantoren

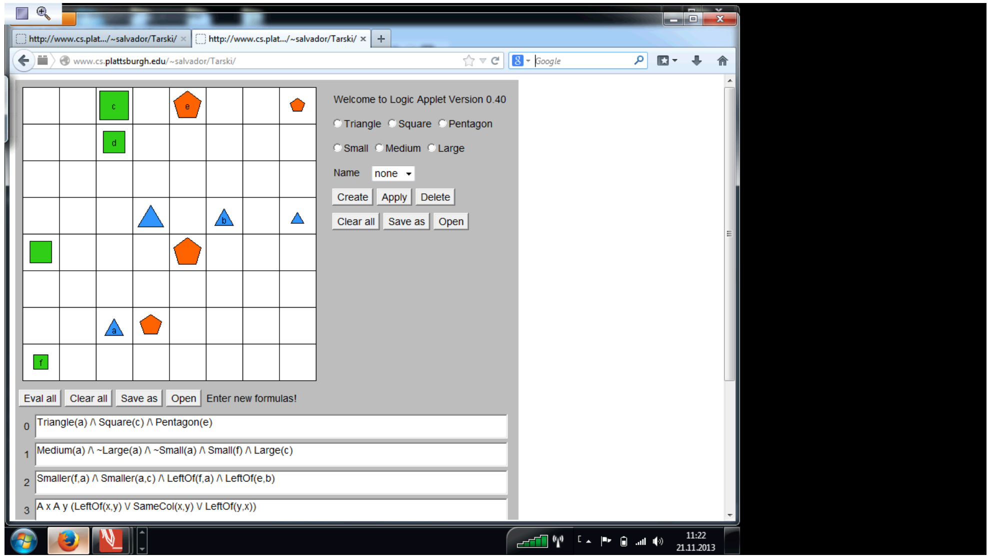
	Schritt	Bewiesen durch
1.	A, V(a) ∧ ¬ H(a) ⊢ V(a) ∧ ¬ H(a)	An.
2.	A, V(a) ∧ ¬ H(a) ⊢ V(a)	Kon.Bes. 1
3.	A, V(a) ∧ ¬ H(a) ⊢ ∀x (V(x) → P(x))	An. (b)
4.	A, V(a) ∧ ¬ H(a) ⊢ V(a) → P(a)	All.Bes. 3
5.	A, V(a) ∧ ¬ H(a) ⊢ P(a)	Imp.Bes 2,4
6.	A, V(a) ∧ ¬ H(a) ⊢ ¬ H(a)	Kon.Bes. 1
7.	A, V(a) ∧ ¬ H(a) ⊢ P(a) ∧ ¬ H(a)	Kon.Ein. 5,6
8.	A, V(a) ∧ ¬ H(a) ⊢ ∃x (P(x) ∧ ¬ H(x))	Exi.Ein.
9.	A ⊢ ∃x (V(x) ∧ ¬ H(x))	An. (a)
10.	A ⊢ ∃x (P(x) ∧ ¬ H(x))	Exi.Ein. 8,9

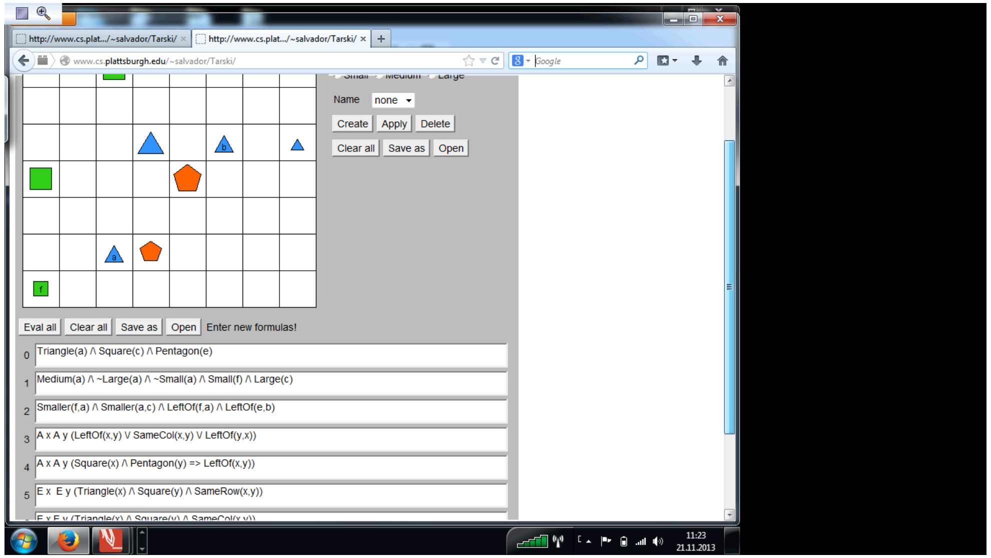
Kapitel II – Grundlagen; Logik

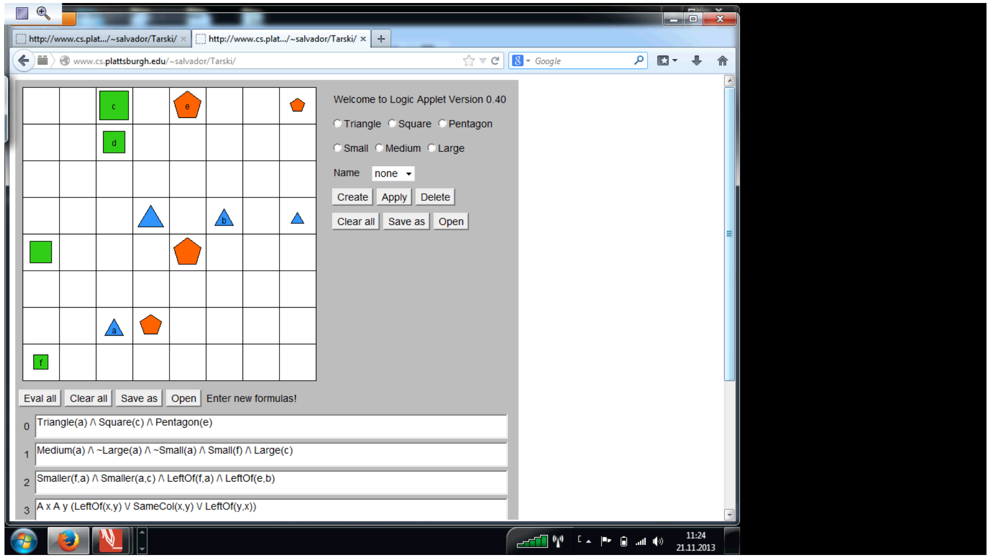
- Zusammenfassung Prädikatenlogik
  - Erweiterung der Aussagenlogik
    - Individuenvariablen und Konstanten
    - Prädikate (mehrstellig)
    - Quantoren
  - Semantik mit Hilfe von Strukturen
  - Tautologie, Widerspruch, Erfüllbarkeit, Äquivalenz
  - Äquivalenzregeln
  - Formalisierung von Aussagen

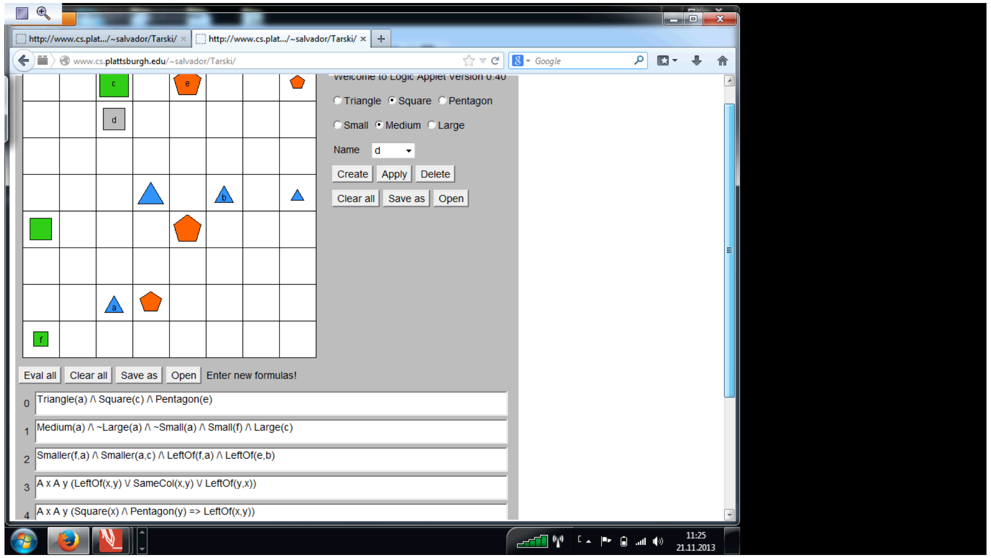

Kapitel II – Grundlagen; Logik
• Zusammenfassung Prädikatenlogik
– Erweiterung der Aussagenlogik
• Individuenvariablen und Konstanten
• Prädikate (mehrstellig)
• Quantoren
– Semantik mit Hilfe von Strukturen
– Tautologie, Widerspruch, Erfüllbarkeit, Äquivalenz
– Äquivalenzregeln
– Formalisierung von Aussagen
62
Vorlesung Diskrete Strukturen WS 13/14
Prof. Dr. J. Esparza – Institut für Informatik, TU München
06-Grundlagen-Praedikatenlogik.pdf*
07-Grundlagen-Beweise.pdf
67 of 67
11:22
21.11.2013

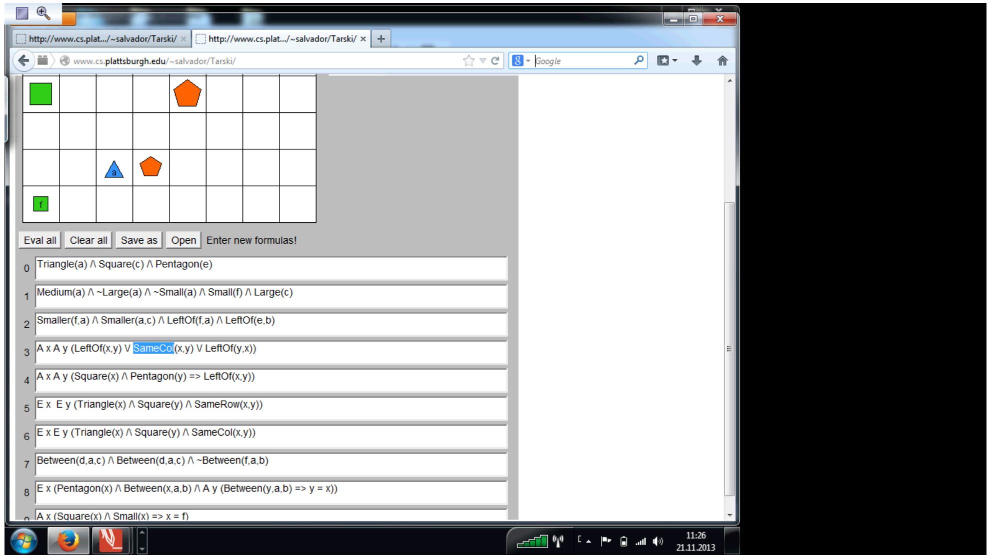

www.cs.plattsburgh.edu/~salvador/Tarski/
Welcome to Logic Applet Version 0.40
Triangle Square Pentagon
Small Medium Large
Name none
Create Apply Delete
Clear all Save as Open
Eval all Clear all Save as Open Enter new formulas!
0 Triangle(a) ∧ Square(c) ∧ Pentagon(e)
1 Medium(a) ∧ ~Large(a) ∧ ~Small(a) ∧ Small(f) ∧ Large(c)
2 Smaller(f,a) ∧ Smaller(a,c) ∧ LeftOf(f,a) ∧ LeftOf(e,b)
3 A x A y (LeftOf(x,y) V SameCol(x,y) V LeftOf(y,x))
11:22
21.11.2013

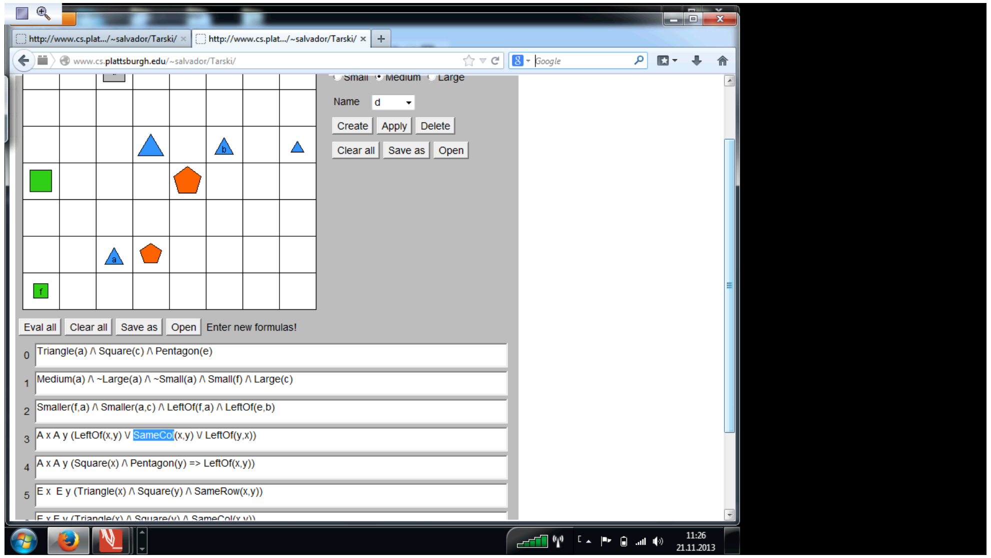

www.cs.plattsburgh.edu/~salvador/Tarski/
Welcome to Logic Applet Version 0.40
Triangle Square Pentagon
Small Medium Large
Name none
Create Apply Delete
Clear all Save as Open
Eval all Clear all Save as Open Enter new formulas!
0 Triangle(a) ∧ Square(c) ∧ Pentagon(e)
1 Medium(a) ∧ ~Large(a) ∧ ~Small(a) ∧ Small(f) ∧ Large(c)
2 Smaller(f,a) ∧ Smaller(a,c) ∧ LeftOf(f,a) ∧ LeftOf(e,b)
3 A x A y (LeftOf(x,y) V SameCol(x,y) V LeftOf(y,x))
11:22
21.11.2013

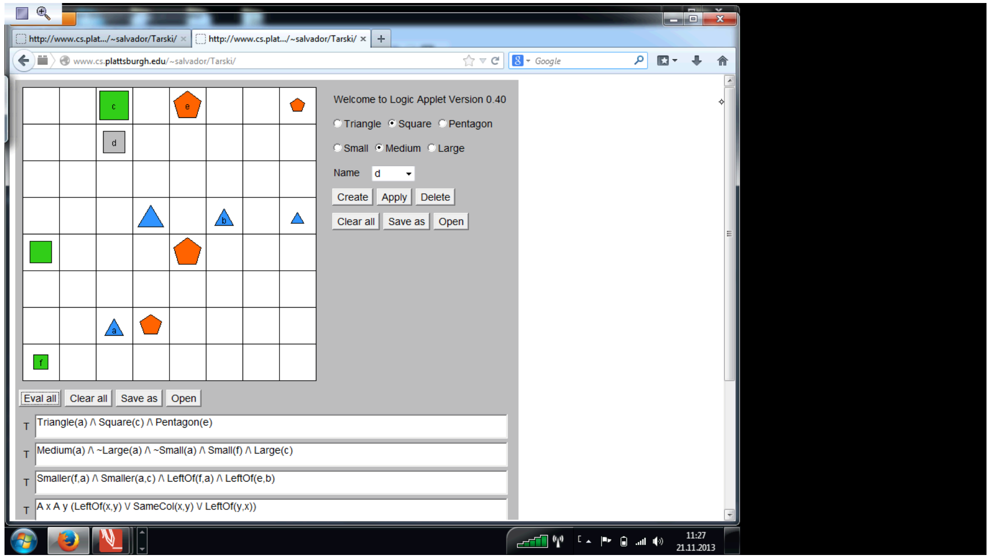

www.cs.plattsburgh.edu/~salvador/Tarski/
Name none
Create Apply Delete
Clear all Save as Open
Eval all Clear all Save as Open Enter new formulas!
0 Triangle(a) ∧ Square(c) ∧ Pentagon(e)
1 Medium(a) ∧ ~Large(a) ∧ ~Small(a) ∧ Small(f) ∧ Large(c)
2 Smaller(f,a) ∧ Smaller(a,c) ∧ LeftOf(f,a) ∧ LeftOf(e,b)
3 A x A y (LeftOf(x,y) V SameCol(x,y) V LeftOf(y,x))
4 A x A y (Square(x) ∧ Pentagon(y) => LeftOf(x,y))
5 E x E y (Triangle(x) ∧ Square(y) ∧ SameRow(x,y))
11:23
21.11.2013

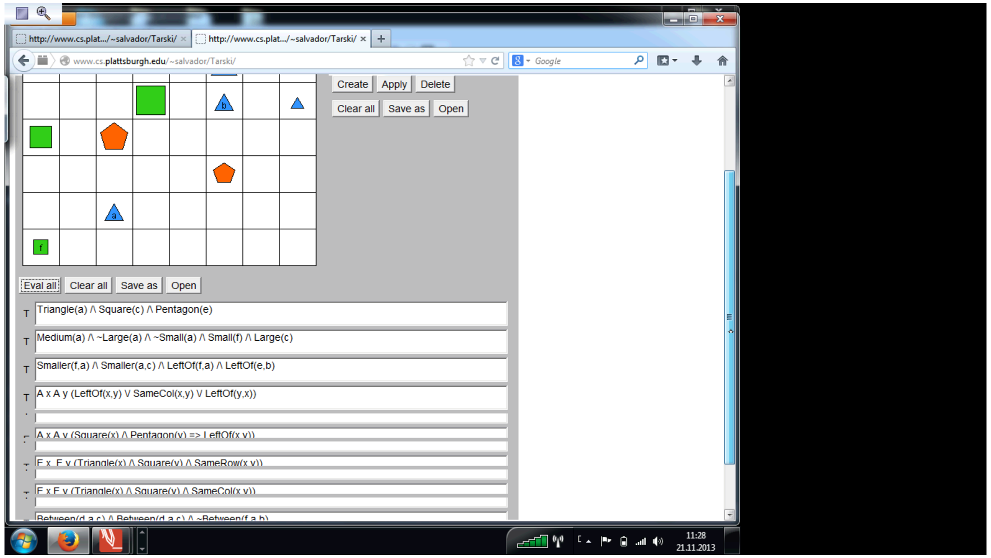

Welcome to Logic Applet Version 0.40
Triangle Square Pentagon
Small Medium Large
Name none
Create Apply Delete
Clear all Save as Open
Eval all Clear all Save as Open Enter new formulas!
0 Triangle(a) ∧ Square(c) ∧ Pentagon(e)
1 Medium(a) ∧ ~Large(a) ∧ ~Small(a) ∧ Small(f) ∧ Large(c)
2 Smaller(f,a) ∧ Smaller(a,c) ∧ LeftOf(f,a) ∧ LeftOf(e,b)
3 A x A y (LeftOf(x,y) V SameCol(x,y) V LeftOf(y,x))
11:24
21.11.2013

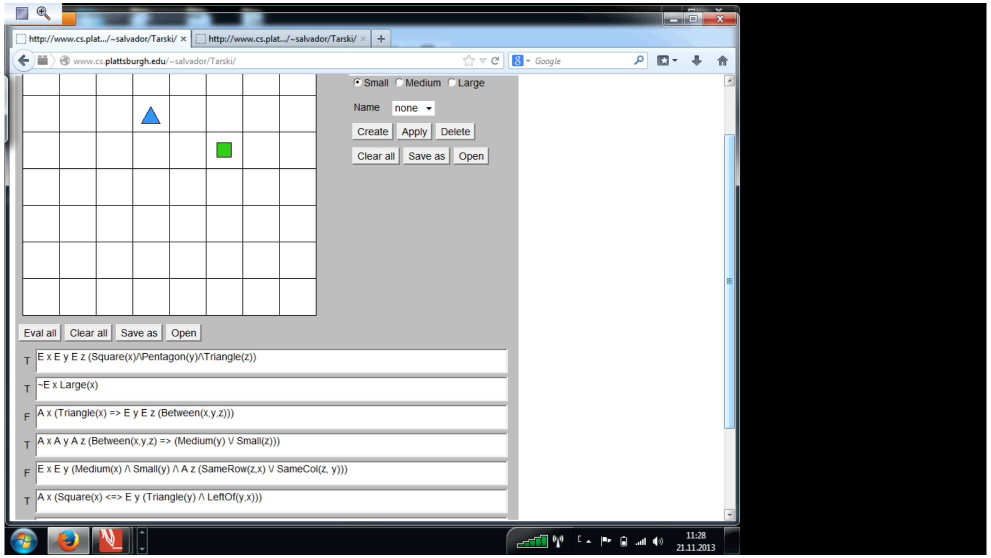

Triangle Square Pentagon
Small Medium Large
Name d
Create Apply Delete
Clear all Save as Open
Eval all Clear all Save as Open Enter new formulas!
0 Triangle(a) ∧ Square(c) ∧ Pentagon(e)
1 Medium(a) ∧ ~Large(a) ∧ ~Small(a) ∧ Small(f) ∧ Large(c)
2 Smaller(f,a) ∧ Smaller(a,c) ∧ LeftOf(f,a) ∧ LeftOf(e,b)
3 A x A y (LeftOf(x,y) V SameCol(x,y) V LeftOf(y,x))
4 A x A y (Square(x) ∧ Pentagon(y) => LeftOf(x,y))
11:25
21.11.2013

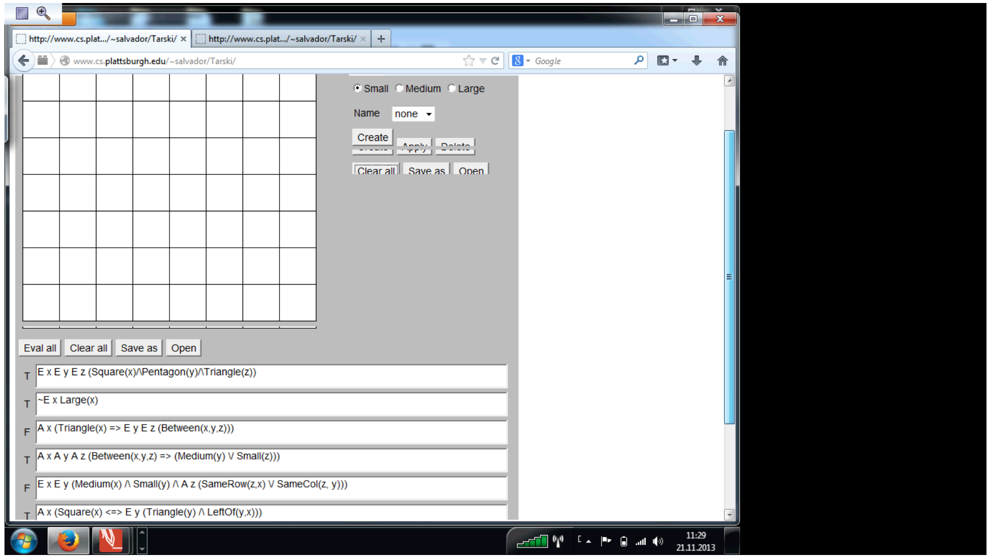

Eval all Clear all Save as Open Enter new formulas!
0 Triangle(a) ∧ Square(c) ∧ Pentagon(e)
1 Medium(a) ∧ ~Large(a) ∧ ~Small(a) ∧ Small(f) ∧ Large(c)
2 Smaller(f,a) ∧ Smaller(a,c) ∧ LeftOf(f,a) ∧ LeftOf(e,b)
3 A x A y (LeftOf(x,y) V SameCol(x,y) V LeftOf(y,x))
4 A x A y (Square(x) ∧ Pentagon(y) => LeftOf(x,y))
5 E x E y (Triangle(x) ∧ Square(y) ∧ SameRow(x,y))
6 E x E y (Triangle(x) ∧ Square(y) ∧ SameCol(x,y))
7 Between(d,a,c) ∧ Between(d,a,c) ∧ ~Between(f,a,b)
8 E x (Pentagon(x) ∧ Between(x,a,b) ∧ A y (Between(y,a,b) => y = x))
11:26
21.11.2013

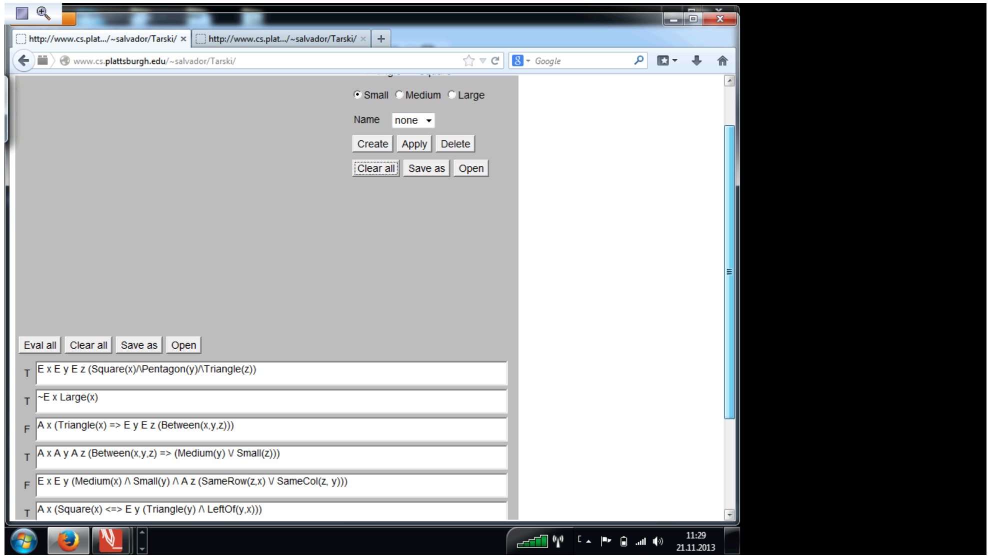

Name d
Create Apply Delete
Clear all Save as Open
Eval all Clear all Save as Open Enter new formulas!
0 Triangle(a) ∧ Square(c) ∧ Pentagon(e)
1 Medium(a) ∧ ~Large(a) ∧ ~Small(a) ∧ Small(f) ∧ Large(c)
2 Smaller(f,a) ∧ Smaller(a,c) ∧ LeftOf(f,a) ∧ LeftOf(e,b)
3 A x A y (LeftOf(x,y) V SameCol(x,y) V LeftOf(y,x))
4 A x A y (Square(x) ∧ Pentagon(y) => LeftOf(x,y))
5 E x E y (Triangle(x) ∧ Square(y) ∧ SameRow(x,y))
11:26
21.11.2013

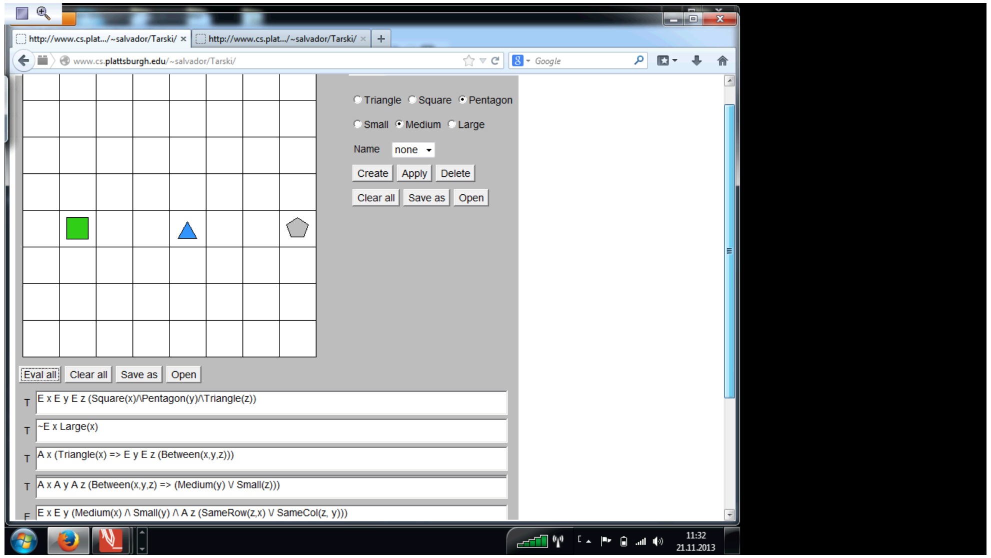

Welcome to Logic Applet Version 0.40
Triangle Square Pentagon
Small Medium Large
Name d
Create Apply Delete
Clear all Save as Open
Eval all Clear all Save as Open
T Triangle(a) ∧ Square(c) ∧ Pentagon(e)
T Medium(a) ∧ ~Large(a) ∧ ~Small(a) ∧ Small(f) ∧ Large(c)
T Smaller(f,a) ∧ Smaller(a,c) ∧ LeftOf(f,a) ∧ LeftOf(e,b)
T A x A y (LeftOf(x,y) V SameCol(x,y) V LeftOf(y,x))
11:27
21.11.2013

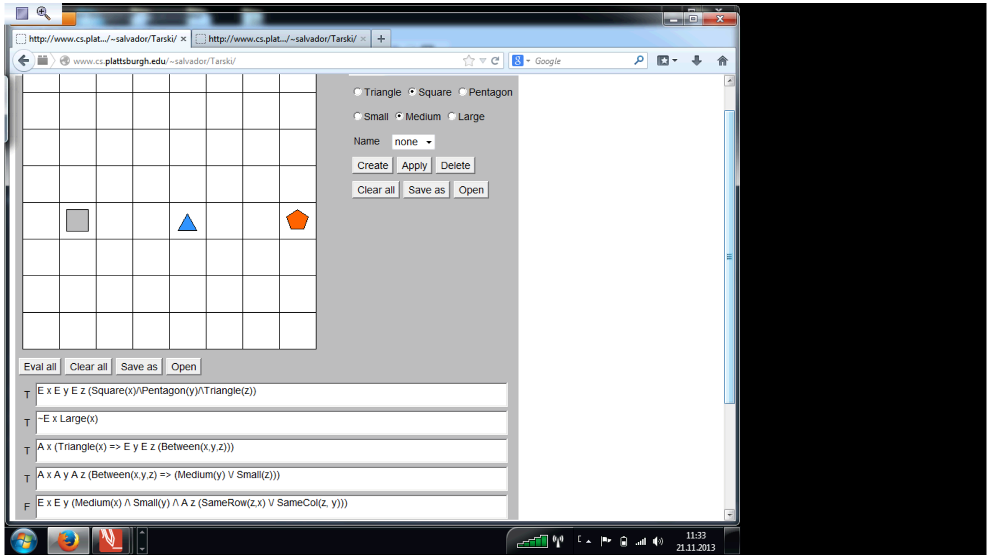

Create Apply Delete
Clear all Save as Open
Eval all Clear all Save as Open
T Triangle(a) ∧ Square(c) ∧ Pentagon(e)
T Medium(a) ∧ ~Large(a) ∧ ~Small(a) ∧ Small(f) ∧ Large(c)
T Smaller(f,a) ∧ Smaller(a,c) ∧ LeftOf(f,a) ∧ LeftOf(e,b)
T A x A y (LeftOf(x,y) V SameCol(x,y) V LeftOf(y,x))
11:28
21.11.2013

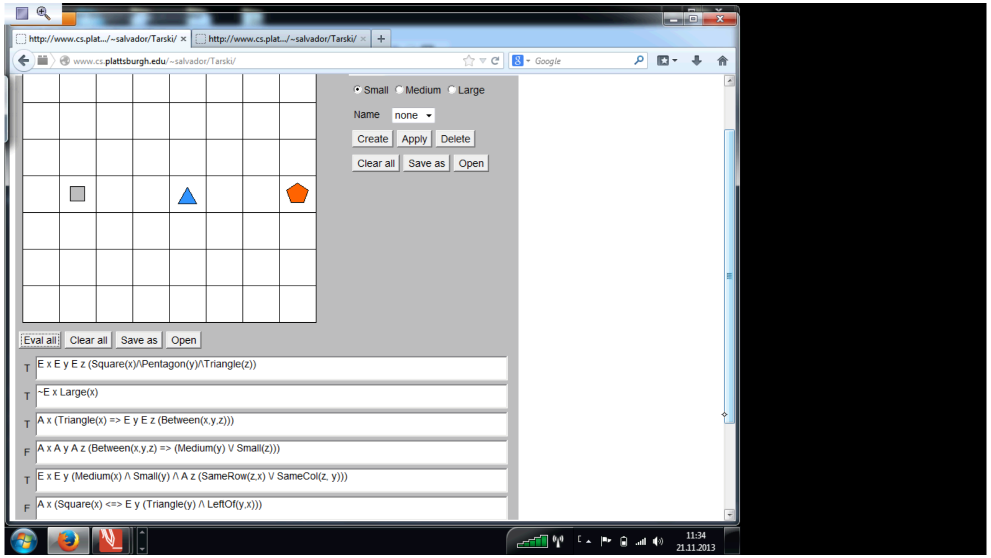

Small Medium Large
Name none
Create Apply Delete
Clear all Save as Open
Eval all Clear all Save as Open
T E x E y E z (Square(x)∧Pentagon(y)∧Triangle(z))
T ~E x Large(x)
F A x (Triangle(x) => E y E z (Between(x,y,z)))
T A x A y A z (Between(x,y,z) => (Medium(y) V Small(z)))
F E x E y (Medium(x) ∧ Small(y) ∧ A z (SameRow(z,x) V SameCol(z, y)))
T A x (Square(x) <=> E y (Triangle(y) ∧ LeftOf(y,x)))
11:28
21.11.2013

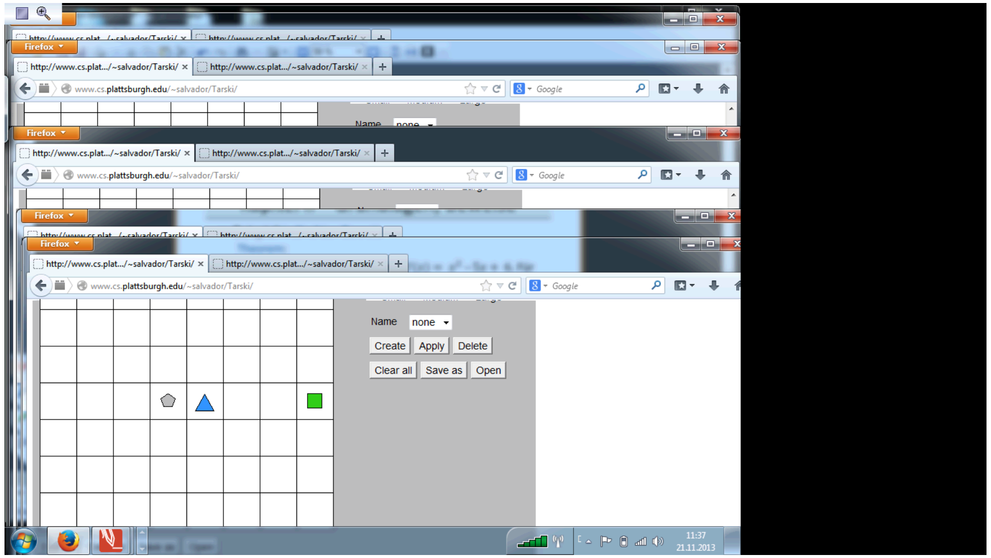

Small Medium Large
Name none
Create
Eval all Clear all Save as Open
T E x E y E z (Square(x)∧Pentagon(y)∧Triangle(z))
T ~E x Large(x)
F A x (Triangle(x) => E y E z (Between(x,y,z)))
T A x A y A z (Between(x,y,z) => (Medium(y) V Small(z)))
F E x E y (Medium(x) ∧ Small(y) ∧ A z (SameRow(z,x) V SameCol(z, y)))
T A x (Square(x) <=> E y (Triangle(y) ∧ LeftOf(y,x)))
11:29
21.11.2013


http://www.cs.plat.../~salvador/Tarski/
www.cs.plattsburgh.edu/~salvador/Tarski/
Small Medium Large
Name none
Create Apply Delete
Clear all Save as Open
Eval all Clear all Save as Open
T E x E y E z (Square(x)/\Pentagon(y)/\Triangle(z))
T ~E x Large(x)
F A x (Triangle(x) => E y E z (Between(x,y,z)))
T A x A y A z (Between(x,y,z) => (Medium(y) V Small(z)))
F E x E y (Medium(x) Λ Small(y) Λ A z (SameRow(z,x) V SameCol(z, y)))
T A x (Square(x) <=> E y (Triangle(y) Λ LeftOf(y,x)))
11:29
21.11.2013

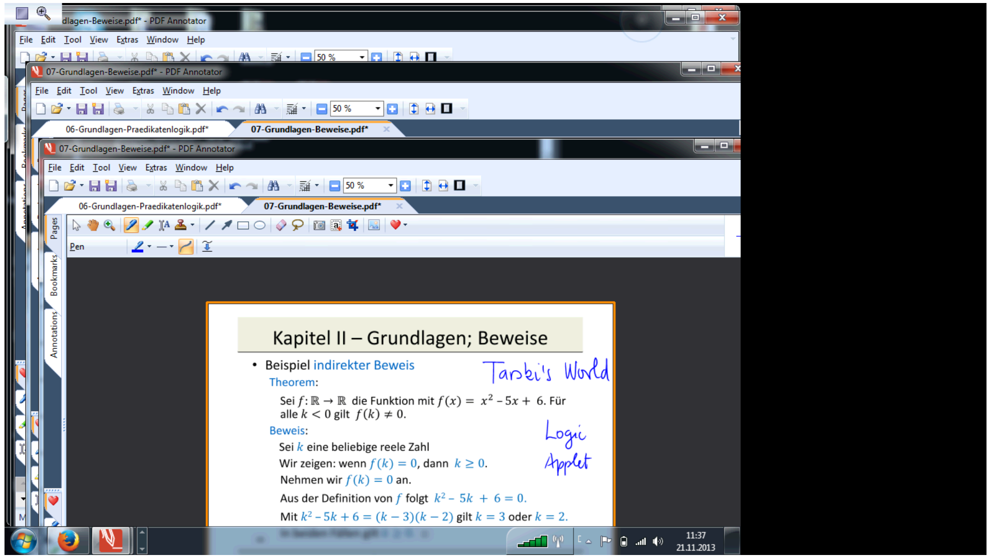

http://www.cs.plat.../~salvador/Tarski/
www.cs.plattsburgh.edu/~salvador/Tarski/
Triangle Square Pentagon
Small Medium Large
Name none
Create Apply Delete
Clear all Save as Open
Eval all Clear all Save as Open
T E x E y E z (Square(x)/\Pentagon(y)/\Triangle(z))
T ~E x Large(x)
T A x (Triangle(x) => E y E z (Between(x,y,z)))
T A x A y A z (Between(x,y,z) => (Medium(y) V Small(z)))
F E x E y (Medium(x) Λ Small(y) Λ A z (SameRow(z,x) V SameCol(z, y)))
11:32
21.11.2013


http://www.cs.plat.../~salvador/Tarski/
www.cs.plattsburgh.edu/~salvador/Tarski/
Triangle Square Pentagon
Small Medium Large
Name none
Create Apply Delete
Clear all Save as Open
Eval all Clear all Save as Open
T E x E y E z (Square(x)/\Pentagon(y)/\Triangle(z))
T ~E x Large(x)
T A x (Triangle(x) => E y E z (Between(x,y,z)))
T A x A y A z (Between(x,y,z) => (Medium(y) V Small(z)))
F E x E y (Medium(x) Λ Small(y) Λ A z (SameRow(z,x) V SameCol(z, y)))
11:33
21.11.2013

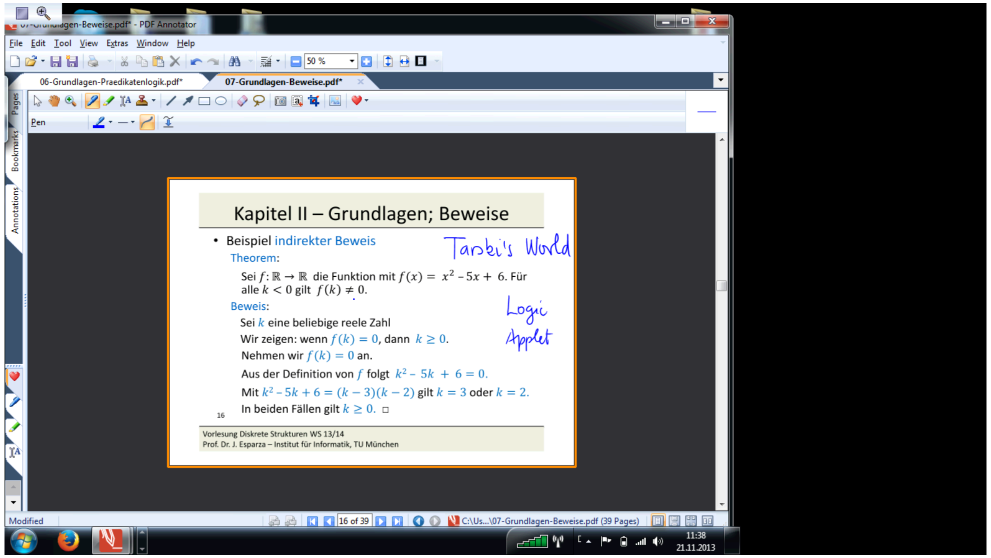

http://www.cs.plat.../~salvador/Tarski/
www.cs.plattsburgh.edu/~salvador/Tarski/
Small Medium Large
Name none
Create Apply Delete
Clear all Save as Open
Eval all Clear all Save as Open
T E x E y E z (Square(x)/\Pentagon(y)/\Triangle(z))
T ~E x Large(x)
T A x (Triangle(x) => E y E z (Between(x,y,z)))
F A x A y A z (Between(x,y,z) => (Medium(y) V Small(z)))
T E x E y (Medium(x) Λ Small(y) Λ A z (SameRow(z,x) V SameCol(z, y)))
F A x (Square(x) <=> E y (Triangle(y) Λ LeftOf(y,x)))
11:34
21.11.2013



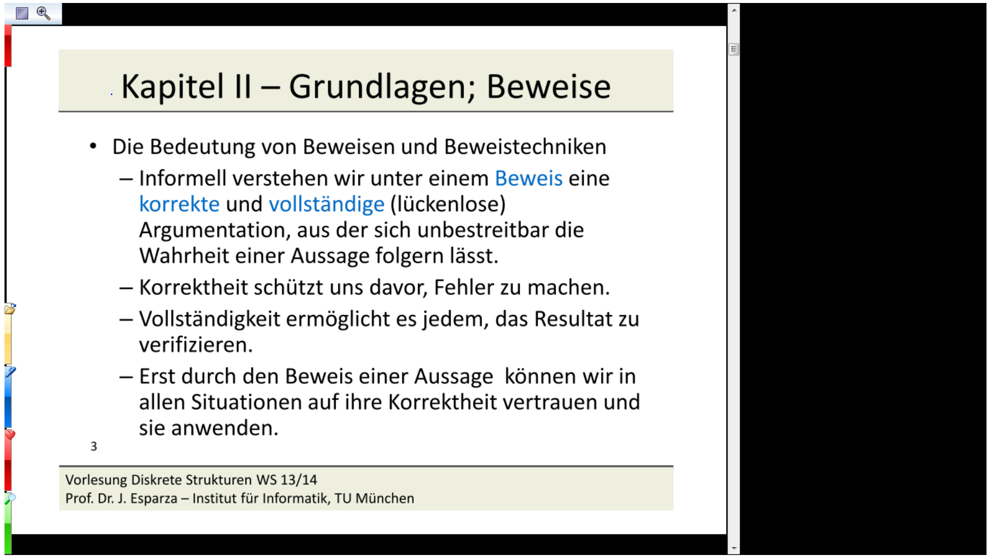
www.cs.plattsburgh.edu/~salvador/Tarski/
Name none
Create Apply Delete
Clear all Save as Open



07-Grundlagen-Beweise.pdf* - PDF Annotator
06-Grundlagen-Praedikatenlogik.pdf*
07-Grundlagen-Beweise.pdf*
Kapitel II – Grundlagen; Beweise
Beispiel indirekter Beweis
Tarski's World
Theorem:
Sei $f: \mathbb{R} \to \mathbb{R}$ die Funktion mit $f(x) = x^2 - 5x + 6$. Für alle $k < 0$ gilt $f(k) \neq 0$.
Beweis:
Logic
Applet
Sei k eine beliebige reele Zahl
Wir zeigen: wenn $f(k) = 0$, dann $k \geq 0$.
Nehmen wir $f(k) = 0$ an.
Aus der Definition von f folgt $k^2 - 5k + 6 = 0$.
Mit $k^2 - 5k + 6 = (k-3)(k-2)$ gilt $k = 3$ oder $k = 2$.
11:37
21.11.2013



07-Grundlagen-Beweise.pdf* - PDF Annotator
06-Grundlagen-Praedikatenlogik.pdf*
07-Grundlagen-Beweise.pdf*
Kapitel II – Grundlagen; Beweise
Beispiel indirekter Beweis
Tarski's World
Theorem:
Sei $f: \mathbb{R} \to \mathbb{R}$ die Funktion mit $f(x) = x^2 - 5x + 6$. Für alle $k < 0$ gilt $f(k) \neq 0$.
Beweis:
Logic
Applet
Sei k eine beliebige reele Zahl
Wir zeigen: wenn $f(k) = 0$, dann $k \geq 0$.
Nehmen wir $f(k) = 0$ an.
Aus der Definition von f folgt $k^2 - 5k + 6 = 0$.
Mit $k^2 - 5k + 6 = (k-3)(k-2)$ gilt $k = 3$ oder $k = 2$.
In beiden Fällen gilt $k \geq 0$. □
16
Vorlesung Diskrete Strukturen WS 13/14
Prof. Dr. J. Esparza – Institut für Informatik, TU München
16 of 39
C:\Us...\07-Grundlagen-Beweise.pdf (39 Pages)
11:38
21.11.2013



Kapitel II – Grundlagen; Beweise
Die Bedeutung von Beweisen und Beweistechniken
– Informell verstehen wir unter einem Beweis eine korrekte und vollständige (lückenlose) Argumentation, aus der sich unbestreitbar die Wahrheit einer Aussage folgern lässt.
– Korrektheit schützt uns davor, Fehler zu machen.
– Vollständigkeit ermöglicht es jedem, das Resultat zu verifizieren.
– Erst durch den Beweis einer Aussage können wir in allen Situationen auf ihre Korrektheit vertrauen und sie anwenden.
3
Vorlesung Diskrete Strukturen WS 13/14
Prof. Dr. J. Esparza – Institut für Informatik, TU München